

# DarkRISCV Software Environment

Generated by Doxygen 1.8.17



<b>1 CoreMark on DarkRISCV</b>	<b>1</b>
1.0.1 config example:	1
1.0.2 make	1
1.0.3 running	1
1.0.4 coremark/MHz	2
<b>2 Software</b>	<b>3</b>
2.1 Tips and Tricks	3
<b>3 DarkRISCV Application</b>	<b>5</b>
<b>4 Namespace Index</b>	<b>7</b>
4.1 Namespace List	7
<b>5 Data Structure Index</b>	<b>9</b>
5.1 Data Structures	9
<b>6 File Index</b>	<b>11</b>
6.1 File List	11
<b>7 Namespace Documentation</b>	<b>13</b>
7.1 programBoard Namespace Reference	13
7.1.1 Variable Documentation	13
7.1.1.1 data	13
7.1.1.2 ser	13
<b>8 Data Structure Documentation</b>	<b>15</b>
8.1 CORE_PORTABLE_S Struct Reference	15
8.1.1 Field Documentation	15
8.1.1.1 portable_id	15
8.2 DARKIO Struct Reference	15
8.2.1 Field Documentation	16
8.2.1.1 board_cm	16
8.2.1.2 board_id	16
8.2.1.3 core_id	16
8.2.1.4 gpio	16
8.2.1.5 gpio_ctrl	17
8.2.1.6 i2c	17
8.2.1.7 irq	17
8.2.1.8 led	17
8.2.1.9 spi	17
8.2.1.10 timer	17
8.2.1.11 timeus	17
8.2.1.12 uart	18
8.3 DARKIO::DARKUART Struct Reference	18

8.3.1 Field Documentation	18
8.3.1.1 baud	18
8.3.1.2 fifo	18
8.3.1.3 stat	18
8.4 I2C_Register Union Reference	19
8.4.1 Field Documentation	19
8.4.1.1 fields	19
8.4.1.2 raw	19
8.5 I2C_RegisterBits Struct Reference	19
8.5.1 Field Documentation	20
8.5.1.1 busy	20
8.5.1.2 data	20
8.5.1.3 n_bytes	20
8.5.1.4 nack	20
8.5.1.5 req_data	20
8.5.1.6 reserved	21
8.5.1.7 slaveAddress	21
8.5.1.8 start	21
8.5.1.9 subaddress	21
8.6 list_data_s Struct Reference	21
8.6.1 Field Documentation	21
8.6.1.1 data16	21
8.6.1.2 idx	22
8.7 list_head_s Struct Reference	22
8.7.1 Field Documentation	22
8.7.1.1 info	22
8.7.1.2 next	23
8.8 MAT_PARAMS_S Struct Reference	23
8.8.1 Field Documentation	23
8.8.1.1 A	23
8.8.1.2 B	23
8.8.1.3 C	23
8.8.1.4 N	24
8.9 RESULTS_S Struct Reference	24
8.9.1 Field Documentation	25
8.9.1.1 crc	25
8.9.1.2 crclist	25
8.9.1.3 crcmatrix	25
8.9.1.4 crcstate	25
8.9.1.5 err	25
8.9.1.6 execs	25
8.9.1.7 iterations	25

8.9.1.8 list . . . . .	26
8.9.1.9 mat . . . . .	26
8.9.1.10 memblock . . . . .	26
8.9.1.11 port . . . . .	26
8.9.1.12 seed1 . . . . .	26
8.9.1.13 seed2 . . . . .	26
8.9.1.14 seed3 . . . . .	26
8.9.1.15 size . . . . .	27
8.10 SPI_Register Union Reference . . . . .	27
8.10.1 Field Documentation . . . . .	27
8.10.1.1 fields . . . . .	27
8.10.1.2 raw . . . . .	28
8.11 SPI_RegisterBits Struct Reference . . . . .	28
8.11.1 Field Documentation . . . . .	28
8.11.1.1 data_received . . . . .	28
8.11.1.2 data_to_send . . . . .	28
8.11.1.3 empty . . . . .	28
8.11.1.4 n_bytes_received . . . . .	29
8.11.1.5 n_bytes_to_send . . . . .	29
8.11.1.6 rx_data_ready . . . . .	29
8.11.1.7 start . . . . .	29
8.11.1.8 tx_ready . . . . .	29
<b>9 File Documentation</b>	<b>31</b>
9.1 badapple/badapple.h File Reference . . . . .	31
9.1.1 Variable Documentation . . . . .	31
9.1.1.1 rle . . . . .	31
9.2 badapple/badapple.txt File Reference . . . . .	32
9.3 badapple/main.c File Reference . . . . .	32
9.3.1 Function Documentation . . . . .	32
9.3.1.1 main() . . . . .	32
9.4 blink/main.c File Reference . . . . .	32
9.4.1 Function Documentation . . . . .	33
9.4.1.1 main() . . . . .	33
9.4.1.2 wait() . . . . .	33
9.4.1.3 wait_100us() . . . . .	33
9.5 darkshell/main.c File Reference . . . . .	33
9.5.1 Function Documentation . . . . .	34
9.5.1.1 main() . . . . .	34
9.6 mandelbrot/main.c File Reference . . . . .	34
9.6.1 Function Documentation . . . . .	34
9.6.1.1 main() . . . . .	34

9.7 coremark/core_portme.c File Reference	35
9.7.1 Macro Definition Documentation	36
9.7.1.1 CLOCKS_PER_SEC	36
9.7.1.2 EE_TICKS_PER_SEC	36
9.7.1.3 GETMYTIME	36
9.7.1.4 MYTIMEDIFF	36
9.7.1.5 SAMPLE_TIME_IMPLEMENTATION	36
9.7.1.6 TIMER_RES_DIVIDER	37
9.7.2 Function Documentation	37
9.7.2.1 barebones_clock()	37
9.7.2.2 get_time()	37
9.7.2.3 portable_fini()	37
9.7.2.4 portable_init()	37
9.7.2.5 start_time()	37
9.7.2.6 stop_time()	38
9.7.2.7 time_in_secs()	38
9.7.3 Variable Documentation	38
9.7.3.1 default_num_contexts	38
9.7.3.2 seed1_volatile	38
9.7.3.3 seed2_volatile	38
9.7.3.4 seed3_volatile	38
9.7.3.5 seed4_volatile	38
9.7.3.6 seed5_volatile	39
9.7.3.7 start_time_val	39
9.7.3.8 stop_time_val	39
9.8 coremark/core_portme.h File Reference	39
9.8.1 Macro Definition Documentation	40
9.8.1.1 align_mem	41
9.8.1.2 COMPILER_FLAGS	41
9.8.1.3 COMPILER_VERSION	41
9.8.1.4 CORETIMETYPE	41
9.8.1.5 HAS_FLOAT	41
9.8.1.6 HAS_PRINTF	41
9.8.1.7 HAS_STDIO	41
9.8.1.8 HAS_TIME_H	42
9.8.1.9 MAIN_HAS_NOARGC	42
9.8.1.10 MAIN_HAS_NORETURN	42
9.8.1.11 MEM_LOCATION	42
9.8.1.12 MEM_METHOD	42
9.8.1.13 MULTITHREAD	42
9.8.1.14 NULL	42
9.8.1.15 SEED_METHOD	42

9.8.1.16 USE_CLOCK	43
9.8.1.17 USE_FORK	43
9.8.1.18 USE_PTHREAD	43
9.8.1.19 USE_SOCKET	43
9.8.1.20 VALIDATION_RUN	43
9.8.2 Typedef Documentation	43
9.8.2.1 core_portable	43
9.8.2.2 CORE_TICKS	43
9.8.2.3 ee_f32	44
9.8.2.4 ee_ptr_int	44
9.8.2.5 ee_s16	44
9.8.2.6 ee_s32	44
9.8.2.7 ee_size_t	44
9.8.2.8 ee_u16	44
9.8.2.9 ee_u32	44
9.8.2.10 ee_u8	44
9.8.3 Function Documentation	45
9.8.3.1 ee_printf()	45
9.8.3.2 portable_fini()	45
9.8.3.3 portable_init()	45
9.8.4 Variable Documentation	45
9.8.4.1 default_num_contexts	45
9.9 coremark/ee_printf.c File Reference	46
9.9.1 Macro Definition Documentation	47
9.9.1.1 HEX_PREP	47
9.9.1.2 is_digit	47
9.9.1.3 LEFT	47
9.9.1.4 PLUS	47
9.9.1.5 SIGN	47
9.9.1.6 SPACE	47
9.9.1.7 UPPERCASE	48
9.9.1.8 ZEROPAD	48
9.9.2 Function Documentation	48
9.9.2.1 eaddr()	48
9.9.2.2 ee_printf()	48
9.9.2.3 ee_vsprintf()	48
9.9.2.4 iaddr()	48
9.9.2.5 number()	49
9.9.2.6 skip_atoi()	49
9.9.2.7 strlen()	49
9.9.2.8 uart_send_char()	49
9.9.3 Variable Documentation	49

9.9.3.1 digits	49
9.9.3.2 upper_digits	49
9.10 coremark/README.md File Reference	50
9.11 darkshell/README.md File Reference	50
9.12 README.md File Reference	50
9.13 coremark/src/core_list_join.c File Reference	50
9.13.1 Typedef Documentation	51
9.13.1.1 list_cmp	51
9.13.2 Function Documentation	51
9.13.2.1 calc_func()	51
9.13.2.2 cmp_complex()	51
9.13.2.3 cmp_idx()	51
9.13.2.4 copy_info()	51
9.13.2.5 core_bench_list()	52
9.13.2.6 core_list_find()	52
9.13.2.7 core_list_init()	52
9.13.2.8 core_list_insert_new()	52
9.13.2.9 core_list_mergesort()	52
9.13.2.10 core_list_remove()	52
9.13.2.11 core_list_reverse()	53
9.13.2.12 core_list_undo_remove()	53
9.14 coremark/src/core_main.c File Reference	53
9.14.1 Macro Definition Documentation	54
9.14.1.1 get_seed	54
9.14.1.2 get_seed_32	54
9.14.2 Function Documentation	54
9.14.2.1 get_seed_args()	54
9.14.2.2 iterate()	54
9.14.2.3 main()	55
9.14.3 Variable Documentation	55
9.14.3.1 list_known_crc	55
9.14.3.2 matrix_known_crc	55
9.14.3.3 mem_name	55
9.14.3.4 state_known_crc	55
9.14.3.5 static_memblk	56
9.15 coremark/src/core_matrix.c File Reference	56
9.15.1 Macro Definition Documentation	57
9.15.1.1 bit_extract	57
9.15.1.2 matrix_big	57
9.15.1.3 matrix_clip	57
9.15.1.4 matrix_test_next	57
9.15.2 Function Documentation	57



9.15.2.1 core_bench_matrix()	57
9.15.2.2 core_init_matrix()	58
9.15.2.3 matrix_add_const()	58
9.15.2.4 matrix_mul_const()	58
9.15.2.5 matrix_mul_matrix()	58
9.15.2.6 matrix_mul_matrix_bitextract()	58
9.15.2.7 matrix_mul_vect()	59
9.15.2.8 matrix_sum()	59
9.15.2.9 matrix_test()	59
9.16 coremark/src/core_state.c File Reference	59
9.16.1 Function Documentation	60
9.16.1.1 core_bench_state()	60
9.16.1.2 core_init_state()	60
9.16.1.3 core_state_transition()	60
9.16.1.4 ee_isdigit()	61
9.16.2 Variable Documentation	61
9.16.2.1 errpat	61
9.16.2.2 floatpat	61
9.16.2.3 intpat	61
9.16.2.4 scipat	61
9.17 coremark/src/core_util.c File Reference	62
9.17.1 Function Documentation	62
9.17.1.1 check_data_types()	62
9.17.1.2 crc16()	63
9.17.1.3 crcu16()	63
9.17.1.4 crcu32()	63
9.17.1.5 crcu8()	63
9.17.1.6 get_seed_args()	63
9.17.1.7 parseval()	63
9.18 coremark/src/coremark.h File Reference	64
9.18.1 Macro Definition Documentation	65
9.18.1.1 ALL_ALGORITHMS_MASK	65
9.18.1.2 ID_LIST	66
9.18.1.3 ID_MATRIX	66
9.18.1.4 ID_STATE	66
9.18.1.5 MAIN_RETURN_TYPE	66
9.18.1.6 MAIN_RETURN_VAL	66
9.18.1.7 MATDAT_INT	66
9.18.1.8 MEM_MALLOC	66
9.18.1.9 MEM_STACK	66
9.18.1.10 MEM_STATIC	67
9.18.1.11 NUM_ALGORITHMS	67

9.18.1.12 SEED_ARG	67
9.18.1.13 SEED_FUNC	67
9.18.1.14 SEED_VOLATILE	67
9.18.1.15 TOTAL_DATA_SIZE	67
9.18.2 Typedef Documentation	67
9.18.2.1 core_results	67
9.18.2.2 core_state_e	68
9.18.2.3 list_data	68
9.18.2.4 list_head	68
9.18.2.5 mat_params	68
9.18.2.6 MATDAT	68
9.18.2.7 MATRES	68
9.18.2.8 secs_ret	68
9.18.3 Enumeration Type Documentation	68
9.18.3.1 CORE_STATE	68
9.18.4 Function Documentation	69
9.18.4.1 check_data_types()	69
9.18.4.2 core_bench_list()	69
9.18.4.3 core_bench_matrix()	69
9.18.4.4 core_bench_state()	69
9.18.4.5 core_init_matrix()	70
9.18.4.6 core_init_state()	70
9.18.4.7 core_list_init()	70
9.18.4.8 crc16()	70
9.18.4.9 crcu16()	70
9.18.4.10 crcu32()	70
9.18.4.11 crcu8()	71
9.18.4.12 get_time()	71
9.18.4.13 iterate()	71
9.18.4.14 parseval()	71
9.18.4.15 portable_free()	71
9.18.4.16 portable_malloc()	71
9.18.4.17 start_time()	71
9.18.4.18 stop_time()	72
9.18.4.19 time_in_secs()	72
9.19 darklibc/i2c.c File Reference	72
9.20 darklibc/i2c_old.c File Reference	72
9.21 darklibc/include/i2c.h File Reference	73
9.21.1 Function Documentation	73
9.21.1.1 i2cReadByte()	74
9.21.1.2 i2cSendByte()	74
9.22 darklibc/include/io.h File Reference	74

9.22.1 Macro Definition Documentation . . . . .	75
9.22.1.1 IRQ_TIMR . . . . .	75
9.22.1.2 IRQ_UART . . . . .	75
9.22.2 Function Documentation . . . . .	76
9.22.2.1 __attribute__() . . . . .	76
9.22.2.2 banner() . . . . .	76
9.22.2.3 board_name() . . . . .	76
9.22.2.4 check4rv32i() . . . . .	76
9.22.2.5 get_mepc() . . . . .	76
9.22.2.6 get_mie() . . . . .	76
9.22.2.7 get_mtvec() . . . . .	77
9.22.2.8 set_mepc() . . . . .	77
9.22.2.9 set_mie() . . . . .	77
9.22.2.10 set_mtvec() . . . . .	77
9.22.3 Variable Documentation . . . . .	77
9.22.3.1 _data . . . . .	77
9.22.3.2 _edata . . . . .	77
9.22.3.3 _etext . . . . .	77
9.22.3.4 _stack . . . . .	78
9.22.3.5 _text . . . . .	78
9.22.3.6 io . . . . .	78
9.22.3.7 kmem . . . . .	78
9.22.3.8 utimers . . . . .	78
9.23 darklibc/include/spi.h File Reference . . . . .	78
9.23.1 Function Documentation . . . . .	79
9.23.1.1 spi_disable() . . . . .	80
9.23.1.2 spi_enable() . . . . .	80
9.23.1.3 spi_init() . . . . .	80
9.23.1.4 spi_read_multiple_bytes() . . . . .	80
9.23.1.5 spi_read_single_byte() . . . . .	80
9.23.1.6 spi_send_receive_data() . . . . .	81
9.23.1.7 spi_set_clock_frequency() . . . . .	81
9.23.1.8 spi_set_data_mode() . . . . .	81
9.23.1.9 spi_transaction_single_byte() . . . . .	81
9.23.1.10 spi_write_multiple_bytes() . . . . .	82
9.23.1.11 spi_write_read_multiple_bytes() . . . . .	82
9.23.1.12 spi_write_read_single_byte() . . . . .	82
9.23.1.13 spi_write_single_byte() . . . . .	83
9.24 darklibc/include/stdio.h File Reference . . . . .	83
9.24.1 Macro Definition Documentation . . . . .	84
9.24.1.1 EBREAK . . . . .	84
9.24.1.2 EOF . . . . .	84

9.24.1.3 NUL	84
9.24.1.4 NULL	85
9.24.2 Function Documentation	85
9.24.2.1 atoi()	85
9.24.2.2 getchar()	85
9.24.2.3 gets()	85
9.24.2.4 mac()	85
9.24.2.5 memcpy()	85
9.24.2.6 memset()	86
9.24.2.7 printf()	86
9.24.2.8 putchar()	86
9.24.2.9 putd()	86
9.24.2.10 puts()	86
9.24.2.11 putstr()	86
9.24.2.12 putx()	87
9.24.2.13 strcmp()	87
9.24.2.14 strlen()	87
9.24.2.15 strncmp()	87
9.24.2.16 strtok()	87
9.24.2.17 usleep()	87
9.24.2.18 xtoi()	88
9.25 darklibc/io.c File Reference	88
9.25.1 Function Documentation	88
9.25.1.1 __attribute__()	88
9.25.1.2 board_name()	89
9.25.1.3 mac()	89
9.25.2 Variable Documentation	89
9.25.2.1 io	89
9.25.2.2 kmem	89
9.25.2.3 utimers	89
9.26 darklibc/spi.c File Reference	90
9.27 darklibc/stdio.c File Reference	90
9.27.1 Function Documentation	91
9.27.1.1 __div_mod_si3()	91
9.27.1.2 __divsi3()	91
9.27.1.3 __modsi3()	91
9.27.1.4 __mulsi3()	92
9.27.1.5 __udiv_udiv_si3()	92
9.27.1.6 __udivsi3()	92
9.27.1.7 __umodsi3()	92
9.27.1.8 __umulsi3()	92
9.27.1.9 atoi()	92

---

9.27.1.10 gets()	93
9.27.1.11 memcpy()	93
9.27.1.12 memset()	93
9.27.1.13 printf()	93
9.27.1.14 putnum()	93
9.27.1.15 puts()	93
9.27.1.16 putstr()	94
9.27.1.17 strcmp()	94
9.27.1.18 strlen()	94
9.27.1.19 strncmp()	94
9.27.1.20 strtok()	94
9.27.1.21 usleep()	94
9.27.1.22 xtoi()	95
9.28 programBoard.py File Reference	95
<b>Index</b>	<b>97</b>



# Chapter 1

## CoreMark on DarkRISCV

### 1.0.1 config example:

**\*\*Because coremark is relatively large, it is nearly 30KB (src/coremark/coremark.o) after compiling with -O2 optimization level. Therefore, the following modifications need to be made before making.\*\***

```
// rtl/config.vh
`ifndef __HARVARD__
    `define MLEN 14 // MEM[13:0] -> 16KBytes LENGTH = 0x4000
`else
    `define MLEN 15 // MEM[14:0] -> 32KBytes LENGTH = 0x8000
`endif
```

### 1.0.2 make

```
make <install> <CROSS=riscv32-unknown-elf CCPATH=/opt/riscv32-gcc/bin ARCH=rv32e APPLICATION=coremark
    HARVARD=1>
```

### 1.0.3 running

**board:scarab\_minispartan6-plus\_lx9 100MHz**

**1.0.3.0.1 GCC -O1:** boot0: text@0+13512 data@16384+2732 stack@32768 (13652 bytes free)

board: scarab minispartan6-plus lx9 (id=13)  
build: Tue, 31 May 2022 10:46:38 +0800 for rv32e

core0: darkriscv@100MHz with: rv32e

uart0: 115200 bps (div=868)

timr0: frequency=1000000Hz (io.timer=99)

CoreMark start in 24029 us.

2K performance run parameters for coremark.

CoreMark Size : 666

Total ticks : 52034539

Total time (secs): 52

Iterations/Sec : 76

Iterations : 4000

Compiler version : GCC11.1.0

Compiler flags : -O1 -DPERFORMANCE\_RUN=1

Memory location : STACK

seedcrc : 0xe9f5

[0]crclist : 0xe714

[0]cromatrix : 0x1fd7

[0]crcstate : 0x8e3a

[0]crcfinal : 0x65c5

Correct operation validated. See README.md for run and reporting rules.

CoreMark finish in 52102812 us.

**1.0.3.0.2 GCC -O2** CoreMark start in 24020 us.  
boot0: text@0+15848 data@16384+2700 stack@32768 (13684 bytes free)  
board: scarab minispartan6-plus lx9 (id=13)  
build: Mon, 30 May 2022 22:35:55 +0800 for rv32e  
core0: darkriscv@100MHz with: rv32e  
uart0: 115200 bps (div=868)  
timr0: frequency=1000000Hz (io.timer=99)  
CoreMark start in 24020 us.  
2K performance run parameters for coremark.  
CoreMark Size : 666  
Total ticks : 44265590  
Total time (secs): 44  
Iterations/Sec : 90  
Iterations : 4000  
Compiler version : GCC11.1.0  
Compiler flags : -O2 -DPERFORMANCE\_RUN=1  
Memory location : STACK  
seedcrc : 0xe9f5  
[0]crclist : 0xe714  
[0]crcmatrix : 0x1fd7  
[0]crcstate : 0x8e3a  
[0]crcfinal : 0x65c5  
Correct operation validated. See README.md for run and reporting rules.  
CoreMark finish in 44333828 us.

## 1.0.4 coremark/MHz

How to calculate the coremark score i.e. coremark/MHz ?

coremark code from [coremark@b24e397](#).



## Chapter 2

# Software

This directory provides support for DarkRISCV software.

The software is 100% written in C language, is compiled by the GCC and lots of support files (elf, assembler, maps, etc) are produced in order to help debug and/or study the RISCV architecture.

### 2.1 Tips and Tricks

As long the FPGA has few BRAMs available, we need write the software thinking about preserve memory space. However, sometimes the code does not help us... anyway, is possible check the memory space used by each function in the firmware with the following script:

```
awk '{
    if($0~/>:/) PTR=$2
    else
    if($0~/:/) DB[PTR]++
} END {
    for(i in DB) print DB[i],i
}' src/darksocv.lst | sort -nr
```

The script will calculate how many instructions each function needs and will print and sort it, producing something like this:

```
456 <main>:
149 <putdx>:
95 <printf>:
62 <strtok>:
62 <gets>:
59 <banner>:
47 <board_name>:
42 <irq_handler>:
...
```

So, with those information, is possible try optimize better the large functions.

TODO:

- add a gdb-stub in order to support UART debug
- add a SREC decoder in order to support application upload via UART
- split the "stdio" in other files
- add more libc features and optimize the existing features



## Chapter 3

# DarkRISCV Application

- darklibc: DarkRISCV C librarys
- darksocv: DarkRISCV test code
- coremark: CoreMark benchmark in DarkRISCV

**For different applications, you need to modify the MLEN in `../rtl/config.vh`**



## Chapter 4

# Namespace Index

### 4.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">programBoard</a> . . . . .	13
--	----



## Chapter 5

# Data Structure Index

### 5.1 Data Structures

Here are the data structures with brief descriptions:

CORE_PORTABLE_S . . . . .	15
DARKIO . . . . .	15
DARKIO::DARKUART . . . . .	18
I2C_Register . . . . .	19
I2C_RegisterBits . . . . .	19
list_data_s . . . . .	21
list_head_s . . . . .	22
MAT_PARAMS_S . . . . .	23
RESULTS_S . . . . .	24
SPI_Register . . . . .	27
SPI_RegisterBits . . . . .	28





## Chapter 6

# File Index

### 6.1 File List

Here is a list of all files with brief descriptions:

<a href="#">programBoard.py</a>	95
<a href="#">badapple/badapple.h</a>	31
<a href="#">badapple/main.c</a>	32
<a href="#">blink/main.c</a>	32
<a href="#">coremark/core_portme.c</a>	35
<a href="#">coremark/core_portme.h</a>	39
<a href="#">coremark/ee_printf.c</a>	46
<a href="#">coremark/src/core_list_join.c</a>	50
<a href="#">coremark/src/core_main.c</a>	53
<a href="#">coremark/src/core_matrix.c</a>	56
<a href="#">coremark/src/core_state.c</a>	59
<a href="#">coremark/src/core_util.c</a>	62
<a href="#">coremark/src/coremark.h</a>	64
<a href="#">darklibc/i2c.c</a>	72
<a href="#">darklibc/i2c_old.c</a>	72
<a href="#">darklibc/io.c</a>	88
<a href="#">darklibc/spi.c</a>	90
<a href="#">darklibc/stdio.c</a>	90
<a href="#">darklibc/include/i2c.h</a>	73
<a href="#">darklibc/include/io.h</a>	74
<a href="#">darklibc/include/spi.h</a>	78
<a href="#">darklibc/include/stdio.h</a>	83
<a href="#">darkshell/main.c</a>	33
<a href="#">mandelbrot/main.c</a>	34



## Chapter 7

# Namespace Documentation

### 7.1 programBoard Namespace Reference

#### Variables

- `ser` = `serial.Serial('COM4', 115200)`
- `data` = `f.read()`

#### 7.1.1 Variable Documentation

##### 7.1.1.1 `data`

```
programBoard.data = f.read()
```

##### 7.1.1.2 `ser`

```
programBoard.ser = serial.Serial('COM4', 115200)
```



## Chapter 8

# Data Structure Documentation

### 8.1 CORE\_PORTABLE\_S Struct Reference

```
#include <core_portme.h>
```

#### Data Fields

- [ee\\_u8 portable\\_id](#)

#### 8.1.1 Field Documentation

##### 8.1.1.1 portable\_id

```
ee_u8 CORE_PORTABLE_S::portable_id
```

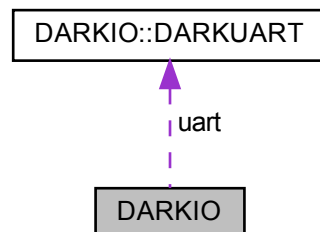
The documentation for this struct was generated from the following file:

- [coremark/core\\_portme.h](#)

### 8.2 DARKIO Struct Reference

```
#include <io.h>
```

Collaboration diagram for DARKIO:



## Data Structures

- struct [DARKUART](#)

## Data Fields

- unsigned char [board\\_id](#)
- unsigned char [board\\_cm](#)
- unsigned char [core\\_id](#)
- unsigned char [irq](#)
- struct [DARKIO::DARKUART](#) [uart](#)
- unsigned short [led](#)
- unsigned short [gpio](#)
- unsigned int [timer](#)
- unsigned int [timeus](#)
- unsigned int [spi](#)
- unsigned int [i2c](#)
- unsigned short [gpio\\_ctrl](#)

### 8.2.1 Field Documentation

#### 8.2.1.1 [board\\_cm](#)

`unsigned char DARKIO::board_cm`

#### 8.2.1.2 [board\\_id](#)

`unsigned char DARKIO::board_id`

#### 8.2.1.3 [core\\_id](#)

`unsigned char DARKIO::core_id`

#### 8.2.1.4 [gpio](#)

`unsigned short DARKIO::gpio`

### 8.2.1.5 gpio\_ctrl

```
unsigned short DARKIO::gpio_ctrl
```

### 8.2.1.6 i2c

```
unsigned int DARKIO::i2c
```

### 8.2.1.7 irq

```
unsigned char DARKIO::irq
```

### 8.2.1.8 led

```
unsigned short DARKIO::led
```

### 8.2.1.9 spi

```
unsigned int DARKIO::spi
```

### 8.2.1.10 timer

```
unsigned int DARKIO::timer
```

### 8.2.1.11 timeus

```
unsigned int DARKIO::timeus
```

#### 8.2.1.12 uart

```
struct DARKIO::DARKUART DARKIO::uart
```

The documentation for this struct was generated from the following file:

- [darklibc/include/io.h](#)

### 8.3 DARKIO::DARKUART Struct Reference

```
#include <io.h>
```

#### Data Fields

- unsigned char [stat](#)
- unsigned char [fifo](#)
- unsigned short [baud](#)

#### 8.3.1 Field Documentation

##### 8.3.1.1 baud

```
unsigned short DARKIO::DARKUART::baud
```

##### 8.3.1.2 fifo

```
unsigned char DARKIO::DARKUART::fifo
```

##### 8.3.1.3 stat

```
unsigned char DARKIO::DARKUART::stat
```

The documentation for this struct was generated from the following file:

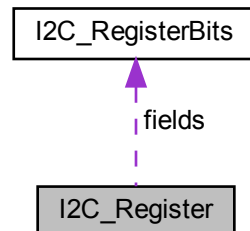
- [darklibc/include/io.h](#)



## 8.4 I2C\_Register Union Reference

```
#include <i2c.h>
```

Collaboration diagram for I2C\_Register:



### Data Fields

- `uint32_t raw`
- `I2C_RegisterBits fields`

### 8.4.1 Field Documentation

#### 8.4.1.1 fields

```
I2C_RegisterBits I2C_Register::fields
```

#### 8.4.1.2 raw

```
uint32_t I2C_Register::raw
```

The documentation for this union was generated from the following file:

- `darklibc/include/i2c.h`

## 8.5 I2C\_RegisterBits Struct Reference

```
#include <i2c.h>
```

## Data Fields

- unsigned int [data](#): 8
- unsigned int [subaddress](#): 8
- unsigned int [slaveAddress](#): 8
- unsigned int [n\\_bytes](#): 3
- unsigned int [start](#): 1
- unsigned int [nack](#): 1
- unsigned int [req\\_data](#): 1
- unsigned int [busy](#): 1
- unsigned int [reserved](#): 1

## 8.5.1 Field Documentation

### 8.5.1.1 busy

```
unsigned int I2C_RegisterBits::busy
```

### 8.5.1.2 data

```
unsigned int I2C_RegisterBits::data
```

### 8.5.1.3 n\_bytes

```
unsigned int I2C_RegisterBits::n_bytes
```

### 8.5.1.4 nack

```
unsigned int I2C_RegisterBits::nack
```

### 8.5.1.5 req\_data

```
unsigned int I2C_RegisterBits::req_data
```

#### 8.5.1.6 reserved

```
unsigned int I2C_RegisterBits::reserved
```

#### 8.5.1.7 slaveAddress

```
unsigned int I2C_RegisterBits::slaveAddress
```

#### 8.5.1.8 start

```
unsigned int I2C_RegisterBits::start
```

#### 8.5.1.9 subaddress

```
unsigned int I2C_RegisterBits::subaddress
```

The documentation for this struct was generated from the following file:

- [darklibc/include/i2c.h](#)

## 8.6 list\_data\_s Struct Reference

```
#include <coremark.h>
```

### Data Fields

- [ee\\_s16 data16](#)
- [ee\\_s16 idx](#)

### 8.6.1 Field Documentation

#### 8.6.1.1 data16

```
ee\_s16 list_data_s::data16
```

### 8.6.1.2 idx

```
ee_s16 list_data_s::idx
```

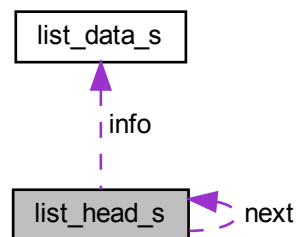
The documentation for this struct was generated from the following file:

- [coremark/src/coremark.h](#)

## 8.7 list\_head\_s Struct Reference

```
#include <coremark.h>
```

Collaboration diagram for list\_head\_s:



### Data Fields

- struct [list\\_head\\_s](#) \* [next](#)
- struct [list\\_data\\_s](#) \* [info](#)

### 8.7.1 Field Documentation

#### 8.7.1.1 info

```
struct list\_data\_s* list_head_s::info
```

### 8.7.1.2 next

```
struct list_head_s* list_head_s::next
```

The documentation for this struct was generated from the following file:

- [coremark/src/coremark.h](#)

## 8.8 MAT\_PARAMS\_S Struct Reference

```
#include <coremark.h>
```

### Data Fields

- `int N`
- `MATDAT * A`
- `MATDAT * B`
- `MATRES * C`

### 8.8.1 Field Documentation

#### 8.8.1.1 A

```
MATDAT* MAT_PARAMS_S::A
```

#### 8.8.1.2 B

```
MATDAT* MAT_PARAMS_S::B
```

#### 8.8.1.3 C

```
MATRES* MAT_PARAMS_S::C
```

#### 8.8.1.4 N

```
int MAT_PARAMS_S::N
```

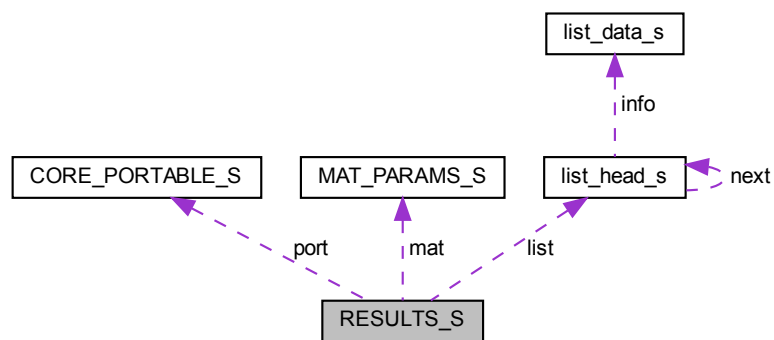
The documentation for this struct was generated from the following file:

- [coremark/src/coremark.h](#)

## 8.9 RESULTS\_S Struct Reference

```
#include <coremark.h>
```

Collaboration diagram for RESULTS\_S:



### Data Fields

- [ee\\_s16 seed1](#)
- [ee\\_s16 seed2](#)
- [ee\\_s16 seed3](#)
- [void \\* memblock \[4\]](#)
- [ee\\_u32 size](#)
- [ee\\_u32 iterations](#)
- [ee\\_u32 execs](#)
- [struct list\\_head\\_s \\* list](#)
- [mat\\_params mat](#)
- [ee\\_u16 crc](#)
- [ee\\_u16 crclist](#)
- [ee\\_u16 crcmatrix](#)
- [ee\\_u16 crcstate](#)
- [ee\\_s16 err](#)
- [core\\_portable port](#)

## 8.9.1 Field Documentation

### 8.9.1.1 crc

ee\_u16 RESULTS\_S::crc

### 8.9.1.2 crclist

ee\_u16 RESULTS\_S::crclist

### 8.9.1.3 crcmatrix

ee\_u16 RESULTS\_S::crcmatrix

### 8.9.1.4 crcstate

ee\_u16 RESULTS\_S::crcstate

### 8.9.1.5 err

ee\_s16 RESULTS\_S::err

### 8.9.1.6 execs

ee\_u32 RESULTS\_S::execs

### 8.9.1.7 iterations

ee\_u32 RESULTS\_S::iterations

#### 8.9.1.8 list

```
struct list_head_s* RESULTS_S::list
```

#### 8.9.1.9 mat

```
mat_params RESULTS_S::mat
```

#### 8.9.1.10 memblock

```
void* RESULTS_S::memblock[4]
```

#### 8.9.1.11 port

```
core_portable RESULTS_S::port
```

#### 8.9.1.12 seed1

```
ee_s16 RESULTS_S::seed1
```

#### 8.9.1.13 seed2

```
ee_s16 RESULTS_S::seed2
```

#### 8.9.1.14 seed3

```
ee_s16 RESULTS_S::seed3
```



### 8.9.1.15 size

```
ee_u32 RESULTS_S::size
```

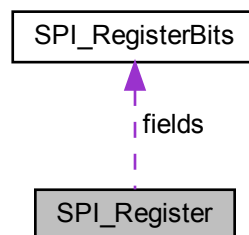
The documentation for this struct was generated from the following file:

- [coremark/src/coremark.h](#)

## 8.10 SPI\_Register Union Reference

```
#include <spi.h>
```

Collaboration diagram for SPI\_Register:



### Data Fields

- `uint32_t` [raw](#)
- [SPI\\_RegisterBits](#) `fields`

### 8.10.1 Field Documentation

#### 8.10.1.1 fields

```
SPI_RegisterBits SPI_Register::fields
```

### 8.10.1.2 raw

```
uint32_t SPI_Register::raw
```

The documentation for this union was generated from the following file:

- [darklibc/include/spi.h](#)

## 8.11 SPI\_RegisterBits Struct Reference

```
#include <spi.h>
```

### Data Fields

- unsigned int [data\\_to\\_send](#): 8
- unsigned int [n\\_bytes\\_to\\_send](#): 3
- unsigned int [start](#): 1
- unsigned int [n\\_bytes\\_received](#): 2
- unsigned int [rx\\_data\\_ready](#): 1
- unsigned int [data\\_received](#): 8
- unsigned int [tx\\_ready](#): 1
- unsigned int [empty](#): 8

### 8.11.1 Field Documentation

#### 8.11.1.1 data\_received

```
unsigned int SPI_RegisterBits::data_received
```

#### 8.11.1.2 data\_to\_send

```
unsigned int SPI_RegisterBits::data_to_send
```

#### 8.11.1.3 empty

```
unsigned int SPI_RegisterBits::empty
```

#### 8.11.1.4 n\_bytes\_received

```
unsigned int SPI_RegisterBits::n_bytes_received
```

#### 8.11.1.5 n\_bytes\_to\_send

```
unsigned int SPI_RegisterBits::n_bytes_to_send
```

#### 8.11.1.6 rx\_data\_ready

```
unsigned int SPI_RegisterBits::rx_data_ready
```

#### 8.11.1.7 start

```
unsigned int SPI_RegisterBits::start
```

#### 8.11.1.8 tx\_ready

```
unsigned int SPI_RegisterBits::tx_ready
```

The documentation for this struct was generated from the following file:

- [darklibc/include/spi.h](#)

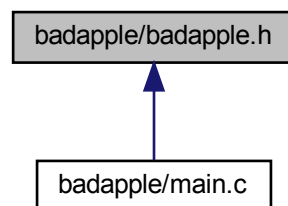


## Chapter 9

# File Documentation

### 9.1 badapple/badapple.h File Reference

This graph shows which files directly or indirectly include this file:



#### Variables

- unsigned char `rle` []

#### 9.1.1 Variable Documentation

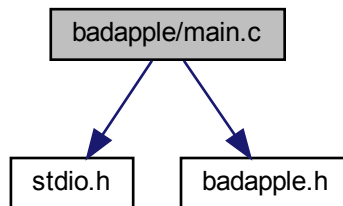
##### 9.1.1.1 `rle`

```
unsigned char rle[]
```

## 9.2 badapple/badapple.txt File Reference

## 9.3 badapple/main.c File Reference

```
#include <stdio.h>
#include "badapple.h"
Include dependency graph for main.c:
```



### Functions

- int `main` ()

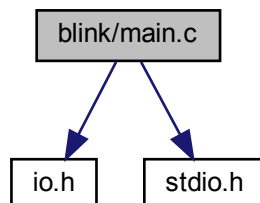
### 9.3.1 Function Documentation

#### 9.3.1.1 `main()`

```
int main ( )
```

## 9.4 blink/main.c File Reference

```
#include <io.h>
#include <stdio.h>
Include dependency graph for main.c:
```



## Functions

- int [main](#) (void)
- void [wait](#) (int cycles)
- void [wait\\_100us](#) ()

### 9.4.1 Function Documentation

#### 9.4.1.1 main()

```
int main (  
    void )
```

#### 9.4.1.2 wait()

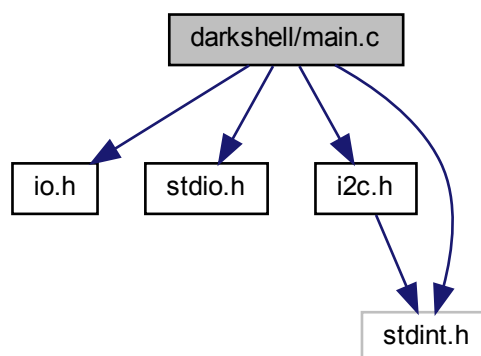
```
void wait (  
    int cycles )
```

#### 9.4.1.3 wait\_100us()

```
void wait_100us ( )
```

## 9.5 darkshell/main.c File Reference

```
#include <io.h>  
#include <stdio.h>  
#include <i2c.h>  
#include <stdint.h>  
Include dependency graph for main.c:
```



## Functions

- int `main` (void)

### 9.5.1 Function Documentation

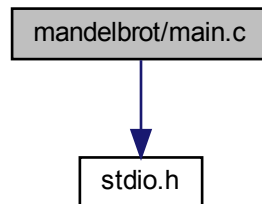
#### 9.5.1.1 `main()`

```
int main (  
    void )
```

## 9.6 mandelbrot/main.c File Reference

```
#include <stdio.h>
```

Include dependency graph for main.c:



## Functions

- int `main` ()

### 9.6.1 Function Documentation

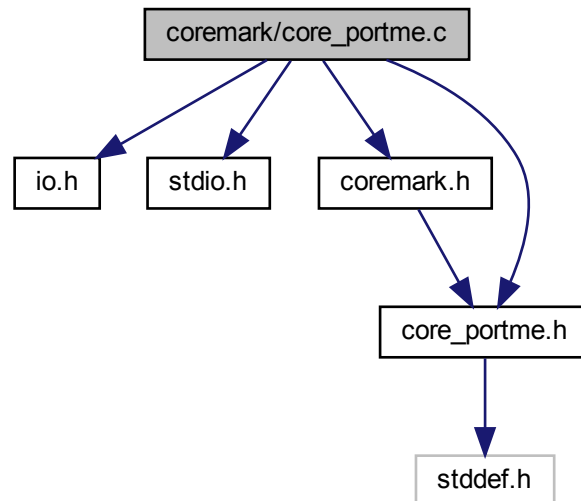
#### 9.6.1.1 `main()`

```
int main ( )
```



## 9.7 coremark/core\_portme.c File Reference

```
#include <io.h>
#include <stdio.h>
#include "coremark.h"
#include "core_portme.h"
Include dependency graph for core_portme.c:
```



### Macros

- `#define CLOCKS_PER_SEC 1000000`
- `#define GETMYTIME(_t) (*_t = barebones_clock())`
- `#define MYTIMEDIFF(fin, ini) ((fin) - (ini))`
- `#define TIMER_RES_DIVIDER 1`
- `#define SAMPLE_TIME_IMPLEMENTATION 1`
- `#define EE_TICKS_PER_SEC (CLOCKS_PER_SEC / TIMER_RES_DIVIDER)`

### Functions

- `CORETIMETYPE barebones_clock ()`
- `void start_time (void)`
- `void stop_time (void)`
- `CORE_TICKS get_time (void)`
- `secs_ret time_in_secs (CORE_TICKS ticks)`
- `void portable_init (core_portable *p, int *argc, char *argv[])`
- `void portable_fini (core_portable *p)`

## Variables

- volatile `ee_s32 seed1_volatile` = 0x3415
- volatile `ee_s32 seed2_volatile` = 0x3415
- volatile `ee_s32 seed3_volatile` = 0x66
- volatile `ee_s32 seed4_volatile` = ITERATIONS
- volatile `ee_s32 seed5_volatile` = 0
- static `CORETIMETYPE start_time_val`
- static `CORETIMETYPE stop_time_val`
- `ee_u32 default_num_contexts` = 1

## 9.7.1 Macro Definition Documentation

### 9.7.1.1 CLOCKS\_PER\_SEC

```
#define CLOCKS_PER_SEC 1000000
```

### 9.7.1.2 EE\_TICKS\_PER\_SEC

```
#define EE_TICKS_PER_SEC (CLOCKS_PER_SEC / TIMER_RES_DIVIDER)
```

### 9.7.1.3 GETMYTIME

```
#define GETMYTIME(  
    _t ) (*_t = barebones_clock())
```

### 9.7.1.4 MYTIMEDIFF

```
#define MYTIMEDIFF(  
    fin,  
    ini ) ((fin) - (ini))
```

### 9.7.1.5 SAMPLE\_TIME\_IMPLEMENTATION

```
#define SAMPLE_TIME_IMPLEMENTATION 1
```

### 9.7.1.6 TIMER\_RES\_DIVIDER

```
#define TIMER_RES_DIVIDER 1
```

## 9.7.2 Function Documentation

### 9.7.2.1 barebones\_clock()

```
CORETIMETYPE barebones_clock ( )
```

### 9.7.2.2 get\_time()

```
CORE_TICKS get_time (
    void )
```

### 9.7.2.3 portable\_fini()

```
void portable_fini (
    core_portable * p )
```

### 9.7.2.4 portable\_init()

```
void portable_init (
    core_portable * p,
    int * argc,
    char * argv[ ] )
```

### 9.7.2.5 start\_time()

```
void start_time (
    void )
```

#### 9.7.2.6 stop\_time()

```
void stop_time (
    void )
```

#### 9.7.2.7 time\_in\_secs()

```
secs_ret time_in_secs (
    CORE_TICKS ticks )
```

### 9.7.3 Variable Documentation

#### 9.7.3.1 default\_num\_contexts

```
ee_u32 default_num_contexts = 1
```

#### 9.7.3.2 seed1\_volatile

```
volatile ee_s32 seed1_volatile = 0x3415
```

#### 9.7.3.3 seed2\_volatile

```
volatile ee_s32 seed2_volatile = 0x3415
```

#### 9.7.3.4 seed3\_volatile

```
volatile ee_s32 seed3_volatile = 0x66
```

#### 9.7.3.5 seed4\_volatile

```
volatile ee_s32 seed4_volatile = ITERATIONS
```

### 9.7.3.6 seed5\_volatile

```
volatile ee_s32 seed5_volatile = 0
```

### 9.7.3.7 start\_time\_val

```
CORETIMETYPE start_time_val [static]
```

Define Host specific (POSIX), or target specific global time variables.

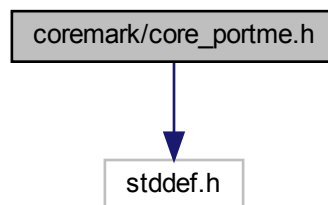
### 9.7.3.8 stop\_time\_val

```
CORETIMETYPE stop_time_val [static]
```

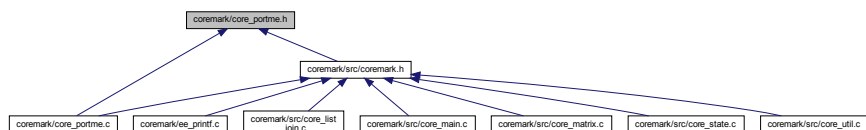
## 9.8 coremark/core\_portme.h File Reference

```
#include <stddef.h>
```

Include dependency graph for core\_portme.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- struct [CORE\\_PORTABLE\\_S](#)

## Macros

- `#define HAS_FLOAT 0`
- `#define HAS_TIME_H 0`
- `#define USE_CLOCK 0`
- `#define HAS_STDIO 0`
- `#define HAS_PRINTF 0`
- `#define COMPILER_VERSION "Please put compiler version here (e.g. gcc 4.1)"`
- `#define COMPILER_FLAGS FLAGS_STR /* "Please put compiler flags here (e.g. -o3)" */`
- `#define MEM_LOCATION "STACK"`
- `#define NULL ((void *)0)`
- `#define align_mem(x) (void *) (4 + (((ee_ptr_int)(x)-1) & ~3))`
- `#define CORETIMETYPE ee_u32`
- `#define SEED_METHOD SEED_VOLATILE`
- `#define MEM_METHOD MEM_STACK`
- `#define MULTITHREAD 1`
- `#define USE_PTHREAD 0`
- `#define USE_FORK 0`
- `#define USE_SOCKET 0`
- `#define MAIN_HAS_NOARGC 1`
- `#define MAIN_HAS_NORETURN 0`
- `#define VALIDATION_RUN 1`

## Typedefs

- `typedef signed short ee_s16`
- `typedef unsigned short ee_u16`
- `typedef signed int ee_s32`
- `typedef double ee_f32`
- `typedef unsigned char ee_u8`
- `typedef unsigned int ee_u32`
- `typedef ee_u32 ee_ptr_int`
- `typedef size_t ee_size_t`
- `typedef ee_u32 CORE_TICKS`
- `typedef struct CORE_PORTABLE_S core_portable`

## Functions

- `void portable_init (core_portable *p, int *argc, char *argv[])`
- `void portable_fini (core_portable *p)`
- `int ee_printf (const char *fmt,...)`

## Variables

- `ee_u32 default_num_contexts`

### 9.8.1 Macro Definition Documentation

### 9.8.1.1 align\_mem

```
#define align_mem(  
    x ) (void *) (4 + (((ee_ptr_int) (x)-1) & ~3))
```

### 9.8.1.2 COMPILER\_FLAGS

```
#define COMPILER_FLAGS FLAGS_STR /* "Please put compiler flags here (e.g. -o3)" */
```

### 9.8.1.3 COMPILER\_VERSION

```
#define COMPILER_VERSION "Please put compiler version here (e.g. gcc 4.1)"
```

### 9.8.1.4 CORETIMETYPE

```
#define CORETIMETYPE ee_u32
```

### 9.8.1.5 HAS\_FLOAT

```
#define HAS_FLOAT 0
```

### 9.8.1.6 HAS\_PRINTF

```
#define HAS_PRINTF 0
```

### 9.8.1.7 HAS\_STDIO

```
#define HAS_STDIO 0
```

#### 9.8.1.8 HAS\_TIME\_H

```
#define HAS_TIME_H 0
```

#### 9.8.1.9 MAIN\_HAS\_NOARGC

```
#define MAIN_HAS_NOARGC 1
```

#### 9.8.1.10 MAIN\_HAS\_NORETURN

```
#define MAIN_HAS_NORETURN 0
```

#### 9.8.1.11 MEM\_LOCATION

```
#define MEM_LOCATION "STACK"
```

#### 9.8.1.12 MEM\_METHOD

```
#define MEM_METHOD MEM\_STACK
```

#### 9.8.1.13 MULTITHREAD

```
#define MULTITHREAD 1
```

#### 9.8.1.14 NULL

```
#define NULL ((void *)0)
```

#### 9.8.1.15 SEED\_METHOD

```
#define SEED_METHOD SEED\_VOLATILE
```



#### 9.8.1.16 USE\_CLOCK

```
#define USE_CLOCK 0
```

#### 9.8.1.17 USE\_FORK

```
#define USE_FORK 0
```

#### 9.8.1.18 USE\_PTHREAD

```
#define USE_PTHREAD 0
```

#### 9.8.1.19 USE\_SOCKET

```
#define USE_SOCKET 0
```

#### 9.8.1.20 VALIDATION\_RUN

```
#define VALIDATION_RUN 1
```

### 9.8.2 Typedef Documentation

#### 9.8.2.1 core\_portable

```
typedef struct CORE_PORTABLE_S core_portable
```

#### 9.8.2.2 CORE\_TICKS

```
typedef ee_u32 CORE_TICKS
```

#### 9.8.2.3 ee\_f32

```
typedef double ee_f32
```

#### 9.8.2.4 ee\_ptr\_int

```
typedef ee_u32 ee_ptr_int
```

#### 9.8.2.5 ee\_s16

```
typedef signed short ee_s16
```

#### 9.8.2.6 ee\_s32

```
typedef signed int ee_s32
```

#### 9.8.2.7 ee\_size\_t

```
typedef size_t ee_size_t
```

#### 9.8.2.8 ee\_u16

```
typedef unsigned short ee_u16
```

#### 9.8.2.9 ee\_u32

```
typedef unsigned int ee_u32
```

#### 9.8.2.10 ee\_u8

```
typedef unsigned char ee_u8
```

## 9.8.3 Function Documentation

### 9.8.3.1 ee\_printf()

```
int ee_printf (
    const char * fmt,
    ... )
```

### 9.8.3.2 portable\_fini()

```
void portable_fini (
    core_portable * p )
```

### 9.8.3.3 portable\_init()

```
void portable_init (
    core_portable * p,
    int * argc,
    char * argv[] )
```

## 9.8.4 Variable Documentation

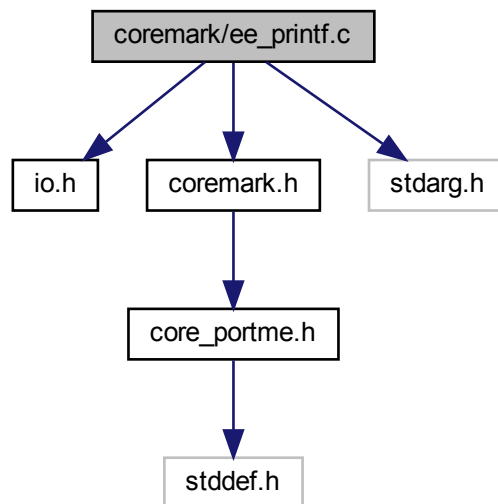
### 9.8.4.1 default\_num\_contexts

ee\_u32 default\_num\_contexts

## 9.9 coremark/ee\_printf.c File Reference

```
#include <io.h>
#include <coremark.h>
#include <stdarg.h>
```

Include dependency graph for ee\_printf.c:



### Macros

- `#define ZEROPAD (1 << 0) /* Pad with zero */`
- `#define SIGN (1 << 1) /* Unsigned/signed long */`
- `#define PLUS (1 << 2) /* Show plus */`
- `#define SPACE (1 << 3) /* Spacer */`
- `#define LEFT (1 << 4) /* Left justified */`
- `#define HEX_PREP (1 << 5) /* 0x */`
- `#define UPPERCASE (1 << 6) /* 'ABCDEF' */`
- `#define is_digit(c) ((c) >= '0' && (c) <= '9')`

### Functions

- static `ee_size_t strlen` (const char \*s, ee\_size\_t count)
- static int `skip_atoi` (const char \*\*s)
- static char \* `number` (char \*str, long num, int base, int size, int precision, int type)
- static char \* `eaddr` (char \*str, unsigned char \*addr, int size, int precision, int type)
- static char \* `iaddr` (char \*str, unsigned char \*addr, int size, int precision, int type)
- static int `ee_vsprintf` (char \*buf, const char \*fmt, va\_list args)
- void `uart_send_char` (char c)
- int `ee_printf` (const char \*fmt,...)

## Variables

- static char \* `digits` = "0123456789abcdefghijklmnopqrstuvwxyz"
- static char \* `upper_digits` = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"

## 9.9.1 Macro Definition Documentation

### 9.9.1.1 HEX\_PREP

```
#define HEX_PREP (1 << 5) /* 0x */
```

### 9.9.1.2 is\_digit

```
#define is_digit(  
    c ) ((c) >= '0' && (c) <= '9')
```

### 9.9.1.3 LEFT

```
#define LEFT (1 << 4) /* Left justified */
```

### 9.9.1.4 PLUS

```
#define PLUS (1 << 2) /* Show plus */
```

### 9.9.1.5 SIGN

```
#define SIGN (1 << 1) /* Unsigned/signed long */
```

### 9.9.1.6 SPACE

```
#define SPACE (1 << 3) /* Spacer */
```

### 9.9.1.7 UPPERCASE

```
#define UPPERCASE (1 << 6) /* 'ABCDEF' */
```

### 9.9.1.8 ZEROPAD

```
#define ZEROPAD (1 << 0) /* Pad with zero */
```

## 9.9.2 Function Documentation

### 9.9.2.1 eaddr()

```
static char* eaddr (  
    char * str,  
    unsigned char * addr,  
    int size,  
    int precision,  
    int type ) [static]
```

### 9.9.2.2 ee\_printf()

```
int ee_printf (  
    const char * fmt,  
    ... )
```

### 9.9.2.3 ee\_vsprintf()

```
static int ee_vsprintf (  
    char * buf,  
    const char * fmt,  
    va_list args ) [static]
```

### 9.9.2.4 iaddr()

```
static char* iaddr (  
    char * str,  
    unsigned char * addr,  
    int size,  
    int precision,  
    int type ) [static]
```

#### 9.9.2.5 number()

```
static char* number (
    char * str,
    long num,
    int base,
    int size,
    int precision,
    int type ) [static]
```

#### 9.9.2.6 skip\_atoi()

```
static int skip_atoi (
    const char ** s ) [static]
```

#### 9.9.2.7 strlen()

```
static ee_size_t strlen (
    const char * s,
    ee_size_t count ) [static]
```

#### 9.9.2.8 uart\_send\_char()

```
void uart_send_char (
    char c )
```

### 9.9.3 Variable Documentation

#### 9.9.3.1 digits

```
char* digits = "0123456789abcdefghijklmnopqrstuvwxyz" [static]
```

#### 9.9.3.2 upper\_digits

```
char* upper_digits = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ" [static]
```

## 9.10 coremark/README.md File Reference

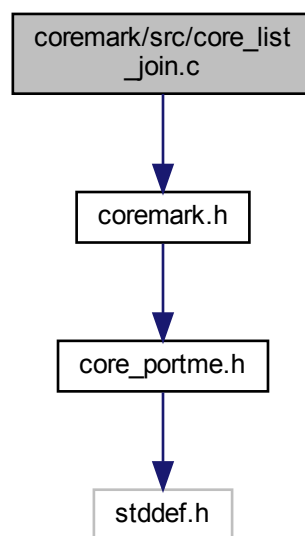
## 9.11 darkshell/README.md File Reference

## 9.12 README.md File Reference

## 9.13 coremark/src/core\_list\_join.c File Reference

```
#include "coremark.h"
```

Include dependency graph for core\_list\_join.c:



### Typedefs

- typedef ee\_s32(\* list\_cmp) (list\_data \*a, list\_data \*b, core\_results \*res)

### Functions

- list\_head \* core\_list\_find (list\_head \*list, list\_data \*info)
- list\_head \* core\_list\_reverse (list\_head \*list)
- list\_head \* core\_list\_remove (list\_head \*item)
- list\_head \* core\_list\_undo\_remove (list\_head \*item\_removed, list\_head \*item\_modified)
- list\_head \* core\_list\_insert\_new (list\_head \*insert\_point, list\_data \*info, list\_head \*\*memblock, list\_data \*\*datablock, list\_head \*memblock\_end, list\_data \*datablock\_end)
- list\_head \* core\_list\_mergesort (list\_head \*list, list\_cmp cmp, core\_results \*res)
- ee\_s16 calc\_func (ee\_s16 \*pdata, core\_results \*res)
- ee\_s32 cmp\_complex (list\_data \*a, list\_data \*b, core\_results \*res)
- ee\_s32 cmp\_idx (list\_data \*a, list\_data \*b, core\_results \*res)
- void copy\_info (list\_data \*to, list\_data \*from)
- ee\_u16 core\_bench\_list (core\_results \*res, ee\_s16 finder\_idx)
- list\_head \* core\_list\_init (ee\_u32 blksize, list\_head \*memblock, ee\_s16 seed)



## 9.13.1 Typedef Documentation

### 9.13.1.1 list\_cmp

```
typedef ee_s32 (* list_cmp) (list_data *a, list_data *b, core_results *res)
```

## 9.13.2 Function Documentation

### 9.13.2.1 calc\_func()

```
ee_s16 calc_func (  
    ee_s16 * pdata,  
    core_results * res )
```

### 9.13.2.2 cmp\_complex()

```
ee_s32 cmp_complex (  
    list_data * a,  
    list_data * b,  
    core_results * res )
```

### 9.13.2.3 cmp\_idx()

```
ee_s32 cmp_idx (  
    list_data * a,  
    list_data * b,  
    core_results * res )
```

### 9.13.2.4 copy\_info()

```
void copy_info (  
    list_data * to,  
    list_data * from )
```

#### 9.13.2.5 core\_bench\_list()

```
ee_u16 core_bench_list (
    core_results * res,
    ee_sl6 finder_idx )
```

#### 9.13.2.6 core\_list\_find()

```
list_head * core_list_find (
    list_head * list,
    list_data * info )
```

#### 9.13.2.7 core\_list\_init()

```
list_head* core_list_init (
    ee_u32 blksize,
    list_head * memblock,
    ee_sl6 seed )
```

#### 9.13.2.8 core\_list\_insert\_new()

```
list_head * core_list_insert_new (
    list_head * insert_point,
    list_data * info,
    list_head ** memblock,
    list_data ** datablock,
    list_head * memblock_end,
    list_data * datablock_end )
```

#### 9.13.2.9 core\_list\_mergesort()

```
list_head * core_list_mergesort (
    list_head * list,
    list_cmp cmp,
    core_results * res )
```

#### 9.13.2.10 core\_list\_remove()

```
list_head * core_list_remove (
    list_head * item )
```

### 9.13.2.11 core\_list\_reverse()

```
list_head * core_list_reverse (
    list_head * list )
```

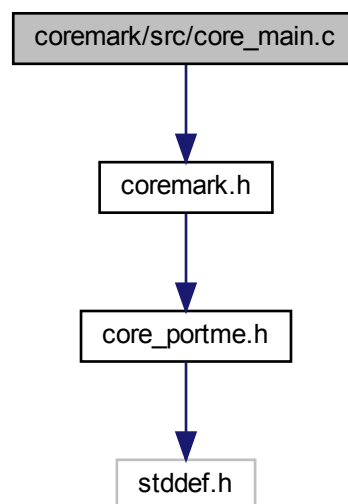
### 9.13.2.12 core\_list\_undo\_remove()

```
list_head * core_list_undo_remove (
    list_head * item_removed,
    list_head * item_modified )
```

## 9.14 coremark/src/core\_main.c File Reference

```
#include "coremark.h"
```

Include dependency graph for core\_main.c:



### Macros

- #define `get_seed(x)` (`ee_s16`) `get_seed_args(x, argc, argv)`
- #define `get_seed_32(x)` `get_seed_args(x, argc, argv)`

### Functions

- void \* `iterate` (void \*pres)
- `ee_s32` `get_seed_args` (int i, int argc, char \*argv[])
- `MAIN_RETURN_TYPE` `main` (int argc, char \*argv[])

## Variables

- static `ee_u16 list_known_crc []`
- static `ee_u16 matrix_known_crc []`
- static `ee_u16 state_known_crc []`
- `ee_u8 static_memblk [TOTAL_DATA_SIZE]`
- `char * mem_name [3] = { "Static", "Heap", "Stack" }`

## 9.14.1 Macro Definition Documentation

### 9.14.1.1 `get_seed`

```
#define get_seed(  
    x ) (ee_s16) get_seed_args(x, argc, argv)
```

### 9.14.1.2 `get_seed_32`

```
#define get_seed_32(  
    x ) get_seed_args(x, argc, argv)
```

## 9.14.2 Function Documentation

### 9.14.2.1 `get_seed_args()`

```
ee_s32 get_seed_args (  
    int i,  
    int argc,  
    char * argv[] )
```

### 9.14.2.2 `iterate()`

```
void* iterate (  
    void * pres )
```

### 9.14.2.3 main()

```
MAIN_RETURN_TYPE main (  
    int argc,  
    char * argv[] )
```

## 9.14.3 Variable Documentation

### 9.14.3.1 list\_known\_crc

```
ee_u16 list_known_crc[] [static]
```

**Initial value:**

```
= { (ee_u16)0xd4b0,  
    (ee_u16)0x3340,  
    (ee_u16)0x6a79,  
    (ee_u16)0xe714,  
    (ee_u16)0xe3c1 }
```

### 9.14.3.2 matrix\_known\_crc

```
ee_u16 matrix_known_crc[] [static]
```

**Initial value:**

```
= { (ee_u16)0xbe52,  
    (ee_u16)0x1199,  
    (ee_u16)0x5608,  
    (ee_u16)0x1fd7,  
    (ee_u16)0x0747 }
```

### 9.14.3.3 mem\_name

```
char* mem_name[3] = { "Static", "Heap", "Stack" }
```

### 9.14.3.4 state\_known\_crc

```
ee_u16 state_known_crc[] [static]
```

**Initial value:**

```
= { (ee_u16)0x5e47,  
    (ee_u16)0x39bf,  
    (ee_u16)0xe5a4,  
    (ee_u16)0x8e3a,  
    (ee_u16)0x8d84 }
```

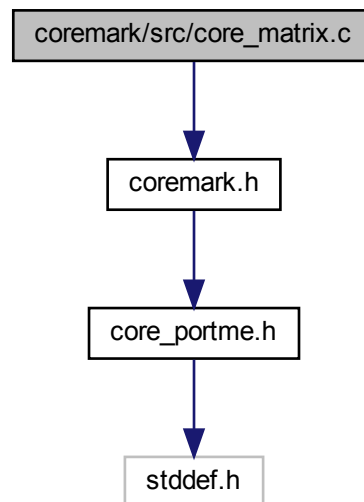
### 9.14.3.5 static\_memblk

```
ee_u8 static_memblk[TOTAL_DATA_SIZE]
```

## 9.15 coremark/src/core\_matrix.c File Reference

```
#include "coremark.h"
```

Include dependency graph for core\_matrix.c:



### Macros

- #define `matrix_test_next(x)`  $(x + 1)$
- #define `matrix_clip(x, y)`  $((y) ? (x) \& 0x0ff : (x) \& 0xffff)$
- #define `matrix_big(x)`  $(0xf000 | (x))$
- #define `bit_extract(x, from, to)`  $((x) >> (from)) \& (\sim(0xffffffff << (to)))$

### Functions

- `ee_s16 matrix_test` (`ee_u32 N`, `MATRES *C`, `MATDAT *A`, `MATDAT *B`, `MATDAT val`)
- `ee_s16 matrix_sum` (`ee_u32 N`, `MATRES *C`, `MATDAT clipval`)
- `void matrix_mul_const` (`ee_u32 N`, `MATRES *C`, `MATDAT *A`, `MATDAT val`)
- `void matrix_mul_vect` (`ee_u32 N`, `MATRES *C`, `MATDAT *A`, `MATDAT *B`)
- `void matrix_mul_matrix` (`ee_u32 N`, `MATRES *C`, `MATDAT *A`, `MATDAT *B`)
- `void matrix_mul_matrix_bitextract` (`ee_u32 N`, `MATRES *C`, `MATDAT *A`, `MATDAT *B`)
- `void matrix_add_const` (`ee_u32 N`, `MATDAT *A`, `MATDAT val`)
- `ee_u16 core_bench_matrix` (`mat_params *p`, `ee_s16 seed`, `ee_u16 crc`)
- `ee_u32 core_init_matrix` (`ee_u32 blksize`, `void *memblk`, `ee_s32 seed`, `mat_params *p`)

## 9.15.1 Macro Definition Documentation

### 9.15.1.1 bit\_extract

```
#define bit_extract(  
    x,  
    from,  
    to ) ((x) >> (from) & (~(0xffffffff << (to))))
```

### 9.15.1.2 matrix\_big

```
#define matrix_big(  
    x ) (0xf000 | (x))
```

### 9.15.1.3 matrix\_clip

```
#define matrix_clip(  
    x,  
    y ) ((y) ? (x)&0x0fff : (x)&0xffff)
```

### 9.15.1.4 matrix\_test\_next

```
#define matrix_test_next(  
    x ) (x + 1)
```

## 9.15.2 Function Documentation

### 9.15.2.1 core\_bench\_matrix()

```
ee_u16 core_bench_matrix (  
    mat_params * p,  
    ee_s16 seed,  
    ee_u16 crc )
```

### 9.15.2.2 core\_init\_matrix()

```
ee_u32 core_init_matrix (
    ee_u32 blksize,
    void * memblk,
    ee_s32 seed,
    mat_params * p )
```

### 9.15.2.3 matrix\_add\_const()

```
void matrix_add_const (
    ee_u32 N,
    MATDAT * A,
    MATDAT val )
```

### 9.15.2.4 matrix\_mul\_const()

```
void matrix_mul_const (
    ee_u32 N,
    MATRES * C,
    MATDAT * A,
    MATDAT val )
```

### 9.15.2.5 matrix\_mul\_matrix()

```
void matrix_mul_matrix (
    ee_u32 N,
    MATRES * C,
    MATDAT * A,
    MATDAT * B )
```

### 9.15.2.6 matrix\_mul\_matrix\_bitextract()

```
void matrix_mul_matrix_bitextract (
    ee_u32 N,
    MATRES * C,
    MATDAT * A,
    MATDAT * B )
```



### 9.15.2.7 matrix\_mul\_vect()

```
void matrix_mul_vect (
    ee_u32 N,
    MATRES * C,
    MATDAT * A,
    MATDAT * B )
```

### 9.15.2.8 matrix\_sum()

```
ee_s16 matrix_sum (
    ee_u32 N,
    MATRES * C,
    MATDAT clipval )
```

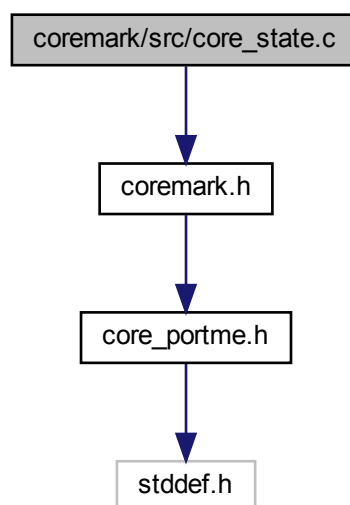
### 9.15.2.9 matrix\_test()

```
ee_s16 matrix_test (
    ee_u32 N,
    MATRES * C,
    MATDAT * A,
    MATDAT * B,
    MATDAT val )
```

## 9.16 coremark/src/core\_state.c File Reference

```
#include "coremark.h"
```

Include dependency graph for core\_state.c:



## Functions

- enum `CORE_STATE` `core_state_transition` (`ee_u8 **instr`, `ee_u32 *transition_count`)
- `ee_u16` `core_bench_state` (`ee_u32` `blksize`, `ee_u8 *memblock`, `ee_s16` `seed1`, `ee_s16` `seed2`, `ee_s16` `step`, `ee_u16` `crc`)
- void `core_init_state` (`ee_u32` `size`, `ee_s16` `seed`, `ee_u8 *p`)
- static `ee_u8` `ee_isdigit` (`ee_u8` `c`)

## Variables

- static `ee_u8 *intpat` [4] = { (`ee_u8 *`)"5012", (`ee_u8 *`)"1234", (`ee_u8 *`)"-874", (`ee_u8 *`)"+122" }
- static `ee_u8 *floatpat` [4]
- static `ee_u8 *scipat` [4]
- static `ee_u8 *errpat` [4]

## 9.16.1 Function Documentation

### 9.16.1.1 `core_bench_state()`

```
ee_u16 core_bench_state (
    ee_u32 blksize,
    ee_u8 * memblock,
    ee_s16 seed1,
    ee_s16 seed2,
    ee_s16 step,
    ee_u16 crc )
```

### 9.16.1.2 `core_init_state()`

```
void core_init_state (
    ee_u32 size,
    ee_s16 seed,
    ee_u8 * p )
```

### 9.16.1.3 `core_state_transition()`

```
enum CORE_STATE core_state_transition (
    ee_u8 ** instr,
    ee_u32 * transition_count )
```

### 9.16.1.4 ee\_isdigit()

```
static ee_u8 ee_isdigit (
    ee_u8 c ) [static]
```

## 9.16.2 Variable Documentation

### 9.16.2.1 errpat

```
ee_u8* errpat[4] [static]
```

#### Initial value:

```
= { (ee_u8 *) "T0.3e-1F",
    (ee_u8 *) "-T.T++Tq",
    (ee_u8 *) "1T3.4e4z",
    (ee_u8 *) "34.0e-T^" }
```

### 9.16.2.2 floatpat

```
ee_u8* floatpat[4] [static]
```

#### Initial value:

```
= { (ee_u8 *) "35.54400",
    (ee_u8 *) ".1234500",
    (ee_u8 *) "-110.700",
    (ee_u8 *) "+0.64400" }
```

### 9.16.2.3 intpat

```
ee_u8* intpat[4] = { (ee_u8 *) "5012", (ee_u8 *) "1234", (ee_u8 *) "-874", (ee_u8 *) "+122" } [static]
```

### 9.16.2.4 scipat

```
ee_u8* scipat[4] [static]
```

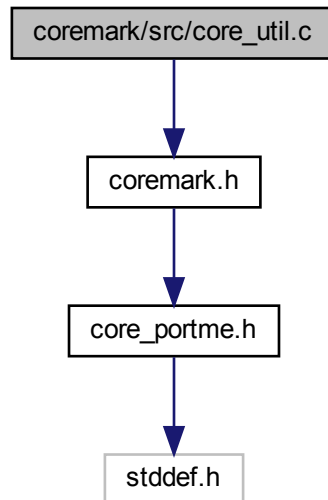
#### Initial value:

```
= { (ee_u8 *) "5.500e+3",
    (ee_u8 *) "-.123e-2",
    (ee_u8 *) "-87e+832",
    (ee_u8 *) "+0.6e-12" }
```

## 9.17 coremark/src/core\_util.c File Reference

```
#include "coremark.h"
```

Include dependency graph for core\_util.c:



### Functions

- [ee\\_s32 parseval](#) (char \*valstring)
- [ee\\_s32 get\\_seed\\_args](#) (int i, int argc, char \*argv[ ])
- [ee\\_u16 crcu8](#) (ee\_u8 data, ee\_u16 crc)
- [ee\\_u16 crcu16](#) (ee\_u16 newval, ee\_u16 crc)
- [ee\\_u16 crcu32](#) (ee\_u32 newval, ee\_u16 crc)
- [ee\\_u16 crc16](#) (ee\_s16 newval, ee\_u16 crc)
- [ee\\_u8 check\\_data\\_types](#) ( )

### 9.17.1 Function Documentation

#### 9.17.1.1 check\_data\_types()

```
ee_u8 check_data_types ( )
```

### 9.17.1.2 crc16()

```
ee_u16 crc16 (
    ee_s16 newval,
    ee_u16 crc )
```

### 9.17.1.3 crcu16()

```
ee_u16 crcu16 (
    ee_u16 newval,
    ee_u16 crc )
```

### 9.17.1.4 crcu32()

```
ee_u16 crcu32 (
    ee_u32 newval,
    ee_u16 crc )
```

### 9.17.1.5 crcu8()

```
ee_u16 crcu8 (
    ee_u8 data,
    ee_u16 crc )
```

### 9.17.1.6 get\_seed\_args()

```
ee_s32 get_seed_args (
    int i,
    int argc,
    char * argv[] )
```

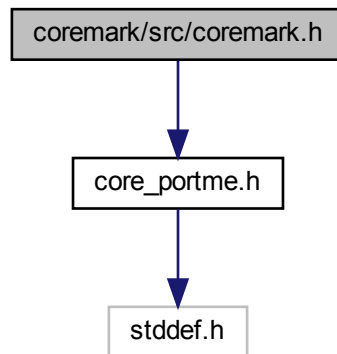
### 9.17.1.7 parseval()

```
ee_s32 parseval (
    char * valstring )
```

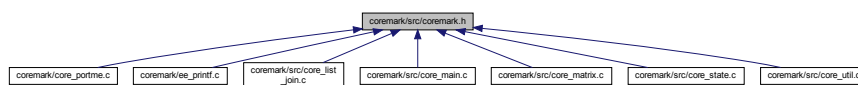
## 9.18 coremark/src/coremark.h File Reference

```
#include "core_portme.h"
```

Include dependency graph for coremark.h:



This graph shows which files directly or indirectly include this file:



### Data Structures

- struct [list\\_data\\_s](#)
- struct [list\\_head\\_s](#)
- struct [MAT\\_PARAMS\\_S](#)
- struct [RESULTS\\_S](#)

### Macros

- #define [TOTAL\\_DATA\\_SIZE](#) 2 \* 1000
- #define [SEED\\_ARG](#) 0
- #define [SEED\\_FUNC](#) 1
- #define [SEED\\_VOLATILE](#) 2
- #define [MEM\\_STATIC](#) 0
- #define [MEM\\_MALLOC](#) 1
- #define [MEM\\_STACK](#) 2
- #define [MAIN\\_RETURN\\_VAL](#) 0
- #define [MAIN\\_RETURN\\_TYPE](#) int
- #define [ID\\_LIST](#) (1 << 0)
- #define [ID\\_MATRIX](#) (1 << 1)
- #define [ID\\_STATE](#) (1 << 2)
- #define [ALL\\_ALGORITHMS\\_MASK](#) (ID\_LIST | ID\_MATRIX | ID\_STATE)
- #define [NUM\\_ALGORITHMS](#) 3
- #define [MATDAT\\_INT](#) 1

## Typedefs

- typedef `ee_u32` `secs_ret`
- typedef struct `list_data_s` `list_data`
- typedef struct `list_head_s` `list_head`
- typedef `ee_s16` `MATDAT`
- typedef `ee_s32` `MATRES`
- typedef struct `MAT_PARAMS_S` `mat_params`
- typedef enum `CORE_STATE` `core_state_e`
- typedef struct `RESULTS_S` `core_results`

## Enumerations

- enum `CORE_STATE` {  
`CORE_START` = 0, `CORE_INVALID`, `CORE_S1`, `CORE_S2`,  
`CORE_INT`, `CORE_FLOAT`, `CORE_EXPONENT`, `CORE_SCIENTIFIC`,  
`NUM_CORE_STATES` }

## Functions

- void \* `iterate` (void \*pres)
- void `start_time` (void)
- void `stop_time` (void)
- `CORE_TICKS` `get_time` (void)
- `secs_ret` `time_in_secs` (`CORE_TICKS` ticks)
- `ee_u16` `crcu8` (`ee_u8` data, `ee_u16` crc)
- `ee_u16` `crc16` (`ee_s16` newval, `ee_u16` crc)
- `ee_u16` `crcu16` (`ee_u16` newval, `ee_u16` crc)
- `ee_u16` `crcu32` (`ee_u32` newval, `ee_u16` crc)
- `ee_u8` `check_data_types` (void)
- void \* `portable_malloc` (`ee_size_t` size)
- void `portable_free` (void \*p)
- `ee_s32` `parseval` (char \*valstring)
- `list_head` \* `core_list_init` (`ee_u32` blksize, `list_head` \*memblock, `ee_s16` seed)
- `ee_u16` `core_bench_list` (`core_results` \*res, `ee_s16` finder\_idx)
- void `core_init_state` (`ee_u32` size, `ee_s16` seed, `ee_u8` \*p)
- `ee_u16` `core_bench_state` (`ee_u32` blksize, `ee_u8` \*memblock, `ee_s16` seed1, `ee_s16` seed2, `ee_s16` step, `ee_u16` crc)
- `ee_u32` `core_init_matrix` (`ee_u32` blksize, void \*membblk, `ee_s32` seed, `mat_params` \*p)
- `ee_u16` `core_bench_matrix` (`mat_params` \*p, `ee_s16` seed, `ee_u16` crc)

### 9.18.1 Macro Definition Documentation

#### 9.18.1.1 ALL\_ALGORITHMS\_MASK

```
#define ALL_ALGORITHMS_MASK (ID_LIST | ID_MATRIX | ID_STATE)
```

#### 9.18.1.2 ID\_LIST

```
#define ID_LIST (1 << 0)
```

#### 9.18.1.3 ID\_MATRIX

```
#define ID_MATRIX (1 << 1)
```

#### 9.18.1.4 ID\_STATE

```
#define ID_STATE (1 << 2)
```

#### 9.18.1.5 MAIN\_RETURN\_TYPE

```
#define MAIN_RETURN_TYPE int
```

#### 9.18.1.6 MAIN\_RETURN\_VAL

```
#define MAIN_RETURN_VAL 0
```

#### 9.18.1.7 MATDAT\_INT

```
#define MATDAT_INT 1
```

#### 9.18.1.8 MEM\_MALLOC

```
#define MEM_MALLOC 1
```

#### 9.18.1.9 MEM\_STACK

```
#define MEM_STACK 2
```



#### 9.18.1.10 MEM\_STATIC

```
#define MEM_STATIC 0
```

#### 9.18.1.11 NUM\_ALGORITHMS

```
#define NUM_ALGORITHMS 3
```

#### 9.18.1.12 SEED\_ARG

```
#define SEED_ARG 0
```

#### 9.18.1.13 SEED\_FUNC

```
#define SEED_FUNC 1
```

#### 9.18.1.14 SEED\_VOLATILE

```
#define SEED_VOLATILE 2
```

#### 9.18.1.15 TOTAL\_DATA\_SIZE

```
#define TOTAL_DATA_SIZE 2 * 1000
```

### 9.18.2 Typedef Documentation

#### 9.18.2.1 core\_results

```
typedef struct RESULTS_S core_results
```

#### 9.18.2.2 core\_state\_e

```
typedef enum CORE_STATE core_state_e
```

#### 9.18.2.3 list\_data

```
typedef struct list_data_s list_data
```

#### 9.18.2.4 list\_head

```
typedef struct list_head_s list_head
```

#### 9.18.2.5 mat\_params

```
typedef struct MAT_PARAMS_S mat_params
```

#### 9.18.2.6 MATDAT

```
typedef ee_s16 MATDAT
```

#### 9.18.2.7 MATRES

```
typedef ee_s32 MATRES
```

#### 9.18.2.8 secs\_ret

```
typedef ee_u32 secs_ret
```

### 9.18.3 Enumeration Type Documentation

#### 9.18.3.1 CORE\_STATE

```
enum CORE_STATE
```

## Enumerator

CORE_START	
CORE_INVALID	
CORE_S1	
CORE_S2	
CORE_INT	
CORE_FLOAT	
CORE_EXPONENT	
CORE_SCIENTIFIC	
NUM_CORE_STATES	

## 9.18.4 Function Documentation

### 9.18.4.1 check\_data\_types()

```
ee_u8 check_data_types (
    void )
```

### 9.18.4.2 core\_bench\_list()

```
ee_u16 core_bench_list (
    core_results * res,
    ee_s16 finder_idx )
```

### 9.18.4.3 core\_bench\_matrix()

```
ee_u16 core_bench_matrix (
    mat_params * p,
    ee_s16 seed,
    ee_u16 crc )
```

### 9.18.4.4 core\_bench\_state()

```
ee_u16 core_bench_state (
    ee_u32 blksize,
    ee_u8 * memblock,
    ee_s16 seed1,
    ee_s16 seed2,
    ee_s16 step,
    ee_u16 crc )
```

#### 9.18.4.5 core\_init\_matrix()

```
ee_u32 core_init_matrix (
    ee_u32 blksize,
    void * memblk,
    ee_s32 seed,
    mat_params * p )
```

#### 9.18.4.6 core\_init\_state()

```
void core_init_state (
    ee_u32 size,
    ee_s16 seed,
    ee_u8 * p )
```

#### 9.18.4.7 core\_list\_init()

```
list_head* core_list_init (
    ee_u32 blksize,
    list_head * memblock,
    ee_s16 seed )
```

#### 9.18.4.8 crc16()

```
ee_u16 crc16 (
    ee_s16 newval,
    ee_u16 crc )
```

#### 9.18.4.9 crcu16()

```
ee_u16 crcu16 (
    ee_u16 newval,
    ee_u16 crc )
```

#### 9.18.4.10 crcu32()

```
ee_u16 crcu32 (
    ee_u32 newval,
    ee_u16 crc )
```

#### 9.18.4.11 crcu8()

```
ee_u16 crcu8 (
    ee_u8 data,
    ee_u16 crc )
```

#### 9.18.4.12 get\_time()

```
CORE_TICKS get_time (
    void )
```

#### 9.18.4.13 iterate()

```
void* iterate (
    void * pres )
```

#### 9.18.4.14 parseval()

```
ee_s32 parseval (
    char * valstring )
```

#### 9.18.4.15 portable\_free()

```
void portable_free (
    void * p )
```

#### 9.18.4.16 portable\_malloc()

```
void* portable_malloc (
    ee_size_t size )
```

#### 9.18.4.17 start\_time()

```
void start_time (
    void )
```

#### 9.18.4.18 stop\_time()

```
void stop_time (
    void )
```

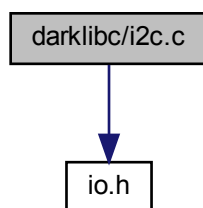
#### 9.18.4.19 time\_in\_secs()

```
secs_ret time_in_secs (
    CORE_TICKS ticks )
```

### 9.19 darklibc/i2c.c File Reference

```
#include <io.h>
```

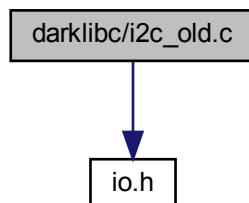
Include dependency graph for i2c.c:



### 9.20 darklibc/i2c\_old.c File Reference

```
#include <io.h>
```

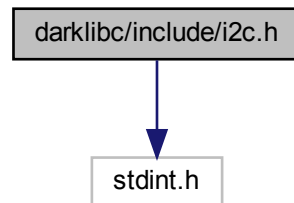
Include dependency graph for i2c\_old.c:



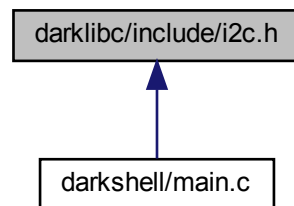
## 9.21 darklibc/include/i2c.h File Reference

```
#include <stdint.h>
```

Include dependency graph for i2c.h:



This graph shows which files directly or indirectly include this file:



### Data Structures

- struct [I2C\\_RegisterBits](#)
- union [I2C\\_Register](#)

### Functions

- void [i2cSendByte](#) (char, char, char)  
*Sends a single byte via I2C.*
- char [i2cReadByte](#) (char, char)  
*Reads a single byte via I2C.*

#### 9.21.1 Function Documentation

### 9.21.1.1 i2cReadByte()

```
char i2cReadByte (
    char ,
    char )
```

Reads a single byte via I2C.

This function reads a byte from a specific slave device and subaddress over I2C.

#### Parameters

<i>slaveAddress</i>	The I2C slave device address.
<i>subaddress</i>	The register/subaddress within the slave device.

#### Returns

char The byte read, or -1 on error.

### 9.21.1.2 i2cSendByte()

```
void i2cSendByte (
    char ,
    char ,
    char )
```

Sends a single byte via I2C.

This function sends a byte to a specific slave device and subaddress over I2C.

#### Parameters

<i>slaveAddress</i>	The I2C slave device address.
<i>subaddress</i>	The register/subaddress within the slave device.
<i>byteToSend</i>	The byte of data to be transmitted.

#### Returns

int 0 on success, -1 if no ACK received or if I2C is busy.

## 9.22 darklibc/include/io.h File Reference

This graph shows which files directly or indirectly include this file:





## Data Structures

- struct [DARKIO](#)
- struct [DARKIO::DARKUART](#)

## Macros

- `#define` [IRQ\\_TIMR](#) 0x80
- `#define` [IRQ\\_UART](#) 0x02

## Functions

- char \* [board\\_name](#) (int)
- int [check4rv32i](#) (void)
- void [set\\_mtvec](#) (void(\*)(void))
- void [set\\_mepc](#) (void(\*)(void))
- void [set\\_mie](#) (int)
- int [get\\_mtvec](#) (void)
- int [get\\_mepc](#) (void)
- int [get\\_mie](#) (void)
- void [banner](#) (void)
- `__attribute__((interrupt("machine")))` void [irq\\_handler](#)(void)

## Variables

- volatile int [utimers](#)
- volatile struct [DARKIO](#) \* [io](#)
- unsigned char [kmem](#) [8192]
- unsigned [\\_text](#)
- unsigned [\\_data](#)
- unsigned [\\_etext](#)
- unsigned [\\_edata](#)
- unsigned [\\_stack](#)

### 9.22.1 Macro Definition Documentation

#### 9.22.1.1 IRQ\_TIMR

```
#define IRQ_TIMR 0x80
```

#### 9.22.1.2 IRQ\_UART

```
#define IRQ_UART 0x02
```

## 9.22.2 Function Documentation

### 9.22.2.1 `__attribute__()`

```
__attribute__ (  
    (interrupt("machine")) )
```

### 9.22.2.2 `banner()`

```
void banner (  
    void )
```

### 9.22.2.3 `board_name()`

```
char* board_name (  
    int )
```

### 9.22.2.4 `check4rv32i()`

```
int check4rv32i (  
    void )
```

### 9.22.2.5 `get_mepc()`

```
int get_mepc (  
    void )
```

### 9.22.2.6 `get_mie()`

```
int get_mie (  
    void )
```

### 9.22.2.7 `get_mtvec()`

```
int get_mtvec (
    void )
```

### 9.22.2.8 `set_mepc()`

```
void set_mepc (
    void(*) (void) f )
```

### 9.22.2.9 `set_mie()`

```
void set_mie (
    int )
```

### 9.22.2.10 `set_mtvec()`

```
void set_mtvec (
    void(*) (void) f )
```

## 9.22.3 Variable Documentation

### 9.22.3.1 `__data`

```
unsigned __data
```

### 9.22.3.2 `__edata`

```
unsigned __edata
```

### 9.22.3.3 `__etext`

```
unsigned __etext
```

#### 9.22.3.4 `_stack`

```
unsigned _stack
```

#### 9.22.3.5 `_text`

```
unsigned _text
```

#### 9.22.3.6 `io`

```
volatile struct DARKIO* io
```

#### 9.22.3.7 `kmem`

```
unsigned char kmem[8192]
```

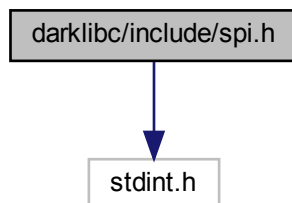
#### 9.22.3.8 `utimers`

```
volatile int utimers
```

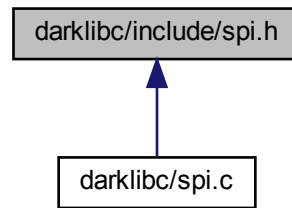
### 9.23 `darklibc/include/spi.h` File Reference

```
#include <stdint.h>
```

Include dependency graph for `spi.h`:



This graph shows which files directly or indirectly include this file:



## Data Structures

- struct [SPI\\_RegisterBits](#)
- union [SPI\\_Register](#)

## Functions

- void [spi\\_init](#) ()
- void [spi\\_set\\_clock\\_frequency](#) (uint32\_t frequency)
- void [spi\\_set\\_data\\_mode](#) (uint8\_t mode)
- void [spi\\_enable](#) ()
- void [spi\\_disable](#) ()
- uint8\_t [spi\\_send\\_receive\\_data](#) (uint8\_t data)
- uint8\_t [spi\\_transaction\\_single\\_byte](#) ([SPI\\_Register](#))  
*Performs a single byte SPI transaction.*
- uint8\_t [spi\\_write\\_read\\_single\\_byte](#) (uint8\_t, uint8\_t, uint8\_t)  
*Transmits a single byte via SPI and simultaneously receives a byte.*
- uint8\_t [spi\\_read\\_single\\_byte](#) (uint8\_t, uint8\_t)  
*Reads a single byte from the SPI bus.*
- void [spi\\_write\\_single\\_byte](#) (uint8\_t, uint8\_t, uint8\_t)  
*Writes a single byte to the SPI bus.*
- void [spi\\_read\\_multiple\\_bytes](#) (uint8\_t \*, uint8\_t, uint8\_t)  
*Reads multiple bytes from the SPI interface.*
- void [spi\\_write\\_multiple\\_bytes](#) (uint8\_t \*, uint8\_t, uint8\_t)  
*Writes multiple bytes to the SPI bus.*
- void [spi\\_write\\_read\\_multiple\\_bytes](#) (uint8\_t \*, uint8\_t \*, uint8\_t, uint8\_t)  
*Writes and reads multiple bytes over SPI.*

### 9.23.1 Function Documentation

#### 9.23.1.1 spi\_disable()

```
void spi_disable ( )
```

#### 9.23.1.2 spi\_enable()

```
void spi_enable ( )
```

#### 9.23.1.3 spi\_init()

```
void spi_init ( )
```

#### 9.23.1.4 spi\_read\_multiple\_bytes()

```
void spi_read_multiple_bytes (
    uint8_t * ,
    uint8_t ,
    uint8_t )
```

Reads multiple bytes from the SPI interface.

This function reads a specified number of bytes from the SPI interface and stores them in the provided buffer.

##### Parameters

<i>rx_data</i>	Pointer to the buffer where the received data will be stored.
<i>length</i>	The number of bytes to read from the SPI interface.
<i>slave</i>	The slave device to communicate with.

#### 9.23.1.5 spi\_read\_single\_byte()

```
uint8_t spi_read_single_byte (
    uint8_t ,
    uint8_t )
```

Reads a single byte from the SPI bus.

This function performs a read operation on the SPI bus and returns the byte that was read. Used to receive data from an SPI slave device.

## Parameters

<i>length</i>	The number of bytes to be in the complete transaction. Should be 1 when not called as part of a multi-byte transaction.
<i>slave</i>	The slave device to communicate with.

## Returns

uint8\_t The byte read from the SPI bus.

**9.23.1.6 spi\_send\_receive\_data()**

```
uint8_t spi_send_receive_data (
    uint8_t data )
```

**9.23.1.7 spi\_set\_clock\_frequency()**

```
void spi_set_clock_frequency (
    uint32_t frequency )
```

**9.23.1.8 spi\_set\_data\_mode()**

```
void spi_set_data_mode (
    uint8_t mode )
```

**9.23.1.9 spi\_transaction\_single\_byte()**

```
uint8_t spi_transaction_single_byte (
    SPI_Register )
```

Performs a single byte SPI transaction.

This function initiates a single byte transaction over the SPI interface using the provided SPI register configuration.

## Parameters

<i>new_spi_register</i>	[ <a href="#">SPI_Register</a> ] The SPI register configuration to be used for the transaction.
-------------------------	---

**Returns**

[uint8\_t] The byte received from the SPI transaction.

**9.23.1.10 spi\_write\_multiple\_bytes()**

```
void spi_write_multiple_bytes (
    uint8_t * ,
    uint8_t ,
    uint8_t )
```

Writes multiple bytes to the SPI bus.

This function sends a sequence of bytes over the SPI bus.

**Parameters**

<i>data</i>	Pointer to the array of bytes to be sent.
<i>length</i>	Number of bytes to be sent.
<i>slave</i>	The slave device to communicate with.

**9.23.1.11 spi\_write\_read\_multiple\_bytes()**

```
void spi_write_read_multiple_bytes (
    uint8_t * ,
    uint8_t * ,
    uint8_t ,
    uint8_t )
```

Writes and reads multiple bytes over SPI.

This function transmits and receives a specified number of bytes over the SPI bus.

**Parameters**

<i>tx_data</i>	Pointer to the data to be transmitted.
<i>rx_data</i>	Pointer to the buffer where received data will be stored.
<i>length</i>	Number of bytes to be transmitted and received.
<i>slave</i>	The slave device to communicate with.

**9.23.1.12 spi\_write\_read\_single\_byte()**

```
uint8_t spi_write_read_single_byte (
    uint8_t ,
```



```
uint8_t ,
uint8_t )
```

Transmits a single byte via SPI and simultaneously receives a byte.

This function sends a single byte of data through the SPI interface and reads a byte of data received from the SPI slave device. Used for full-duplex communication where data is sent and received simultaneously.

#### Parameters

<i>data</i>	The byte of data to be transmitted.
<i>length</i>	The number of bytes to be in the transaction.
<i>length</i>	The number of bytes to be in the complete transaction. Should be 1 when not called as part of a multi-byte transaction.
<i>slave</i>	The slave device to communicate with.

#### Returns

uint8\_t The byte received from the SPI transaction.

#### 9.23.1.13 spi\_write\_single\_byte()

```
void spi_write_single_byte (
    uint8_t ,
    uint8_t ,
    uint8_t )
```

Writes a single byte to the SPI bus.

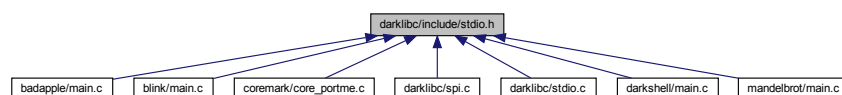
This function sends a single byte of data over the SPI bus.

#### Parameters

<i>data</i>	The byte of data to be sent.
<i>length</i>	The number of bytes to be in the complete transaction. Should be 1 when not called as part of a multi-byte transaction.
<i>slave</i>	The slave device to communicate with.

## 9.24 darklibc/include/stdio.h File Reference

This graph shows which files directly or indirectly include this file:



## Macros

- `#define EOF -1`
- `#define NUL 0`
- `#define NULL (void *)0`
- `#define EBREAK asm("ebreak")`

## Functions

- `int getchar (void)`
- `int putchar (int c)`
- `char * gets (char *p, int s)`
- `void putstr (char *p)`
- `int puts (char *p)`
- `int printf (char *fmt,...)`
- `int strcmp (char *s1, char *s2)`
- `int strncmp (char *s1, char *s2, int len)`
- `int strlen (char *s1)`
- `void putx (unsigned)`
- `void putd (int)`
- `char * memcpy (char *dptr, char *sptr, int len)`
- `char * memset (char *dptr, int c, int len)`
- `char * strtok (char *str, char *dptr)`
- `int atoi (char *)`
- `int xtoi (char *)`
- `int mac (int, short, short)`
- `void usleep (int)`

### 9.24.1 Macro Definition Documentation

#### 9.24.1.1 EBREAK

```
#define EBREAK asm("ebreak")
```

#### 9.24.1.2 EOF

```
#define EOF -1
```

#### 9.24.1.3 NUL

```
#define NUL 0
```

#### 9.24.1.4 NULL

```
#define NULL (void *)0
```

### 9.24.2 Function Documentation

#### 9.24.2.1 atoi()

```
int atoi (  
    char * )
```

#### 9.24.2.2 getchar()

```
int getchar (  
    void )
```

#### 9.24.2.3 gets()

```
char* gets (  
    char * p,  
    int s )
```

#### 9.24.2.4 mac()

```
int mac (  
    int ,  
    short ,  
    short )
```

#### 9.24.2.5 memcpy()

```
char* memcpy (  
    char * dptr,  
    char * sptr,  
    int len )
```

#### 9.24.2.6 `memset()`

```
char* memset (
    char * dptr,
    int c,
    int len )
```

#### 9.24.2.7 `printf()`

```
int printf (
    char * fmt,
    ... )
```

#### 9.24.2.8 `putchar()`

```
int putchar (
    int c )
```

#### 9.24.2.9 `putd()`

```
void putd (
    int )
```

#### 9.24.2.10 `puts()`

```
int puts (
    char * p )
```

#### 9.24.2.11 `putstr()`

```
void putstr (
    char * p )
```

#### 9.24.2.12 putx()

```
void putx (
    unsigned )
```

#### 9.24.2.13 strcmp()

```
int strcmp (
    char * s1,
    char * s2 )
```

#### 9.24.2.14 strlen()

```
int strlen (
    char * s1 )
```

#### 9.24.2.15 strncmp()

```
int strncmp (
    char * s1,
    char * s2,
    int len )
```

#### 9.24.2.16 strtok()

```
char* strtok (
    char * str,
    char * dptr )
```

#### 9.24.2.17 usleep()

```
void usleep (
    int )
```

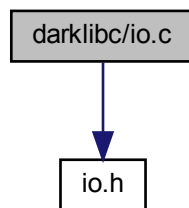
### 9.24.2.18 xtoi()

```
int xtoi (
    char * )
```

## 9.25 darklibc/io.c File Reference

```
#include <io.h>
```

Include dependency graph for io.c:



### Functions

- char \* [board\\_name](#) (int id)
- [\\_\\_attribute\\_\\_](#) ((interrupt("machine")))
- int [mac](#) (int acc, short x, short y)

### Variables

- volatile struct [DARKIO io](#)
- unsigned char [kmem](#) [8192] = "darksocv x86 payload test"
- volatile int [utimers](#) = 0

### 9.25.1 Function Documentation

#### 9.25.1.1 \_\_attribute\_\_()

```
__attribute__ (
    (interrupt("machine")) )
```

### 9.25.1.2 board\_name()

```
char* board_name (
    int id )
```

### 9.25.1.3 mac()

```
int mac (
    int acc,
    short x,
    short y )
```

## 9.25.2 Variable Documentation

### 9.25.2.1 io

```
volatile struct DARKIO io
```

#### Initial value:

```
=
{
    4, 100, 0, 0,
    { 0, 0, 0 },
    0,
    0,
    1000000,
    0,
    0,
    0
}
```

### 9.25.2.2 kmem

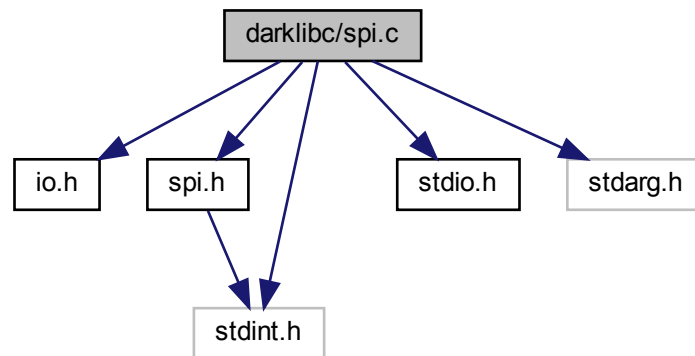
```
unsigned char kmem[8192] = "darksocv x86 payload test"
```

### 9.25.2.3 utimers

```
volatile int utimers = 0
```

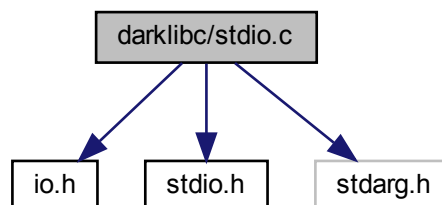
## 9.26 darklibc/spi.c File Reference

```
#include <io.h>
#include <spi.h>
#include <stdio.h>
#include <stdarg.h>
#include <stdint.h>
Include dependency graph for spi.c:
```



## 9.27 darklibc/stdio.c File Reference

```
#include <io.h>
#include <stdio.h>
#include <stdarg.h>
Include dependency graph for stdio.c:
```



## Functions

- `char * gets (char *p, int s)`



- void [putstr](#) (char \*p)
- int [puts](#) (char \*p)
- void [putnum](#) (unsigned i, int base)
- int [printf](#) (char \*fmt,...)
- int [strncmp](#) (char \*s1, char \*s2, int len)
- int [strcmp](#) (char \*s1, char \*s2)
- int [strlen](#) (char \*s1)
- char \* [strtok](#) (char \*str, char \*dptr)
- char \* [memcpy](#) (char \*dptr, char \*sptr, int len)
- char \* [memset](#) (char \*dptr, int c, int len)
- int [atoi](#) (char \*s1)
- int [xtoi](#) (char \*s1)
- unsigned [\\_\\_umulsi3](#) (unsigned x, unsigned y)
- int [\\_\\_mulsi3](#) (int x, int y)
- unsigned [\\_\\_udiv\\_umod\\_si3](#) (unsigned x, unsigned y, int opt)
- int [\\_\\_udivsi3](#) (int x, int y)
- int [\\_\\_umodsi3](#) (int x, int y)
- int [\\_\\_div\\_mod\\_si3](#) (int x, int y, int opt)
- int [\\_\\_divsi3](#) (int x, int y)
- int [\\_\\_modsi3](#) (int x, int y)
- void [usleep](#) (int delay)

## 9.27.1 Function Documentation

### 9.27.1.1 [\\_\\_div\\_mod\\_si3\(\)](#)

```
int __div_mod_si3 (  
    int x,  
    int y,  
    int opt )
```

### 9.27.1.2 [\\_\\_divsi3\(\)](#)

```
int __divsi3 (  
    int x,  
    int y )
```

### 9.27.1.3 [\\_\\_modsi3\(\)](#)

```
int __modsi3 (  
    int x,  
    int y )
```

**9.27.1.4 \_\_mulsi3()**

```
int __mulsi3 (
    int x,
    int y )
```

**9.27.1.5 \_\_udiv\_umod\_si3()**

```
unsigned __udiv_umod_si3 (
    unsigned x,
    unsigned y,
    int opt )
```

**9.27.1.6 \_\_udivsi3()**

```
int __udivsi3 (
    int x,
    int y )
```

**9.27.1.7 \_\_umodsi3()**

```
int __umodsi3 (
    int x,
    int y )
```

**9.27.1.8 \_\_umulsi3()**

```
unsigned __umulsi3 (
    unsigned x,
    unsigned y )
```

**9.27.1.9 atoi()**

```
int atoi (
    char * s1 )
```

### 9.27.1.10 gets()

```
char* gets (
    char * p,
    int s )
```

### 9.27.1.11 memcpy()

```
char* memcpy (
    char * dptr,
    char * sptr,
    int len )
```

### 9.27.1.12 memset()

```
char* memset (
    char * dptr,
    int c,
    int len )
```

### 9.27.1.13 printf()

```
int printf (
    char * fmt,
    ... )
```

### 9.27.1.14 putnum()

```
void putnum (
    unsigned i,
    int base )
```

### 9.27.1.15 puts()

```
int puts (
    char * p )
```

**9.27.1.16 putstr()**

```
void putstr (
    char * p )
```

**9.27.1.17 strcmp()**

```
int strcmp (
    char * s1,
    char * s2 )
```

**9.27.1.18 strlen()**

```
int strlen (
    char * s1 )
```

**9.27.1.19 strncmp()**

```
int strncmp (
    char * s1,
    char * s2,
    int len )
```

**9.27.1.20 strtok()**

```
char* strtok (
    char * str,
    char * dptr )
```

**9.27.1.21 usleep()**

```
void usleep (
    int delay )
```

### 9.27.1.22 xtoi()

```
int xtoi (
    char * s1 )
```

## 9.28 programBoard.py File Reference

### Namespaces

- [programBoard](#)

### Variables

- [programBoard.ser](#) = serial.Serial('COM4', 115200)
- [programBoard.data](#) = f.read()



# Index

- `__attribute__`
  - io.c, [88](#)
  - io.h, [76](#)
- `__div_mod_si3`
  - stdio.c, [91](#)
- `__divsi3`
  - stdio.c, [91](#)
- `__modsi3`
  - stdio.c, [91](#)
- `__mulsi3`
  - stdio.c, [91](#)
- `__udiv_umod_si3`
  - stdio.c, [92](#)
- `__udivsi3`
  - stdio.c, [92](#)
- `__umodsi3`
  - stdio.c, [92](#)
- `__umulsi3`
  - stdio.c, [92](#)
- `_data`
  - io.h, [77](#)
- `_edata`
  - io.h, [77](#)
- `_etext`
  - io.h, [77](#)
- `_stack`
  - io.h, [77](#)
- `_text`
  - io.h, [78](#)

## A

- MAT\_PARAMS\_S, [23](#)
- align\_mem
  - core\_portme.h, [40](#)
- ALL\_ALGORITHMS\_MASK
  - coremark.h, [65](#)
- atoi
  - stdio.c, [92](#)
  - stdio.h, [85](#)

## B

- MAT\_PARAMS\_S, [23](#)
- badapple.h
  - rl, [31](#)
- badapple/badapple.h, [31](#)
- badapple/badapple.txt, [32](#)
- badapple/main.c, [32](#)
- banner
  - io.h, [76](#)
- barebones\_clock

- core\_portme.c, [37](#)

## baud

- DARKIO::DARKUART, [18](#)

## bit\_extract

- core\_matrix.c, [57](#)

## blink/main.c, [32](#)

## board\_cm

- DARKIO, [16](#)

## board\_id

- DARKIO, [16](#)

## board\_name

- io.c, [88](#)

- io.h, [76](#)

## busy

- I2C\_RegisterBits, [20](#)

## C

- MAT\_PARAMS\_S, [23](#)

## calc\_func

- core\_list\_join.c, [51](#)

## check4rv32i

- io.h, [76](#)

## check\_data\_types

- core\_util.c, [62](#)

- coremark.h, [69](#)

## CLOCKS\_PER\_SEC

- core\_portme.c, [36](#)

## cmp\_complex

- core\_list\_join.c, [51](#)

## cmp\_idx

- core\_list\_join.c, [51](#)

## COMPILER\_FLAGS

- core\_portme.h, [41](#)

## COMPILER\_VERSION

- core\_portme.h, [41](#)

## copy\_info

- core\_list\_join.c, [51](#)

## core\_bench\_list

- core\_list\_join.c, [51](#)

- coremark.h, [69](#)

## core\_bench\_matrix

- core\_matrix.c, [57](#)

- coremark.h, [69](#)

## core\_bench\_state

- core\_state.c, [60](#)

- coremark.h, [69](#)

## CORE\_EXPONENT

- coremark.h, [69](#)

## CORE\_FLOAT

- coremark.h, [69](#)

- core\_id
  - DARKIO, 16
- core\_init\_matrix
  - core\_matrix.c, 57
  - coremark.h, 69
- core\_init\_state
  - core\_state.c, 60
  - coremark.h, 70
- CORE\_INT
  - coremark.h, 69
- CORE\_INVALID
  - coremark.h, 69
- core\_list\_find
  - core\_list\_join.c, 52
- core\_list\_init
  - core\_list\_join.c, 52
  - coremark.h, 70
- core\_list\_insert\_new
  - core\_list\_join.c, 52
- core\_list\_join.c
  - calc\_func, 51
  - cmp\_complex, 51
  - cmp\_idx, 51
  - copy\_info, 51
  - core\_bench\_list, 51
  - core\_list\_find, 52
  - core\_list\_init, 52
  - core\_list\_insert\_new, 52
  - core\_list\_mergesort, 52
  - core\_list\_remove, 52
  - core\_list\_reverse, 52
  - core\_list\_undo\_remove, 53
  - list\_cmp, 51
- core\_list\_mergesort
  - core\_list\_join.c, 52
- core\_list\_remove
  - core\_list\_join.c, 52
- core\_list\_reverse
  - core\_list\_join.c, 52
- core\_list\_undo\_remove
  - core\_list\_join.c, 53
- core\_main.c
  - get\_seed, 54
  - get\_seed\_32, 54
  - get\_seed\_args, 54
  - iterate, 54
  - list\_known\_crc, 55
  - main, 54
  - matrix\_known\_crc, 55
  - mem\_name, 55
  - state\_known\_crc, 55
  - static\_memblk, 55
- core\_matrix.c
  - bit\_extract, 57
  - core\_bench\_matrix, 57
  - core\_init\_matrix, 57
  - matrix\_add\_const, 58
  - matrix\_big, 57
  - matrix\_clip, 57
  - matrix\_mul\_const, 58
  - matrix\_mul\_matrix, 58
  - matrix\_mul\_matrix\_bitextract, 58
  - matrix\_mul\_vect, 58
  - matrix\_sum, 59
  - matrix\_test, 59
  - matrix\_test\_next, 57
- core\_portable
  - core\_portme.h, 43
- CORE\_PORTABLE\_S, 15
  - portable\_id, 15
- core\_portme.c
  - barebones\_clock, 37
  - CLOCKS\_PER\_SEC, 36
  - default\_num\_contexts, 38
  - EE\_TICKS\_PER\_SEC, 36
  - get\_time, 37
  - GETMYTIME, 36
  - MYTIMEDIFF, 36
  - portable\_fini, 37
  - portable\_init, 37
  - SAMPLE\_TIME\_IMPLEMENTATION, 36
  - seed1\_volatile, 38
  - seed2\_volatile, 38
  - seed3\_volatile, 38
  - seed4\_volatile, 38
  - seed5\_volatile, 38
  - start\_time, 37
  - start\_time\_val, 39
  - stop\_time, 37
  - stop\_time\_val, 39
  - time\_in\_secs, 38
  - TIMER\_RES\_DIVIDER, 36
- core\_portme.h
  - align\_mem, 40
  - COMPILER\_FLAGS, 41
  - COMPILER\_VERSION, 41
  - core\_portable, 43
  - CORE\_TICKS, 43
  - CORETIMETYPE, 41
  - default\_num\_contexts, 45
  - ee\_f32, 43
  - ee\_printf, 45
  - ee\_ptr\_int, 44
  - ee\_s16, 44
  - ee\_s32, 44
  - ee\_size\_t, 44
  - ee\_u16, 44
  - ee\_u32, 44
  - ee\_u8, 44
  - HAS\_FLOAT, 41
  - HAS\_PRINTF, 41
  - HAS\_STDIO, 41
  - HAS\_TIME\_H, 41
  - MAIN\_HAS\_NOARGC, 42
  - MAIN\_HAS\_NORETURN, 42
  - MEM\_LOCATION, 42



- MEM\_METHOD, 42
- MULTITHREAD, 42
- NULL, 42
- portable\_fini, 45
- portable\_init, 45
- SEED\_METHOD, 42
- USE\_CLOCK, 42
- USE\_FORK, 43
- USE\_PTHREAD, 43
- USE\_SOCKET, 43
- VALIDATION\_RUN, 43
- core\_results
  - coremark.h, 67
- CORE\_S1
  - coremark.h, 69
- CORE\_S2
  - coremark.h, 69
- CORE\_SCIENTIFIC
  - coremark.h, 69
- CORE\_START
  - coremark.h, 69
- CORE\_STATE
  - coremark.h, 68
- core\_state.c
  - core\_bench\_state, 60
  - core\_init\_state, 60
  - core\_state\_transition, 60
  - ee\_isdigit, 60
  - errpat, 61
  - floatpat, 61
  - intpat, 61
  - scipat, 61
- core\_state\_e
  - coremark.h, 67
- core\_state\_transition
  - core\_state.c, 60
- CORE\_TICKS
  - core\_portme.h, 43
- core\_util.c
  - check\_data\_types, 62
  - crc16, 62
  - crcu16, 63
  - crcu32, 63
  - crcu8, 63
  - get\_seed\_args, 63
  - parseval, 63
- coremark.h
  - ALL\_ALGORITHMS\_MASK, 65
  - check\_data\_types, 69
  - core\_bench\_list, 69
  - core\_bench\_matrix, 69
  - core\_bench\_state, 69
  - CORE\_EXPONENT, 69
  - CORE\_FLOAT, 69
  - core\_init\_matrix, 69
  - core\_init\_state, 70
  - CORE\_INT, 69
  - CORE\_INVALID, 69
  - core\_list\_init, 70
  - core\_results, 67
  - CORE\_S1, 69
  - CORE\_S2, 69
  - CORE\_SCIENTIFIC, 69
  - CORE\_START, 69
  - CORE\_STATE, 68
  - core\_state\_e, 67
  - crc16, 70
  - crcu16, 70
  - crcu32, 70
  - crcu8, 70
  - get\_time, 71
  - ID\_LIST, 65
  - ID\_MATRIX, 66
  - ID\_STATE, 66
  - iterate, 71
  - list\_data, 68
  - list\_head, 68
  - MAIN\_RETURN\_TYPE, 66
  - MAIN\_RETURN\_VAL, 66
  - mat\_params, 68
  - MATDAT, 68
  - MATDAT\_INT, 66
  - MATRES, 68
  - MEM\_MALLOC, 66
  - MEM\_STACK, 66
  - MEM\_STATIC, 66
  - NUM\_ALGORITHMS, 67
  - NUM\_CORE\_STATES, 69
  - parseval, 71
  - portable\_free, 71
  - portable\_malloc, 71
  - secs\_ret, 68
  - SEED\_ARG, 67
  - SEED\_FUNC, 67
  - SEED\_VOLATILE, 67
  - start\_time, 71
  - stop\_time, 71
  - time\_in\_secs, 72
  - TOTAL\_DATA\_SIZE, 67
- coremark/core\_portme.c, 35
- coremark/core\_portme.h, 39
- coremark/ee\_printf.c, 46
- coremark/README.md, 50
- coremark/src/core\_list\_join.c, 50
- coremark/src/core\_main.c, 53
- coremark/src/core\_matrix.c, 56
- coremark/src/core\_state.c, 59
- coremark/src/core\_util.c, 62
- coremark/src/coremark.h, 64
- CORETIMETYPE
  - core\_portme.h, 41
- crc
  - RESULTS\_S, 25
- crc16
  - core\_util.c, 62
  - coremark.h, 70

- crclist
  - RESULTS\_S, 25
- crcmatrix
  - RESULTS\_S, 25
- crcstate
  - RESULTS\_S, 25
- crcu16
  - core\_util.c, 63
  - coremark.h, 70
- crcu32
  - core\_util.c, 63
  - coremark.h, 70
- crcu8
  - core\_util.c, 63
  - coremark.h, 70
- DARKIO, 15
  - board\_cm, 16
  - board\_id, 16
  - core\_id, 16
  - gpio, 16
  - gpio\_ctrl, 16
  - i2c, 17
  - irq, 17
  - led, 17
  - spi, 17
  - timer, 17
  - timeus, 17
  - uart, 17
- DARKIO::DARKUART, 18
  - baud, 18
  - fifo, 18
  - stat, 18
- darklibc/i2c.c, 72
- darklibc/i2c\_old.c, 72
- darklibc/include/i2c.h, 73
- darklibc/include/io.h, 74
- darklibc/include/spi.h, 78
- darklibc/include/stdio.h, 83
- darklibc/io.c, 88
- darklibc/spi.c, 90
- darklibc/stdio.c, 90
- darkshell/main.c, 33
- darkshell/README.md, 50
- data
  - I2C\_RegisterBits, 20
  - programBoard, 13
- data16
  - list\_data\_s, 21
- data\_received
  - SPI\_RegisterBits, 28
- data\_to\_send
  - SPI\_RegisterBits, 28
- default\_num\_contexts
  - core\_portme.c, 38
  - core\_portme.h, 45
- digits
  - ee\_printf.c, 49
- eaddr
  - ee\_printf.c, 48
- EBREAK
  - stdio.h, 84
- ee\_f32
  - core\_portme.h, 43
- ee\_isdigit
  - core\_state.c, 60
- ee\_printf
  - core\_portme.h, 45
  - ee\_printf.c, 48
- ee\_printf.c
  - digits, 49
  - eaddr, 48
  - ee\_printf, 48
  - ee\_vsprintf, 48
  - HEX\_PREP, 47
  - iaddr, 48
  - is\_digit, 47
  - LEFT, 47
  - number, 48
  - PLUS, 47
  - SIGN, 47
  - skip\_atoi, 49
  - SPACE, 47
  - strnlen, 49
  - uart\_send\_char, 49
  - upper\_digits, 49
  - UPPERCASE, 47
  - ZEROPAD, 48
- ee\_ptr\_int
  - core\_portme.h, 44
- ee\_s16
  - core\_portme.h, 44
- ee\_s32
  - core\_portme.h, 44
- ee\_size\_t
  - core\_portme.h, 44
- EE\_TICKS\_PER\_SEC
  - core\_portme.c, 36
- ee\_u16
  - core\_portme.h, 44
- ee\_u32
  - core\_portme.h, 44
- ee\_u8
  - core\_portme.h, 44
- ee\_vsprintf
  - ee\_printf.c, 48
- empty
  - SPI\_RegisterBits, 28
- EOF
  - stdio.h, 84
- err
  - RESULTS\_S, 25
- errpat
  - core\_state.c, 61
- execs
  - RESULTS\_S, 25

- fields
  - I2C\_Register, 19
  - SPI\_Register, 27
- fifo
  - DARKIO::DARKUART, 18
- floatpat
  - core\_state.c, 61
- get\_mepc
  - io.h, 76
- get\_mie
  - io.h, 76
- get\_mtvec
  - io.h, 76
- get\_seed
  - core\_main.c, 54
- get\_seed\_32
  - core\_main.c, 54
- get\_seed\_args
  - core\_main.c, 54
  - core\_util.c, 63
- get\_time
  - core\_portme.c, 37
  - coremark.h, 71
- getchar
  - stdio.h, 85
- GETMYTIME
  - core\_portme.c, 36
- gets
  - stdio.c, 92
  - stdio.h, 85
- gpio
  - DARKIO, 16
- gpio\_ctrl
  - DARKIO, 16
- HAS\_FLOAT
  - core\_portme.h, 41
- HAS\_PRINTF
  - core\_portme.h, 41
- HAS\_STDIO
  - core\_portme.h, 41
- HAS\_TIME\_H
  - core\_portme.h, 41
- HEX\_PREP
  - ee\_printf.c, 47
- i2c
  - DARKIO, 17
- i2c.h
  - i2cReadByte, 73
  - i2cSendByte, 74
- I2C\_Register, 19
  - fields, 19
  - raw, 19
- I2C\_RegisterBits, 19
  - busy, 20
  - data, 20
  - n\_bytes, 20
  - nack, 20
  - req\_data, 20
  - reserved, 20
  - slaveAddress, 21
  - start, 21
  - subaddress, 21
- i2cReadByte
  - i2c.h, 73
- i2cSendByte
  - i2c.h, 74
- iaddr
  - ee\_printf.c, 48
- ID\_LIST
  - coremark.h, 65
- ID\_MATRIX
  - coremark.h, 66
- ID\_STATE
  - coremark.h, 66
- idx
  - list\_data\_s, 21
- info
  - list\_head\_s, 22
- intpat
  - core\_state.c, 61
- io
  - io.c, 89
  - io.h, 78
- io.c
  - \_\_attribute\_\_, 88
  - board\_name, 88
  - io, 89
  - kmem, 89
  - mac, 89
  - utimers, 89
- io.h
  - \_\_attribute\_\_, 76
  - \_data, 77
  - \_edata, 77
  - \_etext, 77
  - \_stack, 77
  - \_text, 78
  - banner, 76
  - board\_name, 76
  - check4rv32i, 76
  - get\_mepc, 76
  - get\_mie, 76
  - get\_mtvec, 76
  - io, 78
  - IRQ\_TIMR, 75
  - IRQ\_UART, 75
  - kmem, 78
  - set\_mepc, 77
  - set\_mie, 77
  - set\_mtvec, 77
  - utimers, 78
- irq
  - DARKIO, 17
- IRQ\_TIMR

- io.h, 75
- IRQ\_UART
  - io.h, 75
- is\_digit
  - ee\_printf.c, 47
- iterate
  - core\_main.c, 54
  - coremark.h, 71
- iterations
  - RESULTS\_S, 25
- kmem
  - io.c, 89
  - io.h, 78
- led
  - DARKIO, 17
- LEFT
  - ee\_printf.c, 47
- list
  - RESULTS\_S, 25
- list\_cmp
  - core\_list\_join.c, 51
- list\_data
  - coremark.h, 68
- list\_data\_s, 21
  - data16, 21
  - idx, 21
- list\_head
  - coremark.h, 68
- list\_head\_s, 22
  - info, 22
  - next, 22
- list\_known\_crc
  - core\_main.c, 55
- mac
  - io.c, 89
  - stdio.h, 85
- main
  - core\_main.c, 54
  - main.c, 32–34
- main.c
  - main, 32–34
  - wait, 33
  - wait\_100us, 33
- MAIN\_HAS\_NOARGC
  - core\_portme.h, 42
- MAIN\_HAS\_NORETURN
  - core\_portme.h, 42
- MAIN\_RETURN\_TYPE
  - coremark.h, 66
- MAIN\_RETURN\_VAL
  - coremark.h, 66
- mandelbrot/main.c, 34
- mat
  - RESULTS\_S, 26
- mat\_params
  - coremark.h, 68
- MAT\_PARAMS\_S, 23
  - A, 23
  - B, 23
  - C, 23
  - N, 23
- MATDAT
  - coremark.h, 68
- MATDAT\_INT
  - coremark.h, 66
- MATRES
  - coremark.h, 68
- matrix\_add\_const
  - core\_matrix.c, 58
- matrix\_big
  - core\_matrix.c, 57
- matrix\_clip
  - core\_matrix.c, 57
- matrix\_known\_crc
  - core\_main.c, 55
- matrix\_mul\_const
  - core\_matrix.c, 58
- matrix\_mul\_matrix
  - core\_matrix.c, 58
- matrix\_mul\_matrix\_bitextract
  - core\_matrix.c, 58
- matrix\_mul\_vect
  - core\_matrix.c, 58
- matrix\_sum
  - core\_matrix.c, 59
- matrix\_test
  - core\_matrix.c, 59
- matrix\_test\_next
  - core\_matrix.c, 57
- MEM\_LOCATION
  - core\_portme.h, 42
- MEM\_MALLOC
  - coremark.h, 66
- MEM\_METHOD
  - core\_portme.h, 42
- mem\_name
  - core\_main.c, 55
- MEM\_STACK
  - coremark.h, 66
- MEM\_STATIC
  - coremark.h, 66
- memblock
  - RESULTS\_S, 26
- memcpy
  - stdio.c, 93
  - stdio.h, 85
- memset
  - stdio.c, 93
  - stdio.h, 85
- MULTITHREAD
  - core\_portme.h, 42
- MYTIMEDIFF
  - core\_portme.c, 36
- N

- MAT\_PARAMS\_S, 23
- n\_bytes
  - I2C\_RegisterBits, 20
- n\_bytes\_received
  - SPI\_RegisterBits, 28
- n\_bytes\_to\_send
  - SPI\_RegisterBits, 29
- nack
  - I2C\_RegisterBits, 20
- next
  - list\_head\_s, 22
- NUL
  - stdio.h, 84
- NULL
  - core\_portme.h, 42
  - stdio.h, 84
- NUM\_ALGORITHMS
  - coremark.h, 67
- NUM\_CORE\_STATES
  - coremark.h, 69
- number
  - ee\_printf.c, 48
- parseval
  - core\_util.c, 63
  - coremark.h, 71
- PLUS
  - ee\_printf.c, 47
- port
  - RESULTS\_S, 26
- portable\_fini
  - core\_portme.c, 37
  - core\_portme.h, 45
- portable\_free
  - coremark.h, 71
- portable\_id
  - CORE\_PORTABLE\_S, 15
- portable\_init
  - core\_portme.c, 37
  - core\_portme.h, 45
- portable\_malloc
  - coremark.h, 71
- printf
  - stdio.c, 93
  - stdio.h, 86
- programBoard, 13
  - data, 13
  - ser, 13
- programBoard.py, 95
- putchar
  - stdio.h, 86
- putd
  - stdio.h, 86
- putnum
  - stdio.c, 93
- puts
  - stdio.c, 93
  - stdio.h, 86
- putstr
  - stdio.c, 93
  - stdio.h, 86
- putcx
  - stdio.h, 86
- raw
  - I2C\_Register, 19
  - SPI\_Register, 27
- README.md, 50
- req\_data
  - I2C\_RegisterBits, 20
- reserved
  - I2C\_RegisterBits, 20
- RESULTS\_S, 24
  - crc, 25
  - crclist, 25
  - crcmatrix, 25
  - crcstate, 25
  - err, 25
  - execs, 25
  - iterations, 25
  - list, 25
  - mat, 26
  - memblock, 26
  - port, 26
  - seed1, 26
  - seed2, 26
  - seed3, 26
  - size, 26
- rle
  - badapple.h, 31
- rx\_data\_ready
  - SPI\_RegisterBits, 29
- SAMPLE\_TIME\_IMPLEMENTATION
  - core\_portme.c, 36
- scipat
  - core\_state.c, 61
- secs\_ret
  - coremark.h, 68
- seed1
  - RESULTS\_S, 26
- seed1\_volatile
  - core\_portme.c, 38
- seed2
  - RESULTS\_S, 26
- seed2\_volatile
  - core\_portme.c, 38
- seed3
  - RESULTS\_S, 26
- seed3\_volatile
  - core\_portme.c, 38
- seed4\_volatile
  - core\_portme.c, 38
- seed5\_volatile
  - core\_portme.c, 38
- SEED\_ARG
  - coremark.h, 67
- SEED\_FUNC

- coremark.h, 67
- SEED\_METHOD
  - core\_portme.h, 42
- SEED\_VOLATILE
  - coremark.h, 67
- ser
  - programBoard, 13
- set\_mepc
  - io.h, 77
- set\_mie
  - io.h, 77
- set\_mtvec
  - io.h, 77
- SIGN
  - ee\_printf.c, 47
- size
  - RESULTS\_S, 26
- skip\_atoi
  - ee\_printf.c, 49
- slaveAddress
  - I2C\_RegisterBits, 21
- SPACE
  - ee\_printf.c, 47
- spi
  - DARKIO, 17
- spi.h
  - spi\_disable, 79
  - spi\_enable, 80
  - spi\_init, 80
  - spi\_read\_multiple\_bytes, 80
  - spi\_read\_single\_byte, 80
  - spi\_send\_receive\_data, 81
  - spi\_set\_clock\_frequency, 81
  - spi\_set\_data\_mode, 81
  - spi\_transaction\_single\_byte, 81
  - spi\_write\_multiple\_bytes, 82
  - spi\_write\_read\_multiple\_bytes, 82
  - spi\_write\_read\_single\_byte, 82
  - spi\_write\_single\_byte, 83
- spi\_disable
  - spi.h, 79
- spi\_enable
  - spi.h, 80
- spi\_init
  - spi.h, 80
- spi\_read\_multiple\_bytes
  - spi.h, 80
- spi\_read\_single\_byte
  - spi.h, 80
- SPI\_Register, 27
  - fields, 27
  - raw, 27
- SPI\_RegisterBits, 28
  - data\_received, 28
  - data\_to\_send, 28
  - empty, 28
  - n\_bytes\_received, 28
  - n\_bytes\_to\_send, 29
  - rx\_data\_ready, 29
  - start, 29
  - tx\_ready, 29
- spi\_send\_receive\_data
  - spi.h, 81
- spi\_set\_clock\_frequency
  - spi.h, 81
- spi\_set\_data\_mode
  - spi.h, 81
- spi\_transaction\_single\_byte
  - spi.h, 81
- spi\_write\_multiple\_bytes
  - spi.h, 82
- spi\_write\_read\_multiple\_bytes
  - spi.h, 82
- spi\_write\_read\_single\_byte
  - spi.h, 82
- spi\_write\_single\_byte
  - spi.h, 83
- start
  - I2C\_RegisterBits, 21
  - SPI\_RegisterBits, 29
- start\_time
  - core\_portme.c, 37
  - coremark.h, 71
- start\_time\_val
  - core\_portme.c, 39
- stat
  - DARKIO::DARKUART, 18
- state\_known\_crc
  - core\_main.c, 55
- static\_memblk
  - core\_main.c, 55
- stdio.c
  - \_\_div\_mod\_si3, 91
  - \_\_divsi3, 91
  - \_\_modsi3, 91
  - \_\_mulsi3, 91
  - \_\_udiv\_umod\_si3, 92
  - \_\_udivsi3, 92
  - \_\_umodsi3, 92
  - \_\_umulsi3, 92
  - atoi, 92
  - gets, 92
  - memcpy, 93
  - memset, 93
  - printf, 93
  - putnum, 93
  - puts, 93
  - putstr, 93
  - strcmp, 94
  - strlen, 94
  - strncmp, 94
  - strtok, 94
  - usleep, 94
  - xtoi, 94
- stdio.h
  - atoi, 85

- EBREAK, [84](#)
- EOF, [84](#)
- getchar, [85](#)
- gets, [85](#)
- mac, [85](#)
- memcpy, [85](#)
- memset, [85](#)
- NUL, [84](#)
- NULL, [84](#)
- printf, [86](#)
- putchar, [86](#)
- putd, [86](#)
- puts, [86](#)
- putstr, [86](#)
- putx, [86](#)
- strcmp, [87](#)
- strlen, [87](#)
- strncmp, [87](#)
- strtok, [87](#)
- usleep, [87](#)
- xtoi, [87](#)
- stop\_time
  - core\_portme.c, [37](#)
  - coremark.h, [71](#)
- stop\_time\_val
  - core\_portme.c, [39](#)
- strcmp
  - stdio.c, [94](#)
  - stdio.h, [87](#)
- strlen
  - stdio.c, [94](#)
  - stdio.h, [87](#)
- strncmp
  - stdio.c, [94](#)
  - stdio.h, [87](#)
- strlen
  - ee\_printf.c, [49](#)
- strtok
  - stdio.c, [94](#)
  - stdio.h, [87](#)
- subaddress
  - I2C\_RegisterBits, [21](#)
- time\_in\_secs
  - core\_portme.c, [38](#)
  - coremark.h, [72](#)
- timer
  - DARKIO, [17](#)
- TIMER\_RES\_DIVIDER
  - core\_portme.c, [36](#)
- timeus
  - DARKIO, [17](#)
- TOTAL\_DATA\_SIZE
  - coremark.h, [67](#)
- tx\_ready
  - SPI\_RegisterBits, [29](#)
- uart
  - DARKIO, [17](#)
  - uart\_send\_char
    - ee\_printf.c, [49](#)
  - upper\_digits
    - ee\_printf.c, [49](#)
  - UPPERCASE
    - ee\_printf.c, [47](#)
  - USE\_CLOCK
    - core\_portme.h, [42](#)
  - USE\_FORK
    - core\_portme.h, [43](#)
  - USE\_PTHREAD
    - core\_portme.h, [43](#)
  - USE\_SOCKET
    - core\_portme.h, [43](#)
  - usleep
    - stdio.c, [94](#)
    - stdio.h, [87](#)
  - utimers
    - io.c, [89](#)
    - io.h, [78](#)
- VALIDATION\_RUN
  - core\_portme.h, [43](#)
- wait
  - main.c, [33](#)
- wait\_100us
  - main.c, [33](#)
- xtoi
  - stdio.c, [94](#)
  - stdio.h, [87](#)
- ZEROPAD
  - ee\_printf.c, [48](#)