

MAXREFDES71# Code Documentation

V01.00

Generated by Doxygen 1.8.2

Mon Apr 14 2014 12:22:26

Contents

| | | |
|----------|---|-----------|
| 1 | Main Page | 1 |
| 1.1 | Introduction | 1 |
| 2 | Data Structure Index | 3 |
| 2.1 | Data Structures | 3 |
| 3 | File Index | 5 |
| 3.1 | File List | 5 |
| 4 | Data Structure Documentation | 7 |
| 4.1 | maximDateTime Struct Reference | 7 |
| 4.1.1 | Detailed Description | 7 |
| 4.1.2 | Field Documentation | 7 |
| 4.1.2.1 | uchDate | 7 |
| 4.1.2.2 | uchDow | 7 |
| 4.1.2.3 | uchHour | 8 |
| 4.1.2.4 | uchMinute | 8 |
| 4.1.2.5 | uchMonth | 8 |
| 4.1.2.6 | uchSecond | 8 |
| 4.1.2.7 | uchYear | 8 |
| 4.2 | maximOLEDDisplay Struct Reference | 8 |
| 4.2.1 | Detailed Description | 8 |
| 4.2.2 | Field Documentation | 8 |
| 4.2.2.1 | flippedBuffer | 8 |
| 4.2.2.2 | font | 9 |
| 4.2.2.3 | portStatus | 9 |
| 4.2.2.4 | writeBuffer | 9 |
| 4.2.2.5 | xgpioPort | 9 |
| 5 | File Documentation | 11 |

| | | |
|----------|---|----|
| 5.1 | src/axi_millbrae.c File Reference | 11 |
| 5.1.1 | Detailed Description | 12 |
| 5.1.2 | Function Documentation | 13 |
| 5.1.2.1 | AXI_MILLBRAE_Config_ADC | 13 |
| 5.1.2.2 | AXI_MILLBRAE_Config_DAC | 13 |
| 5.1.2.3 | AXI_MILLBRAE_Interrupt_Disable | 13 |
| 5.1.2.4 | AXI_MILLBRAE_Interrupt_Enable | 14 |
| 5.1.2.5 | AXI_MILLBRAE_Read_Data | 14 |
| 5.1.2.6 | AXI_MILLBRAE_Read_Interrupt_Status | 14 |
| 5.1.2.7 | AXI_MILLBRAE_Single_Convert | 15 |
| 5.1.2.8 | AXI_MILLBRAE_Start_Conversion | 15 |
| 5.1.2.9 | AXI_MILLBRAE_Start_Replication | 15 |
| 5.1.2.10 | AXI_MILLBRAE_Stop_Operation | 16 |
| 5.1.2.11 | AXI_MILLBRAE_Write_Timer_Reg | 16 |
| 5.2 | src/axi_millbrae.h File Reference | 17 |
| 5.2.1 | Macro Definition Documentation | 18 |
| 5.2.1.1 | AXI_MILLBRAE_ADC_CONF_REG_OFFSET | 18 |
| 5.2.1.2 | AXI_MILLBRAE_ADC_DATA1_REG_OFFSET | 18 |
| 5.2.1.3 | AXI_MILLBRAE_ADC_DATA2_REG_OFFSET | 18 |
| 5.2.1.4 | AXI_MILLBRAE_DAC_CONF_REG_OFFSET | 18 |
| 5.2.1.5 | AXI_MILLBRAE_IP_INTR_ENABLE_REG_OFFSET | 18 |
| 5.2.1.6 | AXI_MILLBRAE_IP_INTR_STATUS_REG_OFFSET | 18 |
| 5.2.1.7 | AXI_MILLBRAE_MODE_REG_OFFSET | 19 |
| 5.2.1.8 | AXI_MILLBRAE_mReadReg | 19 |
| 5.2.1.9 | AXI_MILLBRAE_mWriteReg | 19 |
| 5.2.1.10 | AXI_MILLBRAE_START_OPERATION_REG_OFFSET | 19 |
| 5.2.1.11 | AXI_MILLBRAE_TIMER_REG_OFFSET | 20 |
| 5.2.1.12 | AXI_MILLBRAE_USER_SLV_SPACE_OFFSET | 20 |
| 5.2.1.13 | AXI_MILLBRAE_WRITE_ADC_CR_REG_OFFSET | 20 |
| 5.2.1.14 | AXI_MILLBRAE_WRITE_DAC_CR_REG_OFFSET | 20 |
| 5.2.2 | Function Documentation | 20 |
| 5.2.2.1 | AXI_MILLBRAE_Config_ADC | 20 |
| 5.2.2.2 | AXI_MILLBRAE_Config_DAC | 20 |
| 5.2.2.3 | AXI_MILLBRAE_Interrupt_Disable | 21 |
| 5.2.2.4 | AXI_MILLBRAE_Interrupt_Enable | 21 |
| 5.2.2.5 | AXI_MILLBRAE_Read_Data | 21 |
| 5.2.2.6 | AXI_MILLBRAE_Read_Interrupt_Status | 22 |

| | | |
|----------|---|----|
| 5.2.2.7 | AXI_MILLBRAE_Single_Convert | 22 |
| 5.2.2.8 | AXI_MILLBRAE_Start_Conversion | 23 |
| 5.2.2.9 | AXI_MILLBRAE_Start_Replication | 23 |
| 5.2.2.10 | AXI_MILLBRAE_Stop_Operation | 23 |
| 5.2.2.11 | AXI_MILLBRAE_Write_Timer_Reg | 24 |
| 5.3 | src/maximDeviceSpecificUtilities.c File Reference | 24 |
| 5.3.1 | Detailed Description | 25 |
| 5.3.2 | Macro Definition Documentation | 25 |
| 5.3.2.1 | INTC_DEVICE_INT_ID | 25 |
| 5.3.3 | Function Documentation | 25 |
| 5.3.3.1 | continuous_sampling | 25 |
| 5.3.3.2 | SetupInterruptSystem | 26 |
| 5.3.3.3 | signal_replication | 26 |
| 5.3.3.4 | start_sampling | 27 |
| 5.3.4 | Variable Documentation | 27 |
| 5.3.4.1 | g_auSamplesCh1 | 27 |
| 5.3.4.2 | g_auSamplesCh2 | 27 |
| 5.3.4.3 | g_uchReadADCHandlerStop | 27 |
| 5.3.4.4 | g_unCount | 27 |
| 5.3.4.5 | g_unSampleSize | 27 |
| 5.4 | src/maximDeviceSpecificUtilities.h File Reference | 28 |
| 5.4.1 | Detailed Description | 28 |
| 5.4.2 | Function Documentation | 29 |
| 5.4.2.1 | continuous_sampling | 29 |
| 5.4.2.2 | ReadADCHandler | 29 |
| 5.4.2.3 | SetupInterruptSystem | 29 |
| 5.4.2.4 | signal_replication | 29 |
| 5.4.2.5 | start_sampling | 30 |
| 5.5 | src/MAXREFDES71.c File Reference | 30 |
| 5.5.1 | Detailed Description | 31 |
| 5.5.2 | Macro Definition Documentation | 31 |
| 5.5.2.1 | MAJOR_REVISION | 31 |
| 5.5.2.2 | MINOR_REVISION | 32 |
| 5.5.3 | Function Documentation | 32 |
| 5.5.3.1 | main | 32 |
| 5.5.4 | Variable Documentation | 32 |
| 5.5.4.1 | font | 32 |

| | | |
|----------|---|----|
| 5.5.4.2 | <code>g_nSampleRate</code> | 32 |
| 5.5.4.3 | <code>g_sInputString</code> | 32 |
| 5.5.4.4 | <code>g_structureOLED</code> | 32 |
| 5.5.4.5 | <code>g_tempString</code> | 32 |
| 5.5.4.6 | <code>g_unSampleSize</code> | 33 |
| 5.5.4.7 | <code>g_xGpioLed</code> | 33 |
| 5.6 | <code>src/MAXREFDES71.h</code> File Reference | 33 |
| 5.6.1 | Macro Definition Documentation | 34 |
| 5.6.1.1 | <code>ABOUT_ONE_SECOND</code> | 34 |
| 5.6.1.2 | <code>DEFAULT_HYPERTERMINAL_UART_ADDRESS</code> | 34 |
| 5.6.1.3 | <code>DEFAULT_HYPERTERMINAL_UART_ID</code> | 34 |
| 5.6.1.4 | <code>INPUT_STRING_MAX_SIZE</code> | 34 |
| 5.6.2 | Variable Documentation | 34 |
| 5.6.2.1 | <code>g_nSampleRate</code> | 34 |
| 5.6.2.2 | <code>g_sInputString</code> | 34 |
| 5.6.2.3 | <code>g_structureOLED</code> | 34 |
| 5.6.2.4 | <code>g_tempString</code> | 34 |
| 5.6.2.5 | <code>g_uchAtoDConverterChannel</code> | 34 |
| 5.6.2.6 | <code>g_unSampleSize</code> | 34 |
| 5.6.2.7 | <code>g_xGpioLed</code> | 35 |
| 5.7 | <code>src/menu.c</code> File Reference | 35 |
| 5.7.1 | Detailed Description | 35 |
| 5.7.2 | Function Documentation | 36 |
| 5.7.2.1 | <code>menu_cls</code> | 36 |
| 5.7.2.2 | <code>menu_print_ADC_menu</code> | 36 |
| 5.7.2.3 | <code>menu_print_channel_menu</code> | 37 |
| 5.7.2.4 | <code>menu_print_line</code> | 37 |
| 5.7.2.5 | <code>menu_print_main_menu</code> | 37 |
| 5.7.2.6 | <code>menu_print_maxim_banner</code> | 37 |
| 5.7.2.7 | <code>menu_print_maxim_banner_big</code> | 37 |
| 5.7.2.8 | <code>menu_print_prompt</code> | 38 |
| 5.7.2.9 | <code>menu_print_replication_rate_menu</code> | 38 |
| 5.7.2.10 | <code>menu_print_sample_rate_menu</code> | 38 |
| 5.7.2.11 | <code>menu_print_sample_size_menu</code> | 39 |
| 5.7.2.12 | <code>menu_retrieve_keypress</code> | 39 |
| 5.8 | <code>src/menu.h</code> File Reference | 39 |
| 5.8.1 | Detailed Description | 41 |

| | | |
|----------|----------------------------------|----|
| 5.8.2 | Macro Definition Documentation | 41 |
| 5.8.2.1 | ADC_MENU | 41 |
| 5.8.2.2 | BASE_FUNCTION_STATE | 41 |
| 5.8.2.3 | KEYPRESS_ARROW_DOWN | 41 |
| 5.8.2.4 | KEYPRESS_ARROW_LEFT | 41 |
| 5.8.2.5 | KEYPRESS_ARROW_RIGHT | 42 |
| 5.8.2.6 | KEYPRESS_ARROW_UP | 42 |
| 5.8.2.7 | KEYPRESS_END | 42 |
| 5.8.2.8 | MAIN_MENU | 42 |
| 5.8.2.9 | MENU_CHANNEL | 42 |
| 5.8.2.10 | MENU_SAMPLE_SIZE | 42 |
| 5.8.2.11 | MENU_SAMPLE_SPEED | 42 |
| 5.8.2.12 | REP_RATE_MENU | 42 |
| 5.8.2.13 | START_BLOCK_SAMPLING | 43 |
| 5.8.2.14 | START_CONTINUOUS_SAMPLING | 43 |
| 5.8.2.15 | WAIT_KEYPRESS | 43 |
| 5.8.3 | Function Documentation | 43 |
| 5.8.3.1 | menu_cls | 43 |
| 5.8.3.2 | menu_print_ADC_menu | 43 |
| 5.8.3.3 | menu_print_channel_menu | 43 |
| 5.8.3.4 | menu_print_main_menu | 44 |
| 5.8.3.5 | menu_print_maxim_banner | 44 |
| 5.8.3.6 | menu_print_maxim_banner_big | 44 |
| 5.8.3.7 | menu_print_prompt | 44 |
| 5.8.3.8 | menu_print_replication_rate_menu | 45 |
| 5.8.3.9 | menu_print_sample_rate_menu | 45 |
| 5.8.3.10 | menu_print_sample_size_menu | 45 |
| 5.8.3.11 | menu_retrieve_keypress | 45 |
| 5.9 | src/platform.c File Reference | 46 |
| 5.9.1 | Macro Definition Documentation | 46 |
| 5.9.1.1 | UART_BAUD | 46 |
| 5.9.2 | Function Documentation | 46 |
| 5.9.2.1 | cleanup_platform | 46 |
| 5.9.2.2 | disable_caches | 47 |
| 5.9.2.3 | enable_caches | 47 |
| 5.9.2.4 | init_platform | 47 |
| 5.9.2.5 | init_uart | 47 |

| | |
|---|----|
| 5.10 src/platform.h File Reference | 47 |
| 5.10.1 Function Documentation | 47 |
| 5.10.1.1 cleanup_platform | 47 |
| 5.10.1.2 init_platform | 47 |
| 5.11 src/platform_config.h File Reference | 47 |
| 5.12 src/utilities.c File Reference | 47 |
| 5.12.1 Detailed Description | 49 |
| 5.12.2 Macro Definition Documentation | 49 |
| 5.12.2.1 OLED_DATA_COMMAND_B | 49 |
| 5.12.2.2 OLED_RESET_B | 49 |
| 5.12.2.3 OLED_SCLK | 49 |
| 5.12.2.4 OLED_SDIN | 49 |
| 5.12.2.5 OLED_VBAT | 49 |
| 5.12.2.6 OLED_VDD | 49 |
| 5.12.2.7 UART_BASEADDR | 49 |
| 5.12.3 Function Documentation | 50 |
| 5.12.3.1 checkUartEmpty | 50 |
| 5.12.3.2 clearOLEDBuffer | 50 |
| 5.12.3.3 delay | 50 |
| 5.12.3.4 displayOLEDBuffer | 51 |
| 5.12.3.5 flipAndCopyDisplayBuffer | 51 |
| 5.12.3.6 getUartByte | 51 |
| 5.12.3.7 initializeOLED | 52 |
| 5.12.3.8 led_knight_rider | 52 |
| 5.12.3.9 number_raised_to_power | 52 |
| 5.12.3.10 print_asterisks | 53 |
| 5.12.3.11 printfToBufferOLED | 53 |
| 5.12.3.12 printfToOLED | 53 |
| 5.12.3.13 putCharOLED | 54 |
| 5.12.3.14 sendOLEDSPI | 54 |
| 5.12.3.15 sendUartByte | 55 |
| 5.13 src/utilities.h File Reference | 55 |
| 5.13.1 Detailed Description | 56 |
| 5.13.2 Function Documentation | 56 |
| 5.13.2.1 checkUartEmpty | 56 |
| 5.13.2.2 clearOLEDBuffer | 57 |
| 5.13.2.3 delay | 57 |

| | |
|---|----|
| 5.13.2.4 displayOLEDBuffer | 57 |
| 5.13.2.5 flipAndCopyDisplayBuffer | 58 |
| 5.13.2.6 getUartByte | 58 |
| 5.13.2.7 initializeOLED | 58 |
| 5.13.2.8 led_knight_rider | 59 |
| 5.13.2.9 number_raised_to_power | 59 |
| 5.13.2.10print_asterisks | 60 |
| 5.13.2.11printfToBufferOLED | 60 |
| 5.13.2.12printfToOLED | 60 |
| 5.13.2.13putCharOLED | 61 |
| 5.13.2.14sendOLEDSPI | 61 |
| 5.13.2.15sendUartByte | 61 |
| 5.13.3 Variable Documentation | 62 |
| 5.13.3.1 t | 62 |

Chapter 1

Main Page

1.1 Introduction

This is the code documentation for the MAXREFDES71# subsystem reference design.

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

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

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

| | |
|---|-------------------|
| maximDateTime | |
| < Structure to hold time, date, day of week | 7 |
| maximOLEDDisplay | 8 |

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

| | |
|---|----|
| src/ axi_millbrae.c | 11 |
| src/ axi_millbrae.h | 17 |
| src/ maximDeviceSpecificUtilities.c | 24 |
| src/ maximDeviceSpecificUtilities.h | 28 |
| src/ MAXREFDES71.c | 30 |
| src/ MAXREFDES71.h | 33 |
| src/ menu.c | 35 |
| src/ menu.h | 39 |
| src/ platform.c | 46 |
| src/ platform.h | 47 |
| src/ platform_config.h | 47 |
| src/ utilities.c | 47 |
| src/ utilities.h | 55 |

Chapter 4

Data Structure Documentation

4.1 maximDateTime Struct Reference

< Structure to hold time, date, day of week

```
#include <MAXREFDES71.h>
```

Data Fields

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

4.1.1 Detailed Description

< Structure to hold time, date, day of week

Definition at line 84 of file MAXREFDES71.h.

4.1.2 Field Documentation

4.1.2.1 u8 uchDate

Definition at line 90 of file MAXREFDES71.h.

4.1.2.2 u8 uchDow

Definition at line 92 of file MAXREFDES71.h.

4.1.2.3 u8 uchHour

Definition at line 86 of file MAXREFDES71.h.

4.1.2.4 u8 uchMinute

Definition at line 87 of file MAXREFDES71.h.

4.1.2.5 u8 uchMonth

Definition at line 89 of file MAXREFDES71.h.

4.1.2.6 u8 uchSecond

Definition at line 88 of file MAXREFDES71.h.

4.1.2.7 u8 uchYear

Definition at line 91 of file MAXREFDES71.h.

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

- [src/MAXREFDES71.h](#)

4.2 maximOLEDDisplay Struct Reference

```
#include <MAXREFDES71.h>
```

Data Fields

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

4.2.1 Detailed Description

Definition at line 95 of file MAXREFDES71.h.

4.2.2 Field Documentation

4.2.2.1 u8 flippedBuffer[512]

Definition at line 100 of file MAXREFDES71.h.

4.2.2.2 **u8* font**

Definition at line 101 of file MAXREFDES71.h.

4.2.2.3 **u8 portStatus**

Definition at line 98 of file MAXREFDES71.h.

4.2.2.4 **u8 writeBuffer[512]**

Definition at line 99 of file MAXREFDES71.h.

4.2.2.5 **XGpio xgpioPort**

Definition at line 97 of file MAXREFDES71.h.

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

- src/[MAXREFDES71.h](#)

Chapter 5

File Documentation

5.1 src/axi_millbrae.c File Reference

```
#include "axi_millbrae.h"
```

Functions

- void [AXI_MILLBRAE_Config_ADC](#) (u32 BaseAddress, u32 Value)
Config ADC.
- void [AXI_MILLBRAE_Config_DAC](#) (u32 BaseAddress, u32 Value)
Config DAC.
- void [AXI_MILLBRAE_Start_Conversion](#) (u32 BaseAddress)
Start ADC Conversion.
- void [AXI_MILLBRAE_Stop_Operation](#) (u32 BaseAddress)
Stop operation.
- void [AXI_MILLBRAE_Start_Replication](#) (u32 BaseAddress)
Start signal replication.
- void [AXI_MILLBRAE_Write_Timer_Reg](#) (u32 BaseAddress, u32 Value)
Write Value to the Timer register.
- void [AXI_MILLBRAE_Interrupt_Enable](#) (u32 BaseAddress)
Enable the interrupt.
- void [AXI_MILLBRAE_Interrupt_Disable](#) (u32 BaseAddress)
Disable the interrupt.
- u32 [AXI_MILLBRAE_Read_Data](#) (u32 BaseAddress, u8 Channel)
Read sampled data.
- u32 [AXI_MILLBRAE_Read_Interrupt_Status](#) (u32 BaseAddress)
Read interrupt status.
- void [AXI_MILLBRAE_Single_Convert](#) (u32 BaseAddress, u8 Channel, u16 *uSample)
Read one ADC sample.

5.1.1 Detailed Description

```

Project: MAXREFDES71#
Filename: axi_millbrae.c
Description: This file contains the software API definition of the
             axi_millbrae custom IP core.

```

Revision History:

12-17-13 Rev 01.00 GL Initial Release

This code follows the following naming conventions:

char chPmodValue

char (array) sPmodString[16]

float fPmodValue

int nPmodValue

int (array) anPmodValue[16]

u16 uPmodValue

u16 (array) auPmodValue[16]

u8 uchPmodValue

u8 (array) auchPmodBuffer[16]

unsigned int unPmodValue

int * punPmodValue

```

Project: MAXREFDES61#
Filename: axi_millbrae.h
Description: This header file contains the software API prototypes of the
             axi_millbrae custom IP core.

```

Revision History:

12-17-13 Rev 01.00 GL Initial Release

This code follows the following naming conventions:

char chPmodValue

char (array) sPmodString[16]

float fPmodValue

int nPmodValue

int (array) anPmodValue[16]

u16 uPmodValue

u16 (array) auPmodValue[16]

u8 uchPmodValue

u8 (array) auchPmodBuffer[16]

unsigned int unPmodValue

int * punPmodValue

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

5.1.2 Function Documentation

5.1.2.1 void AXI_MILLBRAE_Config_ADC (u32 *BaseAddress*, u32 *Value*)

Config ADC.

Details

This function write a value to the config register of the ADC

Parameters

| | | |
|----|--------------------|--|
| in | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|----|--------------------|--|

Return values

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

Definition at line 66 of file axi_millbrae.c.

5.1.2.2 void AXI_MILLBRAE_Config_DAC (u32 *BaseAddress*, u32 *Value*)

Config DAC.

Details

This function write a value to the config register of the DAC

Parameters

| | | |
|----|--------------------|--|
| in | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|----|--------------------|--|

Return values

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

Definition at line 81 of file axi_millbrae.c.

5.1.2.3 void AXI_MILLBRAE_Interrupt_Disable (u32 *BaseAddress*)

Disable the interrupt.

Details

This function writes a 0 to the interrupt enable register to disable the interrupt

Parameters

| | | |
|----|--------------------|--|
| in | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|----|--------------------|--|

Return values

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

Definition at line 183 of file axi_millbrae.c.

5.1.2.4 void AXI_MILLBRAE_Interrupt_Enable (u32 *BaseAddress*)

Enable the interrupt.

Details

This function writes a 1 to the interrupt enable register to enable the interrupt. When the interrupt is enabled, the AXI_MILLBRAE IP core will raise an interrupt when the sampled data is available.

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

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

Definition at line 166 of file axi_millbrae.c.

5.1.2.5 u32 AXI_MILLBRAE_Read_Data (u32 *BaseAddress*, u8 *Channel*)

Read sampled data.

Details

This function reads the ADC sampled data register. The data register will be overwritten when the new sampled data is available.

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

| | |
|------------|--------------|
| <i>ADC</i> | sampled data |
|------------|--------------|

Definition at line 198 of file axi_millbrae.c.

5.1.2.6 u32 AXI_MILLBRAE_Read_Interrupt_Status (u32 *BaseAddress*)

Read interrupt status.

Details

This function reads the interrupt status register. When the value is one, sampled data is available in the data register. Interrupt status register is a clear-on-read register

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

| | |
|------------------|--------|
| <i>Interrupt</i> | status |
|------------------|--------|

Definition at line 214 of file axi_millbrae.c.

5.1.2.7 void AXI_MILLBRAE_Single_Convert (u32 *BaseAddress*, u8 *Channel*, u16 * *uSample*)

Read one ADC sample.

Details

This function starts an ADC conversion and returns one ADC sample.

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

| | |
|------------|------------|
| <i>One</i> | ADC sample |
|------------|------------|

Definition at line 231 of file axi_millbrae.c.

5.1.2.8 void AXI_MILLBRAE_Start_Conversion (u32 *BaseAddress*)

Start ADC Conversion.

Details

This function writes a 0 to the Mode register of the AXI_MILLBRAE ip core to set the operation mode to ADC conversion mode This function also write a 1 to the start operation register of the AXI_MILLBRAE ip core to start the conversion

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

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

Definition at line 96 of file axi_millbrae.c.

5.1.2.9 void AXI_MILLBRAE_Start_Replication (u32 *BaseAddress*)

Start signal replication.

Details

This function writes a 1 to the Mode register of the AXI_MILLBRAE ip core to set the operation mode to signal replication mode This function also write a 1 to the start operation register of the AXI_MILLBRAE ip core to start the signal replication

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

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

Definition at line 129 of file axi_millbrae.c.

5.1.2.10 void AXI_MILLBRAE_Stop_Operation (u32 *BaseAddress*)

Stop operation.

Details

This function writes a 0 to the start operation register of the AXI_MILLBRAE ip core to stop the operation

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

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

Definition at line 114 of file axi_millbrae.c.

5.1.2.11 void AXI_MILLBRAE_Write_Timer_Reg (u32 *BaseAddress*, u32 *Value*)

Write Value to the Timer register.

Details

This function writes a value to the timer register. The timer register is used to control the sampling rate Sampling rate = 50000000/(Value+1)

Parameters

| | | |
|-----------|--------------------|---|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
| <i>in</i> | <i>Value</i> | - a value to be written to the timer register |

Return values

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

Definition at line 149 of file axi_millbrae.c.

5.2 src/axi_millbrae.h File Reference

```
#include "xbasic_types.h"
#include "xstatus.h"
#include "xil_io.h"
```

Macros

- #define [AXI_MILLBRAE_USER_SLV_SPACE_OFFSET](#) (0x00000000)
- #define [AXI_MILLBRAE_ADC_DATA1_REG_OFFSET](#) ([AXI_MILLBRAE_USER_SLV_SPACE_OFFSET](#) + 0x00000000)
- #define [AXI_MILLBRAE_ADC_DATA2_REG_OFFSET](#) ([AXI_MILLBRAE_USER_SLV_SPACE_OFFSET](#) + 0x00000004)
- #define [AXI_MILLBRAE_MODE_REG_OFFSET](#) ([AXI_MILLBRAE_USER_SLV_SPACE_OFFSET](#) + 0x00000008)
- #define [AXI_MILLBRAE_TIMER_REG_OFFSET](#) ([AXI_MILLBRAE_USER_SLV_SPACE_OFFSET](#) + 0x0000000C)
- #define [AXI_MILLBRAE_ADC_CONF_REG_OFFSET](#) ([AXI_MILLBRAE_USER_SLV_SPACE_OFFSET](#) + 0x00000010)
- #define [AXI_MILLBRAE_DAC_CONF_REG_OFFSET](#) ([AXI_MILLBRAE_USER_SLV_SPACE_OFFSET](#) + 0x00000014)
- #define [AXI_MILLBRAE_WRITE_ADC_CR_REG_OFFSET](#) ([AXI_MILLBRAE_USER_SLV_SPACE_OFFSET](#) + 0x00000018)
- #define [AXI_MILLBRAE_WRITE_DAC_CR_REG_OFFSET](#) ([AXI_MILLBRAE_USER_SLV_SPACE_OFFSET](#) + 0x0000001C)
- #define [AXI_MILLBRAE_START_OPERATION_REG_OFFSET](#) ([AXI_MILLBRAE_USER_SLV_SPACE_OFFSET](#) + 0x00000020)
- #define [AXI_MILLBRAE_IP_INTR_STATUS_REG_OFFSET](#) ([AXI_MILLBRAE_USER_SLV_SPACE_OFFSET](#) + 0x00000024)
- #define [AXI_MILLBRAE_IP_INTR_ENABLE_REG_OFFSET](#) ([AXI_MILLBRAE_USER_SLV_SPACE_OFFSET](#) + 0x00000028)
- #define [AXI_MILLBRAE_mWriteReg](#)(BaseAddress, RegOffset, Data) Xil_Out32((BaseAddress) + (RegOffset), (Xuint32)(Data))
- #define [AXI_MILLBRAE_mReadReg](#)(BaseAddress, RegOffset) Xil_In32((BaseAddress) + (RegOffset))

Functions

- void [AXI_MILLBRAE_Config_ADC](#) (u32 BaseAddress, u32 Value)
Config ADC.
- void [AXI_MILLBRAE_Config_DAC](#) (u32 BaseAddress, u32 Value)
Config DAC.
- void [AXI_MILLBRAE_Start_Conversion](#) (u32 BaseAddress)
Start ADC Conversion.
- void [AXI_MILLBRAE_Stop_Operation](#) (u32 BaseAddress)
Stop operation.
- void [AXI_MILLBRAE_Start_Replication](#) (u32 BaseAddress)

Start signal replication.

- void [AXI_MILLBRAE_Write_Timer_Reg](#) (u32 BaseAddress, u32 Value)

Write Value to the Timer register.

- void [AXI_MILLBRAE_Interrupt_Enable](#) (u32 BaseAddress)

Enable the interrupt.

- void [AXI_MILLBRAE_Interrupt_Disable](#) (u32 BaseAddress)

Disable the interrupt.

- u32 [AXI_MILLBRAE_Read_Data](#) (u32 BaseAddress, u8 Channel)

Read sampled data.

- u32 [AXI_MILLBRAE_Read_Interrupt_Status](#) (u32 BaseAddress)

Read interrupt status.

- void [AXI_MILLBRAE_Single_Convert](#) (u32 BaseAddress, u8 Channel, u16 *uSample)

Read one ADC sample.

5.2.1 Macro Definition Documentation

5.2.1.1 **#define AXI_MILLBRAE_ADC_CONF_REG_OFFSET (AXI_MILLBRAE_USER_SLV_SPACE_OFFSET + 0x00000010)**

Definition at line 80 of file axi_millbrae.h.

5.2.1.2 **#define AXI_MILLBRAE_ADC_DATA1_REG_OFFSET (AXI_MILLBRAE_USER_SLV_SPACE_OFFSET + 0x00000000)**

Definition at line 76 of file axi_millbrae.h.

5.2.1.3 **#define AXI_MILLBRAE_ADC_DATA2_REG_OFFSET (AXI_MILLBRAE_USER_SLV_SPACE_OFFSET + 0x00000004)**

Definition at line 77 of file axi_millbrae.h.

5.2.1.4 **#define AXI_MILLBRAE_DAC_CONF_REG_OFFSET (AXI_MILLBRAE_USER_SLV_SPACE_OFFSET + 0x00000014)**

Definition at line 81 of file axi_millbrae.h.

5.2.1.5 **#define AXI_MILLBRAE_IP_INTR_ENABLE_REG_OFFSET (AXI_MILLBRAE_USER_SLV_SPACE_OFFSET + 0x00000028)**

Definition at line 86 of file axi_millbrae.h.

5.2.1.6 **#define AXI_MILLBRAE_IP_INTR_STATUS_REG_OFFSET (AXI_MILLBRAE_USER_SLV_SPACE_OFFSET + 0x00000024)**

Definition at line 85 of file axi_millbrae.h.

5.2.1.7 **#define AXI_MILLBRAE_MODE_REG_OFFSET (AXI_MILLBRAE_USER_SLV_SPACE_OFFSET + 0x00000008)**

Definition at line 78 of file axi_millbrae.h.

5.2.1.8 **#define AXI_MILLBRAE_mReadReg(BaseAddress, RegOffset) Xil_In32((BaseAddress) + (RegOffset))**

Read a value from a AXI_MILLBRAE register. A 32 bit read is performed. If the component is implemented in a smaller width, only the least significant data is read from the register. The most significant data will be read as 0.

Parameters

| | |
|--------------------|---|
| <i>BaseAddress</i> | is the base address of the AXI_MILLBRAE device. |
| <i>RegOffset</i> | is the register offset from the base to write to. |

Returns

Data is the data from the register.

Note

C-style signature: Xuint32 [AXI_MILLBRAE_mReadReg\(Xuint32 BaseAddress, unsigned RegOffset\)](#)

Definition at line 130 of file axi_millbrae.h.

5.2.1.9 **#define AXI_MILLBRAE_mWriteReg(BaseAddress, RegOffset, Data) Xil_Out32((BaseAddress) + (RegOffset), (Xuint32)(Data))**

Write a value to a AXI_MILLBRAE register. A 32 bit write is performed. If the component is implemented in a smaller width, only the least significant data is written.

Parameters

| | |
|--------------------|---|
| <i>BaseAddress</i> | is the base address of the AXI_MILLBRAE device. |
| <i>RegOffset</i> | is the register offset from the base to write to. |
| <i>Data</i> | is the data written to the register. |

Returns

None.

Note

C-style signature: void [AXI_MILLBRAE_mWriteReg\(Xuint32 BaseAddress, unsigned RegOffset, Xuint32 Data\)](#)

Definition at line 110 of file axi_millbrae.h.

5.2.1.10 **#define AXI_MILLBRAE_START_OPERATION_REG_OFFSET (AXI_MILLBRAE_USER_SLV_SPACE_OFFSET + 0x00000020)**

Definition at line 84 of file axi_millbrae.h.

5.2.1.11 #define AXI_MILLBRAE_TIMER_REG_OFFSET (AXI_MILLBRAE_USER_SLV_SPACE_OFFSET + 0x0000000C)

Definition at line 79 of file axi_millbrae.h.

5.2.1.12 #define AXI_MILLBRAE_USER_SLV_SPACE_OFFSET (0x00000000)

User Logic Slave Space Offsets

Definition at line 75 of file axi_millbrae.h.

5.2.1.13 #define AXI_MILLBRAE_WRITE_ADC_CR_REG_OFFSET (AXI_MILLBRAE_USER_SLV_SPACE_OFFSET + 0x00000018)

Definition at line 82 of file axi_millbrae.h.

5.2.1.14 #define AXI_MILLBRAE_WRITE_DAC_CR_REG_OFFSET (AXI_MILLBRAE_USER_SLV_SPACE_OFFSET + 0x0000001C)

Definition at line 83 of file axi_millbrae.h.

5.2.2 Function Documentation

5.2.2.1 void AXI_MILLBRAE_Config_ADC (u32 *BaseAddress*, u32 *Value*)

Config ADC.

Details

This function write a value to the config register of the ADC

Parameters

| | | |
|----|--------------------|--|
| in | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|----|--------------------|--|

Return values

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

Definition at line 66 of file axi_millbrae.c.

5.2.2.2 void AXI_MILLBRAE_Config_DAC (u32 *BaseAddress*, u32 *Value*)

Config DAC.

Details

This function write a value to the config register of the DAC

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

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

Definition at line 81 of file axi_millbrae.c.

5.2.2.3 void AXI_MILLBRAE_Interrupt_Disable (u32 *BaseAddress*)

Disable the interrupt.

Details

This function writes a 0 to the interrupt enable register to disable the interrupt

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

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

Definition at line 183 of file axi_millbrae.c.

5.2.2.4 void AXI_MILLBRAE_Interrupt_Enable (u32 *BaseAddress*)

Enable the interrupt.

Details

This function writes a 1 to the interrupt enable register to enable the interrupt. When the interrupt is enabled, the AXI_MILLBRAE IP core will raise an interrupt when the sampled data is available.

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

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

Definition at line 166 of file axi_millbrae.c.

5.2.2.5 u32 AXI_MILLBRAE_Read_Data (u32 *BaseAddress*, u8 *Channel*)

Read sampled data.

Details

This function reads the ADC sampled data register. The data register will be overwritten when the new sampled data is available.

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

| | |
|------------|--------------|
| <i>ADC</i> | sampled data |
|------------|--------------|

Definition at line 198 of file axi_millbrae.c.

5.2.2.6 u32 AXI_MILLBRAE_Read_Interrupt_Status (u32 *BaseAddress*)

Read interrupt status.

Details

This function reads the interrupt status register. When the value is one, sampled data is available in the data register. Interrupt status register is a clear-on-read register

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

| | |
|------------------|--------|
| <i>Interrupt</i> | status |
|------------------|--------|

Definition at line 214 of file axi_millbrae.c.

5.2.2.7 void AXI_MILLBRAE_Single_Convert (u32 *BaseAddress*, u8 *Channel*, u16 * *uSample*)

Read one ADC sample.

Details

This function starts an ADC conversion and returns one ADC sample.

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

| | |
|------------|------------|
| <i>One</i> | ADC sample |
|------------|------------|

Definition at line 231 of file axi_millbrae.c.

5.2.2.8 void AXI_MILLBRAE_Start_Conversion (u32 *BaseAddress*)

Start ADC Conversion.

Details

This function writes a 0 to the Mode register of the AXI_MILLBRAE ip core to set the operation mode to ADC conversion mode This function also write a 1 to the start operation register of the AXI_MILLBRAE ip core to start the conversion

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

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

Definition at line 96 of file axi_millbrae.c.

5.2.2.9 void AXI_MILLBRAE_Start_Replication (u32 *BaseAddress*)

Start signal replication.

Details

This function writes a 1 to the Mode register of the AXI_MILLBRAE ip core to set the operation mode to signal replication mode This function also write a 1 to the start operation register of the AXI_MILLBRAE ip core to start the signal replication

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

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

Definition at line 129 of file axi_millbrae.c.

5.2.2.10 void AXI_MILLBRAE_Stop_Operation (u32 *BaseAddress*)

Stop operation.

Details

This function writes a 0 to the start operation register of the AXI_MILLBRAE ip core to stop the operation

Parameters

| | | |
|-----------|--------------------|--|
| <i>in</i> | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
|-----------|--------------------|--|

Return values

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

Definition at line 114 of file axi_millbrae.c.

5.2.2.11 void AXI_MILLBRAE_Write_Timer_Reg (u32 *BaseAddress*, u32 *Value*)

Write Value to the Timer register.

Details

This function writes a value to the timer register. The timer register is used to control the sampling rate Sampling rate = 50000000/(Value+1)

Parameters

| | | |
|----|--------------------|---|
| in | <i>BaseAddress</i> | - Base address of the AXI_MILLBRAE IP core |
| in | <i>Value</i> | - a value to be written to the timer register |

Return values

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

Definition at line 149 of file axi_millbrae.c.

5.3 src/maximDeviceSpecificUtilities.c File Reference

```
#include "xbasic_types.h"
#include "stdio.h"
#include "xscugic.h"
#include "axi_millbrae.h"
#include "utilities.h"
```

Macros

- #define [INTC_DEVICE_INT_ID](#) 91

Functions

- int [SetupInterruptSystem](#) (XScuGic *IntcInstancePtr)
Setup the interrupt handler.
- u32 [start_sampling](#) (u32 unSampleSize, int nSampleRate, u16 *auSamplesCh1, u16 *auSamplesCh2)
Receive a block of samples at a constant rate.
- void [signal_replication](#) (int nRepRate)
Continuously reproduce the analog input signals on the outputs.
- void [continuous_sampling](#) (int nChannel)
Continuously reads the ADC and display the data via the HyperTerminal.

Variables

- u32 [g_unCount](#) =0
- u16 * [g_auSamplesCh1](#)
- u16 * [g_auSamplesCh2](#)
- u32 [g_unSampleSize](#)
- u8 [g_uchReadADCHandlerStop](#) =1

5.3.1 Detailed Description

Project: MAXREFDES71#
 Filename: maximDeviceSpecificUtilities.c
 Description: This module uses the AXI_MILLBRAE custom ip core Ver 1.00a to control the ADC and DAC on MAXREFDES71#. These low level functions could be cut/pasted into the user's application as a starting point for development of an end application.

Revision History:

12-17-13 Rev 01.00 GL Initial Release

This code follows the following naming conventions:

char chPmodValue

char (array) sPmodString[16]

float fPmodValue

int nPmodValue

int (array) anPmodValue[16]

u16 uPmodValue

u16 (array) auPmodValue[16]

u8 uchPmodValue

u8 (array) auchPmodBuffer[16]

unsigned int unPmodValue

int * punPmodValue

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

5.3.2 Macro Definition Documentation

5.3.2.1 #define INTC_DEVICE_INT_ID 91

Definition at line 70 of file maximDeviceSpecificUtilities.c.

5.3.3 Function Documentation

5.3.3.1 void continuous_sampling (int nChannel)

Continuously reads the ADC and display the data via the HyperTerminal.

Details

This function reads the ADC every half of a second and display the data via the Hyperterminal. Press the ESC key to return to the main menu.

Parameters

| | |
|----------------|-------------------------------|
| <i>Channel</i> | - selected ADC channel number |
|----------------|-------------------------------|

Return values

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

Definition at line 225 of file maximDeviceSpecificUtilities.c.

5.3.3.2 int SetupInterruptSystem (XScuGic * IntcInstancePtr)

Setup the interrupt handler.

Details

This function connects the interrupt handler to the processor

Parameters

| | | |
|-----------|-------------------------|---|
| <i>in</i> | <i>*IntcInstancePtr</i> | - IntcInstancePtr is the instance of the interrupt controller |
|-----------|-------------------------|---|

Return values

| | |
|--------------------|--|
| <i>XST_SUCCESS</i> | to indicate success, otherwise XST_FAILURE |
|--------------------|--|

Definition at line 285 of file maximDeviceSpecificUtilities.c.

5.3.3.3 void signal_replication (int nRepRate)

Continuously reproduce the analog input signals on the outputs.

Details

This function reads the ADCs and replicates on the outputs at
Press the ESC key to return to the main menu.

Parameters

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

Return values

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

Definition at line 173 of file maximDeviceSpecificUtilities.c.

5.3.3.4 u32 start_sampling (u32 unSampleSize, int nSampleRate, u16 * auSamplesCh1, u16 * auSamplesCh2)

Receive a block of samples at a constant rate.

Details

This function is used to receive a block of samples at a constant sampling rate. The size of the block is defined in u16 *auSampleSize. And the sampling rate is defined in int nSampleRate. The Sampled data will be stored in an array *auSamplesChx

Parameters

| | | |
|-----|----------------------|----------------|
| in | <i>unSampleSize</i> | - Sample size |
| in | <i>nSampleRate</i> | - Sample rate |
| out | <i>*auSamplesChx</i> | - sampled data |

Return values

| | |
|---------------|---------------------|
| <i>Number</i> | of samples received |
|---------------|---------------------|

Definition at line 81 of file maximDeviceSpecificUtilities.c.

5.3.4 Variable Documentation

5.3.4.1 u16* g_auSamplesCh1

Definition at line 76 of file maximDeviceSpecificUtilities.c.

5.3.4.2 u16* g_auSamplesCh2

Definition at line 77 of file maximDeviceSpecificUtilities.c.

5.3.4.3 u8 g_uchReadADCHandlerStop =1

Definition at line 79 of file maximDeviceSpecificUtilities.c.

5.3.4.4 u32 g_unCount =0

Definition at line 75 of file maximDeviceSpecificUtilities.c.

5.3.4.5 u32 g_unSampleSize

Definition at line 78 of file maximDeviceSpecificUtilities.c.

5.4 src/maximDeviceSpecificUtilities.h File Reference

```
#include "xscugic.h"
```

Functions

- u32 [start_sampling](#) (u32 unSampleSize, int nSampleRate, u16 *auSamplesCh1, u16 *auSamplesCh2)
Receive a block of samples at a constant rate.
- void [continuous_sampling](#) (int nChannel)
Continuously reads the ADC and display the data via the HyperTerminal.
- void [signal_replication](#) (int nRepRate)
Continuously reproduce the analog input signals on the outputs.
- int [SetupInterruptSystem](#) (XScuGic *IntcInstancePtr)
Setup the interrupt handler.
- void [ReadADCHandler](#) (void *CallBackRef)

5.4.1 Detailed Description

```
Project: MAXREFDES71#
Filename: maximDeviceSpecificUtilities.h
Description: This module uses the AXI_MILLBRAE custom ip core Ver 1.00a to
              control the ADC and DAC on MAXREFDES71#. These
              low level functions could be cut/pasted into the user's
              application as a starting point for development of an end
              application.
```

Revision History:

12-17-13 Rev 01.00 GL Initial Release

This code follows the following naming conventions:

char chPmodValue

char (array) sPmodString[16]

float fPmodValue

int nPmodValue

int (array) anPmodValue[16]

u16 uPmodValue

u16 (array) auPmodValue[16]

u8 uchPmodValue

u8 (array) auchPmodBuffer[16]

unsigned int unPmodValue

int * punPmodValue

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

5.4.2 Function Documentation

5.4.2.1 void continuous_sampling (int *nChannel*)

Continuously reads the ADC and display the data via the HyperTerminal.

Details

This function reads the ADC every half of a second and display the data via the Hyperterminal. Press the ESC key to return to the main menu.

Parameters

| | |
|----------------|-------------------------------|
| <i>Channel</i> | - selected ADC channel number |
|----------------|-------------------------------|

Return values

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

Definition at line 225 of file maximDeviceSpecificUtilities.c.

5.4.2.2 void ReadADCHandler (void * *CallBackRef*)

5.4.2.3 int SetupInterruptSystem (XScuGic * *IntcInstancePtr*)

Setup the interrupt handler.

Details

This function connects the interrupt handler to the processor

Parameters

| | | |
|-----------|-------------------------|---|
| <i>in</i> | <i>*IntcInstancePtr</i> | - IntcInstancePtr is the instance of the interrupt controller |
|-----------|-------------------------|---|

Return values

| | |
|--------------------|--|
| <i>XST_SUCCESS</i> | to indicate success, otherwise XST_FAILURE |
|--------------------|--|

Definition at line 285 of file maximDeviceSpecificUtilities.c.

5.4.2.4 void signal_replication (int *nRepRate*)

Continuously reproduce the analog input signals on the outputs.

Details

This function reads the ADCs and replicates on the outputs at
Press the ESC key to return to the main menu.

Parameters

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

Return values

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

Definition at line 173 of file maximDeviceSpecificUtilities.c.

5.4.2.5 u32 start_sampling (u32 unSampleSize, int nSampleRate, u16 * auSamplesCh1, u16 * auSamplesCh2)

Receive a block of samples at a constant rate.

Details

This function is used to receive a block of samples at a constant sampling rate. The size of the block is defined in u16 *auSampleSize. And the sampling rate is defined in int nSampleRate. The Sampled data will be stored in an array *auSamplesChx

Parameters

| | | |
|-----|----------------------|----------------|
| in | <i>unSampleSize</i> | - Sample size |
| in | <i>nSampleRate</i> | - Sample rate |
| out | <i>*auSamplesChx</i> | - sampled data |

Return values

| | |
|---------------|---------------------|
| <i>Number</i> | of samples received |
|---------------|---------------------|

Definition at line 81 of file maximDeviceSpecificUtilities.c.

5.5 src/MAXREFDES71.c File Reference

```
#include "xparameters.h"
#include "stdio.h"
#include "platform.h"
#include "menu.h"
#include "utilities.h"
#include "maximDeviceSpecificUtilities.h"
#include "MAXREFDES71.h"
#include "axi_millbrae.h"
```

Macros

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

Functions

- int [main](#) (void)

Main() function for MAXREFDES71.

Variables

- char [g_sInputString](#) [INPUT_STRING_MAX_SIZE]
- XGpio [g_xGpioLed](#)
- struct [maximOLEDDisplay](#) [g_structureOLED](#)
- char [g_tempString](#) [32]
- u8 [font](#) [1024]
- u32 [g_unSampleSize](#) = 65536
- int [g_nSampleRate](#) = 0

5.5.1 Detailed Description

```
Project: MAXREFDES71#
Filename: MAXREFDES71.c
Description: This module contains the Main application for the
             Avnet Zedboard implementation of the MAXREFDES71
             example program.
```

Revision History:

03-26-13 Rev 01.00 GL Initial Release

This code follows the following naming conventions:

char chPmodValue

char (array) sPmodString[16]

float fPmodValue

int nPmodValue

int (array) anPmodValue[16]

u16 uPmodValue

u16 (array) auPmodValue[16]

u8 uchPmodValue

u8 (array) auchPmodBuffer[16]

unsigned int unPmodValue

int * punPmodValue

Definition in file [MAXREFDES71.c](#).

5.5.2 Macro Definition Documentation

5.5.2.1 #define MAJOR_REVISION 01

Definition at line 85 of file MAXREFDES71.c.

5.5.2.2 `#define MINOR_REVISION 00`

Definition at line 86 of file MAXREFDES71.c.

5.5.3 Function Documentation

5.5.3.1 `int main (void)`

Main() function for MAXREFDES71.

Details

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

Parameters

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

Return values

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

Definition at line 229 of file MAXREFDES71.c.

5.5.4 Variable Documentation

5.5.4.1 `u8 font[1024]`

Definition at line 95 of file MAXREFDES71.c.

5.5.4.2 `int g_nSampleRate = 0`

Definition at line 226 of file MAXREFDES71.c.

5.5.4.3 `char g_sInputString[INPUT_STRING_MAX_SIZE]`

Definition at line 90 of file MAXREFDES71.c.

5.5.4.4 `struct maximOLEDDisplay g_structureOLED`

Definition at line 92 of file MAXREFDES71.c.

5.5.4.5 `char g_tempString[32]`

Definition at line 93 of file MAXREFDES71.c.

5.5.4.6 u32 g_unSampleSize = 65536

Definition at line 225 of file MAXREFDES71.c.

5.5.4.7 XGpio g_xGpioLed

Definition at line 91 of file MAXREFDES71.c.

5.6 src/MAXREFDES71.h File Reference

```
#include "xparameters.h"
#include "stdio.h"
#include "xgpio.h"
#include "xgpio_l.h"
#include "utilities.h"
#include <string.h>
#include "platform.h"
#include "math.h"
```

Data Structures

- struct [maximDateTime](#)
< Structure to hold time, date, day of week
- 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 [INPUT_STRING_MAX_SIZE](#) 16

Variables

- XGpio [g_xGpioLed](#)
- u8 [g_uchAtoDConverterChannel](#)
- u32 [g_unSampleSize](#)
- int [g_nSampleRate](#)
- char [g_sInputString](#) [[INPUT_STRING_MAX_SIZE](#)]
- struct [maximOLEDDisplay](#) [g_structureOLED](#)
- char [g_tempString](#) [32]

5.6.1 Macro Definition Documentation

5.6.1.1 **#define ABOUT_ONE_SECOND 74067512**

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

Definition at line 78 of file MAXREFDES71.h.

5.6.1.2 **#define DEFAULT_HYPERTERMINAL_UART_ADDRESS XPAR_PS7_UART_1_BASEADDR**

Definition at line 76 of file MAXREFDES71.h.

5.6.1.3 **#define DEFAULT_HYPERTERMINAL_UART_ID XPAR_PS7_UART_1_DEVICE_ID**

macro used to abstract Physical Port of Hyperterminal UART

Definition at line 75 of file MAXREFDES71.h.

5.6.1.4 **#define INPUT_STRING_MAX_SIZE 16**

Definition at line 82 of file MAXREFDES71.h.

5.6.2 Variable Documentation

5.6.2.1 **int g_nSampleRate**

Definition at line 226 of file MAXREFDES71.c.

5.6.2.2 **char g_slInputString[INPUT_STRING_MAX_SIZE]**

Definition at line 90 of file MAXREFDES71.c.

5.6.2.3 **struct maximOLEDDisplay g_structureOLED**

Definition at line 92 of file MAXREFDES71.c.

5.6.2.4 **char g_tempString[32]**

Definition at line 93 of file MAXREFDES71.c.

5.6.2.5 **u8 g_uchAtoDConverterChannel**

5.6.2.6 **u32 g_unSampleSize**

Definition at line 78 of file maximDeviceSpecificUtilities.c.

5.6.2.7 XGpio g_xGpioLed

Definition at line 91 of file MAXREFDES71.c.

5.7 src/menu.c File Reference

```
#include <stdio.h>
#include "platform.h"
#include "xparameters.h"
#include "utilities.h"
#include "string.h"
#include "MAXREFDES71.h"
#include "maximDeviceSpecificUtilities.h"
#include "menu.h"
```

Functions

- void [menu_cls](#) ()
Function to clear the screen via Hyperterminal.
- void [menu_print_maxim_banner](#) ()
Print standard Maxim banner at top of Hyperterminal screen.
- void [menu_print_maxim_banner_big](#) ()
Print large Maxim banner at top of Hyperterminal screen.
- void [menu_print_prompt](#) ()
Print a standard prompt for keyboard input "> ".
- void [menu_print_line](#) ()
Print one line of dashes across the screen via Hyperterminal.
- u8 [menu_retrieve_keypress](#) (u32 nUartAddress)
Get a single keypress via Hyperterminal.
- void [menu_print_main_menu](#) ()
Print the main menu listing choice of module to test.
- void [menu_print_ADC_menu](#) ()
Print the ADC conversion menu.
- void [menu_print_channel_menu](#) ()
- void [menu_print_sample_rate_menu](#) ()
Print a menu listing sample speed choices.
- void [menu_print_sample_size_menu](#) ()
Print a menu listing sample size choices.
- void [menu_print_replication_rate_menu](#) ()
Print the replication rate menu listing choice of module to test.

5.7.1 Detailed Description

```
Project: MAXREFDES71#
Filename: menu.c
Description: This module contains all the functions used to
             generate the menus and menu options used to run the
             MAXREFDES71# example firmware.
```

Revision History:

12-17-13 Rev 01.00 GL Initial Release

This code follows the following naming conventions:

char chPmodValue

char (array) sPmodString[16]

float fPmodValue

int nPmodValue

int (array) anPmodValue[16]

u16 uPmodValue

u16 (array) auPmodValue[16]

u8 uchPmodValue

u8 (array) auchPmodBuffer[16]

unsigned int unPmodValue

int * punPmodValue

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

5.7.2 Function Documentation

5.7.2.1 void menu_cls ()

Function to clear the screen via Hyperterminal.

Parameters

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

Return values

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

Definition at line 70 of file menu.c.

5.7.2.2 void menu_print_ADC_menu ()

Print the ADC conversion menu.

Details.**Return values**

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

Definition at line 229 of file menu.c.

5.7.2.3 void menu_print_channel_menu ()

Definition at line 246 of file menu.c.

5.7.2.4 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 155 of file menu.c.

5.7.2.5 void menu_print_main_menu ()

Print the main menu listing choice of module to test.

Details.**Return values**

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

Definition at line 212 of file menu.c.

5.7.2.6 void menu_print_maxim_banner ()

Print standard Maxim banner at top of Hyperterminal screen.

Parameters

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

Return values

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

Definition at line 84 of file menu.c.

5.7.2.7 void menu_print_maxim_banner_big ()

Print large Maxim banner at top of Hyperterminal screen.

Parameters

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

Return values

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

Definition at line 108 of file menu.c.

5.7.2.8 void menu_print_prompt ()

Print a standard prompt for keyboard input ">".

Parameters

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

Return values

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

Definition at line 141 of file menu.c.

5.7.2.9 void menu_print_replication_rate_menu ()

Print the replication rate menu listing choice of module to test.

Details.**Return values**

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

Definition at line 298 of file menu.c.

5.7.2.10 void menu_print_sample_rate_menu ()

Print a menu listing sample speed choices.

Details**Return values**

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

Definition at line 258 of file menu.c.

5.7.2.11 void menu_print_sample_size_menu ()

Print a menu listing sample size choices.

Details**Return values**

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

Definition at line 279 of file menu.c.

5.7.2.12 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 MicroBlaze memory map |
|-----------|---------------------|---|

Return values

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

Definition at line 168 of file menu.c.

5.8 src/menu.h File Reference

```
#include "xbasic_types.h"
#include "stdio.h"
#include "MAXREFDES71.h"
```

Macros

- #define **MAIN_MENU** 0
// Menu state machine state
- #define **WAIT_KEYPRESS** 1
// Menu state machine state
- #define **MENU_SAMPLE_SPEED** 2

```

    // Menu state machine state
• #define MENU_SAMPLE_SIZE 3
    // Menu state machine state
• #define START_CONTINUOUS_SAMPLING 4
    // Menu state machine state
• #define START_BLOCK_SAMPLING 5
    // Menu state machine state
• #define MENU_CHANNEL 6
    // Menu state machine state
• #define ADC_MENU 7
    // Menu state machine state
• #define REP_RATE_MENU 8
    // Menu state machine state
• #define BASE_FUNCTION_STATE 10
    // Menu state machine state
• #define KEYPRESS_ARROW_UP 240
    Assign up-arrow an extended ascii code which won't be used elsewhere.
• #define KEYPRESS_ARROW_DOWN 241
    Assign up-arrow an extended ascii code which won't be used elsewhere.
• #define KEYPRESS_ARROW_LEFT 242
    Assign up-arrow an extended ascii code which won't be used elsewhere.
• #define KEYPRESS_ARROW_RIGHT 243
    Assign up-arrow an extended ascii code which won't be used elsewhere.
• #define KEYPRESS_END 244
    Assign up-arrow an extended ascii code which won't be used elsewhere.

```

Functions

```

• void menu_cls ()
    Function to clear the screen via Hyperterminal.
• void menu_print_maxim_banner ()
    Print standard Maxim banner at top of Hyperterminal screen.
• void menu_print_maxim_banner_big ()
    Print large Maxim banner at top of Hyperterminal screen.
• void menu_print_prompt ()
    Print a standard prompt for keyboard input "> ".
• u8 menu_retrieve_keypress (u32 nUartAddress)
    Get a single keypress via Hyperterminal.
• void menu_print_main_menu ()
    Print the main menu listing choice of module to test.
• void menu_print_ADC_menu ()
    Print the ADC conversion menu.
• void menu_print_channel_menu ()
• void menu_print_sample_rate_menu ()
    Print a menu listing sample speed choices.
• void menu_print_sample_size_menu ()
    Print a menu listing sample size choices.
• void menu_print_replication_rate_menu ()
    Print the replication rate menu listing choice of module to test.

```

5.8.1 Detailed Description

Project: MAXREFDES70#
Filename: menu.h
Description: This module contains all the functions used to generate the menus and menu options used to run the MAXREFDES71# example firmware.

Revision History:

12-17-13 Rev 01.00 GL Initial Release

This code follows the following naming conventions:

char chPmodValue

char (array) sPmodString[16]

float fPmodValue

int nPmodValue

int (array) anPmodValue[16]

u16 uPmodValue

u16 (array) auPmodValue[16]

u8 uchPmodValue

u8 (array) auchPmodBuffer[16]

unsigned int unPmodValue

int * punPmodValue

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

5.8.2 Macro Definition Documentation

5.8.2.1 #define ADC_MENU 7

// Menu state machine state

Definition at line 75 of file menu.h.

5.8.2.2 #define BASE_FUNCTION_STATE 10

// Menu state machine state

Definition at line 77 of file menu.h.

5.8.2.3 #define KEYPRESS_ARROW_DOWN 241

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

Definition at line 80 of file menu.h.

5.8.2.4 #define KEYPRESS_ARROW_LEFT 242

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

Definition at line 81 of file menu.h.

5.8.2.5 #define KEYPRESS_ARROW_RIGHT 243

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

Definition at line 82 of file menu.h.

5.8.2.6 #define KEYPRESS_ARROW_UP 240

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

Definition at line 79 of file menu.h.

5.8.2.7 #define KEYPRESS_END 244

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

Definition at line 83 of file menu.h.

5.8.2.8 #define MAIN_MENU 0

// Menu state machine state

Definition at line 68 of file menu.h.

5.8.2.9 #define MENU_CHANNEL 6

// Menu state machine state

Definition at line 74 of file menu.h.

5.8.2.10 #define MENU_SAMPLE_SIZE 3

// Menu state machine state

Definition at line 71 of file menu.h.

5.8.2.11 #define MENU_SAMPLE_SPEED 2

// Menu state machine state

Definition at line 70 of file menu.h.

5.8.2.12 #define REP_RATE_MENU 8

// Menu state machine state

Definition at line 76 of file menu.h.

5.8.2.13 #define START_BLOCK_SAMPLING 5

// Menu state machine state

Definition at line 73 of file menu.h.

5.8.2.14 #define START_CONTINUOUS_SAMPLING 4

// Menu state machine state

Definition at line 72 of file menu.h.

5.8.2.15 #define WAIT_KEYPRESS 1

// Menu state machine state

Definition at line 69 of file menu.h.

5.8.3 Function Documentation**5.8.3.1 void menu_cls ()**

Function to clear the screen via Hyperterminal.

Parameters

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

Return values

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

Definition at line 70 of file menu.c.

5.8.3.2 void menu_print_ADC_menu ()

Print the ADC conversion menu.

Details.**Return values**

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

Definition at line 229 of file menu.c.

5.8.3.3 void menu_print_channel_menu ()

Definition at line 246 of file menu.c.

5.8.3.4 void menu_print_main_menu ()

Print the main menu listing choice of module to test.

Details.

Return values

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

Definition at line 212 of file menu.c.

5.8.3.5 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 84 of file menu.c.

5.8.3.6 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 108 of file menu.c.

5.8.3.7 void menu_print_prompt ()

Print a standard prompt for keyboard input ">".

Parameters

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

Return values

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

Definition at line 141 of file menu.c.

5.8.3.8 void menu_print_replication_rate_menu ()

Print the replication rate menu listing choice of module to test.

Details.**Return values**

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

Definition at line 298 of file menu.c.

5.8.3.9 void menu_print_sample_rate_menu ()

Print a menu listing sample speed choices.

Details**Return values**

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

Definition at line 258 of file menu.c.

5.8.3.10 void menu_print_sample_size_menu ()

Print a menu listing sample size choices.

Details**Return values**

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

Definition at line 279 of file menu.c.

5.8.3.11 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

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

Return values

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

Definition at line 168 of file menu.c.

5.9 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` ()

5.9.1 Macro Definition Documentation

5.9.1.1 `#define UART_BAUD 9600`

Definition at line 30 of file platform.c.

5.9.2 Function Documentation

5.9.2.1 void `cleanup_platform` ()

Definition at line 78 of file platform.c.

5.9.2.2 void disable_caches ()

Definition at line 49 of file platform.c.

5.9.2.3 void enable_caches ()

Definition at line 33 of file platform.c.

5.9.2.4 void init_platform ()

Definition at line 71 of file platform.c.

5.9.2.5 void init_uart ()

Definition at line 56 of file platform.c.

5.10 src/platform.h File Reference

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

Functions

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

5.10.1 Function Documentation**5.10.1.1 void cleanup_platform ()**

Definition at line 78 of file platform.c.

5.10.1.2 void init_platform ()

Definition at line 71 of file platform.c.

5.11 src/platform_config.h File Reference**5.12 src/utilities.c File Reference**

```
#include "xparameters.h"  
#include "stdio.h"  
#include "utilities.h"  
#include "MAXREFDES71.h"
```

Macros

- #define `UART_BASEADDR` `XPAR_XUARTPS_0_BASEADDR`
- #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 `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.
- int `number_raised_to_power` (int nBase, int nExponent)
Raise nBase to the nExponent power (operates with integers only).
- void `print_asterisks` (int nQuantity)
Print nQuantity of asterisks to the default Hyperterminal UART.
- u8 `getUartByte` (u32 nUartAddress)
Get a byte from either the full UART in the Zynq PS, or a Xilinx UartLite.
- 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.
- void `initializeOLED` (u8 *pFont)
Initializes the OLED display.
- void `sendOLEDSPI` (u8 uchDataToWrite)
Bit bang routine for Zedboard SPI OLED interface.
- 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.

5.12.1 Detailed Description

```

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

```

Revision History:

4-13-12 Rev 1.0 Seth Messimer Initial Release

7-20-12 Rev 1.4 Nathan Young Additional functions

This code follows the following naming conventions.

```

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

```

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

5.12.2 Macro Definition Documentation

5.12.2.1 **#define OLED_DATA_COMMAND_B 0x04**

Definition at line 71 of file [utilities.c](#).

5.12.2.2 **#define OLED_RESET_B 0x08**

Definition at line 70 of file [utilities.c](#).

5.12.2.3 **#define OLED_SCLK 0x01**

Definition at line 73 of file [utilities.c](#).

5.12.2.4 **#define OLED_SDIN 0x02**

Definition at line 72 of file [utilities.c](#).

5.12.2.5 **#define OLED_VBAT 0x20**

Definition at line 68 of file [utilities.c](#).

5.12.2.6 **#define OLED_VDD 0x10;**

Definition at line 69 of file [utilities.c](#).

5.12.2.7 **#define UART_BASEADDR XPAR_XUARTPS_0_BASEADDR**

Definition at line 66 of file [utilities.c](#).

5.12.3 Function Documentation

5.12.3.1 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

| | | |
|----|----------------|---------------------|
| in | unUartAddress. | 32 bit UART address |
|----|----------------|---------------------|

Return values

| | |
|------|-------------------------|
| True | if empty (false if not) |
|------|-------------------------|

Definition at line 215 of file utilities.c.

5.12.3.2 void clearOLEDBuffer (u8 * pauchBuffer)

Clears the OLED display buffer.

Details

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

Parameters

| | | |
|----|--------------|---|
| in | *pauchBuffer | - pointer to an u8 array used for the OLED pixel buffer |
|----|--------------|---|

Return values

| | |
|------|--|
| None | |
|------|--|

Definition at line 415 of file utilities.c.

5.12.3.3 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 | nStopValue | - number of iterations to loop |
|----|------------|--------------------------------|

Return values

| | |
|------|--|
| None | |
|------|--|

Definition at line 75 of file *utilities.c*.

5.12.3.4 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 432 of file *utilities.c*.

5.12.3.5 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>*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 586 of file *utilities.c*.

5.12.3.6 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 174 of file utilities.c.

5.12.3.7 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

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

Return values

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

Definition at line 252 of file utilities.c.

5.12.3.8 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

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

Return values

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

Definition at line 96 of file utilities.c.

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

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

Details

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

Parameters

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

Return values

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

Definition at line 131 of file utilities.c.

5.12.3.10 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 160 of file utilities.c.

5.12.3.11 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 527 of file utilities.c.

5.12.3.12 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 567 of file utilities.c.

5.12.3.13 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 495 of file utilities.c.

5.12.3.14 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 382 of file utilities.c.

5.12.3.15 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 195 of file utilities.c.

5.13 src/utilities.h File Reference

```
#include "xparameters.h"
#include "xbasic_types.h"
#include "stdio.h"
#include "xgpio.h"
#include "xgpio_l.h"
#include "MAXREFDES71.h"
```

Functions

- 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.
- int [number_raised_to_power](#) (int nBase, int nExponent)
Raise nBase to the nExponent power (operates with integers only).
- void [print_asterisks](#) (int nQuantity)
Print nQuantity of asterisks to the default Hyperterminal UART.
- u8 [getUartByte](#) (u32 nUartAddress)
Get a byte from either the full UART in the Zynq PS, or a Xilinx UartLite.
- 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.
- void [initializeOLED](#) (u8 *pFont)
Initializes the OLED display.

- void [sendOLEDSPI](#) (u8 uchDataToWrite)
Bit bang routine for Zedboard SPI OLED interface.
- 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.

Variables

- struct [maximDateTime](#) * t

5.13.1 Detailed Description

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

Revision History:

4-13-12 Rev 1.0 Seth Messimer Initial Release

7-20-12 Rev 1.4 Nathan Young Additional functions

This code follows the following naming conventions.

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

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

5.13.2 Function Documentation

5.13.2.1 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 215 of file utilities.c.

5.13.2.2 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 415 of file utilities.c.

5.13.2.3 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 75 of file utilities.c.

5.13.2.4 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

| |
|------|
| None |
|------|

Definition at line 432 of file utilities.c.

5.13.2.5 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

| |
|------|
| None |
|------|

Definition at line 586 of file utilities.c.

5.13.2.6 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

| | |
|---------|-------------------------------------|
| Returns | the next byte from the Uart Rx FIFO |
|---------|-------------------------------------|

Definition at line 174 of file utilities.c.

5.13.2.7 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 252 of file utilities.c.

5.13.2.8 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 MicroBlaze memory map |
|----|-------------------|--|

Return values

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

Definition at line 96 of file utilities.c.

5.13.2.9 int number_raised_to_power (int nBase, int nExponent)

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

Details

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

Parameters

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

Return values

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

Definition at line 131 of file utilities.c.

5.13.2.10 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

| |
|------|
| None |
|------|

Definition at line 160 of file utilities.c.

5.13.2.11 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 a null terminated character string. Should be generated with sprintf (or similar). |

Return values

| |
|------|
| None |
|------|

Definition at line 527 of file utilities.c.

5.13.2.12 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 a null terminated character string. Should be generated with sprintf (or similar). |

Return values

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

Definition at line 567 of file utilities.c.

5.13.2.13 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

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

Return values

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

Definition at line 495 of file utilities.c.

5.13.2.14 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 382 of file utilities.c.

5.13.2.15 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 195 of file utilities.c.

5.13.3 Variable Documentation

5.13.3.1 struct `maximDateTime`* t

Definition at line 76 of file utilities.h.