MPX Thunder Krakens

R1

Generated by Doxygen 1.8.11

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 function_name Struct Reference

Public Attributes

- char * **nameStr**
- int(* **function**)(int argc, char **argv)
- char * **usage**
- char * **help**

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

- modules/r1/r1.h

Chapter 4

File Documentation

4.1 include/core/serial.h File Reference

Serial - Header.

Macros

- #define **COM1** 0x3f8
- #define **COM2** 0x2f8
- #define **COM3** 0x3e8
- #define **COM4** 0x2e8
- #define **WithoutEcho** 0
- #define **WithEcho** 1

Functions

- int **init_serial** (int device)
- int **serial_println** (const char ∗msg)
- int **serial_print** (const char ∗msg)
- int **set_serial_out** (int device)
- int **set_serial_in** (int device)
- void get_input_line (char ∗buffer, const int buffer_size, const int bWithEcho)

  get_input_line.

4.1.1 Detailed Description

Serial - Header.

Author

Thunder Krakens

Date

February 2nd, 2016

Version

R1

4.1.2 Function Documentation

4.1.2.1 void get_input_line (char * *buffer*, const int *buffer_size*, const int *bWithEcho*)

get_input_line.

Description: Get user's input from keyborad.

Parameters

buffer	- The pointer to the buffer where store the user's input.
buffer_size	- The size of that buffer.
bWithEcho	- With echo or not

Returns

VOID

4.2 include/string.h File Reference

String Handling - Header.

```
#include <system.h>
```

Functions

- int isspace (const char *c)
  Name: isspace.
- void * memset (void *s, int c, size_t n)
  Name: memset.
- char * strcpy (char *s1, const char *s2)
  Name: strcpy.
- char * strcat (char *s1, const char *s2)
  Name: strcat.
- int strlen (const char *s)
  Name: strlen.
- int strcmp (const char *s1, const char *s2)
  Name: strcmp.
- char * strtok (char *s1, const char *s2)
  Name: strtok.
- int atoi (const char *s)
  Name: atoi.
- int sprintf (char *str, const char *format,...)
  Name: sprintf.
- int printf (const char *format,...)
  Name: printf.

4.2.1 Detailed Description

String Handling - Header.

Author

Thunder Krakens

Date

February 2nd, 2016

Version

R1

4.2.2 Function Documentation

4.2.2.1 int atoi (const char * s)

Name: atoi.

Description: Convert an ASCII string to an integer.

Parameters

const	char *s - String.

Returns

integer - The converted integer.

4.2.2.2 int isspace (const char * c)

Name: isspace.

Description: Identifies if its space

Parameters

const	char *c - A constant character

4.2.2.3 void* memset (void * s, int c, size_t n)

Name: memset.

Description: Sets region of memory

Parameters

s	- destination
c	- byte to write
n	- count

Returns

VOID

4.2.2.4 int printf (const char * *format,* ...)

Name: printf.

Description: Print out a formatted string.

%[-x]c output a character, '-' - align right, x - the output width

%[-x]s output a string, '-' - align right, x - the output width

%[{-,+}x]d output a character, '-' - align right, '+' - align right and display '+' sign, x - the output width

%[-x]X (capital 'X') output a hexadecimal number, '-' - align right, x - the output width

Note

Output width will be ignored if width is smaller than actual length.

Parameters

str	- Output string.
format	- The format of the string.
...	- All of the additional parameters.

Returns

vsprintf(str, format, ap) - Return the string with its format and pointer.

4.2.2.5 int sprintf (char * *str,* const char * *format,* ...)

Name: sprintf.

Description: Generate a formatted string.

%[-x]c output a character, '-' - align right, x - the output width

%[-x]s output a string, '-' - align right, x - the output width

%[{-,+}x]d output a character, '-' - align right, '+' - align right and display '+' sign, x - the output width

%[-x]X (capital 'X') output a hexadecimal number, '-' - align right, x - the output width

Note

Output width will be ignored if width is smaller than actual length.

Parameters

str	- Output string.
format	- The format of the string.
...	- All of the additional parameters.

Returns

vsprintf(str, format, ap) - Return the string with its format and pointer.

4.2.2.6 char* strcat (char * *s1*, const char * *s2*)

Name: strcat.

Description: Concatenate the contents of one string onto another.

Parameters

s1	- Destination string
s2	- Source string

Returns

s1 - Destination String

4.2.2.7 int strcmp (const char * *s1*, const char * *s2*)

Name: strcmp.

Description: String comparison.

Parameters

s1	- First string to use for the compare.
s2	- Second string to use for the compare.

Returns

whether they are the same or not.

4.2.2.8 char* strcpy (char * *s1*, const char * *s2*)

Name: strcpy.

Description: Copies one string to another.

Parameters

s1	- Destination string
s2	- Source string

Returns

s1 - Destination String

4.2.2.9 int strlen (const char * *s*)

Name: strlen.

Description: Returns the length of a string.

Parameters

s	- String input.

Returns

count - Length of the String

4.2.2.10 char* strtok (char * *s1*, const char * *s2*)

Name: strtok.

Description: Split string into tokens.

Parameters

s1	- String
s2	- Delimiter

4.3 kernel/core/serial.c File Reference

Serial.

```
#include <stdint.h>
#include <string.h>
#include <core/io.h>
#include <core/serial.h>
```

Macros

- #define **NO_ERROR** 0
- #define **ESC_KEY** 27
- #define **BRACKET_KEY** 91
- #define **ENTER_KEY** 13
- #define **BACKSPACE_KEY** 127
- #define **DEL_KEY_SEQ_3** 51
- #define **DEL_KEY_SEQ_4** 126
- #define **UP_ARROW** 65
- #define **DOWN_ARROW** 66
- #define **RIGHT_ARROW** 67
- #define **LEFT_ARROW** 68

Functions

- int **init_serial** (int device)
- int **serial_println** (const char ∗msg)
- int **serial_print** (const char ∗msg)
- int **set_serial_out** (int device)
- int **set_serial_in** (int device)
- void get_input_line (char ∗buffer, const int buffer_size, const int bWithEcho)

  get_input_line.

Variables

- int **serial_port_out** = 0
- int **serial_port_in** = 0

4.3.1 Detailed Description

Serial.

Author

Thunder Krakens

Date

February 2nd, 2016

Version

R1

4.3.2 Function Documentation

4.3.2.1 void get_input_line (char ∗ *buffer,* const int *buffer_size,* const int *bWithEcho*)

get_input_line.

Description: Get user's input from keyborad.

Parameters

buffer	- The pointer to the buffer where store the user's input.
buffer_size	- The size of that buffer.
bWithEcho	- With echo or not

Returns

VOID

4.4 lib/string.c File Reference

String Handling.

```
#include <system.h>
#include <core/serial.h>
#include "../modules/mpx_supt.h"
#include <string.h>
```

Functions

- int strlen (const char *s)

  Name: strlen.
- char * strcpy (char *s1, const char *s2)

  Name: strcpy.
- int atoi (const char *s)

  Name: atoi.
- int strcmp (const char *s1, const char *s2)

  Name: strcmp.
- int sprintf (char *str, const char *format,...)

  Name: sprintf.
- int printf (const char *format,...)

  Name: printf.
- char * strcat (char *s1, const char *s2)

  Name: strcat.
- int isspace (const char *c)

  Name: isspace.
- void * memset (void *s, int c, size_t n)

  Name: memset.
- char * strtok (char *s1, const char *s2)

  Name: strtok.

4.4.1 Detailed Description

String Handling.

Author

Thunder Krakens

Date

February 2nd, 2016

Version

R1

4.4.2 Function Documentation

4.4.2.1 int atoi (const char * s)

Name: atoi.

Description: Convert an ASCII string to an integer.

Parameters

const	char *s - String.

Returns

integer - The converted integer.

4.4.2.2 int isspace (const char * c)

Name: isspace.

Description: Identifies if its space

Parameters

const	char *c - A constant character

4.4.2.3 void* memset (void * s, int c, size_t n)

Name: memset.

Description: Sets region of memory

Parameters

s	- destination
c	- byte to write
n	- count

Returns

VOID

4.4.2.4 int printf (const char * *format,* ...)

Name: printf.

Description: Print out a formatted string.

%[-x]c output a character, '-' - align right, x - the output width

%[-x]s output a string, '-' - align right, x - the output width

%[{-,+}x]d output a character, '-' - align right, '+' - align right and display '+' sign, x - the output width

%[-x]X (capital 'X') output a hexadecimal number, '-' - align right, x - the output width

Note

Output width will be ignored if width is smaller than actual length.

Parameters

str	- Output string.
format	- The format of the string.
...	- All of the additional parameters.

Returns

vsprintf(str, format, ap) - Return the string with its format and pointer.

4.4.2.5 int sprintf (char * *str,* const char * *format,* ...)

Name: sprintf.

Description: Generate a formatted string.

%[-x]c output a character, '-' - align right, x - the output width

%[-x]s output a string, '-' - align right, x - the output width

%[{-,+}x]d output a character, '-' - align right, '+' - align right and display '+' sign, x - the output width

%[-x]X (capital 'X') output a hexadecimal number, '-' - align right, x - the output width

Note

Output width will be ignored if width is smaller than actual length.

Parameters

str	- Output string.
format	- The format of the string.
...	- All of the additional parameters.

Returns

vsprintf(str, format, ap) - Return the string with its format and pointer.

4.4.2.6 char* strcat (char * *s1*, const char * *s2*)

Name: strcat.

Description: Concatenate the contents of one string onto another.

Parameters

s1	- Destination string
s2	- Source string

Returns

s1 - Destination String

4.4.2.7 int strcmp (const char * *s1*, const char * *s2*)

Name: strcmp.

Description: String comparison.

Parameters

s1	- First string to use for the compare.
s2	- Second string to use for the compare.

Returns

whether they are the same or not.

4.4.2.8 char* strcpy (char * *s1*, const char * *s2*)

Name: strcpy.

Description: Copies one string to another.

Parameters

s1	- Destination string
s2	- Source string

Returns

s1 - Destination String

4.4.2.9 int strlen (const char * *s*)

Name: strlen.

Description: Returns the length of a string.

Parameters

s	- String input.

Returns

count - Length of the String

4.4.2.10 char* strtok (char * *s1*, const char * *s2*)

Name: strtok.

Description: Split string into tokens.

Parameters

s1	- String
s2	- Delimiter

4.5 modules/errno.h File Reference

Type of Erros.

Macros

- #define **E_NOERROR** 0 /* No errors */
- #define **E_INVPARA** 1 /* Invalid parameters had been passed in */
- #define **E_INVSTRF** 2 /* Invalid [Input] string format */

Typedefs

- typedef unsigned int error_t

  Name: error_t.

4.5.1 Detailed Description

Type of Erros.

Author

Thunder Krakens

Date

February 2nd, 2016

Version

R1

This file contains the type of errors the the user may input. The error can be from invalid paramter passed to a function, or invalid input format.

4.5.2 Typedef Documentation

4.5.2.1 typedef unsigned int error_t

Name: error_t.

Description: This file contains the type of errors the the user may input.

4.6 modules/r1/r1.c File Reference

The main code file for Module R1.

```
#include "r1.h"
#include "../mpx_supt.h"
#include "sys_clock.h"
#include <string.h>
#include <core/serial.h>
#include <core/io.h>
```

Macros

- #define **USER_INPUT_BUFFER_SIZE** 1000
- #define **MAX_ARGC** 50
- #define **MOD_VERSION** "R1"
- #define **COMPLETION** "02/05/2016"

Enumerations

- enum CommandPaserStat { **NotWriting**, **NormalWriting**, **DoubleQuoteWriting**, **SingleQuoteWriting** }

  Name: CommandParserStat.

Functions

- int commhand ()

  Name: commhand.
- void command_line_parser (const char *CmdStr, int *argc, char **argv, const int MaxArgNum, const int MaxStrLen)

  Name: command_line_parser.

4.6.1 Detailed Description

The main code file for Module R1.

Author

Thunder Krakens

Date

February 2nd, 2016

Version

R1

This file contains setdate, getdate, settime, gettime, help functions, commandhander and command line parser

4.6.2 Enumeration Type Documentation

4.6.2.1 enum CommandPaserStat

Name: CommandParserStat.

Description: The stats of the command parser

4.6.3 Function Documentation

4.6.3.1 void command_line_parser (const char * *CmdStr,* int * *argc,* char ** *argv,* const int *MaxArgNum,* const int *MaxStrLen*)

Name: command_line_parser.

Description: Splits the complete command line into tokens by space, single quote, or double quote.

Parameters

CmdStr	- The complete input command.
argc	- The number of tokens found.
argv	- The array of tokens.
MaxArgNum	- The maximum number of tokens that array can hold.
MaxStrLen	- The maximum length of each token that string can hold.

Returns

void

4.6.3.2 int commhand ( )

Name: commhand.

Description: Accepts and handles commands from the user.

Parameters

User	input

Returns

0

4.7 modules/r1/r1.h File Reference

The main header for Module R1.

Classes

- struct function_name

Macros

- #define **HELP** 0
- #define **VERSION** 1
- #define **GETTIME** 2
- #define **SETTIME** 3
- #define **GETDATE** 4
- #define **SETDATE** 5
- #define **SHUTDOWN** 6
- #define **NUM_OF_FUNCTIONS** 7

Functions

- int commhand ()

  Name: commhand.
- void command_line_parser (const char *CmdStr, int *argc, char **argv, const int MaxArgNum, const int MaxStrLen)

  Name: command_line_parser.

Variables

- function_name functions [NUM_OF_FUNCTIONS]

  Name: fucntion_name.

4.7.1 Detailed Description

The main header for Module R1.

Author

Thunder Krakens

Date

February 2nd, 2016

Version

R1

This file contains setdate, getdate, settime, gettime, help functions, commandhander and command line parser

4.7.2 Function Documentation

4.7.2.1 void command_line_parser (const char * *CmdStr,* int * *argc,* char ** *argv,* const int *MaxArgNum,* const int *MaxStrLen*)

Name: command_line_parser.

Description: Splits the complete command line into tokens by space, single quote, or double quote.

Parameters

CmdStr	- The complete input command.
argc	- The number of tokens found.
argv	- The array of tokens.
MaxArgNum	- The maximum number of tokens that array can hold.
MaxStrLen	- The maximum length of each token that string can hold.

Returns

void

4.7.2.2 int commhand ( )

Name: commhand.

Description: Accepts and handles commands from the user.

Parameters

User	input

Returns

0

4.7.3 Variable Documentation

4.7.3.1 function_name functions[NUM_OF_FUNCTIONS]

Name: fucntion_name.

Description: Initializes number of fucntions

Parameters

NUM_OF_FUNCTION	- Predefined with 6 fuctions.

4.8 modules/r1/sys_clock.c File Reference

System Clock and Date.

```
#include "sys_clock.h"
#include "r1.h"
#include <string.h>
#include <core/io.h>
```

Macros

- #define **RTC_INDEX_SECOND** 0x00
- #define **RTC_INDEX_SECOND_ALARM** 0x01
- #define **RTC_INDEX_MINUTE** 0x02
- #define **RTC_INDEX_MINUTE_ALARM** 0x03

- #define **RTC_INDEX_HOUR** 0x04
- #define **RTC_INDEX_HOUR_ALARM** 0x05
- #define **RTC_INDEX_DAY_WEEK** 0x06
- #define **RTC_INDEX_DAY_MONTH** 0x07
- #define **RTC_INDEX_MONTH** 0x08
- #define **RTC_INDEX_YEAR** 0x09

Functions

- int set_time_main (int argc, char **argv)
  Name: set_time_main.
- int get_time_main (int argc, char **argv)
  Name: get_time_main.
- error_t set_time_str (const char *timeStr)
  Name: set_time_str.
- void get_time (date_time *dateTimeValues)
  Name: get_time.
- error_t set_time (const date_time *dateTimeValues)
  Name: set_time_str.
- void get_date (date_time *dateTimeValues)
  Name: get_date.
- error_t set_date (const date_time *dateTimeValues)
  Name: set_date.
- int get_date_main (int argc, char **argv)
  Name: get_date_main.
- int set_date_str (const char *str)
  Name: get_date_main.
- int set_date_main (int argc, char **argv)
  Name: set_date_str.

4.8.1 Detailed Description

System Clock and Date.

Author

Thunder Krakens

Date

February 2nd, 2016

Version

R1

The main file that manipulates and controls the system's clock.

4.8.2 Function Documentation

4.8.2.1 void get_date (date_time * *dateTimeValues*)

Name: get_date.

Description: Retrieves system's current date.

Parameters

dateTimeValues	- The value of current date

Returns

VOID

4.8.2.2 int get_date_main (int *argc,* char ** *argv*)

Name: get_date_main.

Description: Retrieves system's current date.

Parameters

argc	- The number of tokens found.
argv	- The array of tokens.

Returns

0

4.8.2.3 void get_time (date_time * *dateTimeValues*)

Name: get_time.

Description: Retrieves system's current time and date.

Parameters

dateTimeValues	- The value of current time and date

Returns

VOID

4.8.2.4 int get_time_main (int *argc,* char ** *argv*)

Name: get_time_main.

Description: Retrieves system's current time.

Parameters

argc	- The number of tokens found.
argv	- The array of tokens.

Returns

0

4.8.2.5 error_t set_date (const date_time * *dateTimeValues*)

Name: set_date.

Description: Sets the date of the system.

Parameters

dateTimeValues	- The value of current time and date

Returns

E_NOERROR - When no error was detected
E_INVPARA - Invalid Parameter

4.8.2.6 int set_date_main (int *argc,* char ** *argv*)

Name: set_date_str.

Description: Sets the date for the system by string.

Parameters

argc	- The number of tokens found.
argv	- The array of tokens.

Returns

0

4.8.2.7 int set_date_str (const char * *str*)

Name: get_date_main.

Description: Retrieves system's current date.

Parameters

argc	- The number of tokens found.

Returns

0

4.8.2.8 error_t set_time (const date_time * *dateTimeValues*)

Name: set_time_str.

Description: Sets the time for the system by string.

Parameters

timeStr	- The string type of current Time.

Returns

dateTimeValues - Returns the set time of the system
E_INVSTRF - Invalid String

4.8.2.9 int set_time_main (int *argc,* char ** *argv*)

Name: set_time_main.

Description: Sets the time for the system.

Parameters

argc	- The number of tokens found.
argv	- The array of tokens.

Returns

0

4.8.2.10 error_t set_time_str (const char * *timeStr*)

Name: set_time_str.

Description: Sets the time for the system by string.

Parameters

timeStr	- The string type of current Time.

Returns

dateTimeValues - Returns the set time of the system
E_INVSTRF - Invalid String