

# Carmel (MAXREFDES18#) Code Documentation

V01.00

Generated by Doxygen 1.8.2

Wed Aug 28 2013 11:28:45



# Contents

<b>1</b>	<b>Main Page</b>	<b>1</b>
1.1	Introduction . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>File Documentation</b>	<b>5</b>
3.1	src/MAXREFDES18.c File Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Macro Definition Documentation . . . . .	6
3.1.2.1	MAJOR_REVISION . . . . .	6
3.1.2.2	MINOR_REVISION . . . . .	6
3.1.3	Function Documentation . . . . .	6
3.1.3.1	main . . . . .	6
3.1.4	Variable Documentation . . . . .	7
3.1.4.1	g_gpButtons . . . . .	7
3.1.4.2	g_PmodPortMuxSettings . . . . .	7
3.1.4.3	g_sInputString . . . . .	7
3.1.4.4	g_unActivePeripheralAddressSPI . . . . .	7
3.1.4.5	g_xGpioLed . . . . .	7
3.1.4.6	g_xGpioPmodPortA . . . . .	7
3.1.4.7	g_xGpioPmodPortB . . . . .	7
3.2	src/MAXREFDES18.h File Reference . . . . .	8
3.2.1	Detailed Description . . . . .	9
3.2.2	Macro Definition Documentation . . . . .	9
3.2.2.1	ABOUT_ONE_SECOND . . . . .	9
3.2.2.2	DEFAULT_HYPERTERMINAL_UART_ADDRESS . . . . .	9
3.2.2.3	DEFAULT_HYPERTERMINAL_UART_ID . . . . .	9
3.2.2.4	INPUT_STRING_MAX_SIZE . . . . .	9

3.2.2.5	PMOD_PORT_TYPE_GPIO	10
3.2.2.6	PMOD_PORT_TYPE_I2C	10
3.2.2.7	PMOD_PORT_TYPE_SPI	10
3.2.2.8	PMOD_PORT_TYPE_UART	10
3.2.2.9	PMOD_TYPE_GPIO	10
3.2.2.10	PMOD_TYPE_I2C	10
3.2.2.11	PMOD_TYPE_SPI	10
3.2.2.12	PMOD_TYPE_UART	10
3.2.3	Variable Documentation	11
3.2.3.1	g_slInputString	11
3.2.3.2	g_xGpioPmodPortA	11
3.2.3.3	g_xGpioPmodPortB	11
3.3	src/menu.c File Reference	11
3.3.1	Detailed Description	12
3.3.2	Function Documentation	12
3.3.2.1	menu_cls	12
3.3.2.2	menu_get_direct_entry	13
3.3.2.3	menu_print_current_range_menu	13
3.3.2.4	menu_print_line	13
3.3.2.5	menu_print_maxim_banner	14
3.3.2.6	menu_print_maxim_banner_big	14
3.3.2.7	menu_print_mode_menu	14
3.3.2.8	menu_print_prompt	14
3.3.2.9	menu_print_voltage_range_menu	15
3.3.2.10	menu_retrieve_keypress	15
3.4	src/menu.h File Reference	15
3.4.1	Detailed Description	17
3.4.2	Macro Definition Documentation	17
3.4.2.1	CURRENT_RANGE_0_20_MENU	17
3.4.2.2	CURRENT_RANGE_20_20_MENU	17
3.4.2.3	CURRENT_RANGE_4_20_MENU	18
3.4.2.4	CURRENT_RANGE_MENU	18
3.4.2.5	KEYPRESS_ARROW_DOWN	18
3.4.2.6	KEYPRESS_ARROW_LEFT	18
3.4.2.7	KEYPRESS_ARROW_RIGHT	18
3.4.2.8	KEYPRESS_ARROW_UP	18
3.4.2.9	KEYPRESS_END	18

3.4.2.10	MODE_MENU	18
3.4.2.11	VOLTAGE_RANGE_0_10_MENU	18
3.4.2.12	VOLTAGE_RANGE_0_5_MENU	19
3.4.2.13	VOLTAGE_RANGE_10_10_MENU	19
3.4.2.14	VOLTAGE_RANGE_MENU	19
3.4.2.15	WAIT_KEYPRESS	19
3.4.3	Function Documentation	19
3.4.3.1	menu_cls	19
3.4.3.2	menu_get_direct_entry	19
3.4.3.3	menu_print_current_range_menu	20
3.4.3.4	menu_print_maxim_banner	20
3.4.3.5	menu_print_maxim_banner_big	20
3.4.3.6	menu_print_mode_menu	21
3.4.3.7	menu_print_prompt	21
3.4.3.8	menu_print_voltage_range_menu	21
3.4.3.9	menu_retrieve_keypress	21
3.5	src/platform.c File Reference	22
3.5.1	Macro Definition Documentation	22
3.5.1.1	UART_BAUD	22
3.5.2	Function Documentation	22
3.5.2.1	cleanup_platform	22
3.5.2.2	disable_caches	22
3.5.2.3	enable_caches	22
3.5.2.4	init_platform	23
3.5.2.5	init_uart	23
3.6	src/platform.h File Reference	23
3.6.1	Function Documentation	23
3.6.1.1	cleanup_platform	23
3.6.1.2	init_platform	23
3.7	src/platform_config.h File Reference	23
3.8	src/utilities.c File Reference	23
3.8.1	Detailed Description	24
3.8.2	Function Documentation	24
3.8.2.1	delay	24
3.8.2.2	GetLine	25
3.8.2.3	led_knight_rider	25
3.8.2.4	max_configure_PMOD_port	26

3.8.2.5	number_raised_to_power . . . . .	26
3.8.2.6	print_asterisks . . . . .	26
3.8.2.7	receive_byte_with_timeout . . . . .	27
3.8.2.8	SpiRW . . . . .	27
3.9	src/utilities.h File Reference . . . . .	28
3.9.1	Detailed Description . . . . .	28
3.9.2	Function Documentation . . . . .	29
3.9.2.1	delay . . . . .	29
3.9.2.2	GetLine . . . . .	29
3.9.2.3	led_knight_rider . . . . .	30
3.9.2.4	max_configure_PMOD_port . . . . .	30
3.9.2.5	number_raised_to_power . . . . .	31
3.9.2.6	print_asterisks . . . . .	31
3.9.2.7	receive_byte_with_timeout . . . . .	31
3.9.2.8	SpiRW . . . . .	32

# 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

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

src/ <a href="#">MAXREFDES18.c</a> . . . . .	5
src/ <a href="#">MAXREFDES18.h</a> . . . . .	8
src/ <a href="#">menu.c</a> . . . . .	11
src/ <a href="#">menu.h</a> . . . . .	15
src/ <a href="#">platform.c</a> . . . . .	22
src/ <a href="#">platform.h</a> . . . . .	23
src/ <a href="#">platform_config.h</a> . . . . .	23
src/ <a href="#">utilities.c</a> . . . . .	23
src/ <a href="#">utilities.h</a> . . . . .	28



## Chapter 3

# File Documentation

### 3.1 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

- char g\_sInputString [INPUT\_STRING\_MAX\_SIZE]
- XGpio g\_xGpioPmodPortA
- XGpio g\_xGpioPmodPortB
- XGpio g\_gpButtons
- XGpio g\_xGpioLed
- u8 g\_PmodPortMuxSettings =0
- u32 g\_unActivePeripheralAddressSPI =XPAR\_SPI\_0\_BASEADDR

#### 3.1.1 Detailed Description

Project: Carmel (MAXREFDES18#)  
Filename: MAXREFDES18.c  
Description: This module contains the Main application for the

Avnet LX9 Board 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](#).

### 3.1.2 Macro Definition Documentation

#### 3.1.2.1 #define MAJOR\_REVISION 1

Definition at line 82 of file MAXREFDES18.c.

#### 3.1.2.2 #define MINOR\_REVISION 0

Definition at line 83 of file MAXREFDES18.c.

### 3.1.3 Function Documentation

#### 3.1.3.1 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

None	
------	--

## Return values

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

Definition at line 96 of file MAXREFDES18.c.

### 3.1.4 Variable Documentation

#### 3.1.4.1 XGpio g\_gpButtons

Definition at line 90 of file MAXREFDES18.c.

#### 3.1.4.2 u8 g\_PmodPortMuxSettings =0

Definition at line 92 of file MAXREFDES18.c.

#### 3.1.4.3 char g\_sInputString[INPUT\_STRING\_MAX\_SIZE]

Definition at line 86 of file MAXREFDES18.c.

#### 3.1.4.4 u32 g\_unActivePeripheralAddressSPI =XPAR\_SPI\_0\_BASEADDR

Definition at line 94 of file MAXREFDES18.c.

#### 3.1.4.5 XGpio g\_xGpioLed

Definition at line 91 of file MAXREFDES18.c.

#### 3.1.4.6 XGpio g\_xGpioPmodPortA

Definition at line 87 of file MAXREFDES18.c.

#### 3.1.4.7 XGpio g\_xGpioPmodPortB

Definition at line 88 of file MAXREFDES18.c.

## 3.2 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"
```

### 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 [INPUT\\_STRING\\_MAX\\_SIZE](#) 16

### Variables

- XGpio [g\\_xGpioPmodPortA](#)
- XGpio [g\\_xGpioPmodPortB](#)
- char [g\\_sInputString](#) [INPUT\_STRING\_MAX\_SIZE]

### 3.2.1 Detailed Description

```
Project: Carmel (MAXREFDES18#)
Filename: MAXREFDES18.h
Description: This module contains the Main application for the
             Avnet LX9 Board 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](#).

### 3.2.2 Macro Definition Documentation

#### 3.2.2.1 **#define ABOUT\_ONE\_SECOND 3508380**

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

Definition at line 78 of file MAXREFDES18.h.

#### 3.2.2.2 **#define DEFAULT\_HYPERTERMINAL\_UART\_ADDRESS XPAR\_USB\_UART\_BASEADDR**

Definition at line 76 of file MAXREFDES18.h.

#### 3.2.2.3 **#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 MAXREFDES18.h.

#### 3.2.2.4 **#define INPUT\_STRING\_MAX\_SIZE 16**

Definition at line 92 of file MAXREFDES18.h.

**3.2.2.5 #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 MAXREFDES18.h.

**3.2.2.6 #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 MAXREFDES18.h.

**3.2.2.7 #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 MAXREFDES18.h.

**3.2.2.8 #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 MAXREFDES18.h.

**3.2.2.9 #define PMOD\_TYPE\_GPIO 2**

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

**3.2.2.10 #define PMOD\_TYPE\_I2C 3**

value which specifies I2C in maximProduct structure  
Definition at line 85 of file MAXREFDES18.h.

**3.2.2.11 #define PMOD\_TYPE\_SPI 1**

value which specifies SPI in maximProduct structure  
Definition at line 83 of file MAXREFDES18.h.

**3.2.2.12 #define PMOD\_TYPE\_UART 0**

value which specifies UART in maximProduct structure  
Definition at line 82 of file MAXREFDES18.h.



### 3.2.3 Variable Documentation

#### 3.2.3.1 char g\_sInputString[INPUT\_STRING\_MAX\_SIZE]

Definition at line 86 of file MAXREFDES18.c.

#### 3.2.3.2 XGpio g\_xGpioPmodPortA

Definition at line 87 of file MAXREFDES18.c.

#### 3.2.3.3 XGpio g\_xGpioPmodPortB

Definition at line 88 of file MAXREFDES18.c.

## 3.3 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.*

### 3.3.1 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](#).

### 3.3.2 Function Documentation

#### 3.3.2.1 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.

**3.3.2.2 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.

**3.3.2.3 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.

**3.3.2.4 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.

**3.3.2.5 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.

**3.3.2.6 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.

**3.3.2.7 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.

**3.3.2.8 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.

**3.3.2.9 void menu\_print\_voltage\_range\_menu ( )**

Print the voltage range menu.

## Parameters

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

## Return values

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

Definition at line 307 of file menu.c.

**3.3.2.10 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.

**3.4 src/menu.h File Reference**

```
#include "xbasic_types.h"
#include "xspi_1.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 `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.*

### 3.4.1 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](#).

### 3.4.2 Macro Definition Documentation

#### 3.4.2.1 `#define CURRENT_RANGE_0_20_MENU 5`

Menu state machine state.

Definition at line 74 of file menu.h.

#### 3.4.2.2 `#define CURRENT_RANGE_20_20_MENU 4`

Menu state machine state.

Definition at line 73 of file menu.h.

#### **3.4.2.3 #define CURRENT\_RANGE\_4\_20\_MENU 6**

Menu state machine state.

Definition at line 75 of file menu.h.

#### **3.4.2.4 #define CURRENT\_RANGE\_MENU 2**

Menu state machine state.

Definition at line 71 of file menu.h.

#### **3.4.2.5 #define KEYPRESS\_ARROW\_DOWN 241**

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

Definition at line 81 of file menu.h.

#### **3.4.2.6 #define KEYPRESS\_ARROW\_LEFT 242**

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

Definition at line 82 of file menu.h.

#### **3.4.2.7 #define KEYPRESS\_ARROW\_RIGHT 243**

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

Definition at line 83 of file menu.h.

#### **3.4.2.8 #define KEYPRESS\_ARROW\_UP 240**

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

Definition at line 80 of file menu.h.

#### **3.4.2.9 #define KEYPRESS\_END 244**

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

Definition at line 84 of file menu.h.

#### **3.4.2.10 #define MODE\_MENU 0**

Menu state machine state.

Definition at line 69 of file menu.h.

#### **3.4.2.11 #define VOLTAGE\_RANGE\_0\_10\_MENU 14**

Menu state machine state.

Definition at line 77 of file menu.h.



**3.4.2.12 #define VOLTAGE\_RANGE\_0\_5\_MENU 15**

Menu state machine state.

Definition at line 78 of file menu.h.

**3.4.2.13 #define VOLTAGE\_RANGE\_10\_10\_MENU 13**

Menu state machine state.

Definition at line 76 of file menu.h.

**3.4.2.14 #define VOLTAGE\_RANGE\_MENU 3**

Menu state machine state.

Definition at line 72 of file menu.h.

**3.4.2.15 #define WAIT\_KEYPRESS 1**

Menu state machine state.

Definition at line 70 of file menu.h.

**3.4.3 Function Documentation****3.4.3.1 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.

**3.4.3.2 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.

**3.4.3.3 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.

**3.4.3.4 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.

**3.4.3.5 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.

#### 3.4.3.6 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.

#### 3.4.3.7 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.

#### 3.4.3.8 void menu\_print\_voltage\_range\_menu ( )

Print the voltage range menu.

##### Parameters

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

##### Return values

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

Definition at line 307 of file menu.c.

#### 3.4.3.9 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.

## 3.5 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](#) ()

### 3.5.1 Macro Definition Documentation

#### 3.5.1.1 #define UART\_BAUD 9600

Definition at line 30 of file platform.c.

### 3.5.2 Function Documentation

#### 3.5.2.1 void cleanup\_platform ( )

Definition at line 78 of file platform.c.

#### 3.5.2.2 void disable\_caches ( )

Definition at line 49 of file platform.c.

#### 3.5.2.3 void enable\_caches ( )

Definition at line 33 of file platform.c.

**3.5.2.4 void init\_platform ( )**

Definition at line 71 of file platform.c.

**3.5.2.5 void init\_uart ( )**

Definition at line 56 of file platform.c.

**3.6 src/platform.h File Reference**

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

**Functions**

- void [init\\_platform](#) ()
- void [cleanup\\_platform](#) ()

**3.6.1 Function Documentation****3.6.1.1 void cleanup\_platform ( )**

Definition at line 78 of file platform.c.

**3.6.1.2 void init\_platform ( )**

Definition at line 71 of file platform.c.

**3.7 src/platform\_config.h File Reference****3.8 src/utilities.c File Reference**

```
#include "utilities.h"
#include "MAXREFDES18.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, 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\\_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).*

### 3.8.1 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](#).

### 3.8.2 Function Documentation

#### 3.8.2.1 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 166 of file utilities.c.

**3.8.2.2 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 347 of file utilities.c.

**3.8.2.3 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 the memory map
----	-------------------	---

**Return values**

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

Definition at line 187 of file utilities.c.

### 3.8.2.4 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 222 of file utilities.c.

### 3.8.2.5 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<sup>Exponent</sup></i>
--------------------------------

Definition at line 262 of file utilities.c.

### 3.8.2.6 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.

### 3.8.2.7 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 291 of file utilities.c.

### 3.8.2.8 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>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
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 78 of file utilities.c.

### 3.9 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

- 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\\_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 [print\\_asterisks](#) (int nQuantity)  
*Print nQuantity of asterisks to the default Hyperterminal UART.*

#### 3.9.1 Detailed Description

```
Project: Carmel (MAXREFDES18)
Filename: utilities.h
Description: This module contains the utility functions for the
             LX9 board 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](#).

### 3.9.2 Function Documentation

#### 3.9.2.1 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 166 of file utilities.c.

#### 3.9.2.2 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 347 of file utilities.c.

**3.9.2.3 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 the memory map
----	-------------------	---

**Return values**

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

Definition at line 187 of file utilities.c.

**3.9.2.4 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 222 of file utilities.c.

**3.9.2.5 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 ( $nBase^{nExponent}$ ) 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**

$Base^{Exponent}$
-------------------

Definition at line 262 of file utilities.c.

**3.9.2.6 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.

**3.9.2.7 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 <i>uchRxData</i>

**Return values**

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

Definition at line 291 of file utilities.c.

### 3.9.2.8 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 78 of file utilities.c.

# Index

CURRENT\_RANGE\_MENU  
    menu.h, [18](#)  
cleanup\_platform  
    platform.c, [22](#)  
    platform.h, [23](#)  
  
delay  
    utilities.c, [24](#)  
    utilities.h, [29](#)  
disable\_caches  
    platform.c, [22](#)  
  
enable\_caches  
    platform.c, [22](#)  
  
g\_PmodPortMuxSettings  
    MAXREFDES18.c, [7](#)  
g\_gpButtons  
    MAXREFDES18.c, [7](#)  
g\_sInputString  
    MAXREFDES18.c, [7](#)  
    MAXREFDES18.h, [11](#)  
g\_unActivePeripheralAddressSPI  
    MAXREFDES18.c, [7](#)  
g\_xGpioLed  
    MAXREFDES18.c, [7](#)  
g\_xGpioPmodPortA  
    MAXREFDES18.c, [7](#)  
    MAXREFDES18.h, [11](#)  
g\_xGpioPmodPortB  
    MAXREFDES18.c, [7](#)  
    MAXREFDES18.h, [11](#)  
GetLine  
    utilities.c, [25](#)  
    utilities.h, [29](#)  
  
init\_platform  
    platform.c, [22](#)  
    platform.h, [23](#)  
init\_uart  
    platform.c, [23](#)  
  
KEYPRESS\_ARROW\_DOWN  
    menu.h, [18](#)  
KEYPRESS\_ARROW\_LEFT  
    menu.h, [18](#)

KEYPRESS\_ARROW\_RIGHT  
    menu.h, [18](#)  
KEYPRESS\_ARROW\_UP  
    menu.h, [18](#)  
KEYPRESS\_END  
    menu.h, [18](#)  
  
led\_knight\_rider  
    utilities.c, [25](#)  
    utilities.h, [30](#)  
  
MAJOR\_REVISION  
    MAXREFDES18.c, [6](#)  
MAXREFDES18.c  
    g\_PmodPortMuxSettings, [7](#)  
    g\_gpButtons, [7](#)  
    g\_sInputString, [7](#)  
    g\_unActivePeripheralAddressSPI, [7](#)  
    g\_xGpioLed, [7](#)  
    g\_xGpioPmodPortA, [7](#)  
    g\_xGpioPmodPortB, [7](#)  
    main, [6](#)  
MAXREFDES18.h  
    g\_sInputString, [11](#)  
    g\_xGpioPmodPortA, [11](#)  
    g\_xGpioPmodPortB, [11](#)  
MINOR\_REVISION  
    MAXREFDES18.c, [6](#)  
MODE\_MENU  
    menu.h, [18](#)  
main  
    MAXREFDES18.c, [6](#)  
max\_configure\_PMOD\_port  
    utilities.c, [25](#)  
    utilities.h, [30](#)  
menu.c  
    menu\_cls, [12](#)  
    menu\_get\_direct\_entry, [12](#)  
    menu\_print\_current\_range\_menu, [13](#)  
    menu\_print\_line, [13](#)  
    menu\_print\_maxim\_banner, [13](#)  
    menu\_print\_maxim\_banner\_big, [14](#)  
    menu\_print\_mode\_menu, [14](#)  
    menu\_print\_prompt, [14](#)  
    menu\_print\_voltage\_range\_menu, [15](#)  
    menu\_retrieve\_keypress, [15](#)

- menu.h
  - CURRENT\_RANGE\_MENU, 18
  - KEYPRESS\_ARROW\_DOWN, 18
  - KEYPRESS\_ARROW\_LEFT, 18
  - KEYPRESS\_ARROW\_UP, 18
  - KEYPRESS\_END, 18
  - MODE\_MENU, 18
  - menu\_cls, 19
  - menu\_get\_direct\_entry, 19
  - menu\_print\_current\_range\_menu, 20
  - menu\_print\_maxim\_banner, 20
  - menu\_print\_maxim\_banner\_big, 20
  - menu\_print\_mode\_menu, 20
  - menu\_print\_prompt, 21
  - menu\_print\_voltage\_range\_menu, 21
  - menu\_retrieve\_keypress, 21
  - VOLTAGE\_RANGE\_MENU, 19
  - WAIT\_KEYPRESS, 19
- menu\_cls
  - menu.c, 12
  - menu.h, 19
- menu\_get\_direct\_entry
  - menu.c, 12
  - menu.h, 19
- menu\_print\_current\_range\_menu
  - menu.c, 13
  - menu.h, 20
- menu\_print\_line
  - menu.c, 13
- menu\_print\_maxim\_banner
  - menu.c, 13
  - menu.h, 20
- menu\_print\_maxim\_banner\_big
  - menu.c, 14
  - menu.h, 20
- menu\_print\_mode\_menu
  - menu.c, 14
  - menu.h, 20
- menu\_print\_prompt
  - menu.c, 14
  - menu.h, 21
- menu\_print\_voltage\_range\_menu
  - menu.c, 15
  - menu.h, 21
- menu\_retrieve\_keypress
  - menu.c, 15
  - menu.h, 21
- number\_raised\_to\_power
  - utilities.c, 26
  - utilities.h, 31
- PMOD\_TYPE\_GPIO
  - MAXREFDES18.h, 10
- PMOD\_TYPE\_I2C
  - MAXREFDES18.h, 10
- PMOD\_TYPE\_SPI
  - MAXREFDES18.h, 10
- PMOD\_TYPE\_UART
  - MAXREFDES18.h, 10
- platform.c
  - cleanup\_platform, 22
  - disable\_caches, 22
  - enable\_caches, 22
  - init\_platform, 22
  - init\_uart, 23
  - UART\_BAUD, 22
- platform.h
  - cleanup\_platform, 23
  - init\_platform, 23
- print\_asterisks
  - utilities.c, 26
  - utilities.h, 31
- receive\_byte\_with\_timeout
  - utilities.c, 27
  - utilities.h, 31
- SpiRW
  - utilities.c, 27
  - utilities.h, 32
- src/MAXREFDES18.c, 5
- src/MAXREFDES18.h, 8
- src/menu.c, 11
- src/menu.h, 15
- src/platform.c, 22
- src/platform.h, 23
- src/platform\_config.h, 23
- src/utilities.c, 23
- src/utilities.h, 28
- UART\_BAUD
  - platform.c, 22
- utilities.c
  - delay, 24
  - GetLine, 25
  - led\_knight\_rider, 25
  - max\_configure\_PMOD\_port, 25
  - number\_raised\_to\_power, 26
  - print\_asterisks, 26
  - receive\_byte\_with\_timeout, 27
  - SpiRW, 27
- utilities.h
  - delay, 29
  - GetLine, 29
  - led\_knight\_rider, 30
  - max\_configure\_PMOD\_port, 30
  - number\_raised\_to\_power, 31
  - print\_asterisks, 31
  - receive\_byte\_with\_timeout, 31



SpiRW, [32](#)

VOLTAGE\_RANGE\_MENU  
menu.h, [19](#)

WAIT\_KEYPRESS  
menu.h, [19](#)