DarkRISCV Software Environment

Generated by Doxygen 1.8.17

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
`ifdef __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]crcmatrix     : 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 funcion 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 funcions.

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:

Chapter 5

Data Structure Index

5.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

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`

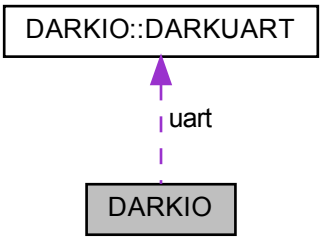
The documentation for this struct was generated from the following file:

- 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
```

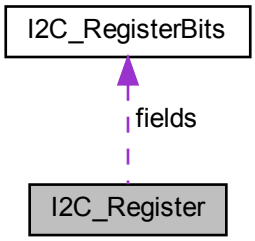
The documentation for this struct was generated from the following file:

- 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
```

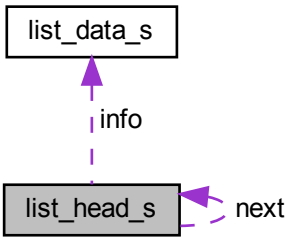
The documentation for this struct was generated from the following file:

- 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
```

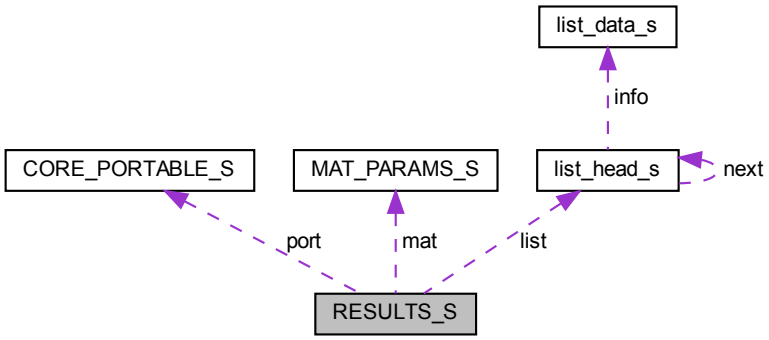
The documentation for this struct was generated from the following file:

- 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
```

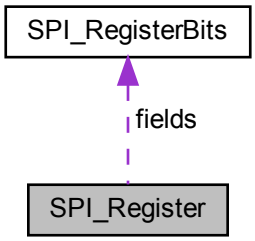
The documentation for this struct was generated from the following file:

- 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:

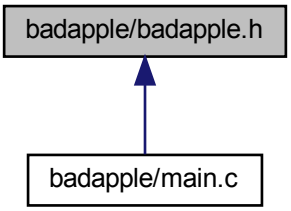
- darklibc/include/spi.h

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

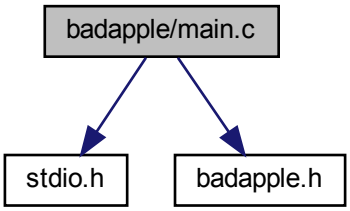
```
unsigned char rle[]
```

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

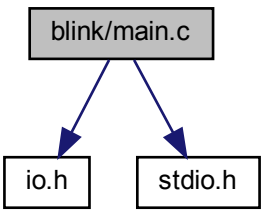
9.3.1.1 main()

```
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 )
```

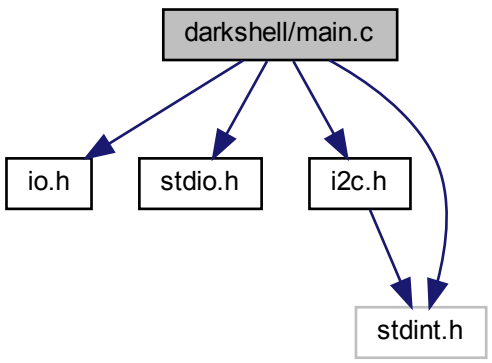
9.4.1.3 wait_100us()

```
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

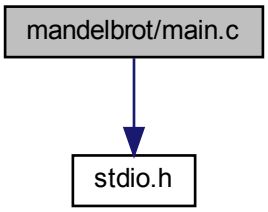
9.5.1.1 main()

```
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

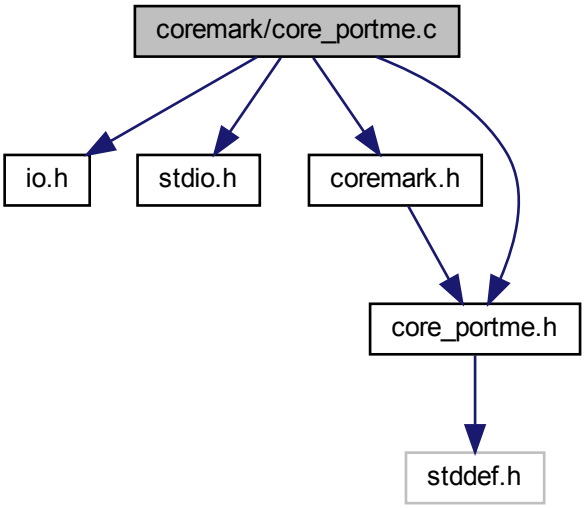
9.6.1.1 main()

```
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.

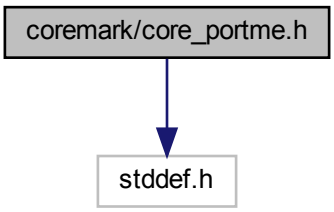
9.7.3.8 stop_time_val

```
CORETIMETYPE stop_time_val  [static]
```

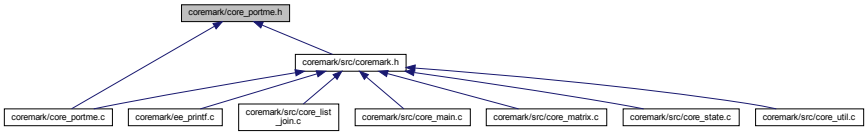
9.8 coremark/core_portme.h File Reference

```
#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

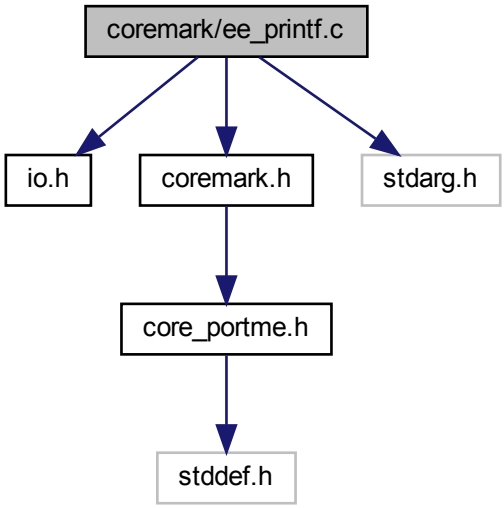
9.8.4.1 default_num_contexts

```
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 strnlen (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 strnlen()

```
static ee_size_t strnlen (
             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

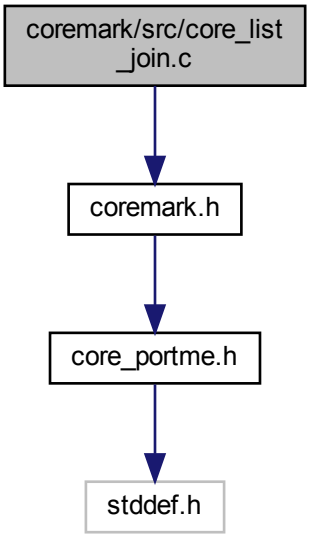
9.11 darkshell/README.md File Reference

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_s16 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_s16 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 )
```

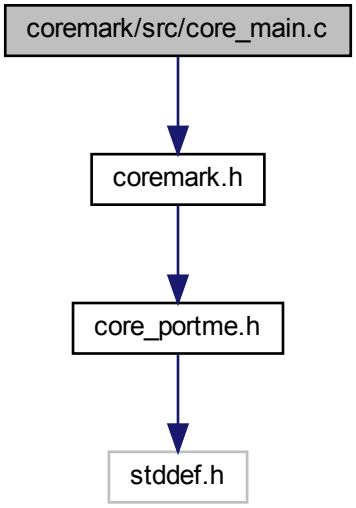
9.13.2.12 core_list_undo_remove()

```
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:

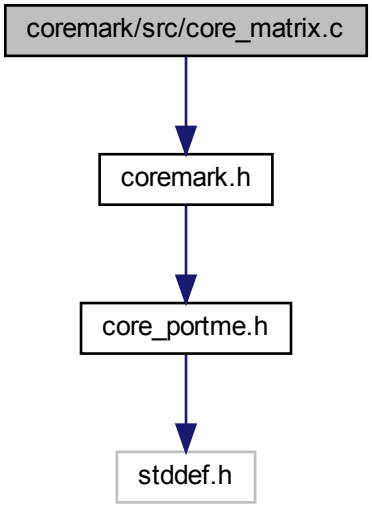
```
= { (ee_u16)0x5e47,
                                    (ee_u16)0x39bf,
                                    (ee_u16)0xe5a4,
                                    (ee_u16)0x8e3a,
                                    (ee_u16)0x8d84 }
```

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)&0x0ffff)
- #define matrix_big(x) (0xf000 | (x))
- #define bit_extract(x, from, to) (((x) >> (from)) & (~(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)&0x0ff :  (x)&0x0ffff)
```

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 )
```

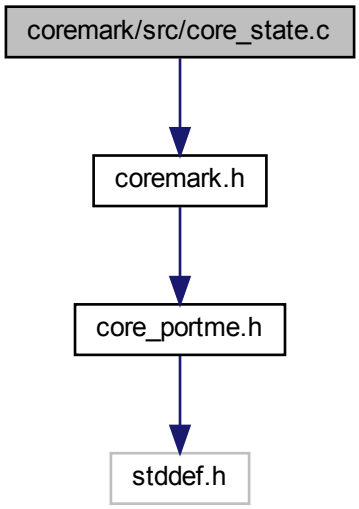
9.15.2.9 matrix_test()

```
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]
```

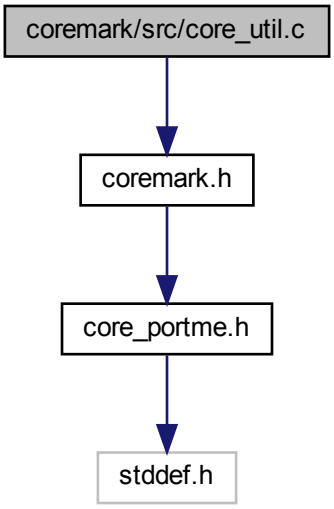
Initial value:

```
= { (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[] )
```

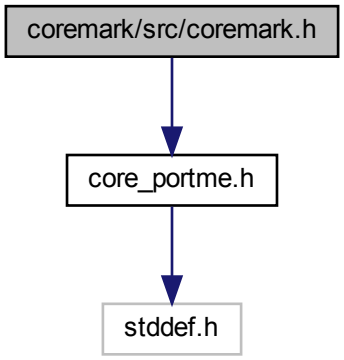
9.17.1.7 parseval()

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

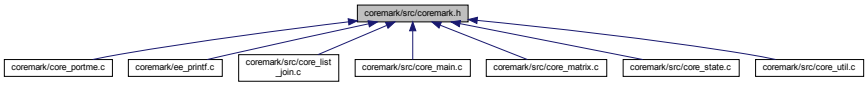
9.18 coremark/src/coremark.h File Reference

```
#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 *memblk, 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  )
```

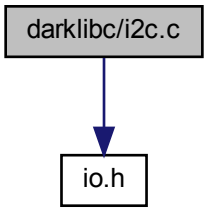
9.18.4.19 time_in_secs()

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

9.19 darklibc/i2c.c File Reference

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

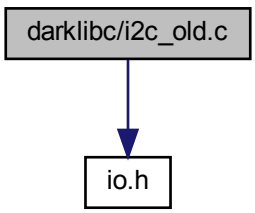
Include dependency graph for i2c.c:



9.20 darklibc/i2c_old.c File Reference

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

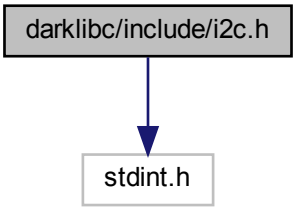
Include dependency graph for i2c_old.c:



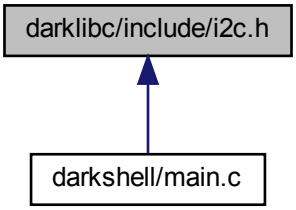
9.21 darklibc/include/i2c.h File Reference

```
#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

slaveAddress	The I2C slave device address.
subaddress	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

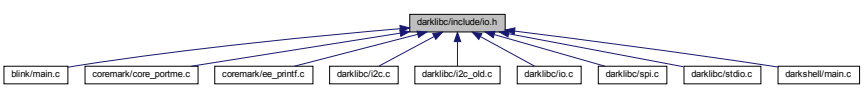
slaveAddress	The I2C slave device address.
subaddress	The register/subaddress within the slave device.
byteToSend	The byte of data to be transmitted.

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(*f)(void))
- void set_mepc (void(*f)(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]
```

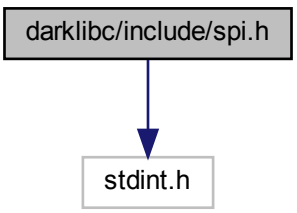
9.22.3.8 utimers

```
volatile int utimers
```

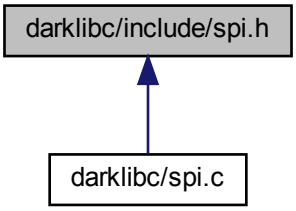
9.23 darklibc/include/spi.h File Reference

```
#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

rx_data	Pointer to the buffer where the received data will be stored.
length	The number of bytes to read from the SPI interface.
slave	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

length	The number of bytes to be in the complete transaction. Should be 1 when not called as part of a multi-byte transaction.
slave	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

new_spi_register	[SPI_Register] 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

data	Pointer to the array of bytes to be sent.
length	Number of bytes to be sent.
slave	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

tx_data	Pointer to the data to be transmitted.
rx_data	Pointer to the buffer where received data will be stored.
length	Number of bytes to be transmitted and received.
slave	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

data	The byte of data to be transmitted.
length	The number of bytes to be in the transaction.
length	The number of bytes to be in the complete transaction. Should be 1 when not called as part of a multi-byte transaction.
slave	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.

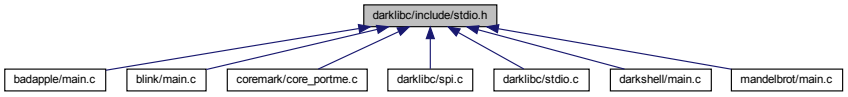
This function sends a single byte of data over the SPI bus.

Parameters

data	The byte of data to be sent.
length	The number of bytes to be in the complete transaction. Should be 1 when not called as part of a multi-byte transaction.
slave	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()

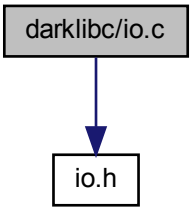
```
void usleep (
            int  )
```

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"
```

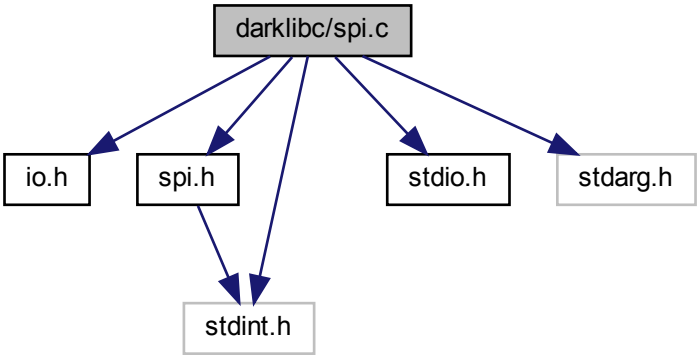
9.25.2.3 utimers

```
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>
```

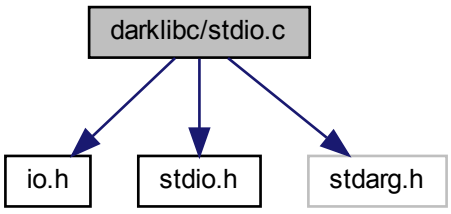
Include dependency graph for spi.c:



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