

Carmel (MAXREFDES18#) Code Documentation

V01.00

Generated by Doxygen 1.8.2

Wed Aug 28 2013 11:47:36

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	maximOLEDDisplay Struct Reference	7
4.1.1	Detailed Description	7
4.1.2	Field Documentation	7
4.1.2.1	flippedBuffer	7
4.1.2.2	font	7
4.1.2.3	portStatus	7
4.1.2.4	writeBuffer	8
4.1.2.5	xgpioPort	8
5	File Documentation	9
5.1	src/MAXREFDES18.c File Reference	9
5.1.1	Detailed Description	10
5.1.2	Macro Definition Documentation	10
5.1.2.1	MAJOR_REVISION	10
5.1.2.2	MINOR_REVISION	10
5.1.3	Function Documentation	10
5.1.3.1	main	10
5.1.4	Variable Documentation	11
5.1.4.1	font	11
5.1.4.2	g_auchPortType	11

5.1.4.3	g_nActivePmodPort	11
5.1.4.4	g_pActiveGpioPort	11
5.1.4.5	g_PmodPortMuxSettings	11
5.1.4.6	g_sInputString	11
5.1.4.7	g_structureOLED	11
5.1.4.8	g_tempString	11
5.1.4.9	g_unActivePeripheralAddressSPI	12
5.1.4.10	g_unSevenSegmentValue	12
5.1.4.11	g_xGpioLed	12
5.1.4.12	g_xGpioPmodPortA	12
5.1.4.13	g_xGpioPmodPortB	12
5.1.4.14	g_xGpioPmodPortC	12
5.1.4.15	g_xGpioPmodPortD	12
5.1.4.16	g_xGpioPmodPortMuxIO	12
5.1.4.17	g_xGpioSevenSegment	12
5.2	src/MAXREFDES18.h File Reference	13
5.2.1	Detailed Description	14
5.2.2	Macro Definition Documentation	15
5.2.2.1	ABOUT_ONE_SECOND	15
5.2.2.2	BIT_BANG_SPI_CLK	15
5.2.2.3	BIT_BANG_SPI_CLRB	15
5.2.2.4	BIT_BANG_SPI_MISO	15
5.2.2.5	BIT_BANG_SPI_MOSI	15
5.2.2.6	BIT_BANG_SPI_SSB	15
5.2.2.7	DEFAULT_HYPERTERMINAL_UART_ADDRESS	15
5.2.2.8	DEFAULT_HYPERTERMINAL_UART_ID	15
5.2.2.9	INPUT_STRING_MAX_SIZE	15
5.2.2.10	PMOD_PORT_TYPE_GPIO	16
5.2.2.11	PMOD_PORT_TYPE_I2C	16
5.2.2.12	PMOD_PORT_TYPE_SPI	16
5.2.2.13	PMOD_PORT_TYPE_UART	16
5.2.2.14	PMOD_TYPE_GPIO	16
5.2.2.15	PMOD_TYPE_I2C	16
5.2.2.16	PMOD_TYPE_SPI	16
5.2.2.17	PMOD_TYPE_UART	16
5.2.3	Variable Documentation	17
5.2.3.1	g_auchPortType	17

5.2.3.2	g_nActivePModPort	17
5.2.3.3	g_pActiveGPIOPort	17
5.2.3.4	g_sInputString	17
5.2.3.5	g_structureOLED	17
5.2.3.6	g_tempString	17
5.2.3.7	g_unActivePeripheralAddressI2C	17
5.2.3.8	g_unActivePeripheralAddressSPI	17
5.2.3.9	g_unActivePeripheralAddressUart	17
5.2.3.10	g_xGpioPmodPortA	17
5.2.3.11	g_xGpioPmodPortB	17
5.2.3.12	g_xGpioPmodPortC	17
5.2.3.13	g_xGpioPmodPortD	18
5.2.3.14	g_xGpioPmodPortMuxIO	18
5.3	src/menu.c File Reference	18
5.3.1	Detailed Description	19
5.3.2	Function Documentation	19
5.3.2.1	menu_cls	19
5.3.2.2	menu_get_direct_entry	19
5.3.2.3	menu_print_current_range_menu	20
5.3.2.4	menu_print_line	20
5.3.2.5	menu_print_maxim_banner	20
5.3.2.6	menu_print_maxim_banner_big	21
5.3.2.7	menu_print_mode_menu	21
5.3.2.8	menu_print_prompt	21
5.3.2.9	menu_print_voltage_range_menu	21
5.3.2.10	menu_retrieve_keypress	22
5.4	src/menu.h File Reference	22
5.4.1	Detailed Description	24
5.4.2	Macro Definition Documentation	24
5.4.2.1	BASE_FUNCTION_STATE	24
5.4.2.2	CURRENT_RANGE_0_20_MENU	24
5.4.2.3	CURRENT_RANGE_20_20_MENU	24
5.4.2.4	CURRENT_RANGE_4_20_MENU	24
5.4.2.5	CURRENT_RANGE_MENU	25
5.4.2.6	KEYPRESS_ARROW_DOWN	25
5.4.2.7	KEYPRESS_ARROW_LEFT	25
5.4.2.8	KEYPRESS_ARROW_RIGHT	25

5.4.2.9	KEYPRESS_ARROW_UP	25
5.4.2.10	KEYPRESS_END	25
5.4.2.11	MODE_MENU	25
5.4.2.12	VOLTAGE_RANGE_0_10_MENU	25
5.4.2.13	VOLTAGE_RANGE_0_5_MENU	26
5.4.2.14	VOLTAGE_RANGE_10_10_MENU	26
5.4.2.15	VOLTAGE_RANGE_MENU	26
5.4.2.16	WAIT_KEYPRESS	26
5.4.3	Function Documentation	26
5.4.3.1	menu_cls	26
5.4.3.2	menu_get_direct_entry	26
5.4.3.3	menu_print_current_range_menu	27
5.4.3.4	menu_print_maxim_banner	27
5.4.3.5	menu_print_maxim_banner_big	27
5.4.3.6	menu_print_mode_menu	28
5.4.3.7	menu_print_prompt	28
5.4.3.8	menu_print_voltage_range_menu	28
5.4.3.9	menu_retrieve_keypress	28
5.5	src/platform.c File Reference	29
5.5.1	Macro Definition Documentation	29
5.5.1.1	UART_BAUD	29
5.5.2	Function Documentation	29
5.5.2.1	cleanup_platform	29
5.5.2.2	disable_caches	29
5.5.2.3	enable_caches	29
5.5.2.4	init_platform	30
5.5.2.5	init_uart	30
5.6	src/platform.h File Reference	30
5.6.1	Function Documentation	30
5.6.1.1	cleanup_platform	30
5.6.1.2	init_platform	30
5.7	src/platform_config.h File Reference	30
5.7.1	Macro Definition Documentation	30
5.7.1.1	STDOUT_IS_PS7_UART	30
5.7.1.2	UART_DEVICE_ID	30
5.8	src/utilities.c File Reference	31
5.8.1	Detailed Description	32

5.8.2	Macro Definition Documentation	32
5.8.2.1	OLED_DATA_COMMAND_B	32
5.8.2.2	OLED_RESET_B	32
5.8.2.3	OLED_SCLK	33
5.8.2.4	OLED_SDIN	33
5.8.2.5	OLED_VBAT	33
5.8.2.6	OLED_VDD	33
5.8.3	Function Documentation	33
5.8.3.1	checkUartEmpty	33
5.8.3.2	clearOLEDBuffer	33
5.8.3.3	delay	34
5.8.3.4	displayOLEDBuffer	34
5.8.3.5	flipAndCopyDisplayBuffer	34
5.8.3.6	GetLine	35
5.8.3.7	getUartByte	35
5.8.3.8	initializeOLED	35
5.8.3.9	led_knight_rider	36
5.8.3.10	max_configure_PMOD_port	36
5.8.3.11	max_set_PMOD_port	37
5.8.3.12	number_raised_to_power	37
5.8.3.13	print_asterisks	37
5.8.3.14	printfToBufferOLED	38
5.8.3.15	printfToOLED	38
5.8.3.16	putCharOLED	38
5.8.3.17	receive_byte_with_timeout	39
5.8.3.18	sendOLEDSPI	39
5.8.3.19	sendUartByte	39
5.8.3.20	SpiRW	40
5.9	src/utilities.h File Reference	41
5.9.1	Detailed Description	42
5.9.2	Function Documentation	42
5.9.2.1	checkUartEmpty	42
5.9.2.2	clearOLEDBuffer	43
5.9.2.3	delay	43
5.9.2.4	displayOLEDBuffer	43
5.9.2.5	flipAndCopyDisplayBuffer	44
5.9.2.6	GetLine	44

5.9.2.7	getUartByte	45
5.9.2.8	initializeOLED	45
5.9.2.9	led_knight_rider	45
5.9.2.10	max_configure_PMOD_port	46
5.9.2.11	max_set_PMOD_port	46
5.9.2.12	number_raised_to_power	46
5.9.2.13	print_asterisks	47
5.9.2.14	printfToBufferOLED	47
5.9.2.15	printfToOLED	47
5.9.2.16	putCharOLED	48
5.9.2.17	receive_byte_with_timeout	48
5.9.2.18	sendOLEDSPI	49
5.9.2.19	sendUartByte	49
5.9.2.20	SpiRW	49

Chapter 1

Main Page

1.1 Introduction

This is the code documentation for the Carmel (MAXREFDES18#) 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:

maximOLEDDisplay	7
----------------------------------	---

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

src/ MAXREFDES18.c	9
src/ MAXREFDES18.h	13
src/ menu.c	18
src/ menu.h	22
src/ platform.c	29
src/ platform.h	30
src/ platform_config.h	30
src/ utilities.c	31
src/ utilities.h	41

Chapter 4

Data Structure Documentation

4.1 maximOLEDDisplay Struct Reference

```
#include <MAXREFDES18.h>
```

Data Fields

- XGpio [xgpioPort](#)
- u8 [portStatus](#)
- u8 [writeBuffer](#) [512]
- u8 [flippedBuffer](#) [512]
- u8 * [font](#)

4.1.1 Detailed Description

Definition at line 100 of file MAXREFDES18.h.

4.1.2 Field Documentation

4.1.2.1 u8 flippedBuffer[512]

Definition at line 105 of file MAXREFDES18.h.

4.1.2.2 u8* font

Definition at line 106 of file MAXREFDES18.h.

4.1.2.3 u8 portStatus

Definition at line 103 of file MAXREFDES18.h.

4.1.2.4 u8 writeBuffer[512]

Definition at line 104 of file MAXREFDES18.h.

4.1.2.5 XGpio xgpioPort

Definition at line 102 of file MAXREFDES18.h.

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

- src/[MAXREFDES18.h](#)

Chapter 5

File Documentation

5.1 src/MAXREFDES18.c File Reference

```
#include <stdio.h>
#include "platform.h"
#include "menu.h"
#include "utilities.h"
#include "MAXREFDES18.h"
```

Macros

- #define MAJOR_REVISION 1
- #define MINOR_REVISION 0

Functions

- int main ()
Main() function for MAXREFDES18.

Variables

- XGpio g_xGpioSevenSegment
- unsigned int g_unSevenSegmentValue =0
- char g_slInputString [INPUT_STRING_MAX_SIZE]
- XGpio g_xGpioPmodPortA
- XGpio g_xGpioPmodPortB
- XGpio g_xGpioPmodPortC
- XGpio g_xGpioPmodPortD
- XGpio g_xGpioPmodPortMuxIO
- XGpio g_xGpioLed
- struct maximOLEDDisplay g_structureOLED
- char g_tempString [32]
- u8 g_PmodPortMuxSettings =0
- int g_nActivePmodPort =0

- u32 [g_unActivePeripheralAddressSPI](#) = XPAR_SPI_0_BASEADDR
- XGpio * [g_pActiveGPIOPort](#) = &[g_xGpioPmodPortA](#)
- unsigned char [g_auchPortType](#) [4]
- u8 [font](#) [1024]

5.1.1 Detailed Description

```
Project: Carmel (MAXREFDES18#)
Filename: MAXREFDES18.c
Description: This module contains the Main application for the
             Avnet ZedBoard implementation of the MAXREFDES18
             example program.
```

Revision History:

08-19-13 Rev 01.00 MG 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 [MAXREFDES18.c](#).

5.1.2 Macro Definition Documentation

5.1.2.1 #define MAJOR_REVISION 1

Definition at line 83 of file MAXREFDES18.c.

5.1.2.2 #define MINOR_REVISION 0

Definition at line 84 of file MAXREFDES18.c.

5.1.3 Function Documentation

5.1.3.1 int main ()

Main() function for MAXREFDES18.

Details

This function sets up and initializes the FPGA and hardware, displays the menu via Hyperterminal (or equivalent Terminal program i.e. Teraterm).

Parameters

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

Return values

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

Definition at line 235 of file MAXREFDES18.c.

5.1.4 Variable Documentation**5.1.4.1 u8 font[1024]**

Definition at line 105 of file MAXREFDES18.c.

5.1.4.2 unsigned char g_auchPortType[4]

Definition at line 103 of file MAXREFDES18.c.

5.1.4.3 int g_nActivePModPort =0

Definition at line 100 of file MAXREFDES18.c.

5.1.4.4 XGpio* g_pActiveGPIOPort = &g_xGpioPmodPortA

Definition at line 102 of file MAXREFDES18.c.

5.1.4.5 u8 g_PmodPortMuxSettings =0

Definition at line 99 of file MAXREFDES18.c.

5.1.4.6 char g_sInputString[INPUT_STRING_MAX_SIZE]

Definition at line 89 of file MAXREFDES18.c.

5.1.4.7 struct maximOLEDDisplay g_structureOLED

Definition at line 96 of file MAXREFDES18.c.

5.1.4.8 char g_tempString[32]

Definition at line 97 of file MAXREFDES18.c.

5.1.4.9 u32 g_unActivePeripheralAddressSPI =XPAR_SPI_0_BASEADDR

Definition at line 101 of file MAXREFDES18.c.

5.1.4.10 unsigned int g_unSevenSegmentValue =0

Definition at line 88 of file MAXREFDES18.c.

5.1.4.11 XGpio g_xGpioLed

Definition at line 95 of file MAXREFDES18.c.

5.1.4.12 XGpio g_xGpioPmodPortA

Definition at line 90 of file MAXREFDES18.c.

5.1.4.13 XGpio g_xGpioPmodPortB

Definition at line 91 of file MAXREFDES18.c.

5.1.4.14 XGpio g_xGpioPmodPortC

Definition at line 92 of file MAXREFDES18.c.

5.1.4.15 XGpio g_xGpioPmodPortD

Definition at line 93 of file MAXREFDES18.c.

5.1.4.16 XGpio g_xGpioPmodPortMuxIO

Definition at line 94 of file MAXREFDES18.c.

5.1.4.17 XGpio g_xGpioSevenSegment

Definition at line 87 of file MAXREFDES18.c.

5.2 src/MAXREFDES18.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"
#include "math.h"
```

Data Structures

- struct [maximOLEDDisplay](#)

Macros

- #define [DEFAULT_HYPERTERMINAL_UART_ID](#) XPAR_PS7_UART_1_DEVICE_ID
macro used to abstract Physical Port of Hyperterminal UART
- #define [DEFAULT_HYPERTERMINAL_UART_ADDRESS](#) XPAR_PS7_UART_1_BASEADDR
- #define [ABOUT_ONE_SECOND](#) 74067512
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 [INPUT_STRING_MAX_SIZE](#) 16

Variables

- XGpio [g_xGpioPmodPortA](#)
- XGpio [g_xGpioPmodPortB](#)
- XGpio [g_xGpioPmodPortC](#)
- XGpio [g_xGpioPmodPortD](#)
- XGpio [g_xGpioPmodPortMuxIO](#)
- char [g_sInputString](#) [[INPUT_STRING_MAX_SIZE](#)]
- int [g_nActivePmodPort](#)
- u32 [g_unActivePeripheralAddressI2C](#)
- u32 [g_unActivePeripheralAddressSPI](#)
- u32 [g_unActivePeripheralAddressUart](#)
- XGpio * [g_pActiveGPIOPort](#)
- struct [maximOLEDDisplay](#) [g_structureOLED](#)
- char [g_tempString](#) [32]
- unsigned char [g_auchPortType](#) [4]

5.2.1 Detailed Description

```
Project: Carmel (MAXREFDES18#)
Filename: MAXREFDES18.h
Description: This module contains the Main application for the
            Avnet ZedBoard implementation of the MAXREFDES18
            example program.
```

Revision History:

08-19-13 Rev 01.00 MG 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 [MAXREFDES18.h](#).

5.2.2 Macro Definition Documentation

5.2.2.1 **#define ABOUT_ONE_SECOND 74067512**

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

Definition at line 80 of file MAXREFDES18.h.

5.2.2.2 **#define BIT_BANG_SPI_CLK 0x08**

bit mask for SPI CLK used in max_GPIO_bit_bang_SPI_write

Definition at line 94 of file MAXREFDES18.h.

5.2.2.3 **#define BIT_BANG_SPI_CLRB 0x04**

bit mask for SPI /CLR used in max_GPIO_bit_bang_SPI_write

Definition at line 95 of file MAXREFDES18.h.

5.2.2.4 **#define BIT_BANG_SPI_MISO 0x04**

bit mask for SPI MISO used in max_GPIO_bit_bang_SPI_write

Definition at line 96 of file MAXREFDES18.h.

5.2.2.5 **#define BIT_BANG_SPI_MOSI 0x02**

bit mask for SPI MOSI used in max_GPIO_bit_bang_SPI_write

Definition at line 97 of file MAXREFDES18.h.

5.2.2.6 **#define BIT_BANG_SPI_SSB 0x01**

bit mask for SPI SS used in max_GPIO_bit_bang_SPI_write

Definition at line 98 of file MAXREFDES18.h.

5.2.2.7 **#define DEFAULT_HYPERTERMINAL_UART_ADDRESS XPAR_PS7_UART_1_BASEADDR**

Definition at line 78 of file MAXREFDES18.h.

5.2.2.8 **#define DEFAULT_HYPERTERMINAL_UART_ID XPAR_PS7_UART_1_DEVICE_ID**

macro used to abstract Physical Port of Hyperterminal UART

Definition at line 77 of file MAXREFDES18.h.

5.2.2.9 **#define INPUT_STRING_MAX_SIZE 16**

Definition at line 109 of file MAXREFDES18.h.

5.2.2.10 #define PMOD_PORT_TYPE_GPIO 0x02

value to switch GPIO controller into a Pmod port in max_configure_PMOD_port
Definition at line 91 of file MAXREFDES18.h.

5.2.2.11 #define PMOD_PORT_TYPE_I2C 0x03

value to switch I2C controller into a Pmod port in max_configure_PMOD_port
Definition at line 92 of file MAXREFDES18.h.

5.2.2.12 #define PMOD_PORT_TYPE_SPI 0x01

value to switch SPI controller into a Pmod port in max_configure_PMOD_port
Definition at line 90 of file MAXREFDES18.h.

5.2.2.13 #define PMOD_PORT_TYPE_UART 0x00

value to switch UART controller into a Pmod port in max_configure_PMOD_port
Definition at line 89 of file MAXREFDES18.h.

5.2.2.14 #define PMOD_TYPE_GPIO 2

value which specifies GPIO in maximProduct structure
Definition at line 86 of file MAXREFDES18.h.

5.2.2.15 #define PMOD_TYPE_I2C 3

value which specifies I2C in maximProduct structure
Definition at line 87 of file MAXREFDES18.h.

5.2.2.16 #define PMOD_TYPE_SPI 1

value which specifies SPI in maximProduct structure
Definition at line 85 of file MAXREFDES18.h.

5.2.2.17 #define PMOD_TYPE_UART 0

value which specifies UART in maximProduct structure
Definition at line 84 of file MAXREFDES18.h.

5.2.3 Variable Documentation

5.2.3.1 unsigned char g_auchPortType[4]

Definition at line 103 of file MAXREFDES18.c.

5.2.3.2 int g_nActivePModPort

Definition at line 100 of file MAXREFDES18.c.

5.2.3.3 XGpio* g_pActiveGPIOPort

Definition at line 102 of file MAXREFDES18.c.

5.2.3.4 char g_sInputString[INPUT_STRING_MAX_SIZE]

Definition at line 89 of file MAXREFDES18.c.

5.2.3.5 struct maximOLEDDisplay g_structureOLED

Definition at line 96 of file MAXREFDES18.c.

5.2.3.6 char g_tempString[32]

Definition at line 97 of file MAXREFDES18.c.

5.2.3.7 u32 g_unActivePeripheralAddressI2C

5.2.3.8 u32 g_unActivePeripheralAddressSPI

Definition at line 101 of file MAXREFDES18.c.

5.2.3.9 u32 g_unActivePeripheralAddressUart

5.2.3.10 XGpio g_xGpioPmodPortA

Definition at line 90 of file MAXREFDES18.c.

5.2.3.11 XGpio g_xGpioPmodPortB

Definition at line 91 of file MAXREFDES18.c.

5.2.3.12 XGpio g_xGpioPmodPortC

Definition at line 92 of file MAXREFDES18.c.

5.2.3.13 XGpio g_xGpioPmodPortD

Definition at line 93 of file MAXREFDES18.c.

5.2.3.14 XGpio g_xGpioPmodPortMuxIO

Definition at line 94 of file MAXREFDES18.c.

5.3 src/menu.c File Reference

```
#include <stdio.h>
#include "platform.h"
#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 "MAXREFDES18.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.
- unsigned int [menu_get_direct_entry](#) (u32 nUartAddress, int nNumberBits)
Retrieve keyboard entry of a value via the Hyperterminal connected UART.
- u8 [menu_retrieve_keypress](#) (u32 nUartAddress)
Get a single keypress via Hyperterminal.
- void [menu_print_mode_menu](#) ()
Print the mode menu.
- void [menu_print_current_range_menu](#) ()
Print the current range menu.
- void [menu_print_voltage_range_menu](#) ()
Print the voltage range menu.

5.3.1 Detailed Description

Project: Carmel (MAXREFDES18#)
 Filename: menu.c
 Description: This module contains all the functions used to generate the menus and menu options used to run the MAXREFDES18# example firmware.

Revision History:

08-19-13 Rev 01.00 MG 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](#).

5.3.2 Function Documentation

5.3.2.1 void menu_cls ()

Function to clear the screen via Hyperterminal.

Parameters

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

Return values

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

Definition at line 75 of file menu.c.

5.3.2.2 unsigned int menu_get_direct_entry (u32 nUartAddress, int nNumberBits)

Retrieve keyboard entry of a value via the Hyperterminal connected UART.

Details

In most cases, this function is used to directly populate a register with a value. int nNumberBits equals the number

of bits in the register. The value input by the user is capped to the maximum allowable for the given number of bits (e.g. - 6 bits => '64' max)

Parameters

in	<i>nUartAddress</i>	- address of the UART peripheral in the memory map
in	<i>nNumberBits</i>	- number of bits in register populate

Return values

<i>Value</i>	entered
--------------	---------

Definition at line 171 of file menu.c.

5.3.2.3 void menu_print_current_range_menu ()

Print the current range menu.

Parameters

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

Return values

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

Definition at line 287 of file menu.c.

5.3.2.4 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 159 of file menu.c.

5.3.2.5 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 88 of file menu.c.

5.3.2.6 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 112 of file menu.c.

5.3.2.7 void menu_print_mode_menu ()

Print the mode menu.

Parameters

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

Return values

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

Definition at line 269 of file menu.c.

5.3.2.8 void menu_print_prompt ()

Print a standard prompt for keyboard input ">>".

Parameters

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

Return values

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

Definition at line 145 of file menu.c.

5.3.2.9 void menu_print_voltage_range_menu ()

Print the voltage range menu.

Parameters

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

Return values

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

Definition at line 308 of file menu.c.

5.3.2.10 u8 menu_retrieve_keypress (u32 nUartAddress)

Get a single keypress via Hyperterminal.

Details

Returns ascii character corresponding to keypress with some preprocessing.
 Escape sequences (Arrow keys and END) are mapped to decimal 240-244 (see defines)
 Characters "0"-"9" converted to numbers 0-9
 Lower case "a"-"z" converted to uppercase "A"-"Z"

Parameters

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

Return values

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

Definition at line 225 of file menu.c.

5.4 src/menu.h File Reference

```
#include "xbasic_types.h"
#include "xspi_l.h"
#include "stdio.h"
#include "MAXREFDES18.h"
```

Macros

- #define **MODE_MENU** 0
Menu state machine state.
- #define **WAIT_KEYPRESS** 1
Menu state machine state.
- #define **BASE_FUNCTION_STATE** 10
Menu state machine state.
- #define **CURRENT_RANGE_MENU** 2
Menu state machine state.

- #define `VOLTAGE_RANGE_MENU` 3
Menu state machine state.
- #define `CURRENT_RANGE_20_20_MENU` 4
Menu state machine state.
- #define `CURRENT_RANGE_0_20_MENU` 5
Menu state machine state.
- #define `CURRENT_RANGE_4_20_MENU` 6
Menu state machine state.
- #define `VOLTAGE_RANGE_10_10_MENU` 13
Menu state machine state.
- #define `VOLTAGE_RANGE_0_10_MENU` 14
Menu state machine state.
- #define `VOLTAGE_RANGE_0_5_MENU` 15
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 ">>".
- unsigned int `menu_get_direct_entry` (u32 nUartAddress, int nNumberBits)
Retrieve keyboard entry of a value via the Hyperterminal connected UART.
- u8 `menu_retrieve_keypress` (u32 nUartAddress)
Get a single keypress via Hyperterminal.
- void `menu_print_mode_menu` ()
Print the mode menu.
- void `menu_print_current_range_menu` ()
Print the current range menu.
- void `menu_print_voltage_range_menu` ()
Print the voltage range menu.

5.4.1 Detailed Description

```
Project: Carmel (MAXREFDES18#)
Filename: menu.h
Description: This module contains all the functions used to
             generate the menus and menu options used to run the
             MAXREFDES18# example firmware.
```

Revision History:

08-19-13 Rev 01.00 MG 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](#).

5.4.2 Macro Definition Documentation

5.4.2.1 #define BASE_FUNCTION_STATE 10

Menu state machine state.

Definition at line 71 of file menu.h.

5.4.2.2 #define CURRENT_RANGE_0_20_MENU 5

Menu state machine state.

Definition at line 76 of file menu.h.

5.4.2.3 #define CURRENT_RANGE_20_20_MENU 4

Menu state machine state.

Definition at line 75 of file menu.h.

5.4.2.4 #define CURRENT_RANGE_4_20_MENU 6

Menu state machine state.

Definition at line 77 of file menu.h.

5.4.2.5 #define CURRENT_RANGE_MENU 2

Menu state machine state.

Definition at line 72 of file menu.h.

5.4.2.6 #define KEYPRESS_ARROW_DOWN 241

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

Definition at line 84 of file menu.h.

5.4.2.7 #define KEYPRESS_ARROW_LEFT 242

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

Definition at line 85 of file menu.h.

5.4.2.8 #define KEYPRESS_ARROW_RIGHT 243

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

Definition at line 86 of file menu.h.

5.4.2.9 #define KEYPRESS_ARROW_UP 240

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

Definition at line 83 of file menu.h.

5.4.2.10 #define KEYPRESS_END 244

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

Definition at line 87 of file menu.h.

5.4.2.11 #define MODE_MENU 0

Menu state machine state.

Definition at line 69 of file menu.h.

5.4.2.12 #define VOLTAGE_RANGE_0_10_MENU 14

Menu state machine state.

Definition at line 80 of file menu.h.

5.4.2.13 #define VOLTAGE_RANGE_0_5_MENU 15

Menu state machine state.

Definition at line 81 of file menu.h.

5.4.2.14 #define VOLTAGE_RANGE_10_10_MENU 13

Menu state machine state.

Definition at line 79 of file menu.h.

5.4.2.15 #define VOLTAGE_RANGE_MENU 3

Menu state machine state.

Definition at line 73 of file menu.h.

5.4.2.16 #define WAIT_KEYPRESS 1

Menu state machine state.

Definition at line 70 of file menu.h.

5.4.3 Function Documentation**5.4.3.1 void menu_cls ()**

Function to clear the screen via Hyperterminal.

Parameters

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

Return values

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

Definition at line 75 of file menu.c.

5.4.3.2 unsigned int menu_get_direct_entry (u32 nUartAddress, int nNumberBits)

Retrieve keyboard entry of a value via the Hyperterminal connected UART.

Details

In most cases, this function is used to directly populate a register with a value. int nNumberBits equals the number of bits in the register. The value input by the user is capped to the maximum allowable for the given number of bits (e.g. - 6 bits => '64' max)

Parameters

in	<i>nUartAddress</i>	- address of the UART peripheral in the memory map
in	<i>nNumberBits</i>	- number of bits in register populate

Return values

<i>Value</i>	entered
--------------	---------

Definition at line 171 of file menu.c.

5.4.3.3 void menu_print_current_range_menu ()

Print the current range menu.

Parameters

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

Return values

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

Definition at line 287 of file menu.c.

5.4.3.4 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 88 of file menu.c.

5.4.3.5 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 112 of file menu.c.

5.4.3.6 void menu_print_mode_menu ()

Print the mode menu.

Parameters

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

Return values

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

Definition at line 269 of file menu.c.

5.4.3.7 void menu_print_prompt ()

Print a standard prompt for keyboard input ">>".

Parameters

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

Return values

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

Definition at line 145 of file menu.c.

5.4.3.8 void menu_print_voltage_range_menu ()

Print the voltage range menu.

Parameters

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

Return values

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

Definition at line 308 of file menu.c.

5.4.3.9 u8 menu_retrieve_keypress (u32 nUartAddress)

Get a single keypress via Hyperterminal.

Details

Returns ascii character corresponding to keypress with some preprocessing.
Escape sequences (Arrow keys and END) are mapped to decimal 240-244 (see defines)
Characters "0"-"9" converted to numbers 0-9
Lower case "a"-"z" converted to uppercase "A"-"Z"

Parameters

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

Return values

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

Definition at line 225 of file menu.c.

5.5 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](#) ()

5.5.1 Macro Definition Documentation

5.5.1.1 #define UART_BAUD 9600

Definition at line 28 of file platform.c.

5.5.2 Function Documentation

5.5.2.1 void cleanup_platform ()

Definition at line 70 of file platform.c.

5.5.2.2 void disable_caches ()

Definition at line 47 of file platform.c.

5.5.2.3 void enable_caches ()

Definition at line 31 of file platform.c.

5.5.2.4 void init_platform ()

Definition at line 63 of file platform.c.

5.5.2.5 void init_uart ()

Definition at line 54 of file platform.c.

5.6 src/platform.h File Reference

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

Functions

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

5.6.1 Function Documentation

5.6.1.1 void cleanup_platform ()

Definition at line 70 of file platform.c.

5.6.1.2 void init_platform ()

Definition at line 63 of file platform.c.

5.7 src/platform_config.h File Reference

Macros

- #define [STDOUT_IS_PS7_UART](#)
- #define [UART_DEVICE_ID](#) 0

5.7.1 Macro Definition Documentation

5.7.1.1 #define STDOUT_IS_PS7_UART

Definition at line 4 of file platform_config.h.

5.7.1.2 #define UART_DEVICE_ID 0

Definition at line 5 of file platform_config.h.

5.8 src/utilities.c File Reference

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

Macros

- #define `OLED_VBAT` 0x20
- #define `OLED_VDD` 0x10;
- #define `OLED_RESET_B` 0x08
- #define `OLED_DATA_COMMAND_B` 0x04
- #define `OLED_SDIN` 0x02
- #define `OLED_SCLK` 0x01

Functions

- void `sendUartByte` (u32 unUartAddress, u8 uchByte)
Send a byte to the UART. Either directs to the full UART in the Zynq PS, or a Xilinx UartLite.
- u8 `checkUartEmpty` (u32 unUartAddress)
Check if either Uart is empty. Either directs to the full UART in the Zynq PS, or a Xilinx UartLite.
- u8 `getUartByte` (u32 nUartAddress)
Get a byte from either the full UART in the Zynq PS, or a Xilinx UartLite.
- 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, u32 unSpiSS)
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_set_PMOD_port` (int nPortNumber, u8 uchPortType)
Configure driving peripherals on each Pmod port.
- void `max_configure_PMOD_port` (u8 uchPmodPortA, u8 uchPmodPortB, u8 uchPmodPortC, u8 uchPmodPortD)
Configure driving peripherals on each Pmod port.
- 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 `sendOLEDSPI` (u8 uchDataToWrite)
Bit bang routine for Zedboard SPI OLED interface.
- void `initializeOLED` (u8 *pFont)
Initializes the OLED display.
- void `clearOLEDBuffer` (u8 *pauchBuffer)

Clears the OLED display buffer.

- void [displayOLEDBuffer](#) (u8 *pauchBuffer)

Copies the display buffer into the OLED display.

- void [putCharOLED](#) (int x, int y, char chCharacter)

Places a single ASCII character into the OLED display buffer.

- void [printfToBufferOLED](#) (int x, int y, char *chString)

Printf-like function to copy an ASCII string into the OLED display buffer.

- void [printfToOLED](#) (int x, int y, char *chString)

Printf-like function to copy an ASCII string into the OLED display buffer.

- void [flipAndCopyDisplayBuffer](#) (u8 *pauchSourceBuffer, u8 *pauchDestinationBuffer)

Copies the pixels from the source display buffer, into the destination buffer, flipping their location 180 degrees.

5.8.1 Detailed Description

```
Project: Carmel (MAXREFDES18)
Filename: utilities.c
Description: This module contains the utility functions for the
             ZedBoard implementation of the example program
             for the MAXREFDES18.
```

Revision History:

08-19-2013 Rev 01.00 MG 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 [utilities.c](#).

5.8.2 Macro Definition Documentation

5.8.2.1 #define OLED_DATA_COMMAND_B 0x04

Definition at line 66 of file utilities.c.

5.8.2.2 #define OLED_RESET_B 0x08

Definition at line 65 of file utilities.c.

5.8.2.3 #define OLED_SCLK 0x01

Definition at line 68 of file *utilities.c*.

5.8.2.4 #define OLED_SDIN 0x02

Definition at line 67 of file *utilities.c*.

5.8.2.5 #define OLED_VBAT 0x20

Definition at line 63 of file *utilities.c*.

5.8.2.6 #define OLED_VDD 0x10;

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

5.8.3 Function Documentation**5.8.3.1 u8 checkUartEmpty (u32 *unUartAddress*)**

Check if either Uart is empty. Either directs to 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>True</i>	if empty (false if not)
-------------	-------------------------

Definition at line 89 of file *utilities.c*.

5.8.3.2 void clearOLEDBuffer (u8 * *pauchBuffer*)

Clears the OLED display buffer.

Details

This function clears the display buffer by setting all pixels to zero.

Parameters

<i>in</i>	<i>*pauchBuffer</i>	- pointer to an u8 array used for the OLED pixel buffer
-----------	---------------------	---

Return values

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

Definition at line 633 of file *utilities.c*.

5.8.3.3 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

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

Return values

<i>None</i>

Definition at line 245 of file utilities.c.

5.8.3.4 void displayOLEDBuffer (u8 * *pauchBuffer*)

Copies the display buffer into the OLED display.

Details

This function writes the the display buffer into the OLED display using the page mode setting.
Note: The Zedboard OLED display contains a Solomon SSD1306 display controller.

Parameters

in	* <i>pauchBuffer</i>	- pointer to an u8 array used for the OLED pixel buffer
----	----------------------	---

Return values

<i>None</i>

Definition at line 650 of file utilities.c.

5.8.3.5 void flipAndCopyDisplayBuffer (u8 * *pauchSourceBuffer*, u8 * *pauchDestinationBuffer*)

Copies the pixels from the source display buffer, into the destination buffer, flipping their location 180 degrees.

Details

The (0,0) pixel location on the OLED display is in the bottom right corner. This function flips the display so that the (0,0) is in the top left corner and (131,31) is in the bottom right.

Parameters

in	* <i>pauchSourceBuffer</i>	- pointer to an u8 buffer used as the source display buffer
in	* <i>pauchDestinationBuffer</i>	- pointer to an u8 buffer used as the destination display buffer

Return values

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

Definition at line 804 of file utilities.c.

5.8.3.6 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 445 of file utilities.c.

5.8.3.7 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 123 of file utilities.c.

5.8.3.8 void initializeOLED (u8 * *pFont*)

Initializes the OLED display.

Details

This function initializes the OLED display and the global data structure that holds the OLED display buffer and font.

Parameters

in	<i>*pFont</i>	- pointer to an u8 array containing the ASCII display font
----	---------------	--

Return values

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

Definition at line 504 of file utilities.c.

5.8.3.9 void led_knight_rider (XGpio * pLED_GPIO, int nNumberOfTimes)

Blink a row of LEDs nNumberOfTimes times.

Details

The Digilent NEXYS-3 and Zedboard have 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	*pLED_GPIO	- address of the GPIO peripheral driving the LEDs in the memory map
----	------------	---

Return values

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

Definition at line 266 of file utilities.c.

5.8.3.10 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 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	uchPmodPortA	- 2-bit number to define configuration for Pmod port A
in	uchPmodPortB	- 2-bit number to define configuration for Pmod port B
in	uchPmodPortC	- 2-bit number to define configuration for Pmod port C
in	uchPmodPortD	- 2-bit number to define configuration for Pmod port D

Return values

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

Definition at line 325 of file utilities.c.

5.8.3.11 void max_set_PMOD_port (int *nPortNumber*, u8 *uchPortType*)

Configure driving peripherals on each Pmod port.

Details

The Maxim HDL hardware design includes a multiplexer on each PMOD port to allow I2C, SPI, GPIO, and UART functionality to be selected on each port.

This function should be called any time a new PMOD type is plugged into a PMOD. Valid settings for uchPortType follow:

PMOD_PORT_TYPE_UART, PMOD_PORT_TYPE_SPI, PMOD_PORT_TYPE_GPIO, PMOD_PORT_TYPE_I2C

Parameters

in	<i>nPortNumber</i>	- integer number of the port to set (0=A, 1=B, 2=C, 3=D)
in	<i>uchPortType</i>	- u8 number to set type of PMOD port

Definition at line 301 of file utilities.c.

5.8.3.12 int number_raised_to_power (int *nBase*, int *nExponent*)

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

Details

Many applications will not have math.h included due to limited memory space. This is a simple functions to implement ($nBase ^ nExponent$)

Parameters

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

Return values

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

Definition at line 361 of file utilities.c.

5.8.3.13 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 143 of file utilities.c.

5.8.3.14 void printfToBufferOLED (int x, int y, char * chString)

Printf-like function to copy an ASCII string into the OLED display buffer.

Details

This function copies a pre-formatted ASCII string into the OLED display buffer, but does not take any action to display the buffer

Parameters

in	x	- The starting position of string within the 16x4 character display
in	y	- The starting position of string within the 16x4 character display
in	*chString	- pointer to an null terminated character string. Should be generated with sprintf (or similar).

Return values

None

Definition at line 745 of file utilities.c.

5.8.3.15 void printfToOLED (int x, int y, char * chString)

Printf-like function to copy an ASCII string into the OLED display buffer.

Details

This function copies a pre-formatted ASCII string into the OLED display buffer, flips/rotates the display pixels, and then sends the buffer to the OLED display

Parameters

in	x	- The starting position of string within the 16x4 character display
in	y	- The starting position of string within the 16x4 character display
in	*chString	- pointer to an null terminated character string. Should be generated with sprintf (or similar).

Return values

None

Definition at line 785 of file utilities.c.

5.8.3.16 void putCharOLED (int x, int y, char chCharacter)

Places a single ASCII character into the OLED display buffer.

Details

This function copies an ASCII character from the font table into the OLED display buffer (configured as a 16x4 character display).

Parameters

in	<i>x</i>	- The x position of the character location
in	<i>y</i>	- The y position of the character location
in	<i>chCharacter</i>	- ASCII character to be printed (ex: 0x41 = 'A')

Return values

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

Definition at line 713 of file utilities.c.

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

Return values

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

Definition at line 389 of file utilities.c.

5.8.3.18 void sendOLEDSPI (u8 uchDataToWrite)

Bit bang routine for Zedboard SPI OLED interface.

Details

This function sends an (unsigned) byte (msb first) to the OLED SPI port

Parameters

in	<i>uchDataToWrite</i>	- u8 value to be written to OLED SPI
----	-----------------------	--------------------------------------

Return values

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

Definition at line 472 of file utilities.c.

5.8.3.19 void sendUartByte (u32 unUartAddress, u8 uchByte)

Send a byte to the UART. Either directs to the full UART in the Zynq PS, or a Xilinx UartLite.

Parameters

in	<i>unUartAddress.</i>	32 bit UART address
in	<i>uchByte.</i>	The byte to be sent to the UART

Return values

<i>None</i>

Definition at line 70 of file utilities.c.

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

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>unPeripheral-AddressSPI</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
in	<i>unSpiSS</i>	- one-hot 32-bit slave select register value. 0x1 = MAX15500, 0x2= MAX5316.

Return values

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

Definition at line 157 of file utilities.c.

5.9 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 "MAXREFDES18.h"
```

Functions

- void [sendUartByte](#) (u32 unUartAddress, u8 uchByte)
Send a byte to the UART. Either directs to the full UART in the Zynq PS, or a Xilinx UartLite.
- u8 [checkUartEmpty](#) (u32 unUartAddress)
Check if either Uart is empty. Either directs to the full UART in the Zynq PS, or a Xilinx UartLite.
- u8 [getUartByte](#) (u32 nUartAddress)
Get a byte from either the full UART in the Zynq PS, or a Xilinx UartLite.
- 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, u32 unSpiSS)
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_set_PMOD_port](#) (int nPortNumber, u8 uchPortType)
Configure driving peripherals on each Pmod port.
- void [max_configure_PMOD_port](#) (u8 uchPmodPortA, u8 uchPmodPortB, u8 uchPmodPortC, u8 uchPmodPortD)
Configure driving peripherals on each Pmod port.
- 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 [sendOLEDSPI](#) (u8 uchDataToWrite)
Bit bang routine for Zedboard SPI OLED interface.
- void [initializeOLED](#) (u8 *pFont)
Initializes the OLED display.
- void [clearOLEDBuffer](#) (u8 *pauchBuffer)
Clears the OLED display buffer.
- void [displayOLEDBuffer](#) (u8 *pauchBuffer)

Copies the display buffer into the OLED display.

- void [putCharOLED](#) (int x, int y, char chCharacter)

Places a single ASCII character into the OLED display buffer.

- void [printfToBufferOLED](#) (int x, int y, char *chString)

Printf-like function to copy an ASCII string into the OLED display buffer.

- void [printfToOLED](#) (int x, int y, char *chString)

Printf-like function to copy an ASCII string into the OLED display buffer.

- void [flipAndCopyDisplayBuffer](#) (u8 *pauchSourceBuffer, u8 *pauchDestinationBuffer)

Copies the pixels from the source display buffer, into the destination buffer, flipping their location 180 degrees.

5.9.1 Detailed Description

```
Project: Carmel (MAXREFDES18)
Filename: utilities.h
Description: This module contains the utility functions for the
             ZedBoard implementation of the example program
             for the MAXREFDES18.
```

Revision History:

08-19-2013 Rev 01.00 MG 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 [utilities.h](#).

5.9.2 Function Documentation

5.9.2.1 u8 checkUartEmpty (u32 unUartAddress)

Check if either Uart is empty. Either directs to the full UART in the Zynq PS, or a Xilinx UartLite.

Parameters

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

Return values

<i>True</i>	if empty (false if not)
-------------	-------------------------

Definition at line 89 of file utilities.c.

5.9.2.2 void clearOLEDBuffer (u8 * *pauchBuffer*)

Clears the OLED display buffer.

Details

This function clears the display buffer by setting all pixels to zero.

Parameters

<i>in</i>	<i>*pauchBuffer</i>	- pointer to an u8 array used for the OLED pixel buffer
-----------	---------------------	---

Return values

<i>None</i>

Definition at line 633 of file utilities.c.

5.9.2.3 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 245 of file utilities.c.

5.9.2.4 void displayOLEDBuffer (u8 * *pauchBuffer*)

Copies the display buffer into the OLED display.

Details

This function writes the the display buffer into the OLED display using the page mode setting.
Note: The Zedboard OLED display contains a Solomon SSD1306 display controller.

Parameters

in	<i>*pauchBuffer</i>	- pointer to an u8 array used for the OLED pixel buffer
----	---------------------	---

Return values

<i>None</i>

Definition at line 650 of file utilities.c.

5.9.2.5 void flipAndCopyDisplayBuffer (u8 * *pauchSourceBuffer*, u8 * *pauchDestinationBuffer*)

Copies the pixels from the source display buffer, into the destination buffer, flipping their location 180 degrees.

Details

The (0,0) pixel location on the OLED display is in the bottom right corner. This function flips the display so that the (0,0) is in the top left corner and (131,31) is in the bottom right.

Parameters

in	<i>*pauchSource-Buffer</i>	- pointer to an u8 buffer used as the source display buffer
in	<i>*pauch-DestinationBuffer</i>	- pointer to an u8 buffer used as the destination display buffer

Return values

<i>None</i>

Definition at line 804 of file utilities.c.

5.9.2.6 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 445 of file utilities.c.

5.9.2.7 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 123 of file utilities.c.

5.9.2.8 void initializeOLED (u8 * pFont)

Initializes the OLED display.

Details

This function initializes the OLED display and the global data structure that holds the OLED display buffer and font.

Parameters

in	<i>*pFont</i>	- pointer to an u8 array containing the ASCII display font
----	---------------	--

Return values

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

Definition at line 504 of file utilities.c.

5.9.2.9 void led_knight_rider (XGpio * pLED_GPIO, int nNumberOfTimes)

Blink a row of LEDs nNumberOfTimes times.

Details

The Digilent NEXYS-3 and Zedboard have 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 the memory map
----	-------------------	---

Return values

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

Definition at line 266 of file utilities.c.

5.9.2.10 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 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 325 of file utilities.c.

5.9.2.11 void max_set_PMOD_port (int nPortNumber, u8 uchPortType)

Configure driving peripherals on each Pmod port.

Details

The Maxim HDL hardware design includes a multiplexer on each PMOD port to allow I2C, SPI, GPIO, and UART functionality to be selected on each port.

This function should be called any time a new PMOD type is plugged into a PMOD. Valid settings for uchPortType follow:

PMOD_PORT_TYPE_UART, PMOD_PORT_TYPE_SPI, PMOD_PORT_TYPE_GPIO, PMOD_PORT_TYPE_I2C

Parameters

in	<i>nPortNumber</i>	- integer number of the port to set (0=A, 1=B, 2=C, 3=D)
in	<i>uchPortType</i>	- u8 number to set type of PMOD port

Definition at line 301 of file utilities.c.

5.9.2.12 int number_raised_to_power (int nBase, int nExponent)

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

Details

Many applications will not have math.h included due to limited memory space. This is a simple functions to implement ($nBase^{nExponent}$)

Parameters

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

Return values

$Base^{Exponent}$

Definition at line 361 of file utilities.c.

5.9.2.13 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 143 of file utilities.c.

5.9.2.14 void printfToBufferOLED (int *x*, int *y*, char * *chString*)

Printf-like function to copy an ASCII string into the OLED display buffer.

Details

This function copies a pre-formatted ASCII string into the OLED display buffer, but does not take any action to display the buffer

Parameters

in	<i>x</i>	- The starting position of string within the 16x4 character display
in	<i>y</i>	- The starting position of string within the 16x4 character display
in	<i>*chString</i>	- pointer to an null terminated character string. Should be generated with sprintf (or similar).

Return values

<i>None</i>

Definition at line 745 of file utilities.c.

5.9.2.15 void printfToOLED (int *x*, int *y*, char * *chString*)

Printf-like function to copy an ASCII string into the OLED display buffer.

Details

This function copies a pre-formatted ASCII string into the OLED display buffer, flips/rotates the display pixels, and then sends the buffer to the OLED display

Parameters

in	<i>x</i>	- The starting position of string within the 16x4 character display
in	<i>y</i>	- The starting position of string within the 16x4 character display
in	<i>*chString</i>	- pointer to an null terminated character string. Should be generated with sprintf (or similar).

Return values

<i>None</i>

Definition at line 785 of file utilities.c.

5.9.2.16 void putCharOLED (int x, int y, char chCharacter)

Places a single ASCII character into the OLED display buffer.

Details

This function copies an ASCII character from the font table into the OLED display buffer (configured as a 16x4 character display).

Parameters

in	<i>x</i>	- The x position of the character location
in	<i>y</i>	- The y position of the character location
in	<i>chCharacter</i>	- ASCII character to be printed (ex: 0x41 = 'A')

Return values

<i>None</i>

Definition at line 713 of file utilities.c.

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

Return values

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

Definition at line 389 of file utilities.c.

5.9.2.18 void sendOLEDSPI (u8 uchDataToWrite)

Bit bang routine for Zedboard SPI OLED interface.

Details

This function sends an (unsigned) byte (msb first) to the OLED SPI port

Parameters

<i>in</i>	<i>uchDataToWrite</i>	- u8 value to be written to OLED SPI
-----------	-----------------------	--------------------------------------

Return values

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

Definition at line 472 of file utilities.c.

5.9.2.19 void sendUartByte (u32 unUartAddress, u8 uchByte)

Send a byte to the UART. Either directs to the full UART in the Zynq PS, or a Xilinx UartLite.

Parameters

<i>in</i>	<i>unUartAddress.</i>	32 bit UART address
<i>in</i>	<i>uchByte.</i>	The byte to be sent to the UART

Return values

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

Definition at line 70 of file utilities.c.

5.9.2.20 int SpiRW (u32 unPeripheralAddressSPI, unsigned int unCPHA, unsigned int unCPOL, u8 * uchWriteBuf, u8 * uchReadBuf, int unNumBytes, u8 uchCsActiveHigh, u32 unSpiSS)

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 uchWriteBuf will be clocked out (MSB first) onto the MOSI pin Data from the MISO pin will be placed into the uchReadBuf uchCsActiveHigh==TRUE allows SS configurations to be used uchCsActiveHigh==FALSE allows SS# configurations to be used

Parameters

in	<i>unPeripheral-AddressSPI</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
in	<i>unSpiSS</i>	- one-hot 32-bit slave select register value. 0x1 = MAX15500, 0x2= MAX5316.

Return values

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

Definition at line 157 of file utilities.c.

Index

BASE_FUNCTION_STATE
menu.h, 24

CURRENT_RANGE_MENU
menu.h, 25

checkUartEmpty
utilities.c, 33
utilities.h, 42

cleanup_platform
platform.c, 29
platform.h, 30

clearOLEDBuffer
utilities.c, 33
utilities.h, 43

delay
utilities.c, 33
utilities.h, 43

disable_caches
platform.c, 29

displayOLEDBuffer
utilities.c, 34
utilities.h, 43

enable_caches
platform.c, 29

flipAndCopyDisplayBuffer
utilities.c, 34
utilities.h, 44

flippedBuffer
maximOLEDDisplay, 7

font
maximOLEDDisplay, 7
MAXREFDES18.c, 11

g_PmodPortMuxSettings
MAXREFDES18.c, 11

g_auchPortType
MAXREFDES18.c, 11
MAXREFDES18.h, 17

g_nActivePmodPort
MAXREFDES18.c, 11
MAXREFDES18.h, 17

g_pActiveGPIOPort
MAXREFDES18.c, 11

MAXREFDES18.h, 17

g_sInputString
MAXREFDES18.c, 11
MAXREFDES18.h, 17

g_structureOLED
MAXREFDES18.c, 11
MAXREFDES18.h, 17

g_tempString
MAXREFDES18.c, 11
MAXREFDES18.h, 17

g_unActivePeripheralAddressI2C
MAXREFDES18.h, 17

g_unActivePeripheralAddressSPI
MAXREFDES18.c, 11
MAXREFDES18.h, 17

g_unActivePeripheralAddressUart
MAXREFDES18.h, 17

g_unSevenSegmentValue
MAXREFDES18.c, 12

g_xGpioLed
MAXREFDES18.c, 12

g_xGpioPmodPortA
MAXREFDES18.c, 12
MAXREFDES18.h, 17

g_xGpioPmodPortB
MAXREFDES18.c, 12
MAXREFDES18.h, 17

g_xGpioPmodPortC
MAXREFDES18.c, 12
MAXREFDES18.h, 17

g_xGpioPmodPortD
MAXREFDES18.c, 12
MAXREFDES18.h, 17

g_xGpioPmodPortMuxIO
MAXREFDES18.c, 12
MAXREFDES18.h, 18

g_xGpioSevenSegment
MAXREFDES18.c, 12

GetLine
utilities.c, 35
utilities.h, 44

getUartByte
utilities.c, 35
utilities.h, 44

init_platform

- platform.c, 29
- platform.h, 30
- init_uart
 - platform.c, 30
- initializeOLED
 - utilities.c, 35
 - utilities.h, 45
- KEYPRESS_ARROW_DOWN
 - menu.h, 25
- KEYPRESS_ARROW_LEFT
 - menu.h, 25
- KEYPRESS_ARROW_RIGHT
 - menu.h, 25
- KEYPRESS_ARROW_UP
 - menu.h, 25
- KEYPRESS_END
 - menu.h, 25
- led_knight_rider
 - utilities.c, 36
 - utilities.h, 45
- MAJOR_REVISION
 - MAXREFDES18.c, 10
- MAXREFDES18.c
 - font, 11
 - g_PmodPortMuxSettings, 11
 - g_auchPortType, 11
 - g_nActivePModPort, 11
 - g_pActiveGPIOPort, 11
 - g_sInputString, 11
 - g_structureOLED, 11
 - g_tempString, 11
 - g_unActivePeripheralAddressSPI, 11
 - g_unSevenSegmentValue, 12
 - g_xGpioLed, 12
 - g_xGpioPmodPortA, 12
 - g_xGpioPmodPortB, 12
 - g_xGpioPmodPortC, 12
 - g_xGpioPmodPortD, 12
 - g_xGpioPmodPortMuxIO, 12
 - g_xGpioSevenSegment, 12
 - main, 10
- MAXREFDES18.h
 - g_auchPortType, 17
 - g_nActivePModPort, 17
 - g_pActiveGPIOPort, 17
 - g_sInputString, 17
 - g_structureOLED, 17
 - g_tempString, 17
 - g_unActivePeripheralAddressI2C, 17
 - g_unActivePeripheralAddressSPI, 17
 - g_unActivePeripheralAddressUart, 17
 - g_xGpioPmodPortA, 17
 - g_xGpioPmodPortB, 17
 - g_xGpioPmodPortC, 17
 - g_xGpioPmodPortD, 17
 - g_xGpioPmodPortMuxIO, 18
- MINOR_REVISION
 - MAXREFDES18.c, 10
- MODE_MENU
 - menu.h, 25
- main
 - MAXREFDES18.c, 10
- max_configure_PMOD_port
 - utilities.c, 36
 - utilities.h, 45
- max_set_PMOD_port
 - utilities.c, 36
 - utilities.h, 46
- maximOLEDDisplay, 7
 - flippedBuffer, 7
 - font, 7
 - portStatus, 7
 - writeBuffer, 7
 - xgpioPort, 8
- menu.c
 - menu_cls, 19
 - menu_get_direct_entry, 19
 - menu_print_current_range_menu, 20
 - menu_print_line, 20
 - menu_print_maxim_banner, 20
 - menu_print_maxim_banner_big, 21
 - menu_print_mode_menu, 21
 - menu_print_prompt, 21
 - menu_print_voltage_range_menu, 21
 - menu_retrieve_keypress, 22
- menu.h
 - BASE_FUNCTION_STATE, 24
 - CURRENT_RANGE_MENU, 25
 - KEYPRESS_ARROW_DOWN, 25
 - KEYPRESS_ARROW_LEFT, 25
 - KEYPRESS_ARROW_UP, 25
 - KEYPRESS_END, 25
 - MODE_MENU, 25
 - menu_cls, 26
 - menu_get_direct_entry, 26
 - menu_print_current_range_menu, 27
 - menu_print_maxim_banner, 27
 - menu_print_maxim_banner_big, 27
 - menu_print_mode_menu, 27
 - menu_print_prompt, 28
 - menu_print_voltage_range_menu, 28
 - menu_retrieve_keypress, 28
 - VOLTAGE_RANGE_MENU, 26
 - WAIT_KEYPRESS, 26
- menu_cls
 - menu.c, 19

- menu.h, 26
- menu_get_direct_entry
 - menu.c, 19
 - menu.h, 26
- menu_print_current_range_menu
 - menu.c, 20
 - menu.h, 27
- menu_print_line
 - menu.c, 20
- menu_print_maxim_banner
 - menu.c, 20
 - menu.h, 27
- menu_print_maxim_banner_big
 - menu.c, 21
 - menu.h, 27
- menu_print_mode_menu
 - menu.c, 21
 - menu.h, 27
- menu_print_prompt
 - menu.c, 21
 - menu.h, 28
- menu_print_voltage_range_menu
 - menu.c, 21
 - menu.h, 28
- menu_retrieve_keypress
 - menu.c, 22
 - menu.h, 28
- number_raised_to_power
 - utilities.c, 37
 - utilities.h, 46
- OLED_RESET_B
 - utilities.c, 32
- OLED_SCLK
 - utilities.c, 32
- OLED_SDIN
 - utilities.c, 33
- OLED_VBAT
 - utilities.c, 33
- OLED_VDD
 - utilities.c, 33
- PMOD_TYPE_GPIO
 - MAXREFDES18.h, 16
- PMOD_TYPE_I2C
 - MAXREFDES18.h, 16
- PMOD_TYPE_SPI
 - MAXREFDES18.h, 16
- PMOD_TYPE_UART
 - MAXREFDES18.h, 16
- platform.c
 - cleanup_platform, 29
 - disable_caches, 29
 - enable_caches, 29
 - init_platform, 29
 - init_uart, 30
 - UART_BAUD, 29
- platform.h
 - cleanup_platform, 30
 - init_platform, 30
- platform_config.h
 - UART_DEVICE_ID, 30
- portStatus
 - maximOLEDDisplay, 7
- print_asterisks
 - utilities.c, 37
 - utilities.h, 47
- printfToBufferOLED
 - utilities.c, 37
 - utilities.h, 47
- printfToOLED
 - utilities.c, 38
 - utilities.h, 47
- putCharOLED
 - utilities.c, 38
 - utilities.h, 48
- receive_byte_with_timeout
 - utilities.c, 39
 - utilities.h, 48
- STDOUT_IS_PS7_UART
 - platform_config.h, 30
- sendOLEDSPI
 - utilities.c, 39
 - utilities.h, 49
- sendUartByte
 - utilities.c, 39
 - utilities.h, 49
- SpiRW
 - utilities.c, 40
 - utilities.h, 49
- src/MAXREFDES18.c, 9
- src/MAXREFDES18.h, 13
- src/menu.c, 18
- src/menu.h, 22
- src/platform.c, 29
- src/platform.h, 30
- src/platform_config.h, 30
- src/utilities.c, 31
- src/utilities.h, 41
- UART_BAUD
 - platform.c, 29
- UART_DEVICE_ID
 - platform_config.h, 30
- utilities.c
 - checkUartEmpty, 33
 - clearOLEDBuffer, 33

- delay, [33](#)
- displayOLEDBuffer, [34](#)
- flipAndCopyDisplayBuffer, [34](#)
- GetLine, [35](#)
- getUartByte, [35](#)
- initializeOLED, [35](#)
- led_knight_rider, [36](#)
- max_configure_PMOD_port, [36](#)
- max_set_PMOD_port, [36](#)
- number_raised_to_power, [37](#)
- OLED_RESET_B, [32](#)
- OLED_SCLK, [32](#)
- OLED_SDIN, [33](#)
- OLED_VBAT, [33](#)
- OLED_VDD, [33](#)
- print_asterisks, [37](#)
- printfToBufferOLED, [37](#)
- printfToOLED, [38](#)
- putCharOLED, [38](#)
- receive_byte_with_timeout, [39](#)
- sendOLEDSPI, [39](#)
- sendUartByte, [39](#)
- SpiRW, [40](#)
- utilities.h
 - checkUartEmpty, [42](#)
 - clearOLEDBuffer, [43](#)
 - delay, [43](#)
 - displayOLEDBuffer, [43](#)
 - flipAndCopyDisplayBuffer, [44](#)
 - GetLine, [44](#)
 - getUartByte, [44](#)
 - initializeOLED, [45](#)
 - led_knight_rider, [45](#)
 - max_configure_PMOD_port, [45](#)
 - max_set_PMOD_port, [46](#)
 - number_raised_to_power, [46](#)
 - print_asterisks, [47](#)
 - printfToBufferOLED, [47](#)
 - printfToOLED, [47](#)
 - putCharOLED, [48](#)
 - receive_byte_with_timeout, [48](#)
 - sendOLEDSPI, [49](#)
 - sendUartByte, [49](#)
 - SpiRW, [49](#)
- VOLTAGE_RANGE_MENU
 - menu.h, [26](#)
- WAIT_KEYPRESS
 - menu.h, [26](#)
- writeBuffer
 - maximOLEDDisplay, [7](#)
- xgpioPort
 - maximOLEDDisplay, [8](#)