ZolOS Functions

|  |  |
| --- | --- |
| Last edited: | 14 Sep 2025 |

ZolOS functions are called via indirection and a jump table, making them available to user programs.

The key files used in setting this up are:

| File | Provides | Notes |
| --- | --- | --- |
| LIB/cfg\_main.asm | Sets addresses of the OS function names – eg, OSGETKEY = $FF00 | Include this in user programs so that the names, with their corresponding addresses, are defined. |
| LIB/cfg\_page\_2.asm | Sets addresses of the call vectors – OSGETKEY\_VEC = $0200 | Include this is user programs to define the call vectors. |
| ROM/z64-main.asm | Creates the main jump table by putting indirect jump instructions at the addresses defined in cfg\_main.asm. |  |
| ROM/include/os\_call\_vectors.asm | Sets up the vectors by putting the real addresses of the various assembly code routines into the call vector table. |  |

NB: The first two files don’t add any size to user programs by being included – they simply define labels for locations.

*For A, X and Y columns*: O = overwritten, P = preserved, – = not affected or not applicable.

NB: These apply only to the top-level functions. Any sub-routines called from within them may affect A, X or Y.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| OS Function | Source file & function | On Entry | On Exit / Notes | A | X | Y |
| READ | | | | | | |
| OSGETINP  Creates an input loop waiting for the null received flag to be set | funcs\_io.asm  get\_input | Resets STDIN\_IDX to 0.  Sets first byte of STDIN\_BUF to 0. | Clears the null received flag. |  |  |  |
| OSGETKEY  Get a single character from STDIN\_BUF | funcs\_io.asm  getkey | Resets STDIN\_IDX to 0. | Key ASCII code in FUNC\_RESULT.  0 means just <return> was entered.  STDIN\_IDX and STDIN\_BUF are reset. | O | – | – |
| OSRDASC  Wrapper to OSRDBYTE. Reads next printable char (including space) from STDIN\_BUF | funcs\_io.asm  read\_ascii | Uses STDIN\_IDX to get next char. | FUNC\_RESULT contains char code.  FUNC\_ERR contains error code.  STDIN\_IDX updated.  STDIN\_BUF not affected. | O | O | – |
| OSRDBYTE  Reads next byte from STDIN\_BUF | funcs\_io.asm    read\_byte | Uses STDIN\_IDX to get next char. | FUNC\_RESULT contains char code.  FUNC\_ERR contains error code.  STDIN\_IDX updated.  STDIN\_BUF not affected. | O | O | – |
| OSRDHBYTE  Reads 2 ASCII chars from STDIN\_BUF and converts to 8-bit value. | funcs\_io.asm    read\_hex\_byte | Uses STDIN\_IDX to get next char.  Expects pair of ASCII chars in STDIN\_BUF | FUNC\_RESULT contains value.  FUNC\_ERR contains error code. | P | P | P |
| OSRDHADDR  Read a two-byte hex address from STDIN\_BUF and converts to 16-bit value. | funcs\_io.asm    read\_hex\_addr | Expects nul- or space-terminated string of ASCII hex characters in STDIN\_BUF.  Uses OSRDHBYTE to get each pair of chars & convert to value. | FUNC\_RES\_L/H contain 16-bit value.  FUNC\_ERR contains last error raised by OSRDHBYTE | P | – | P |
| OSRDCH  Wrapper to OSRDBYTE. Reads next non-space printable char from STDIN\_BUF | funcs\_io.asm    read\_char | Uses STDIN\_IDX to get next char. | FUNC\_RESULT contains char code.  FUNC\_ERR contains error code.  STDIN\_IDX updated.  STDIN\_BUF not affected. | O | O | – |
| OSRDINT16  Read a 16-bit decimal integer from STDIN\_BUF | funcs\_io.asm    read\_int16 | Uses STDIN\_IDX to get next char. | FUNC\_RES\_L/H contain 16-bit number.  FUNC\_ERR contains error code.  STDIN\_IDX updated.  STDIN\_BUF not affected. | P | P | P |
| OSRDFNAME  Reads string from STDIN\_BUF. Checks conforms to filename specs. | funcs\_io.asm    read\_filename | Assumes next data in STDIN\_BUF pointed to by STDIN\_IDX is a filename. | STR\_BUF contains nul-terminated filename.  FUNC\_ERR contains error code.  STDIN\_IDX updated. | P | P | P |
| OSRDSTR  Reads string from STDIN\_BUF | funcs\_io.asm    read\_string | Assumes next data in STDIN\_BUF pointed to by STDIN\_IDX is a filename. | STR\_BUF contains nul-terminated string.  FUNC\_ERR contains error code.  STDIN\_IDX updated. | P | P | P |
| WRITE | | | | | | |
| OSWRBUF  Write STDOUT\_BUF to output stream |  | STDOUT\_BUF must contain null-terminated stream of characters. |  | O | O | – |
| OSWRCH  Write single character to output stream |  | A contains ASCII value of character. |  | – | – | – |
| OSWRERR  Write OS error string to output stream | funcs\_io.asm    os\_print\_error | The error code must be in FUNC\_ERR |  | O | O | – |
| OSWRMSG  Write text pointed to by MSG\_VEC to output stream |  | MSG\_VEC and MSG\_VEC+1 must contain address of a null-terminated message string. |  | P | – | P |
| OSWROP  Write to Output Port on DUART board | funcs\_uart\_sc28L92.asm    duart\_writeOP | A contains value (0 or 1) to be set on pin.    X contains pin number constant – eg, SC28L92\_OP2 |  | P | P | – |
| OSWRSBUF  Write STR\_BUF to output stream |  | STR\_BUF must contain a nul-terminated string. |  | P | – | – |
| OSSOAPP  Append string to STDOUT\_BUF | funcs\_io.asm    stdout\_append | Assumes STDOUT\_IDX points to next char in buffer.  MSG\_VEC/+1 must point to the string | FUNC\_ERR contains err code. 0 = success.  STDOUT\_IDX updated | O | P | P |
| OSSOCH  Append character to STDOUT\_BUF | funcs\_io.asm    stdout\_add\_char | ASCII character code in A. | STDOUT\_IDX updated | P | P | – |
| CONVERSIONS | | | | | | |
| OSB2BIN  Convert a 1-byte integer value to a binary string representation | funcs\_conv.asm    byte\_to\_bin | A must contain value to be converted. | STR\_BUF contains 9 bytes containing binary characters plus nul terminator | O | P | P |
| OSB2HEX  Converts 8-bit value to 2-char hex string representation | funcs\_conv.asm    byte\_to\_hex\_str | A must contain value to be converted. | STR\_BUF contains 3 bytes containing hex characters plus nul terminator | O | – | – |
| OSB2ISTR  Converts 8-bit value to decimal integer string representation | funcs\_conv.asm    byte\_to\_int\_str | A must contain value to be converted. | STR\_BUF contains integer string plus nul terminator  FUNC\_RESULT contains number of digits (not including null terminator) | P | P | P |
| OSHEX2B  Converts 2-char hex string to byte value | funcs\_conv.asm    hex\_str\_to\_byte | BYTE\_CONV\_L/H must contain ASCII hex codes for low/high nibbles. | FUNC\_RESULT contains byte value  FUNC\_ERR contains error code generated by OSHEX2DEC | P | P | – |
| OSU16ISTR  Converts a 16-bit value to a decimal string | funcs\_conv.asm    uint16\_to\_int\_str | MATH\_TMP\_A\_L/H contains 16-bit value | STR\_BUF contains nul-terminated decimal string | P | P | P |
| OSU16HEX  Converts a 16-bit value to a 4-char hex string | funcs\_conv.asm    uint16\_to\_hex\_str | TMP\_ADDR\_A\_L/H contains 16-bit value | STR\_BUF contains nul-terminated hex string | P | – | – |
| OSHEX2DEC  Converts 1-byte integer representing a hex char (ie, '0' to 'F') to integer value (0-15) | funcs\_conv.asm    asc\_hex\_to\_dec | A contains ASCI character value | A contains numeric value  FUNC\_ERR contains error code | O | P | – |
| LCD | | | | | | |
| OSLCDCH  LCD write char |  | A contains ASCII value of character |  | P | – | – |
| OSLCDCLS  LCD clear screen |  |  |  | O | – | – |
| OSLCDERR  LCD write OS error string |  | FUNC\_ERR is assumed to contain an error code |  | O | O | – |
| OSLCDMSG  LCD write text pointed to by MSG\_VEC |  | MSG\_VEC and MSG\_VEC+1 must contain address of a null-terminated message string. |  | P | P | P |
| OSLCDB2HEX  Print byte value as hex |  | A must contain byte value | Uses STR\_BUF as temporary store | O | – | – |
| OSLCDSBUF  Print contents of STR\_BUF to LCD |  | STR\_BUF must contain a nul-terminated string. |  | O | – | – |
| OSLCDSC  LCD Set Cursor |  | X should contain the X param in range 0-15.  Y should be 0 or 1. |  | O | – | O |
| OSLCDWRBUF  Write STDOUT\_BUF to LCD |  | STDOUT\_BUF must contain a nul-terminated string. |  | O | – | – |
| PARALLEL / PRINTER | | | | | | |
| OSPRTBUF  Print contents of STDOUT\_BUF | funcs\_prt.asm    prt\_stdout\_buf | STDOUT\_BUF should contain a nul-terminated string. Calls OSPRTMSG. | FUNC\_RESULT will contain a result code  Wrapper to OSPRTMSG  A is overwritten | O | – | – |
| OSPRTCH  Print character | funcs\_prt.asm    prt\_char | A must contain ASCII char code. |  | O | – | – |
| OSPRTCHK  Check printer state | funcs\_prt.asm    prt\_check\_state |  | FUNC\_RESULT contains one of following error codes:  0 (available/no error)  ERR\_PRT\_STATE\_OL  ERR\_PRT\_STATE\_PE  ERR\_PRT\_STATE\_ERR | O | – | P |
| OSPRTINIT  Initialise the printer VIA | funcs\_prt.asm    prt\_init |  |  | O | – | – |
| OSPRTMSG  Print string pointed to by MSG\_VEC | funcs\_prt.asm    prt\_msg | MSG\_VEC/+1 should contain pointer to a nul-terminated string | FUNC\_RESULT will contain a result code | O | – | O |
| OSPRTSBUF  Print contents of STR\_BUF | funcs\_prt.asm    prt\_str\_buf | STR\_BUF should contain a nul-terminated string. Calls OSPRTMSG. | FUNC\_RESULT will contain a result code  Wrapper to OSPRTMSG | O | – | – |
|  |  |  |  |  |  |  |
| ZolaDOS | | | | | | |
| OSZDDEL  Delete a file on the ZolaDOS server. | funcs\_ZolaDOS    zd\_delfile | STR\_BUF must contain nul-terminated filename | FUNC\_ERR contains error code (0 if successful). | O | – | – |
| OSZDLOAD  Load a file from the ZolaDOS server into memory at USR\_START | funcs\_ZolaDOS    zd\_loadfile | STR\_BUF must contain nul-terminated filename  FILE\_ADDR/+1 must contain address to which data will be loaded | FUNC\_ERR contains error code (0 if successful).  LOMEM is set. | O | – | – |
| OSZDSAVE  Save a block of memory to a file. | funcs\_ZolaDOS    zd\_save\_data | TMP\_ADDR\_A/+1 must contain start address of memory  TMP\_ADDR\_B/+1 must contain end address of memory  STR\_BUF must contain nul-terminated filename | FUNC\_ERR contains error code (0 if successful). | O | – | – |
| MISC | | | | | | |
| OSDELAY  General-purpose delay function. Blocking | funcs\_4x20\_lcd.asm  delay | LCDV\_TIMER\_INTVL/+1 contains 16-bit delay value (in ms) |  | P | – | – |
| SPI | | | | | | |
| OSSPIEXCH  Performs an SPI byte exchange | funcs\_spi65    spi\_exchange\_byte | A contains byte to be sent | A contains byte received | O | – | – |
| OSRDDATE  Read date from RTC |  |  | Date data starting at RTC\_DAT\_BUF |  |  |  |
| OSRDTIME  Read time from RTC |  |  | Time data starting at RTC\_CLK\_BUF |  |  |  |
|  |  |  |  |  |  |  |
| OSSFTRST  Soft reset |  | -- | Use direct JMP (not JSR or vectored/indirect) | – | – | – |
| OSHRDRST  Hard reset |  | -- | Use direct JMP (not JSR or vectored/indirect) | – | – | – |
|  |  |  |  |  |  |  |