OS Call Summary

Last edited: 7 Jun 2025

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 Converts 8-bit value to 2-char hex 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	0	-	-
OSLCDSBUF Print contents of STR_BUF to LCD		STR_BUF must contain a nul-terminated string.		0	-	-
OSLCDSC LCD Set Cursor		X should contain the X param in range 0-15. Y should be 0 or 1.		0	-	0
OSLCDWRBUF Write STDOUT_BUF to LCD		STDOUT_BUF must contain a nul-terminated string.		0	-	-
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	0	-	-
OSPRTCH Print character	funcs_prt.asm prt_char	A must contain ASCII char code.		0	-	-

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	0	-	P
OSPRTINIT Initialise the printer VIA	funcs_prt.asm prt_init			0	-	-
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	0	-	0
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	0	-	-
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	-	-

OSUSRINT For vectoring user-program interrupts		-- to come --				
SPI						
OSSPIEXCH Performs an SPI byte exchange	funcs_spi65 spi_exchange_byte	**A** contains byte to be sent	**A** contains byte received	0	-	-
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)	-	-	-