

Sonoma (MAXREFDES14#) Code Documentation

V01.00

Generated by Doxygen 1.8.2

Fri Aug 2 2013 16:24:18

Contents

1	Main Page	1
1.1	Introduction	1
2	Data Structure Index	3
2.1	Data Structures	3
3	File Index	5
3.1	File List	5
4	Data Structure Documentation	7
4.1	maximDateTime Struct Reference	7
4.1.1	Detailed Description	7
4.1.2	Field Documentation	7
4.1.2.1	uchDate	7
4.1.2.2	uchDow	7
4.1.2.3	uchHour	8
4.1.2.4	uchMinute	8
4.1.2.5	uchMonth	8
4.1.2.6	uchSecond	8
4.1.2.7	uchYear	8
4.2	maximProduct Struct Reference	8
4.2.1	Detailed Description	8
4.2.2	Field Documentation	8
4.2.2.1	sPartName	8
4.2.2.2	sPartNumber	9
4.2.2.3	uchKeypress	9
4.2.2.4	uchPmodType	9
5	File Documentation	11
5.1	src/maximDeviceSpecificUtilities.c File Reference	11

5.1.1	Detailed Description	11
5.1.2	Function Documentation	12
5.1.2.1	read_reg	12
5.1.2.2	write_reg	12
5.2	src/maximDeviceSpecificUtilities.h File Reference	13
5.2.1	Detailed Description	13
5.2.2	Function Documentation	13
5.2.2.1	read_reg	13
5.2.2.2	write_reg	14
5.3	src/MAXREFDES14.c File Reference	14
5.3.1	Detailed Description	15
5.3.2	Macro Definition Documentation	15
5.3.2.1	MAJOR_REVISION	15
5.3.2.2	MINOR_REVISION	15
5.3.3	Function Documentation	15
5.3.3.1	main	15
5.3.4	Variable Documentation	16
5.3.4.1	g_gpButtons	16
5.3.4.2	g_PmodPortMuxSettings	16
5.3.4.3	g_slInputString	16
5.3.4.4	g_unSevenSegmentValue	16
5.3.4.5	g_xGpioLed	16
5.3.4.6	g_xGpioPmodPortA	16
5.3.4.7	g_xGpioPmodPortB	16
5.3.4.8	g_xGpioSevenSegment	16
5.4	src/MAXREFDES14.h File Reference	17
5.4.1	Macro Definition Documentation	18
5.4.1.1	ABOUT_ONE_SECOND	18
5.4.1.2	BIT_BANG_SPI_CLK	18
5.4.1.3	BIT_BANG_SPI_CLRB	18
5.4.1.4	BIT_BANG_SPI_MISO	18
5.4.1.5	BIT_BANG_SPI_MOSI	19
5.4.1.6	BIT_BANG_SPI_SSB	19
5.4.1.7	DE	19
5.4.1.8	DEFAULT_HYPERTERMINAL_UART_ADDRESS	19
5.4.1.9	DEFAULT_HYPERTERMINAL_UART_ID	19
5.4.1.10	DI	19

5.4.1.11	INPUT_STRING_MAX_SIZE	19
5.4.1.12	PMOD_PORT_TYPE_GPIO	19
5.4.1.13	PMOD_PORT_TYPE_I2C	19
5.4.1.14	PMOD_PORT_TYPE_SPI	20
5.4.1.15	PMOD_PORT_TYPE_UART	20
5.4.1.16	PMOD_TYPE_GPIO	20
5.4.1.17	PMOD_TYPE_I2C	20
5.4.1.18	PMOD_TYPE_SPI	20
5.4.1.19	PMOD_TYPE_UART	20
5.4.1.20	RE_B	20
5.4.1.21	RO	20
5.4.2	Variable Documentation	21
5.4.2.1	g_sInputString	21
5.4.2.2	g_xGpioPmodPortA	21
5.4.2.3	g_xGpioPmodPortB	21
5.5	src/menu.c File Reference	21
5.5.1	Detailed Description	22
5.5.2	Function Documentation	22
5.5.2.1	menu_cls	22
5.5.2.2	menu_print_line	22
5.5.2.3	menu_print_main_menu	23
5.5.2.4	menu_print_maxim_banner	23
5.5.2.5	menu_print_maxim_banner_big	23
5.5.2.6	menu_print_prompt	24
5.5.2.7	menu_read_reg	24
5.5.2.8	menu_retrieve_keypress	24
5.6	src/menu.h File Reference	25
5.6.1	Detailed Description	26
5.6.2	Macro Definition Documentation	26
5.6.2.1	BASE_FUNCTION_STATE	26
5.6.2.2	KEYPRESS_ARROW_DOWN	26
5.6.2.3	KEYPRESS_ARROW_LEFT	26
5.6.2.4	KEYPRESS_ARROW_RIGHT	26
5.6.2.5	KEYPRESS_ARROW_UP	27
5.6.2.6	KEYPRESS_END	27
5.6.2.7	MAIN_MENU	27
5.6.2.8	MENU_READ_REG	27

5.6.2.9	MENU_REG_ADDR	27
5.6.2.10	MENU_WRITE_REG	27
5.6.2.11	WAIT_KEYPRESS	27
5.6.3	Function Documentation	27
5.6.3.1	menu_cls	27
5.6.3.2	menu_print_main_menu	28
5.6.3.3	menu_print_maxim_banner	28
5.6.3.4	menu_print_maxim_banner_big	28
5.6.3.5	menu_print_prompt	29
5.6.3.6	menu_retrieve_keypress	29
5.7	src/platform.c File Reference	29
5.7.1	Macro Definition Documentation	30
5.7.1.1	UART_BAUD	30
5.7.2	Function Documentation	30
5.7.2.1	cleanup_platform	30
5.7.2.2	disable_caches	30
5.7.2.3	enable_caches	30
5.7.2.4	init_platform	30
5.7.2.5	init_uart	30
5.8	src/platform.h File Reference	30
5.8.1	Function Documentation	30
5.8.1.1	cleanup_platform	30
5.8.1.2	init_platform	31
5.9	src/platform_config.h File Reference	31
5.10	src/utilities.c File Reference	31
5.10.1	Detailed Description	31
5.10.2	Function Documentation	32
5.10.2.1	check_if_i2c_device_is_present	32
5.10.2.2	delay	32
5.10.2.3	GetLine	33
5.10.2.4	getUartByte	33
5.10.2.5	htoi	33
5.10.2.6	led_knight_rider	33
5.10.2.7	max_configure_PMOD_port	34
5.10.2.8	number_raised_to_power	34
5.10.2.9	print_asterisks	35
5.10.2.10	printf_temp	35

5.10.2.11receive_byte_with_timeout	35
5.10.2.12SpiRW	36
5.11 src/utilities.h File Reference	36
5.11.1 Detailed Description	37
5.11.2 Function Documentation	38
5.11.2.1 check_if_i2c_device_is_present	38
5.11.2.2 delay	38
5.11.2.3 GetLine	38
5.11.2.4 getUartByte	39
5.11.2.5 htoi	39
5.11.2.6 led_knight_rider	39
5.11.2.7 max_configure_PMOD_port	40
5.11.2.8 number_raised_to_power	40
5.11.2.9 print_asterisks	40
5.11.2.10print_seven_segment_four_digits	41
5.11.2.11print_seven_segment_number	41
5.11.2.12print_seven_segment_temperature	41
5.11.2.13print_seven_segment_time	41
5.11.2.14printf_temp	41
5.11.2.15receive_byte_with_timeout	41
5.11.2.16set_seven_segment_character	42
5.11.2.17SpiRW	42
5.11.3 Variable Documentation	42
5.11.3.1 t	42

Chapter 1

Main Page

1.1 Introduction

This is the code documentation for the Sonoma (MAXREFDES14#) subsystem reference design.

The Files page contains the File List page and the Globals page.

The Globals page contains the Functions, Variables, and Macros sub-pages.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

maximDateTime	
< Structure to hold time, date, day of week	7
maximProduct	
< Structure holding info on each module for menus	8

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

src/ maximDeviceSpecificUtilities.c	11
src/ maximDeviceSpecificUtilities.h	13
src/ MAXREFDES14.c	14
src/ MAXREFDES14.h	17
src/ menu.c	21
src/ menu.h	25
src/ platform.c	29
src/ platform.h	30
src/ platform_config.h	31
src/ utilities.c	31
src/ utilities.h	36

Chapter 4

Data Structure Documentation

4.1 maximDateTime Struct Reference

< Structure to hold time, date, day of week

```
#include <MAXREFDES14.h>
```

Data Fields

- u8 [uchHour](#)
- u8 [uchMinute](#)
- u8 [uchSecond](#)
- u8 [uchMonth](#)
- u8 [uchDate](#)
- u8 [uchYear](#)
- u8 [uchDow](#)

4.1.1 Detailed Description

< Structure to hold time, date, day of week

Definition at line 113 of file MAXREFDES14.h.

4.1.2 Field Documentation

4.1.2.1 u8 uchDate

Definition at line 119 of file MAXREFDES14.h.

4.1.2.2 u8 uchDow

Definition at line 121 of file MAXREFDES14.h.

4.1.2.3 u8 uchHour

Definition at line 115 of file MAXREFDES14.h.

4.1.2.4 u8 uchMinute

Definition at line 116 of file MAXREFDES14.h.

4.1.2.5 u8 uchMonth

Definition at line 118 of file MAXREFDES14.h.

4.1.2.6 u8 uchSecond

Definition at line 117 of file MAXREFDES14.h.

4.1.2.7 u8 uchYear

Definition at line 120 of file MAXREFDES14.h.

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

- src/[MAXREFDES14.h](#)

4.2 maximProduct Struct Reference

< Structure holding info on each module for menus

```
#include <MAXREFDES14.h>
```

Data Fields

- char [sPartNumber](#) [10]
- char [sPartName](#) [80]
- u8 [uchPmodType](#)
- u8 [uchKeypress](#)

4.2.1 Detailed Description

< Structure holding info on each module for menus

Definition at line 105 of file MAXREFDES14.h.

4.2.2 Field Documentation

4.2.2.1 char sPartName[80]

Definition at line 108 of file MAXREFDES14.h.

4.2.2.2 **char sPartNumber[10]**

Definition at line 107 of file MAXREFDES14.h.

4.2.2.3 **u8 uchKeypress**

Definition at line 110 of file MAXREFDES14.h.

4.2.2.4 **u8 uchPmodType**

Definition at line 109 of file MAXREFDES14.h.

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

- src/[MAXREFDES14.h](#)

Chapter 5

File Documentation

5.1 src/maximDeviceSpecificUtilities.c File Reference

```
#include "xbasic_types.h"
#include "stdio.h"
#include "maximDeviceSpecificUtilities.h"
#include "utilities.h"
```

Functions

- u32 [read_reg](#) (u8 uchRegAddr)
Reads the value of a register and display the data via the HyperTerminal.
- void [write_reg](#) (u8 uchRegAddr, u32 unRegValue)
Write a value to a register.

5.1.1 Detailed Description

```
Project: Sonoma (MAXREFDES14#)
Filename: maximDeviceSpecificUtilities.c
Description: This module uses the Xilinx XSpi core Ver 3.03a peripheral to make
             custom functions for the Sonoma MAXREFDES14#. These
             low level functions could be cut/pasted into the user's
             application as a starting point for development of an end
             application.
```

Revision History:

05-31-13 Rev 01.00 GL Initial Release

This code follows the following naming conventions:

char chPmodValue

char (array) sPmodString[16]

float fPmodValue

int nPmodValue

int (array) anPmodValue[16]

u16 uPmodValue

u16 (array) auPmodValue[16]

u8 uchPmodValue

u8 (array) auchPmodBuffer[16]

unsigned int unPmodValue

int * punPmodValue

Definition in file [maximDeviceSpecificUtilities.c](#).

5.1.2 Function Documentation

5.1.2.1 u32 read_reg (u8 uchRegAddr)

Reads the value of a register and display the data via the HyperTerminal.

Details

Reads the value of a register and display the data via the HyperTerminal

Parameters

in	<i>uchRegAddr</i>	- Register Address
----	-------------------	--------------------

Return values

<i>Register</i>	Value
-----------------	-------

Definition at line 69 of file maximDeviceSpecificUtilities.c.

5.1.2.2 void write_reg (u8 uchRegAddr, u32 unRegValue)

Write a value to a register.

Details

Write a value to a register

Parameters

in	<i>uchRegAddr</i>	- Register Address
in	<i>uchRegAddr</i>	- Register Value

Return values

<i>None</i>

Definition at line 141 of file maximDeviceSpecificUtilities.c.

5.2 src/maximDeviceSpecificUtilities.h File Reference

Functions

- u32 [read_reg](#) (u8 uchRegAddr)
Reads the value of a register and display the data via the HyperTerminal.
- void [write_reg](#) (u8 uchRegAddr, u32 unRegValue)
Write a value to a register.

5.2.1 Detailed Description

```
Project: Sonoma (MAXREFDES14#)
Filename: maximDeviceSpecificUtilities.h
Description: This module uses the Xilinx XSpi core Ver 3.03a peripheral to make
             custom functions for the Sonoma MAXREFDES14#. These
             low level functions could be cut/pasted into the user's
             application as a starting point for development of an end
             application.
```

Revision History:

05-31-13 Rev 01.00 GL Initial Release

This code follows the following naming conventions:

char chPmodValue

char (array) sPmodString[16]

float fPmodValue

int nPmodValue

int (array) anPmodValue[16]

u16 uPmodValue

u16 (array) auPmodValue[16]

u8 uchPmodValue

u8 (array) auchPmodBuffer[16]

unsigned int unPmodValue

int * punPmodValue

Definition in file [maximDeviceSpecificUtilities.h](#).

5.2.2 Function Documentation

5.2.2.1 u32 read_reg (u8 uchRegAddr)

Reads the value of a register and display the data via the HyperTerminal.

Details

Reads the value of a register and display the data via the HyperTerminal

Parameters

in	<i>uchRegAddr</i>	- Register Address
----	-------------------	--------------------

Return values

<i>Register</i>	Value
-----------------	-------

Definition at line 69 of file maximDeviceSpecificUtilities.c.

5.2.2.2 void write_reg (u8 uchRegAddr, u32 unRegValue)

Write a value to a register.

Details

Write a value to a register

Parameters

in	<i>uchRegAddr</i>	- Register Address
in	<i>uchRegAddr</i>	- Register Value

Return values

<i>None</i>

Definition at line 141 of file maximDeviceSpecificUtilities.c.

5.3 src/MAXREFDES14.c File Reference

```
#include <stdio.h>
#include "xparameters.h"
#include "platform.h"
#include "menu.h"
#include "utilities.h"
#include "maximDeviceSpecificUtilities.h"
#include "MAXREFDES14.h"
```

Macros

- #define [MAJOR_REVISION](#) 01
- #define [MINOR_REVISION](#) 00

Functions

- int [main](#) ()
Main() function for Analog Essentials example program.

Variables

- XGpio [g_xGpioSevenSegment](#)
- unsigned int [g_unSevenSegmentValue](#) =0
- char [g_slInputString](#) [INPUT_STRING_MAX_SIZE]
- XGpio [g_xGpioPmodPortA](#)
- XGpio [g_xGpioPmodPortB](#)
- XGpio [g_gpButtons](#)
- XGpio [g_xGpioLed](#)
- u8 [g_PmodPortMuxSettings](#) =0

5.3.1 Detailed Description

```
Project: Sonoma (MAXREFDES14#)
Filename: MAXREFDES14.c
Description: This module contains the Main application for the
             AVNET LX9 MicroBoard implementation of the MAXREFDES14
             example program.
```

Revision History:

05-31-13 Rev 01.00 GL Initial Release

This code follows the following naming conventions.

char chPmodValue char (array) sPmodString[16] float fPmodValue int nPmodValue int (array) anPmodValue[16] u16
uPmodValue u8 uchPmodValue u8 (array) auchPmodBuffer[16] unsigned int unPmodValue int * punPmodValue

Definition in file [MAXREFDES14.c](#).

5.3.2 Macro Definition Documentation

5.3.2.1 `#define MAJOR_REVISION 01`

Definition at line 83 of file MAXREFDES14.c.

5.3.2.2 `#define MINOR_REVISION 00`

Definition at line 84 of file MAXREFDES14.c.

5.3.3 Function Documentation

5.3.3.1 `int main ( )`

Main() function for Analog Essentials example program.

Details

This function sets up and initializes the FPGA and hardware, displays the root menu via Hyperterminal, then dispatches individual demo programs for specific module based on user's keypress selection.

Parameters

<i>None</i>	
-------------	--

Return values

<i>Always</i>	TRUE
---------------	------

Definition at line 97 of file MAXREFDES14.c.

5.3.4 Variable Documentation

5.3.4.1 XGpio g_gpButtons

Definition at line 93 of file MAXREFDES14.c.

5.3.4.2 u8 g_PmodPortMuxSettings =0

Definition at line 95 of file MAXREFDES14.c.

5.3.4.3 char g_sInputString[INPUT_STRING_MAX_SIZE]

Definition at line 89 of file MAXREFDES14.c.

5.3.4.4 unsigned int g_unSevenSegmentValue =0

Definition at line 88 of file MAXREFDES14.c.

5.3.4.5 XGpio g_xGpioLed

Definition at line 94 of file MAXREFDES14.c.

5.3.4.6 XGpio g_xGpioPmodPortA

Definition at line 90 of file MAXREFDES14.c.

5.3.4.7 XGpio g_xGpioPmodPortB

Definition at line 91 of file MAXREFDES14.c.

5.3.4.8 XGpio g_xGpioSevenSegment

Definition at line 87 of file MAXREFDES14.c.

5.4 src/MAXREFDES14.h File Reference

```
#include "xgpio.h"
#include "xgpio_l.h"
#include "xparameters.h"
#include "xuartlite.h"
#include "xspi_l.h"
#include "xspi.h"
#include "xiic_l.h"
#include "utilities.h"
#include <string.h>
#include <stdio.h>
#include "platform.h"
```

Data Structures

- struct [maximProduct](#)
< Structure holding info on each module for menus
- struct [maximDateTime](#)
< Structure to hold time, date, day of week

Macros

- #define [DEFAULT_HYPERTERMINAL_UART_ID](#) XPAR_USB_UART_DEVICE_ID
macro used to abstract Physical Port of Hyperterminal UART
- #define [DEFAULT_HYPERTERMINAL_UART_ADDRESS](#) XPAR_USB_UART_BASEADDR
- #define [ABOUT_ONE_SECOND](#) 3508380
approx 1 second delay when used as argument with function delay(numberCyclesToDelay)
- #define [PMOD_TYPE_UART](#) 0
value which specifies UART in [maximProduct](#) structure
- #define [PMOD_TYPE_SPI](#) 1
value which specifies SPI in [maximProduct](#) structure
- #define [PMOD_TYPE_GPIO](#) 2
value which specifies GPIO in [maximProduct](#) structure
- #define [PMOD_TYPE_I2C](#) 3
value which specifies I2C in [maximProduct](#) structure
- #define [PMOD_PORT_TYPE_UART](#) 0x00
value to switch UART controller into a Pmod port in max_configure_PMOD_port
- #define [PMOD_PORT_TYPE_SPI](#) 0x01
value to switch SPI controller into a Pmod port in max_configure_PMOD_port
- #define [PMOD_PORT_TYPE_GPIO](#) 0x02
value to switch GPIO controller into a Pmod port in max_configure_PMOD_port
- #define [PMOD_PORT_TYPE_I2C](#) 0x03
value to switch I2C controller into a Pmod port in max_configure_PMOD_port
- #define [BIT_BANG_SPI_CLK](#) 0x08
bit mask for SPI CLK used in max_GPIO_bit_bang_SPI_write
- #define [BIT_BANG_SPI_CLRB](#) 0x04

- bit mask for SPI /CLR used in max_GPIO_bit_bang_SPI_write*
- #define [BIT_BANG_SPI_MISO](#) 0x04
- bit mask for SPI MISO used in max_GPIO_bit_bang_SPI_write*
- #define [BIT_BANG_SPI_MOSI](#) 0x02
- bit mask for SPI MOSI used in max_GPIO_bit_bang_SPI_write*
- #define [BIT_BANG_SPI_SSB](#) 0x01
- bit mask for SPI SS used in max_GPIO_bit_bang_SPI_write*
- #define [RE_B](#) 0x00000001
- bit mask for RS-485 /RE used in menu_MAX14840*
- #define [DI](#) 0x00000002
- bit mask for RS-485 DI used in menu_MAX14840*
- #define [RO](#) 0x00000004
- bit mask for RS-485 RO used in menu_MAX14840*
- #define [DE](#) 0x00000008
- bit mask for RS-485 DE used in menu_MAX14840*
- #define [INPUT_STRING_MAX_SIZE](#) 16

Variables

- XGpio [g_xGpioPmodPortA](#)
- XGpio [g_xGpioPmodPortB](#)
- char [g_sInputString](#) [[INPUT_STRING_MAX_SIZE](#)]

5.4.1 Macro Definition Documentation

5.4.1.1 #define ABOUT_ONE_SECOND 3508380

approx 1 second delay when used as argument with function delay(numberCyclesToDelay)

Definition at line 78 of file MAXREFDES14.h.

5.4.1.2 #define BIT_BANG_SPI_CLK 0x08

bit mask for SPI CLK used in max_GPIO_bit_bang_SPI_write

Definition at line 92 of file MAXREFDES14.h.

5.4.1.3 #define BIT_BANG_SPI_CLRB 0x04

bit mask for SPI /CLR used in max_GPIO_bit_bang_SPI_write

Definition at line 93 of file MAXREFDES14.h.

5.4.1.4 #define BIT_BANG_SPI_MISO 0x04

bit mask for SPI MISO used in max_GPIO_bit_bang_SPI_write

Definition at line 94 of file MAXREFDES14.h.

5.4.1.5 #define BIT_BANG_SPI_MOSI 0x02

bit mask for SPI MOSI used in max_GPIO_bit_bang_SPI_write

Definition at line 95 of file MAXREFDES14.h.

5.4.1.6 #define BIT_BANG_SPI_SSB 0x01

bit mask for SPI SS used in max_GPIO_bit_bang_SPI_write

Definition at line 96 of file MAXREFDES14.h.

5.4.1.7 #define DE 0x00000008

bit mask for RS-485 DE used in menu_MAX14840

Definition at line 101 of file MAXREFDES14.h.

5.4.1.8 #define DEFAULT_HYPERTERMINAL_UART_ADDRESS XPAR_USB_UART_BASEADDR

Definition at line 76 of file MAXREFDES14.h.

5.4.1.9 #define DEFAULT_HYPERTERMINAL_UART_ID XPAR_USB_UART_DEVICE_ID

macro used to abstract Physical Port of Hyperterminal UART

Definition at line 75 of file MAXREFDES14.h.

5.4.1.10 #define DI 0x00000002

bit mask for RS-485 DI used in menu_MAX14840

Definition at line 99 of file MAXREFDES14.h.

5.4.1.11 #define INPUT_STRING_MAX_SIZE 16

Definition at line 124 of file MAXREFDES14.h.

5.4.1.12 #define PMOD_PORT_TYPE_GPIO 0x02

value to switch GPIO controller into a Pmod port in max_configure_PMOD_port

Definition at line 89 of file MAXREFDES14.h.

5.4.1.13 #define PMOD_PORT_TYPE_I2C 0x03

value to switch I2C controller into a Pmod port in max_configure_PMOD_port

Definition at line 90 of file MAXREFDES14.h.

5.4.1.14 #define PMOD_PORT_TYPE_SPI 0x01

value to switch SPI controller into a Pmod port in max_configure_PMOD_port

Definition at line 88 of file MAXREFDES14.h.

5.4.1.15 #define PMOD_PORT_TYPE_UART 0x00

value to switch UART controller into a Pmod port in max_configure_PMOD_port

Definition at line 87 of file MAXREFDES14.h.

5.4.1.16 #define PMOD_TYPE_GPIO 2

value which specifies GPIO in [maximProduct](#) structure

Definition at line 84 of file MAXREFDES14.h.

5.4.1.17 #define PMOD_TYPE_I2C 3

value which specifies I2C in [maximProduct](#) structure

Definition at line 85 of file MAXREFDES14.h.

5.4.1.18 #define PMOD_TYPE_SPI 1

value which specifies SPI in [maximProduct](#) structure

Definition at line 83 of file MAXREFDES14.h.

5.4.1.19 #define PMOD_TYPE_UART 0

value which specifies UART in [maximProduct](#) structure

Definition at line 82 of file MAXREFDES14.h.

5.4.1.20 #define RE_B 0x00000001

bit mask for RS-485 /RE used in menu_MAX14840

Definition at line 98 of file MAXREFDES14.h.

5.4.1.21 #define RO 0x00000004

bit mask for RS-485 RO used in menu_MAX14840

Definition at line 100 of file MAXREFDES14.h.

5.4.2 Variable Documentation

5.4.2.1 char g_sInputString[INPUT_STRING_MAX_SIZE]

Definition at line 89 of file MAXREFDES14.c.

5.4.2.2 XGpio g_xGpioPmodPortA

Definition at line 90 of file MAXREFDES14.c.

5.4.2.3 XGpio g_xGpioPmodPortB

Definition at line 91 of file MAXREFDES14.c.

5.5 src/menu.c File Reference

```
#include <stdio.h>
#include "platform.h"
#include "xparameters.h"
#include "utilities.h"
#include "string.h"
#include "MAXREFDES14.h"
#include "maximDeviceSpecificUtilities.h"
#include "menu.h"
```

Functions

- void [menu_cls](#) ()
Function to clear the screen via Hyperterminal.
- void [menu_print_maxim_banner](#) ()
Print standard Maxim banner at top of Hyperterminal screen.
- void [menu_print_maxim_banner_big](#) ()
Print large Maxim banner at top of Hyperterminal screen.
- void [menu_print_prompt](#) ()
Print a standard prompt for keyboard input ">".
- void [menu_print_line](#) ()
Print one line of dashes across the screen via Hyperterminal.
- u8 [menu_retrieve_keypress](#) (u32 nUartAddress)
Get a single keypress via Hyperterminal.
- void [menu_print_main_menu](#) ()
Print the main menu listing choice of module to test.
- void [menu_read_reg](#) ()
Print a menu listing read modes.

5.5.1 Detailed Description

```

Project: Sonoma (MAXREFDES14#)
Filename: menu.c
Description: This module contains all the functions used to
             generate the menus and menu options used to run the
             MAXREFDES14# example firmware.

```

Revision History:

05-31-13 Rev 01.00 GL Initial Release

This code follows the following naming conventions:

char chPmodValue

char (array) sPmodString[16]

float fPmodValue

int nPmodValue

int (array) anPmodValue[16]

u16 uPmodValue

u16 (array) auPmodValue[16]

u8 uchPmodValue

u8 (array) auchPmodBuffer[16]

unsigned int unPmodValue

int * punPmodValue

Definition in file [menu.c](#).

5.5.2 Function Documentation

5.5.2.1 void menu_cls ()

Function to clear the screen via Hyperterminal.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

Definition at line 70 of file menu.c.

5.5.2.2 void menu_print_line ()

Print one line of dashes across the screen via Hyperterminal.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

Definition at line 155 of file menu.c.

5.5.2.3 void menu_print_main_menu ()

Print the main menu listing choice of module to test.

Details.**Return values**

<i>None</i>	
-------------	--

Definition at line 189 of file menu.c.

5.5.2.4 void menu_print_maxim_banner ()

Print standard Maxim banner at top of Hyperterminal screen.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

Definition at line 84 of file menu.c.

5.5.2.5 void menu_print_maxim_banner_big ()

Print large Maxim banner at top of Hyperterminal screen.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

Definition at line 108 of file menu.c.

5.5.2.6 void menu_print_prompt ()

Print a standard prompt for keyboard input ">".

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

Definition at line 141 of file menu.c.

5.5.2.7 void menu_read_reg ()

Print a menu listing read modes.

Details**Return values**

<i>None</i>	
-------------	--

Definition at line 207 of file menu.c.

5.5.2.8 u8 menu_retrieve_keypress (u32 nUartAddress)

Get a single keypress via Hyperterminal.

Details

Returns ascii character corresponding to keypress with some preprocessing.
Lower case "a"-"z" converted to uppercase "A"-"Z"

Parameters

<i>in</i>	<i>nUartAddress</i>	- address of the UART peripheral in MicroBlaze memory map
-----------	---------------------	---

Return values

<i>Character, partially</i>	decoded.
-----------------------------	----------

Definition at line 167 of file menu.c.

5.6 src/menu.h File Reference

```
#include "xbasic_types.h"
#include "stdio.h"
#include "MAXREFDES14.h"
```

Macros

- #define [MAIN_MENU](#) 0
// Menu state machine state
- #define [MENU_REG_ADDR](#) 1
// Menu state machine state
- #define [MENU_READ_REG](#) 2
// Menu state machine state
- #define [MENU_WRITE_REG](#) 3
// Menu state machine state
- #define [WAIT_KEYPRESS](#) 6
// Menu state machine state
- #define [BASE_FUNCTION_STATE](#) 10
// Menu state machine state
- #define [KEYPRESS_ARROW_UP](#) 240
Assign up-arrow an extended ascii code which won't be used elsewhere.
- #define [KEYPRESS_ARROW_DOWN](#) 241
Assign up-arrow an extended ascii code which won't be used elsewhere.
- #define [KEYPRESS_ARROW_LEFT](#) 242
Assign up-arrow an extended ascii code which won't be used elsewhere.
- #define [KEYPRESS_ARROW_RIGHT](#) 243
Assign up-arrow an extended ascii code which won't be used elsewhere.
- #define [KEYPRESS_END](#) 244
Assign up-arrow an extended ascii code which won't be used elsewhere.

Functions

- void [menu_cls](#) ()
Function to clear the screen via Hyperterminal.
- void [menu_print_maxim_banner](#) ()
Print standard Maxim banner at top of Hyperterminal screen.
- void [menu_print_maxim_banner_big](#) ()
Print large Maxim banner at top of Hyperterminal screen.
- void [menu_print_prompt](#) ()
Print a standard prompt for keyboard input ">".
- u8 [menu_retrieve_keypress](#) (u32 nUartAddress)
Get a single keypress via Hyperterminal.
- void [menu_print_main_menu](#) ()
Print the main menu listing choice of module to test.

5.6.1 Detailed Description

```

Project: Sonoma (MAXREFDES14#)
Filename: menu.h
Description: This module contains all the functions used to
             generate the menus and menu options used to run the
             MAXREFDES14# example firmware.

```

Revision History:

05-31-13 Rev 01.00 GL Initial Release

This code follows the following naming conventions:

char chPmodValue

char (array) sPmodString[16]

float fPmodValue

int nPmodValue

int (array) anPmodValue[16]

u16 uPmodValue

u16 (array) auPmodValue[16]

u8 uchPmodValue

u8 (array) auchPmodBuffer[16]

unsigned int unPmodValue

int * punPmodValue

Definition in file [menu.h](#).

5.6.2 Macro Definition Documentation

5.6.2.1 #define BASE_FUNCTION_STATE 10

// Menu state machine state

Definition at line 73 of file menu.h.

5.6.2.2 #define KEYPRESS_ARROW_DOWN 241

Assign up-arrow an extended ascii code which won't be used elsewhere.

Definition at line 76 of file menu.h.

5.6.2.3 #define KEYPRESS_ARROW_LEFT 242

Assign up-arrow an extended ascii code which won't be used elsewhere.

Definition at line 77 of file menu.h.

5.6.2.4 #define KEYPRESS_ARROW_RIGHT 243

Assign up-arrow an extended ascii code which won't be used elsewhere.

Definition at line 78 of file menu.h.

5.6.2.5 #define KEYPRESS_ARROW_UP 240

Assign up-arrow an extended ascii code which won't be used elsewhere.

Definition at line 75 of file menu.h.

5.6.2.6 #define KEYPRESS_END 244

Assign up-arrow an extended ascii code which won't be used elsewhere.

Definition at line 79 of file menu.h.

5.6.2.7 #define MAIN_MENU 0

// Menu state machine state

Definition at line 68 of file menu.h.

5.6.2.8 #define MENU_READ_REG 2

// Menu state machine state

Definition at line 70 of file menu.h.

5.6.2.9 #define MENU_REG_ADDR 1

// Menu state machine state

Definition at line 69 of file menu.h.

5.6.2.10 #define MENU_WRITE_REG 3

// Menu state machine state

Definition at line 71 of file menu.h.

5.6.2.11 #define WAIT_KEYPRESS 6

// Menu state machine state

Definition at line 72 of file menu.h.

5.6.3 Function Documentation

5.6.3.1 void menu_cls ()

Function to clear the screen via Hyperterminal.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

Definition at line 70 of file menu.c.

5.6.3.2 void menu_print_main_menu ()

Print the main menu listing choice of module to test.

Details.**Return values**

<i>None</i>	
-------------	--

Definition at line 189 of file menu.c.

5.6.3.3 void menu_print_maxim_banner ()

Print standard Maxim banner at top of Hyperterminal screen.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

Definition at line 84 of file menu.c.

5.6.3.4 void menu_print_maxim_banner_big ()

Print large Maxim banner at top of Hyperterminal screen.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

Definition at line 108 of file menu.c.

5.6.3.5 void menu_print_prompt ()

Print a standard prompt for keyboard input ">".

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

Definition at line 141 of file menu.c.

5.6.3.6 u8 menu_retrieve_keypress (u32 nUartAddress)

Get a single keypress via Hyperterminal.

Details

Returns ascii character corresponding to keypress with some preprocessing.
Lower case "a"-"z" converted to uppercase "A"-"Z"

Parameters

<i>in</i>	<i>nUartAddress</i>	- address of the UART peripheral in MicroBlaze memory map
-----------	---------------------	---

Return values

<i>Character, partially</i>	decoded.
-----------------------------	----------

Definition at line 167 of file menu.c.

5.7 src/platform.c File Reference

```
#include "xparameters.h"
#include "xil_cache.h"
#include "platform_config.h"
```

Macros

- #define [UART_BAUD](#) 9600

Functions

- void [enable_caches](#) ()
- void [disable_caches](#) ()
- void [init_uart](#) ()

- void [init_platform](#) ()
- void [cleanup_platform](#) ()

5.7.1 Macro Definition Documentation

5.7.1.1 `#define UART_BAUD 9600`

Definition at line 30 of file platform.c.

5.7.2 Function Documentation

5.7.2.1 `void cleanup_platform ( )`

Definition at line 78 of file platform.c.

5.7.2.2 `void disable_caches ( )`

Definition at line 49 of file platform.c.

5.7.2.3 `void enable_caches ( )`

Definition at line 33 of file platform.c.

5.7.2.4 `void init_platform ( )`

Definition at line 71 of file platform.c.

5.7.2.5 `void init_uart ( )`

Definition at line 56 of file platform.c.

5.8 `src/platform.h` File Reference

```
#include "platform_config.h"
```

Functions

- void [init_platform](#) ()
- void [cleanup_platform](#) ()

5.8.1 Function Documentation

5.8.1.1 `void cleanup_platform ( )`

Definition at line 78 of file platform.c.

5.8.1.2 void init_platform ()

Definition at line 71 of file platform.c.

5.9 src/platform_config.h File Reference

5.10 src/utilities.c File Reference

```
#include "utilities.h"
#include "MAXREFDES14.h"
```

Functions

- void [print_asterisks](#) (int nQuantity)
Print nQuantity of asterisks to the default Hyperterminal UART.
- int [SpiRW](#) (u32 unPeripheralAddressSPI, unsigned int unCPHA, unsigned int unCPOL, u8 *auchWriteBuf, u8 *auchReadBuf, int unNumBytes, u8 uchCsActiveHigh)
Perform a SPI read or write.
- void [delay](#) (int nStopValue)
Loop for nStopValue iterations to provide a delay.
- void [led_knight_rider](#) (XGpio *pLED_GPIO, int nNumberOfTimes)
Blink a row of LEDs nNumberOfTimes times.
- void [max_configure_PMOD_port](#) (u8 uchPmodPortA, u8 uchPmodPortB, u8 uchPmodPortC, u8 uchPmodPortD)
Configure driving peripherals on each Pmod port.
- int [check_if_i2c_device_is_present](#) ()
Test for SDA and SCL pullups.
- int [number_raised_to_power](#) (int nBase, int nExponent)
Raise nBase to the nExponent power (operates with integers only).
- int [receive_byte_with_timeout](#) (u32 unUartAddress, int nTimeoutInTenthsOfSeconds, u8 *uchRxData)
*Receive a byte from the UART located at *pUartAddress.*
- int [GetLine](#) (char *sInputString, unsigned int unMaxSize)
Retrieve a line of characters from the default Hyperterminal UART (DEFAULT_HYPERTERMINAL_UART).
- void [printf_temp](#) (float fTemp, u8 uchPrintCelsius, u8 uchAddCarriageReturn)
Print temperature as xx.x as degrees F or C.
- u8 [getUartByte](#) (u32 nUartAddress)
Get a byte from either the full UART in the Zynq PS, or a Xilinx UartLite.
- int [atoi](#) (char *sStr)

5.10.1 Detailed Description

```
Project: Maxim Plug-in Peripheral Modules
Filename: utilities.c
Description: This module contains a collection of general utility
             functions which are not specific to any particular
             module.
```

Revision History:

4-13-12 Rev 1.0 Seth Messimer Initial Release

7-20-12 Rev 1.4 Nathan Young Additional functions

This code follows the following naming conventions.

char chPmodValue char (array) sPmodString[16] float fPmodValue int nPmodValue int (array) anPmodValue[16] u16
uPmodValue u8 uchPmodValue u8 (array) auchPmodBuffer[16] unsigned int unPmodValue int * punPmodValue

Definition in file [utilities.c](#).

5.10.2 Function Documentation

5.10.2.1 int check_if_i2c_device_is_present ()

Test for SDA and SCL pullups.

Details

This function places the PMOD A port into GPIO mode briefly in order to sample the SDA and SCL lines. If they are high, then this implies that a pull-up is present, and hence a PMOD board is likely present.

Parameters

<i>None</i>	
-------------	--

Return values

<i>TRUE</i>	if pull-ups present, <i>FALSE</i> if pullups not present
-------------	--

Definition at line 261 of file utilities.c.

5.10.2.2 void delay (int nStopValue)

Loop for nStopValue iterations to provide a delay.

Details

It is commonly used with the constant 'ABOUT_ONE_SECOND' defined in maximPMOD.h for setting approximate delays.

Parameters

<i>in</i>	<i>nStopValue</i>	- number of iterations to loop
-----------	-------------------	--------------------------------

Return values

<i>None</i>	
-------------	--

Definition at line 165 of file utilities.c.

5.10.2.3 int GetLine (char * *sInputString*, unsigned int *unMaxSize*)

Retrieve a line of characters from the default Hyperterminal UART (DEFAULT_HYPERTERMINAL_UART).

Details

Maximum number of characters can be specified. Function will timeout after 10 seconds.

Parameters

in	<i>sInputString</i>	- pointer to buffer for input string
in	<i>unMaxSize</i>	- maximum number of characters to input

Return values

<i>TRUE</i>	if operation succeeded
-------------	------------------------

Definition at line 393 of file utilities.c.

5.10.2.4 u8 getUartByte (u32 *nUartAddress*)

Get a byte from either the full UART in the Zynq PS, or a Xilinx UartLite.

Parameters

in	<i>unUartAddress.</i>	32 bit UART address
----	-----------------------	---------------------

Return values

<i>Returns</i>	the next byte from the Uart Rx FIFO
----------------	-------------------------------------

Definition at line 442 of file utilities.c.

5.10.2.5 int htoi (char * *sStr*)

Definition at line 457 of file utilities.c.

5.10.2.6 void led_knight_rider (XGpio * *pLED_GPIO*, int *nNumberOfTimes*)

Blink a row of LEDs *nNumberOfTimes* times.

Details

The Digilent NEXYS-3 board has 8 green LEDs located above the toggle switches. This function blinks them back/forth (a bit like the KITT car from Knight Rider)

Parameters

in	<i>*pLED_GPIO</i>	- address of the GPIO peripheral driving the LEDs in MicroBlaze memory map
----	-------------------	--

Return values

<i>None</i>	
-------------	--

Definition at line 186 of file utilities.c.

5.10.2.7 void max_configure_PMOD_port (u8 uchPmodPortA, u8 uchPmodPortB, u8 uchPmodPortC, u8 uchPmodPortD)

Configure driving peripherals on each Pmod port.

Details

The Maxim HDL hardware design for the Nexys 3 includes a multiplexer on each PMOD port to allow I2C, SPI, GPIO, and UART functionality to be selected on each port. The standard configuration is (PortA = I2C, PortB = SPI, PortC = GPIO, PortD = UART) This function is used to set the standard configuration in the [main\(\)](#) function, as well as to optionally change the port config. The 2-bit number used to define the port configuration is encoded as follows:

00=UART, 01=SPI, 10=GPIO and 11=I2C

Parameters

in	<i>uchPmodPortA</i>	- 2-bit number to define configuration for Pmod port A
in	<i>uchPmodPortB</i>	- 2-bit number to define configuration for Pmod port B
in	<i>uchPmodPortC</i>	- 2-bit number to define configuration for Pmod port C
in	<i>uchPmodPortD</i>	- 2-bit number to define configuration for Pmod port D

Return values

<i>TRUE</i>	if operation succeeded
-------------	------------------------

Definition at line 221 of file utilities.c.

5.10.2.8 int number_raised_to_power (int nBase, int nExponent)

Raise nBase to the nExponent power (operates with integers only).

Details

Many Microblaze applications will not have math.h included due to limited memory space. This is a simple functions to implement $(nBase \wedge nExponent)$ Some Maxim devices (such as MAX44009) return values in mantissa + (power of 2) exponent format.

Parameters

in	<i>nBase</i>	- base
in	<i>nExponent</i>	- exponent

Return values

$Base \wedge Exponent$	
------------------------	--

Definition at line 308 of file *utilities.c*.

5.10.2.9 void print_asterisks (int *nQuantity*)

Print *nQuantity* of asterisks to the default Hyperterminal UART.

Parameters

in	<i>nQuantity</i>	- number of asterisks to print
----	------------------	--------------------------------

Return values

<i>None</i>

Definition at line 64 of file *utilities.c*.

5.10.2.10 void printf_temp (float *fTemp*, u8 *uchPrintCelsius*, u8 *uchAddCarriageReturn*)

Print temperature as xx.x as degrees F or C.

Details

This function prints the temperature with .1 degree resolution in either degrees Celsius (*uchPrintCelsius*==TRUE) -or- Fahrenheit (*uchPrintCelsius*==FALSE)

Parameters

in	<i>fTemp</i>	- temperature to print in celsius
in	<i>uchPrintCelsius</i>	- true = print as celsius, false = print as fahrenheit
in	<i>uchAddCarriageReturn</i>	- true = add a carriage return

Return values

<i>None</i>

Definition at line 420 of file *utilities.c*.

5.10.2.11 int receive_byte_with_timeout (u32 *unUartAddress*, int *nTimeoutInTenthsOfSeconds*, u8 * *uchRxData*)

Receive a byte from the UART located at **pUartAddress*.

Parameters

in	<i>unUartAddress</i>	- address of the UART peripheral in MicroBlaze memory map
in	<i>nTimeoutInTenthsOfSeconds</i>	- amount of time to allow before TIMEOUT
out	* <i>uchRxData</i>	- received data is stored at <i>uchRxData</i>

Return values

<i>TRUE</i>	if operation succeeded
-------------	------------------------

Definition at line 337 of file utilities.c.

5.10.2.12 **int SpiRW (u32 *unPeripheralAddressSPI*, unsigned int *unCPHA*, unsigned int *unCPOL*, u8 * *auchWriteBuf*, u8 * *auchReadBuf*, int *unNumBytes*, u8 *uchCsActiveHigh*)**

Perform a SPI read or write.

Details

This function provides a combination SPI Read and Write to the chosen SPI port in the design CPHA and CPOL can be set to 0 or 1 Pointers are provided to u8 buffers containing the data to be written and received Data in the *auchWriteBuf* will be clocked out (MSB first) onto the MOSI pin Data from the MISO pin will be placed into the *auchReadBuf* *uchCsActiveHigh*==TRUE allows SS configurations to be used *uchCsActiveHigh*==FALSE allows SS# configurations to be used

Parameters

in	<i>unPeripheralAddressSPI</i>	-
in	<i>unCPHA</i>	- phase of SCK (edge to trigger on). 0=Leading edge, 1=Trailing edge
in	<i>unCPOL</i>	- polarity of SCK. 0=Active high, 1=Active low
in	<i>auchWriteBuf</i>	- pointer to write data buffer
in	<i>auchReadBuf</i>	- pointer to read data buffer
in	<i>unNumBytes</i>	- number of bytes to transfer
in	<i>uchCsActiveHigh</i>	- polarity of slave select 0=active low, 1=active high

Return values

<i>Always</i>	returns 0
---------------	-----------

Definition at line 78 of file utilities.c.

5.11 **src/utilities.h** File Reference

```
#include "xbasic_types.h"
#include "xspi_l.h"
#include "stdio.h"
#include "xiic_l.h"
#include "xuartlite_i.h"
#include "xparameters.h"
#include "xgpio.h"
#include "xgpio_l.h"
#include "MAXREFDES14.h"
```

Functions

- int [SpiRW](#) (u32 unPeripheralAddressSPI, unsigned int unCPHA, unsigned int unCPOL, u8 *auchWriteBuf, u8 *auchReadBuf, int unNumBytes, u8 uchCsActiveHigh)
Perform a SPI read or write.
- void [delay](#) (int nStopValue)
Loop for nStopValue iterations to provide a delay.
- void [led_knight_rider](#) (XGpio *pLED_GPIO, int nNumberOfTimes)
Blink a row of LEDs nNumberOfTimes times.
- void [max_configure_PMOD_port](#) (u8 uchPmodPortA, u8 uchPmodPortB, u8 uchPmodPortC, u8 uchPmodPortD)
Configure driving peripherals on each Pmod port.
- int [check_if_i2c_device_is_present](#) ()
Test for SDA and SCL pullups.
- int [number_raised_to_power](#) (int nBase, int nExponent)
Raise nBase to the nExponent power (operates with integers only).
- int [receive_byte_with_timeout](#) (u32 unUartAddress, int nTimeoutInTenthsOfSeconds, u8 *uchRxData)
*Receive a byte from the UART located at *pUartAddress.*
- void [set_seven_segment_character](#) (u8 uchDigitNumber, u8 uchValue, u8 uchDecimalActive)
- void [print_seven_segment_number](#) (float fNumber)
- void [print_seven_segment_time](#) (struct [maximDateTime](#) *t)
- int [GetLine](#) (char *sInputString, unsigned int unMaxSize)
Retrieve a line of characters from the default Hyperterminal UART (DEFAULT_HYPERTERMINAL_UART).
- void [print_asterisks](#) (int nQuantity)
Print nQuantity of asterisks to the default Hyperterminal UART.
- void [printf_temp](#) (float fTemp, u8 uchPrintCelsius, u8 uchAddCarriageReturn)
Print temperature as xx.x as degrees F or C.
- void [print_seven_segment_temperature](#) (float temp, u8 uchPrintCelsius)
- void [print_seven_segment_four_digits](#) (char *puchArray)
- int [htoi](#) (char *sStr)
- u8 [getUartByte](#) (u32 nUartAddress)
Get a byte from either the full UART in the Zynq PS, or a Xilinx UartLite.

Variables

- struct [maximDateTime](#) * t

5.11.1 Detailed Description

```
Project: Maxim Plug-in Peripheral Modules
Filename: utilities.h
Description: This module contains a collection of general utility
             functions which are not specific to any particular
             module.
```

Revision History:

4-13-12 Rev 1.0 Seth Messimer Initial Release

7-20-12 Rev 1.4 Nathan Young Additional functions

This code follows the following naming conventions.

```
char chPmodValue char (array) sPmodString[16] float fPmodValue int nPmodValue int (array) anPmodValue[16] u16
uPmodValue u8 uchPmodValue u8 (array) auchPmodBuffer[16] unsigned int unPmodValue int * punPmodValue
```

Definition in file [utilities.h](#).

5.11.2 Function Documentation

5.11.2.1 int check_if_i2c_device_is_present ()

Test for SDA and SCL pullups.

Details

This function places the PMOD A port into GPIO mode briefly in order to sample the SDA and SCL lines. If they are high, then this implies that a pull-up is present, and hence a PMOD board is likely present.

Parameters

<i>None</i>	
-------------	--

Return values

<i>TRUE</i>	if pull-ups present, <i>FALSE</i> if pullups not present
-------------	--

Definition at line 261 of file utilities.c.

5.11.2.2 void delay (int nStopValue)

Loop for nStopValue iterations to provide a delay.

Details

It is commonly used with the constant 'ABOUT_ONE_SECOND' defined in maximPMOD.h for setting approximate delays.

Parameters

<i>in</i>	<i>nStopValue</i>	- number of iterations to loop
-----------	-------------------	--------------------------------

Return values

<i>None</i>	
-------------	--

Definition at line 165 of file utilities.c.

5.11.2.3 int GetLine (char * sInputString, unsigned int unMaxSize)

Retrieve a line of characters from the default Hyperterminal UART (DEFAULT_HYPERTERMINAL_UART).

Details

Maximum number of characters can be specified. Function will timeout after 10 seconds.

Parameters

<i>in</i>	<i>sInputString</i>	- pointer to buffer for input string
<i>in</i>	<i>unMaxSize</i>	- maximum number of characters to input

Return values

<i>TRUE</i>	if operation succeeded
-------------	------------------------

Definition at line 393 of file *utilities.c*.

5.11.2.4 u8 getUartByte (u32 nUartAddress)

Get a byte from either the full UART in the Zynq PS, or a Xilinx UartLite.

Parameters

<i>in</i>	<i>unUartAddress</i>	32 bit UART address
-----------	----------------------	---------------------

Return values

<i>Returns</i>	the next byte from the Uart Rx FIFO
----------------	-------------------------------------

Definition at line 442 of file *utilities.c*.

5.11.2.5 int htoi (char * sStr)

Definition at line 457 of file *utilities.c*.

5.11.2.6 void led_knight_rider (XGpio * pLED_GPIO, int nNumberOfTimes)

Blink a row of LEDs nNumberOfTimes times.

Details

The Digilent NEXYS-3 board has 8 green LEDs located above the toggle switches. This function blinks them back/forth (a bit like the KITT car from Knight Rider)

Parameters

<i>in</i>	<i>*pLED_GPIO</i>	- address of the GPIO peripheral driving the LEDs in MicroBlaze memory map
-----------	-------------------	--

Return values

<i>None</i>	
-------------	--

Definition at line 186 of file *utilities.c*.

5.11.2.7 void max_configure_PMOD_port (u8 uchPmodPortA, u8 uchPmodPortB, u8 uchPmodPortC, u8 uchPmodPortD)

Configure driving peripherals on each Pmod port.

Details

The Maxim HDL hardware design for the Nexys 3 includes a multiplexer on each PMOD port to allow I2C, SPI, GPIO, and UART functionality to be selected on each port. The standard configuration is (PortA = I2C, PortB = SPI, PortC = GPIO, PortD = UART) This function is used to set the standard configuration in the [main\(\)](#) function, as well as to optionally change the port config. The 2-bit number used to define the port configuration is encoded as follows:

00=UART, 01=SPI, 10=GPIO and 11=I2C

Parameters

in	<i>uchPmodPortA</i>	- 2-bit number to define configuration for Pmod port A
in	<i>uchPmodPortB</i>	- 2-bit number to define configuration for Pmod port B
in	<i>uchPmodPortC</i>	- 2-bit number to define configuration for Pmod port C
in	<i>uchPmodPortD</i>	- 2-bit number to define configuration for Pmod port D

Return values

<i>TRUE</i>	if operation succeeded
-------------	------------------------

Definition at line 221 of file utilities.c.

5.11.2.8 int number_raised_to_power (int nBase, int nExponent)

Raise nBase to the nExponent power (operates with integers only).

Details

Many Microblaze applications will not have math.h included due to limited memory space. This is a simple functions to implement $(nBase \wedge nExponent)$ Some Maxim devices (such as MAX44009) return values in mantissa + (power of 2) exponent format.

Parameters

in	<i>nBase</i>	- base
in	<i>nExponent</i>	- exponent

Return values

<i>Base^{Exponent}</i>	
--------------------------------	--

Definition at line 308 of file utilities.c.

5.11.2.9 void print_asterisks (int nQuantity)

Print nQuantity of asterisks to the default Hyperterminal UART.

Parameters

in	<i>nQuantity</i>	- number of asterisks to print
----	------------------	--------------------------------

Return values

<i>None</i>

Definition at line 64 of file utilities.c.

5.11.2.10 void print_seven_segment_four_digits (char * *puchArray*)

5.11.2.11 void print_seven_segment_number (float *fNumber*)

5.11.2.12 void print_seven_segment_temperature (float *temp*, u8 *uchPrintCelsius*)

5.11.2.13 void print_seven_segment_time (struct maximDateTime * *t*)

5.11.2.14 void printf_temp (float *fTemp*, u8 *uchPrintCelsius*, u8 *uchAddCarriageReturn*)

Print temperature as xx.x as degrees F or C.

Details

This function prints the temperature with .1 degree resolution in either degrees Celsius (*uchPrintCelsius*==TRUE) -or- Fahrenheit (*uchPrintCelsius*==FALSE)

Parameters

in	<i>fTemp</i>	- temperature to print in celsius
in	<i>uchPrintCelsius</i>	- true = print as celsius, false = print as fahrenheit
in	<i>uchAddCarriageReturn</i>	- true = add a carriage return

Return values

<i>None</i>

Definition at line 420 of file utilities.c.

5.11.2.15 int receive_byte_with_timeout (u32 *unUartAddress*, int *nTimeoutInTenthsOfSeconds*, u8 * *uchRxData*)

Receive a byte from the UART located at *pUartAddress.

Parameters

in	<i>unUartAddress</i>	- address of the UART peripheral in MicroBlaze memory map
in	<i>nTimeoutInTenthsOfSeconds</i>	- amount of time to allow before TIMEOUT
out	<i>*uchRxData</i>	- received data is stored at <i>uchRxData</i>

Return values

<i>TRUE</i>	if operation succeeded
-------------	------------------------

Definition at line 337 of file utilities.c.

5.11.2.16 void set_seven_segment_character (u8 uchDigitNumber, u8 uchValue, u8 uchDecimalActive)

5.11.2.17 int SpiRW (u32 unPeripheralAddressSPI, unsigned int unCPHA, unsigned int unCPOL, u8 * auchWriteBuf, u8 * auchReadBuf, int unNumBytes, u8 uchCsActiveHigh)

Perform a SPI read or write.

Details

This function provides a combination SPI Read and Write to the chosen SPI port in the design CPHA and CPOL can be set to 0 or 1 Pointers are provided to u8 buffers containing the data to be written and received Data in the auchWriteBuf will be clocked out (MSB first) onto the MOSI pin Data from the MISO pin will be placed into the auchReadBuf uchCsActiveHigh==TRUE allows SS configurations to be used uchCsActiveHigh==FALSE allows SS# configurations to be used

Parameters

in	<i>unPeripheralAddressSPI</i>	-
in	<i>unCPHA</i>	- phase of SCK (edge to trigger on). 0=Leading edge, 1=Trailing edge
in	<i>unCPOL</i>	- polarity of SCK. 0=Active high, 1=Active low
in	<i>auchWriteBuf</i>	- pointer to write data buffer
in	<i>auchReadBuf</i>	- pointer to read data buffer
in	<i>unNumBytes</i>	- number of bytes to transfer
in	<i>uchCsActiveHigh</i>	- polarity of slave select 0=active low, 1=active high

Return values

<i>Always</i>	returns 0
---------------	-----------

Definition at line 78 of file utilities.c.

5.11.3 Variable Documentation

5.11.3.1 struct maximDateTime* t

Definition at line 101 of file utilities.h.

Index

BASE_FUNCTION_STATE
 menu.h, 26

check_if_i2c_device_is_present
 utilities.c, 32
 utilities.h, 38

cleanup_platform
 platform.c, 30
 platform.h, 30

DE
 MAXREFDES14.h, 19

DI
 MAXREFDES14.h, 19

delay
 utilities.c, 32
 utilities.h, 38

disable_caches
 platform.c, 30

enable_caches
 platform.c, 30

g_PmodPortMuxSettings
 MAXREFDES14.c, 16

g_gpButtons
 MAXREFDES14.c, 16

g_slInputString
 MAXREFDES14.c, 16
 MAXREFDES14.h, 21

g_unSevenSegmentValue
 MAXREFDES14.c, 16

g_xGpioLed
 MAXREFDES14.c, 16

g_xGpioPmodPortA
 MAXREFDES14.c, 16
 MAXREFDES14.h, 21

g_xGpioPmodPortB
 MAXREFDES14.c, 16
 MAXREFDES14.h, 21

g_xGpioSevenSegment
 MAXREFDES14.c, 16

GetLine
 utilities.c, 32
 utilities.h, 38

getUartByte
 utilities.c, 33
 utilities.h, 39

htoi
 utilities.c, 33
 utilities.h, 39

init_platform
 platform.c, 30
 platform.h, 30

init_uart
 platform.c, 30

KEYPRESS_ARROW_DOWN
 menu.h, 26

KEYPRESS_ARROW_LEFT
 menu.h, 26

KEYPRESS_ARROW_RIGHT
 menu.h, 26

KEYPRESS_ARROW_UP
 menu.h, 27

KEYPRESS_END
 menu.h, 27

led_knight_rider
 utilities.c, 33
 utilities.h, 39

MAIN_MENU
 menu.h, 27

MAJOR_REVISION
 MAXREFDES14.c, 15

MAXREFDES14.c
 g_PmodPortMuxSettings, 16
 g_gpButtons, 16
 g_slInputString, 16
 g_unSevenSegmentValue, 16
 g_xGpioLed, 16
 g_xGpioPmodPortA, 16
 g_xGpioPmodPortB, 16
 g_xGpioSevenSegment, 16
 main, 15

MAXREFDES14.h
 DE, 19
 DI, 19
 g_slInputString, 21

- g_xGpioPmodPortA, 21
 - g_xGpioPmodPortB, 21
 - RE_B, 20
 - RO, 20
- MENU_READ_REG
 - menu.h, 27
- MENU_REG_ADDR
 - menu.h, 27
- MENU_WRITE_REG
 - menu.h, 27
- MINOR_REVISION
 - MAXREFDES14.c, 15
- main
 - MAXREFDES14.c, 15
- max_configure_PMOD_port
 - utilities.c, 34
 - utilities.h, 39
- maximDateTime, 7
 - uchDate, 7
 - uchDow, 7
 - uchHour, 7
 - uchMinute, 8
 - uchMonth, 8
 - uchSecond, 8
 - uchYear, 8
- maximDeviceSpecificUtilities.c
 - read_reg, 12
 - write_reg, 12
- maximDeviceSpecificUtilities.h
 - read_reg, 13
 - write_reg, 14
- maximProduct, 8
 - sPartName, 8
 - sPartNumber, 8
 - uchKeypress, 9
 - uchPmodType, 9
- menu.c
 - menu_cls, 22
 - menu_print_line, 22
 - menu_print_main_menu, 23
 - menu_print_maxim_banner, 23
 - menu_print_maxim_banner_big, 23
 - menu_print_prompt, 23
 - menu_read_reg, 24
 - menu_retrieve_keypress, 24
- menu.h
 - BASE_FUNCTION_STATE, 26
 - KEYPRESS_ARROW_DOWN, 26
 - KEYPRESS_ARROW_LEFT, 26
 - KEYPRESS_ARROW_UP, 27
 - KEYPRESS_END, 27
 - MAIN_MENU, 27
 - MENU_READ_REG, 27
 - MENU_REG_ADDR, 27
 - MENU_WRITE_REG, 27
 - menu_cls, 27
 - menu_print_main_menu, 28
 - menu_print_maxim_banner, 28
 - menu_print_maxim_banner_big, 28
 - menu_print_prompt, 28
 - menu_retrieve_keypress, 29
 - WAIT_KEYPRESS, 27
- menu_cls
 - menu.c, 22
 - menu.h, 27
- menu_print_line
 - menu.c, 22
- menu_print_main_menu
 - menu.c, 23
 - menu.h, 28
- menu_print_maxim_banner
 - menu.c, 23
 - menu.h, 28
- menu_print_maxim_banner_big
 - menu.c, 23
 - menu.h, 28
- menu_print_prompt
 - menu.c, 23
 - menu.h, 28
- menu_read_reg
 - menu.c, 24
- menu_retrieve_keypress
 - menu.c, 24
 - menu.h, 29
- number_raised_to_power
 - utilities.c, 34
 - utilities.h, 40
- PMOD_TYPE_GPIO
 - MAXREFDES14.h, 20
- PMOD_TYPE_I2C
 - MAXREFDES14.h, 20
- PMOD_TYPE_SPI
 - MAXREFDES14.h, 20
- PMOD_TYPE_UART
 - MAXREFDES14.h, 20
- platform.c
 - cleanup_platform, 30
 - disable_caches, 30
 - enable_caches, 30
 - init_platform, 30
 - init_uart, 30
 - UART_BAUD, 30
- platform.h
 - cleanup_platform, 30
 - init_platform, 30
- print_asterisks
 - utilities.c, 35

- utilities.h, 40
- print_seven_segment_four_digits
 - utilities.h, 41
- print_seven_segment_number
 - utilities.h, 41
- print_seven_segment_temperature
 - utilities.h, 41
- print_seven_segment_time
 - utilities.h, 41
- printf_temp
 - utilities.c, 35
 - utilities.h, 41
- RE_B
 - MAXREFDES14.h, 20
- RO
 - MAXREFDES14.h, 20
- read_reg
 - maximDeviceSpecificUtilities.c, 12
 - maximDeviceSpecificUtilities.h, 13
- receive_byte_with_timeout
 - utilities.c, 35
 - utilities.h, 41
- sPartName
 - maximProduct, 8
- sPartNumber
 - maximProduct, 8
- set_seven_segment_character
 - utilities.h, 42
- SpiRW
 - utilities.c, 36
 - utilities.h, 42
- src/MAXREFDES14.c, 14
- src/MAXREFDES14.h, 17
- src/maximDeviceSpecificUtilities.c, 11
- src/maximDeviceSpecificUtilities.h, 13
- src/menu.c, 21
- src/menu.h, 25
- src/platform.c, 29
- src/platform.h, 30
- src/platform_config.h, 31
- src/utilities.c, 31
- src/utilities.h, 36
- t
 - utilities.h, 42
- UART_BAUD
 - platform.c, 30
- uchDate
 - maximDateTime, 7
- uchDow
 - maximDateTime, 7
- uchHour
 - maximDateTime, 7
- uchKeypress
 - maximProduct, 9
- uchMinute
 - maximDateTime, 8
- uchMonth
 - maximDateTime, 8
- uchPmodType
 - maximProduct, 9
- uchSecond
 - maximDateTime, 8
- uchYear
 - maximDateTime, 8
- utilities.c
 - check_if_i2c_device_is_present, 32
 - delay, 32
 - GetLine, 32
 - getUartByte, 33
 - htoi, 33
 - led_knight_rider, 33
 - max_configure_PMOD_port, 34
 - number_raised_to_power, 34
 - print_asterisks, 35
 - printf_temp, 35
 - receive_byte_with_timeout, 35
 - SpiRW, 36
- utilities.h
 - check_if_i2c_device_is_present, 38
 - delay, 38
 - GetLine, 38
 - getUartByte, 39
 - htoi, 39
 - led_knight_rider, 39
 - max_configure_PMOD_port, 39
 - number_raised_to_power, 40
 - print_asterisks, 40
 - print_seven_segment_four_digits, 41
 - print_seven_segment_number, 41
 - print_seven_segment_temperature, 41
 - print_seven_segment_time, 41
 - printf_temp, 41
 - receive_byte_with_timeout, 41
 - set_seven_segment_character, 42
 - SpiRW, 42
 - t, 42
- WAIT_KEYPRESS
 - menu.h, 27
- write_reg
 - maximDeviceSpecificUtilities.c, 12
 - maximDeviceSpecificUtilities.h, 14