

DS5203 FPGA Board

RTLlib Reference

Release 2021-A – May 2021

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	info@dspace.de
Web:	http://www.dspace.com

How to Contact dSPACE Support

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: <http://www.dspace.com/go/locations>
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.
Tel.: +49 5251 1638-941 or e-mail: support@dspace.de

You can also use the support request form: <http://www.dspace.com/go/supportrequest>. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the relevant dSPACE License ID or the serial number of the CmContainer in your support request.

Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit <http://www.dspace.com/go/patches> for software updates and patches.

Important Notice

This publication contains proprietary information that is protected by copyright. All rights are reserved. The publication may be printed for personal or internal use provided all the proprietary markings are retained on all printed copies. In all other cases, the publication must not be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of dSPACE GmbH.

© 2009 - 2021 by:
dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

This publication and the contents hereof are subject to change without notice.

AUTERA, ConfigurationDesk, ControlDesk, MicroAutoBox, MicroLabBox, SCALEXIO, SIMPHERA, SYNECT, SystemDesk, TargetLink and VEOS are registered trademarks of dSPACE GmbH in the United States or other countries, or both. Other brand names or product names are trademarks or registered trademarks of their respective companies or organizations.

Contents

About This Reference	5
General Information on the DS5203 Real-Time Library	7
Overview of the DS5203 Real-Time Library.....	7
Data Types	9
ds5203_prgr_dat.....	9
Macros	13
Base Address of the I/O Board.....	13
Board Initialization	15
ds5203_init.....	15
Identification Functions	19
ds5203_get_board_rev.....	19
ds5203_get_fw_id.....	20
ds5203_get_pb_id.....	22
ds5203_get_appl_id.....	23
ds5203_get_appl_id_string.....	24
Interrupt Functions	27
ds5203_enable_int.....	27
ds5203_disable_int.....	29
ds5203_ack_int.....	30
ds5203_pending_int.....	31
ds5203_read_int.....	33
Data Exchange Functions	35
ds5203_read_reg.....	35
ds5203_write_reg.....	37
ds5203_read_reg_grp.....	39

ds5203_write_reg_grp.....	41
ds5203_read_buf.....	43
ds5203_write_buf.....	45
 FPGA Programming Functions	 47
ds5203_program.....	47
 Function Execution Times	 53
Information on the Test Environment.....	53
Measured Execution Times.....	54
 Index	 57

About This Reference

Content

This RTLib Reference (Real-Time Library) gives detailed descriptions of the C functions needed to program a DS5203 FPGA Board. The C functions can be used to program RTI-specific Simulink S-functions, or to implement your control models manually using C programs.

Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
	Indicates a hazard that, if not avoided, could result in property damage.
	Indicates important information that you should take into account to avoid malfunctions.
	Indicates tips that can make your work easier.
	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

Special folders

Some software products use the following special folders:

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>

or

%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>

Documents folder A standard folder for user-specific documents.

%USERPROFILE%\Documents\dSPACE\<ProductName>\<VersionNumber>

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\<ProductName>

Accessing dSPACE Help and PDF Files


After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com.

To access the Web version, you must have a *mydSPACE* account.

PDF files You can access PDF files via the  icon in dSPACE Help. The PDF opens on the first page.

General Information on the DS5203 Real-Time Library

Introduction

The DS5203 Real-Time Library provides the C functions which make the capabilities of the DS5203 FPGA Board available for RTI-specific Simulink S-functions or for handcoded control models.

Overview of the DS5203 Real-Time Library

Introduction

The DS5203 Real-Time Library provides C functions to perform FPGA communication access for various use cases.

Basics

The DS5203 FPGA Board provides a Xilinx® FPGA that allows you to execute applications separately from the processor application at a very high sample rate. To integrate the FPGA application into the dSPACE real-time environment, you must use the template files of the handcode interface or the RTI FPGA Programming Blockset. It provides Simulink blocks to model the data exchange between the FPGA and the external I/O that you connect to the DS5203 FPGA Board, and between the FPGA and the processor application. You can use the blockset to implement the communication on the FPGA side and the processor side.

With the RTLib functions, you can only implement the communication on the processor side and program the FPGA. You cannot implement the FPGA application via RTLib. An FPGA application must exist to run a handcoded RTLib application.

Information on the RTLib functions

Board initialization Before the initialization of the DS5203 FPGA Board, the processor board must be initialized and an FPGA application must be running. Because the board initialization is terminated if there is no FPGA application

running, one must be programmed beforehand or loaded to the flash memory of the DS5203 FPGA Board. Board initialization succeeds only if the FPGA application is compatible with the processor application.

Identification To avoid hardware damage, the components used (processor application, FPGA application, FPGA framework, piggyback module) must be compatible with each other. To check their compatibility, their identifiers can be read and compared using the identification functions.

Interrupt handling The DS5203 FPGA Board provides 8 interrupt channels which you can handle using the RTLib interrupt functions.

Data exchange With the RTLib, you can implement the processor's read and write access to the data storage that is defined in the FPGA framework. Data exchange with the FPGA application requires data type conversion, which is configured by the `scaling` and `mode` parameters.

The DS5203 FPGA Board provides the following channel types and channel numbers:

- 32 Buffer In channels
- 32 Buffer Out channels
- 128 Register In channels
- 128 Register Out channels

The maximum number of elements in a buffer is specified in the FPGA framework. The configuration of a buffer is valid for all its elements. Buffers are accessed sequentially from the FPGA.

A register has a data width of 32 bits. You can specify groups of registers, which can be accessed synchronously by the FPGA. Each register in a group is configured separately.

Programming The FPGA application that you build, must be programmed to the FPGA. You can do this by loading it into the flash memory of the DS5203 FPGA Board or into the RAM of the FPGA. While the RAM of the FPGA must be programmed on each power up, the flash application is automatically loaded to the FPGA when you power up the board.

The FPGA application is represented by the FPGA model INI file that contains the bitstream and further relevant information. When you program the FPGA you must specify whether the bitstream you want to load is in compressed form.

Related topics

References

Board Initialization.....	15
Data Exchange Functions.....	35
FPGA Programming Functions.....	47
Identification Functions.....	19
Interrupt Functions.....	27

Data Types

Introduction

RTLib5203 provides a specific data structure for the programming options.

ds5203_prgr_dat

Syntax

```
typedef struct
{
    UInt32 model_blkst_major
    UInt32 model_blkst_minor
    UInt32 model_blkst_maintenance
    UInt8 *bitstream_pt
    UInt32 bitstream_length
    UInt32 bitstream_length_unpckd
    UInt32 *compatible_boards_list_maj_pt
    UInt32 *compatible_boards_list_min_pt
    UInt32 compatible_boards_list_length
    UInt32 fw_id
    UInt32 mode
} ds5203_prgr_dat;
```

Include file

ds5203.h

Purpose

To specify the characteristics of the programming function.

Description

The structure defines the version of the RTI FPGA Programming Blockset used, the length of the bitstream in compressed and uncompressed format, a list of compatible board revisions, the identifier of the FPGA framework and the programming mode. The created model INI file contains a C function that provides the preconfigured data structure. It is not necessary to edit the data structure manually.

Members

model_blkst_major Defines the major version of the RTI FPGA Programming Blockset with which the model INI file was created.

model_blkst_minor Defines the minor version of the RTI FPGA Programming Blockset with which the model INI file was created.

model_blkst_maintenance Defines the maintenance version of the RTI FPGA Programming Blockset with which the model INI file was created.

bitstream_pt Defines the address of variable containing the bitstream from the implemented FPGA application.

bitstream_length Defines the length of the compressed bitstream in bytes.

bitstream_length_unpckd Defines the length of the decompressed bitstream. It is only evaluated if the specified bitstream is in compressed mode.

compatible_boards_list_maj_pt Defines the address of the list where the major versions of the compatible boards are stored in.

compatible_boards_list_min_pt Defines the address of the list where the minor versions of the compatible boards are stored in.

compatible_boards_list_length Defines the number of boards in the list.

fw_id Defines the identifier of the FPGA framework. This must be the identifier that you get when you use the **ds5203_get_fw_id** function in the running FPGA application.

mode Defines the mode for programming the DS5203 FPGA Board. You can combine the following predefined symbols by using the logical operator OR.

Symbol	Meaning
DS5203_PRGRM_COM	To decompress the bitstream before loading, if it is compressed. It is represented by 0x04 in the model INI file.
DS5203_PRGRM_SRM	To load the FPGA application to the RAM/SRAM of the FPGA. Because the RAM is a volatile buffer, you must load the application at each board startup. It is represented by 0x02 in the model INI file.
DS5203_PRGRM_FLSH	To store the FPGA application in the flash memory of the DS5203 FPGA Board. When you start the board, the FPGA application is automatically loaded to the FPGA. It is represented by 0x01 in the model INI file.
DS5203_PRGRM_DBGOUT	To enable information messages about the programming state. It is represented by 0x08 in the model INI file.
DS5203_PRGRM_FRCE_PRGR	To disable the version checks (force programming). It is recommended not to use this programming mode because of an unexpected behavior of the FPGA application. It is represented by 0x10 in the model INI file.

For an example, how to use the data structure, refer to [ds5203_program](#) on page 47.

Related topics

References

ds5203_program.....	47
-------------------------------------	--------------------

Macros

Introduction

The base address of an I/O board in a PHS-bus-based system has to be defined by using the `DSxxxx_n_BASE` macro.

Base Address of the I/O Board

DSxxxx_n_BASE macros

When using I/O board functions, you always need the board's base address as a parameter. This address can easily be obtained by using the `DSxxxx_n_BASE` macros, where `DSxxxx` is the board name (for example, `DS2001`) and `n` is an index which counts boards of the same type. The board with the lowest base address is given index 1. The other boards of the same type are given consecutive numbers in order of their base addresses. The macros refer to an internal data structure which holds the addresses of all I/O boards in the system. The initialization function of the processor board (named `init`) creates this data structure. Hence, when you change an I/O board base address, it is not necessary to recompile the code of your application. For more information on the processor board's initialization function, refer to [DS1006 RTLib Reference](#) or [DS1007 RTLib Reference](#).

Note

The `DSxxxx_n_BASE` macros can be used only after the processor board's initialization function `init` is called.

Example

This example demonstrates the use of the `DSxxxx_n_BASE` macros. There are two `DS2001` boards, two `DS2101` boards, and one `DS2002` board connected to a PHS bus. Their base addresses have been set to distinct addresses. The following table shows the I/O boards, their base addresses, and the macros which can be used as base addresses:

Board	Base Address	Macro
DS2001	00H	DS2001_1_BASE
DS2002	20H	DS2002_1_BASE
DS2101	80H	DS2101_1_BASE
DS2001	90H	DS2001_2_BASE
DS2101	A0H	DS2101_2_BASE

Board Initialization

Introduction

Before you can use the DS5203 FPGA Board, you have to perform the initialization process.

Note

- The processor board must be initialized before you can initialize the DS5203 FPGA Board.
- It is assumed that an FPGA application is already loaded when you call the board initialization function. If the FPGA application is not already programmed into the flash memory of the FPGA, you must use **ds5203_program**.

ds5203_init

Syntax

```
void ds5203_init(
    phs_addr_t base,
    UInt64* fpga_appl_id)
```

Include file

ds5203.h

Purpose

To initialize the DS5203 FPGA Board with default settings.

Description

This function initializes the DS5203 FPGA Board:

- PHS++ mode is enabled
- Checks whether the piggyback module matches the DS5203 FPGA firmware version.

- Checks whether the processor application is compatible with the FPGA application.
- Enables background detection of DS5203 hardware and firmware errors.

Note

An FPGA application must be loaded before calling this function.

Parameters

base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to [Base Address of the I/O Board](#) on page 13).

fpga_appl_id Specifies the pointer to the variable containing the identifier of the FPGA application that is generated by the FPGA build process.

Return value

None

Messages

The following messages are defined:

ID	Type	Message	Description
201	Error	ds5203_init(): Invalid PHS-bus base address 0x??	The value of the base parameter is not a valid PHS-bus address. This error can occur if the PHS-bus connection of the I/O board is missing. Check the connection.
300	Error	ds5203_init(0x??): Board not found!	No DS5203 board could be found at the specified PHS-bus address.
301	Error	ds5203_init(0x??): Board is not responding! Hardware reset failed.	The DS5203 board has not booted after power-up.
302	Error	ds5203_init(0x??): No FPGA application loaded.	The board's FPGA is not configured with an application.
303	Error	ds5203_init(0x??): FPGA is not responding! FPGA application initialization failed.	The board's FPGA was configured but failed to boot and initialize the FPGA application.
304	Error	ds5203_init(0x??): Piggyback family 0x??? is not matching DS5203 FPGA framework family 0x???!	The piggyback family does not match the DS5203 FPGA framework family.
305	Error	ds5203_init(0x??): Piggyback revision 0x??? is not matching DS5203 FPGA framework revision 0x???	The piggyback revision does not match the DS5203 FPGA framework revision.
306	Error	ds5203_init(0x??): FPGA application ID 0x??? expected by processor application does not match ID 0x??? of running FPGA application!	The process and FPGA applications are incompatible due to non-matching FPGA application IDs.
307	Error	ds5203_init(0x??): Fatal hardware error of DS5203 board!	The DS5203 status indicates a fatal hardware error.

Execution Times

For information on the execution times, refer to [Function Execution Times](#) on page 53.

Example

This example shows how to initialize the first DS5203 FPGA Board connected to the PHS bus:

```
void main(void)
{
    UInt64 fpga_appl_id = {0x48C01235, 0x592DC2};

    init();
    ds5203_init(DS5203_1_BASE, &fpga_appl_id);
    ...
}
```

Related topics**References**

[Overview of the DS5203 Real-Time Library.....7](#)

Identification Functions

Introduction Functions to get the identifiers of the various components associated with the DS5203 FPGA Board.

Where to go from here

Information in this section

ds5203_get_board_rev.....	19
To read the DS5203 FPGA Board revision.	
ds5203_get_fw_id.....	20
To read the FPGA framework identifier.	
ds5203_get_pb_id.....	22
To read the FPGA piggyback identifier.	
ds5203_get_appl_id.....	23
To read the FPGA application identifier.	
ds5203_get_appl_id_string.....	24
To read the description of the running FPGA application.	

ds5203_get_board_rev

Syntax

```
void ds5203_get_board_rev(
    phs_addr_t base,
    UInt32* major,
    UInt32* minor)
```

Include file

ds5203.h

Purpose	To read the DS5203 FPGA Board revision.										
Description	This function reads out the revision of the DS5203 FPGA Board. It consists of a major and a minor revision number.										
Parameters	<p>base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to Base Address of the I/O Board on page 13).</p> <p>major Specifies the pointer to the variable containing the major revision number of the DS5203 FPGA Board.</p> <p>minor Specifies the pointer to the variable containing the minor revision number of the DS5203 FPGA Board.</p>										
Return value	None										
Execution Times	For information on the execution times, refer to Function Execution Times on page 53.										
Example	<p>This example shows how to get the major and minor board revision numbers:</p> <pre>ds5203_get_board_rev(DS5203_1_BASE, &major, &minor);</pre>										
Related topics	<p>References</p> <table> <tr> <td>ds5203_get_appl_id.....</td><td>23</td></tr> <tr> <td>ds5203_get_appl_id_string.....</td><td>24</td></tr> <tr> <td>ds5203_get_fw_id.....</td><td>20</td></tr> <tr> <td>ds5203_get_pb_id.....</td><td>22</td></tr> <tr> <td>ds5203_init.....</td><td>15</td></tr> </table>	ds5203_get_appl_id	23	ds5203_get_appl_id_string	24	ds5203_get_fw_id	20	ds5203_get_pb_id	22	ds5203_init	15
ds5203_get_appl_id	23										
ds5203_get_appl_id_string	24										
ds5203_get_fw_id	20										
ds5203_get_pb_id	22										
ds5203_init	15										

ds5203_get_fw_id

Syntax

```
void ds5203_get_fw_id(
    phs_addr_t base,
    UInt32* family,
    UInt32* major,
    UInt32* minor)
```

Include file	ds5203.h
Purpose	To read the FPGA framework identifier.
Description	<p>This function reads the identifier of the FPGA framework. It consists of a family identification number, the major revision number and the minor revision number. The framework corresponds to the firmware of the FPGA. For information on the framework, refer to Overview of the DS5203 Real-Time Library on page 7.</p> <div> NOTICE To avoid hardware damage, you must guarantee the compatibility of the piggyback module with the loaded framework (see ds5203_get_pb_id on page 22). </div>
Parameters	<p>base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to Base Address of the I/O Board on page 13).</p> <p>family Specifies the pointer to the variable containing the family identification number of the FPGA framework.</p> <p>major Specifies the pointer to the variable containing the major revision number of the FPGA framework.</p> <p>minor Specifies the pointer to the variable containing the minor revision number of the FPGA framework.</p>
Return value	None
Execution Times	For information on the execution times, refer to Function Execution Times on page 53.
Example	<p>This example shows how to get the framework identifier:</p> <pre>ds5203_get_fw_id(DS5203_1_BASE, &family, &major, &minor);</pre>

Related topics

References

ds5203_get_appl_id	23
ds5203_get_appl_id_string	24
ds5203_get_board_rev	19
ds5203_get_pb_id	22
ds5203_init	15

ds5203_get_pb_id

Syntax

```
void ds5203_get_pb_id(
    phs_addr_t base,
    UInt32* family,
    UInt32* major,
    UInt32* minor)
```

Include file

ds5203.h

Purpose

To read the FPGA piggyback identifier.

Description

This function reads the identifier of the piggyback module that is mounted on the DS5203 FPGA Board.

NOTICE

To avoid hardware damage, you must guarantee the compatibility of the piggyback module with the loaded framework (see [ds5203_get_fw_id](#) on page 20).

Parameters

base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to [Base Address of the I/O Board](#) on page 13).

family Specifies the pointer to the variable containing the family identifier of the piggyback module.

major Specifies the pointer to the variable containing the major revision number of the piggyback module.

minor Specifies the pointer to the variable containing the minor revision number of the piggyback module.

Return value	None										
Execution Times	For information on the execution times, refer to Function Execution Times on page 53.										
Example	<p>This example shows how to get the piggyback identifier:</p> <pre>ds5203_get_pb_id(DS5203_1_BASE, &family, &major, &minor);</pre>										
Related topics	<p>References</p> <table> <tr> <td>ds5203_get_appl_id.....</td><td>23</td></tr> <tr> <td>ds5203_get_appl_id_string.....</td><td>24</td></tr> <tr> <td>ds5203_get_board_rev.....</td><td>19</td></tr> <tr> <td>ds5203_get_fw_id.....</td><td>20</td></tr> <tr> <td>ds5203_init.....</td><td>15</td></tr> </table>	ds5203_get_appl_id	23	ds5203_get_appl_id_string	24	ds5203_get_board_rev	19	ds5203_get_fw_id	20	ds5203_init	15
ds5203_get_appl_id	23										
ds5203_get_appl_id_string	24										
ds5203_get_board_rev	19										
ds5203_get_fw_id	20										
ds5203_init	15										

ds5203_get_appl_id

Syntax	<pre>void ds5203_get_appl_id(phs_addr_t base, UInt64* appl_id)</pre>
Include file	ds5203.h
Purpose	To read the FPGA application identifier.
Description	This function reads the identifier of the running FPGA application.
Parameters	<p>base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to Base Address of the I/O Board on page 13).</p> <p>appl_id Specifies the pointer to the variable containing the identifier of the FPGA application.</p>
Return value	None

Execution Times	For information on the execution times, refer to Function Execution Times on page 53.
------------------------	---

Example	This example shows how to get the identifier of the running application:
----------------	--

```
ds5203_get_appl_id(DS5203_1_BASE, &appl_id);
```

Related topics**References**

ds5203_get_appl_id_string	24
ds5203_get_board_rev	19
ds5203_get_fw_id	20
ds5203_get_pb_id	22
ds5203_init	15

ds5203_get_appl_id_string

Syntax

```
void ds5203_get_appl_id_string(  
    phs_addr_t base,  
    char* id_string)
```

Include file	ds5203.h
---------------------	----------

Purpose	To read the description of the running FPGA application.
----------------	--

Description	This function reads a string that describes the running FPGA application.
--------------------	---

Parameters	<p>base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to Base Address of the I/O Board on page 13).</p> <p>id_string Specifies the pointer to the variable containing the description of the FPGA application. It can have a size of 256 characters.</p>
-------------------	---

Return value	None
---------------------	------

Execution Times	For information on the execution times, refer to Function Execution Times on page 53.
------------------------	---

Example

This example shows how to get the description of the running FPGA application:

```
...  
char id_string[256];  
...  
ds5203_get_appl_id_string(DS5203_1_BASE, id_string);
```

Related topics**References**

ds5203_get_appl_id.....	23
ds5203_get_board_rev.....	19
ds5203_get_fw_id.....	20
ds5203_get_pb_id.....	22
ds5203_init.....	15

Interrupt Functions

Introduction Functions to access the interrupt sources of the DS5203 FPGA Board.

Where to go from here

Information in this section

ds5203_enable_int.....	27
To enable the interrupt channels of the DS5203 FPGA Board.	
ds5203_disable_int.....	29
To disable the interrupt channels of the DS5203 FPGA Board.	
ds5203_ack_int.....	30
To acknowledge the pending interrupts of the DS5203 FPGA Board.	
ds5203_pending_int.....	31
To query if interrupts are pending on the DS5203 FPGA Board.	
ds5203_read_int.....	33
To read the interrupt source register of the DS5203 FPGA Board.	

ds5203_enable_int

Syntax

```
void ds5203_enable_int(
    phs_addr_t base,
    UInt32 int_src)
```

Include file

ds5203.h

Purpose

To enable the interrupt channels of the DS5203 FPGA Board.

Description

The DS5203 FPGA Board provides 8 interrupt channels that you must enable before you can use them as interrupt sources. Each channel can be enabled separately.

Parameters

base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to [Base Address of the I/O Board](#) on page 13).

int_src Specifies the bit mask of interrupt channels to be enabled. If you want to enable more than one interrupt channel, you can combine the following predefined symbols by using the logical operator OR.

Symbols	Meaning
DS5203_INT_SRC_1	Interrupt on channel 1
DS5203_INT_SRC_2	Interrupt on channel 2
DS5203_INT_SRC_3	Interrupt on channel 3
DS5203_INT_SRC_4	Interrupt on channel 4
DS5203_INT_SRC_5	Interrupt on channel 5
DS5203_INT_SRC_6	Interrupt on channel 6
DS5203_INT_SRC_7	Interrupt on channel 7
DS5203_INT_SRC_8	Interrupt on channel 8

Return value

None

Execution Times

For information on the execution times, refer to [Function Execution Times](#) on page 53.

Example

This example shows how to register an interrupt service routine and how to enable the interrupt on channel 1:

```
/* initialize and enable DS5203 PHS-bus interrupts */
error = install_phs_int_vector(DS5203_1_BASE, 0, isr_0);
RTLIB_INT_ENABLE();
ds5203_enable_int(DS5203_1_BASE, DS5203_INT_SRC_1);
```

Related topics**References**

ds5203_ack_int	30
ds5203_disable_int	29
ds5203_init	15
ds5203_pending_int	31
ds5203_read_int	33

ds5203_disable_int

Syntax

```
void ds5203_disable_int(
    phs_addr_t base,
    UInt32 int_src)
```

Include file

ds5203.h

Purpose

To disable the interrupt channels of the DS5203 FPGA Board.

Description

The DS5203 FPGA Board provides 8 interrupt channels. If you have enabled them by using the **ds5203_enable_int** function, you can disable them again.

Parameters

base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to [Base Address of the I/O Board](#) on page 13).

int_src Specifies the bit mask of interrupt channels to be disabled. If you want to disable more than one interrupt channel, you can combine the following predefined symbols by using the logical operator OR.

Symbols	Meaning
DS5203_INT_SRC_1	Interrupt on channel 1
DS5203_INT_SRC_2	Interrupt on channel 2
DS5203_INT_SRC_3	Interrupt on channel 3
DS5203_INT_SRC_4	Interrupt on channel 4
DS5203_INT_SRC_5	Interrupt on channel 5
DS5203_INT_SRC_6	Interrupt on channel 6
DS5203_INT_SRC_7	Interrupt on channel 7
DS5203_INT_SRC_8	Interrupt on channel 8

Return value

None

Execution Times

For information on the execution times, refer to [Function Execution Times](#) on page 53.

Example

This example shows how to disable the interrupts of channels 1 and 3:

```
ds5203_disable_int(DS5203_1_BASE, DS5203_INT_SRC_1|DS5203_INT_SRC_3);
```

Related topics

References

ds5203_ack_int	30
ds5203_enable_int	27
ds5203_init	15
ds5203_pending_int	31
ds5203_read_int	33

ds5203_ack_int

Syntax

```
__INLINE void ds5203_ack_int(  
    phs_addr_t base,  
    UInt32 int_src)
```

Include file

ds5203.h

Purpose

To acknowledge the pending interrupts of the DS5203 FPGA Board.

Description

This function acknowledges the interrupt on the specified channel(s). The acknowledgement resets the associated flag in the interrupt flag register. An interrupt that has been evaluated must always be acknowledged.

Note

The **ds5203_enable_int** function must be called before interrupts can be generated and acknowledged.

Parameters

base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to [Base Address of the I/O Board](#) on page 13).

int_src Specifies the bit mask of the interrupt channels to be acknowledged. If you want to acknowledge more than one interrupt channel, you can combine the following predefined symbols by using the logical operator OR.

Symbols	Meaning
DS5203_INT_SRC_1	Interrupt on channel 1
DS5203_INT_SRC_2	Interrupt on channel 2
DS5203_INT_SRC_3	Interrupt on channel 3
DS5203_INT_SRC_4	Interrupt on channel 4

Symbols	Meaning
DS5203_INT_SRC_5	Interrupt on channel 5
DS5203_INT_SRC_6	Interrupt on channel 6
DS5203_INT_SRC_7	Interrupt on channel 7
DS5203_INT_SRC_8	Interrupt on channel 8

Return value None

Execution Times For information on the execution times, refer to [Function Execution Times](#) on page 53.

Example This example shows how to acknowledge the interrupts on channels 1 and 3:

```
ds5203_ack_int(DS5203_1_BASE, DS5203_INT_SRC_1|DS5203_INT_SRC_3);
```

Related topics

References

ds5203_disable_int	29
ds5203_enable_int	27
ds5203_init	15
ds5203_pending_int	31
ds5203_read_int	33

ds5203_pending_int

Syntax

```
__INLINE UInt32 ds5203_pending_int(
    phs_addr_t base,
    UInt32 int_src)
```

Include file ds5203.h

Purpose To query if interrupts are pending on the DS5203 FPGA Board.

Description This function returns the pending status for the specified interrupt channels. An interrupt is pending if it was requested but not yet acknowledged by the program. If the function returns 0, no interrupt is pending.

Note

The `ds5203_enable_int` function must be called before interrupts can be generated and acknowledged.

Parameters

base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to [Base Address of the I/O Board](#) on page 13).

int_src Specifies the bit mask of interrupt channels to be checked for pending status. If you want to check more than one interrupt channel, you can combine the following predefined symbols by using the logical operator OR.

Symbols	Meaning
DS5203_INT_SRC_1	Interrupt on channel 1
DS5203_INT_SRC_2	Interrupt on channel 2
DS5203_INT_SRC_3	Interrupt on channel 3
DS5203_INT_SRC_4	Interrupt on channel 4
DS5203_INT_SRC_5	Interrupt on channel 5
DS5203_INT_SRC_6	Interrupt on channel 6
DS5203_INT_SRC_7	Interrupt on channel 7
DS5203_INT_SRC_8	Interrupt on channel 8

Return value

This function returns:

- 0 if no interrupt channel is pending.
- > 0 if at least one of the specified interrupt channels is pending.

Execution Times

For information on the execution times, refer to [Function Execution Times](#) on page 53.

Example

This example shows how to query the pending status for channels 1 and 3:

```
pending_status = ds5203_pending_int(DS5203_1_BASE, DS5203_INT_SRC_1 |
DS5203_INT_SRC_3);
```

Related topics**References**

ds5203_ack_int	30
ds5203_disable_int	29
ds5203_enable_int	27
ds5203_init	15
ds5203_read_int	33

ds5203_read_int

Syntax

```
__INLINE void ds5203_read_int(  
    phs_addr_t base,  
    UInt32* int_src)
```

Include file

ds5203.h

Purpose

To read the interrupt source register of the DS5203 FPGA Board.

Description

The `int_src` parameter returns a bit mask containing information on which interrupt channels have requested an interrupt. Use the predefined symbols `DS5203_INT_SRC_1` ... `DS5203_INT_SRC_8` to test the respective interrupt channels.

Note

The `ds5203_enable_int` function must be called before interrupts can be generated and acknowledged.

Parameters

base

Specifies the PHS-bus base address of the board, see `DSxxxx_n_BASE` macros (refer to [Base Address of the I/O Board](#) on page 13).

int_src

Specifies the pointer to the variable containing the bit mask with the pending interrupt channels. You can use the following predefined symbols to access the entries of the bit mask.

Symbols	Meaning
DS5203_INT_SRC_1	Interrupt on channel 1
DS5203_INT_SRC_2	Interrupt on channel 2
DS5203_INT_SRC_3	Interrupt on channel 3
DS5203_INT_SRC_4	Interrupt on channel 4
DS5203_INT_SRC_5	Interrupt on channel 5
DS5203_INT_SRC_6	Interrupt on channel 6
DS5203_INT_SRC_7	Interrupt on channel 7
DS5203_INT_SRC_8	Interrupt on channel 8

Return value

None

Execution Times For information on the execution times, refer to [Function Execution Times](#) on page 53.

Example This example shows how to read the interrupt source register:

```
ds5203_read_int(DS5203_1_BASE, &int_src);
if (int_src & DS5203_INT_SRC_1)
{ ... }
```

Related topics	References
	ds5203_ack_int..... 30
	ds5203_disable_int..... 29
	ds5203_enable_int..... 27
	ds5203_init..... 15
	ds5203_pending_int..... 31

Data Exchange Functions

Introduction Functions to exchange data between the FPGA application running on the DS5203 FPGA Board and the processor board.

Where to go from here **Information in this section**

ds5203_read_reg.....	35
To read data from an exchange register of the DS5203 FPGA Board.	
ds5203_write_reg.....	37
To write data to an exchange register of the DS5203 FPGA Board.	
ds5203_read_reg_grp.....	39
To read from a group of exchange registers of the DS5203 FPGA Board.	
ds5203_write_reg_grp.....	41
To write to a group of exchange registers of the DS5203 FPGA Board.	
ds5203_read_buf.....	43
To read data from a buffer of the DS5203 FPGA Board.	
ds5203_write_buf.....	45
To write data to a buffer of the DS5203 FPGA Board.	

ds5203_read_reg

Syntax

```
__INLINE void ds5203_read_reg(  
    phs_addr_t base,  
    UInt32 reg_nr,  
    dsfloat* data,  
    dsfloat* scaling,  
    UInt32 mode)
```

Include file	ds5203.h																				
Purpose	To read data from an exchange register of the DS5203 FPGA Board.																				
Description	This function reads data from a specified exchange register and scales the data by a specified factor.																				
Parameters	<p>base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to Base Address of the I/O Board on page 13).</p> <p>reg_nr Specifies the number of the data register in the range 1 ... 128.</p> <p>data Specifies the pointer to the variable containing the read register value.</p> <p>scaling Specifies the pointer to the variable containing the scaling factor. Use one of the predefined symbols.</p> <table><tr><th>Symbol</th><th>Meaning</th></tr><tr><td>DS5203_READ_SCALE_BIN_PT_0</td><td>Scaling factor: $1/2^0 = 1.00$</td></tr><tr><td>DS5203_READ_SCALE_BIN_PT_1</td><td>Scaling factor: $1/2^1 = 0.50$</td></tr><tr><td>DS5203_READ_SCALE_BIN_PT_2</td><td>Scaling factor: $1/2^2 = 0.25$</td></tr><tr><td>DS5203_READ_SCALE_BIN_PT_3</td><td>Scaling factor: $1/2^3 = 0.125$</td></tr><tr><td>...</td><td>...</td></tr><tr><td>DS5203_READ_SCALE_BIN_PT_32</td><td>Scaling factor: $1/2^{32} = 2.328e^{-10}$</td></tr></table> <p>mode Specifies the data processing mode. Use one of the predefined symbols.</p> <table><tr><th>Symbol</th><th>Meaning</th></tr><tr><td>DS5203_DATA_MODE_SIGNED</td><td>The data is processed as a signed data type.</td></tr><tr><td>DS5203_DATA_MODE_UNSIGNED</td><td>The data is processed as an unsigned data type.</td></tr></table>	Symbol	Meaning	DS5203_READ_SCALE_BIN_PT_0	Scaling factor: $1/2^0 = 1.00$	DS5203_READ_SCALE_BIN_PT_1	Scaling factor: $1/2^1 = 0.50$	DS5203_READ_SCALE_BIN_PT_2	Scaling factor: $1/2^2 = 0.25$	DS5203_READ_SCALE_BIN_PT_3	Scaling factor: $1/2^3 = 0.125$	DS5203_READ_SCALE_BIN_PT_32	Scaling factor: $1/2^{32} = 2.328e^{-10}$	Symbol	Meaning	DS5203_DATA_MODE_SIGNED	The data is processed as a signed data type.	DS5203_DATA_MODE_UNSIGNED	The data is processed as an unsigned data type.
Symbol	Meaning																				
DS5203_READ_SCALE_BIN_PT_0	Scaling factor: $1/2^0 = 1.00$																				
DS5203_READ_SCALE_BIN_PT_1	Scaling factor: $1/2^1 = 0.50$																				
DS5203_READ_SCALE_BIN_PT_2	Scaling factor: $1/2^2 = 0.25$																				
DS5203_READ_SCALE_BIN_PT_3	Scaling factor: $1/2^3 = 0.125$																				
...	...																				
DS5203_READ_SCALE_BIN_PT_32	Scaling factor: $1/2^{32} = 2.328e^{-10}$																				
Symbol	Meaning																				
DS5203_DATA_MODE_SIGNED	The data is processed as a signed data type.																				
DS5203_DATA_MODE_UNSIGNED	The data is processed as an unsigned data type.																				
Return value	None																				
Execution Times	For information on the execution times, refer to Function Execution Times on page 53.																				

Example	<p>This example shows how to read data in signed mode with a scaling factor of $1/2^{16}$ from register 1:</p> <pre>dsfloat scaling_factor = DS5203_READ_SCALE_BIN_PT_16; ds5203_read_reg(DS5203_1_BASE, 1, &data, &scaling_factor, DS5203_DATA_MODE_SIGNED);</pre>
---------	--

Related topics	<p>References</p> <table> <tr> <td>ds5203_init.....</td> <td>15</td> </tr> <tr> <td>ds5203_read_buf.....</td> <td>43</td> </tr> <tr> <td>ds5203_read_reg_grp.....</td> <td>39</td> </tr> <tr> <td>ds5203_write_buf.....</td> <td>45</td> </tr> <tr> <td>ds5203_write_reg.....</td> <td>37</td> </tr> <tr> <td>ds5203_write_reg_grp.....</td> <td>41</td> </tr> </table>	ds5203_init.....	15	ds5203_read_buf.....	43	ds5203_read_reg_grp.....	39	ds5203_write_buf.....	45	ds5203_write_reg.....	37	ds5203_write_reg_grp.....	41
ds5203_init.....	15												
ds5203_read_buf.....	43												
ds5203_read_reg_grp.....	39												
ds5203_write_buf.....	45												
ds5203_write_reg.....	37												
ds5203_write_reg_grp.....	41												

ds5203_write_reg

Syntax	<pre>__INLINE void ds5203_write_reg(phs_addr_t base, UInt32 reg_nr, dsfloat* data, dsfloat* scaling, UInt32 mode)</pre>
--------	--

Include file	ds5203.h
Purpose	To write data to an exchange register of the DS5203 FPGA Board.
Description	This function writes data to a specified exchange register and scales the data by a specified factor.
Parameters	<p>base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to Base Address of the I/O Board on page 13).</p> <p>reg_nr Specifies the number of the register in the range 1 ... 128.</p> <p>data Specifies the pointer to the variable containing the data that you want to write to the register.</p>

scaling Specifies the pointer to the variable containing the scaling factor. Use one of the predefined symbols.

Symbol	Meaning
DS5203_WRITE_SCALE_BIN_PT_0	Scaling factor: $2^0=1$
DS5203_WRITE_SCALE_BIN_PT_1	Scaling factor: $2^1=2$
DS5203_WRITE_SCALE_BIN_PT_2	Scaling factor: $2^2=4$
DS5203_WRITE_SCALE_BIN_PT_3	Scaling factor: $2^3=8$
...	...
DS5203_WRITE_SCALE_BIN_PT_32	Scaling factor: $2^{32}=4,294,967,296$

mode Specifies the data processing mode. Use one of the predefined symbols.

Symbol	Meaning
DS5203_DATA_MODE_SIGNED	The data is processed as a signed data type.
DS5203_DATA_MODE_UNSIGNED	The data is processed as an unsigned data type.

Return value None

Execution Times For information on the execution times, refer to [Function Execution Times](#) on page 53.

Example This example shows how to write data in signed mode with a scaling factor of 2^{16} to register 1:

```
dsfloat scaling_factor = DS5203_WRITE_SCALE_BIN_PT_16;
ds5203_write_reg(
    DS5203_1_BASE,
    1,
    &data,
    &scaling_factor,
    DS5203_DATA_MODE_SIGNED);
```

Related topics

References

ds5203_init	15
ds5203_read_buf	43
ds5203_read_reg	35
ds5203_read_reg_grp	39
ds5203_write_buf	45
ds5203_write_reg_grp	41

ds5203_read_reg_grp

Syntax

```
__INLINE void ds5203_read_reg_grp(  
    phs_addr_t base,  
    UInt32 grp_nr,  
    UInt32 grp_size,  
    UInt32* reg_nr,  
    dsfloat* data,  
    dsfloat* scaling,  
    UInt32* mode)
```

Include file

ds5203.h

Purpose

To read from a group of exchange registers of the DS5203 FPGA Board.

Description

This function reads a group of data from several specified exchange registers and scales each data value by a specified factor. The data can be assumed to be consistent, which means that each single date of the group is read from the FPGA application at the same point in time.

Parameters

base

Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to [Base Address of the I/O Board](#) on page 13).

grp_nr

Specifies the number of the data group in the range 1 ... 63.

grp_size

Specifies the group size. The number of values is in the range 1 ... 128.

reg_nr

Specifies the pointer to the variable containing the number of each group register in the range 1 ... 128. The variable itself is an array of **grp_size** size.

Note

For best performance, the registers in the group should be in ascending order without gaps.

data

Specifies the pointer to the variable containing the register values as an array of **grp_size** size.

scaling

Specifies the pointer to the variable containing the scaling factors as an array of **grp_size** size. Use the predefined symbols.

Symbol	Meaning
DS5203_READ_SCALE_BIN_PT_0	Scaling factor: $1/2^0 = 1.00$
DS5203_READ_SCALE_BIN_PT_1	Scaling factor: $1/2^1 = 0.50$

Symbol	Meaning
DS5203_READ_SCALE_BIN_PT_2	Scaling factor: $1/2^2 = 0.25$
DS5203_READ_SCALE_BIN_PT_3	Scaling factor: $1/2^3 = 0.125$
...	...
DS5203_READ_SCALE_BIN_PT_32	Scaling factor: $1/2^{32} = 2.328e^{-10}$

mode Specifies the pointer to the variable containing the data modes as an array of **grp_size** size. Use the predefined symbols.

Symbol	Meaning
DS5203_DATA_MODE_SIGNED	The data is processed as a signed data type.
DS5203_DATA_MODE_UNSIGNED	The data is processed as an unsigned data type.

Return value None

Execution Times For information on the execution times, refer to [Function Execution Times](#) on page 53.

Example This example shows how to read data in signed mode from data group 1. It contains 1 value that is scaled by $1/2^{16}$:

```
dsfloat scaling_factor = DS5203_READ_SCALE_BIN_PT_16;
UInt32 mode = DS5203_DATA_MODE_SIGNED;
dsfloat data;
...
ds5203_read_reg_grp(
    DS5203_1_BASE,
    1,
    1,
    &reg_nr,
    &data,
    &scaling_factor,
    &mode);
```

Related topics

References

ds5203_init	15
ds5203_read_buf	43
ds5203_read_reg	35
ds5203_write_buf	45
ds5203_write_reg	37
ds5203_write_reg_grp	41

ds5203_write_reg_grp

Syntax

```
__INLINE void ds5203_write_reg_grp(
    phs_addr_t base,
    UInt32 grp_nr,
    UInt32 grp_size,
    UInt32* reg_nr,
    dsfloat* data,
    dsfloat* scaling,
    UInt32* mode)
```

Include file

ds5203.h

Purpose

To write to a group of exchange registers of the DS5203 FPGA Board.

Description

This function writes data to several specified exchange registers and scales each value in the group by a specified factor. All the data in the group can be assumed to be consistent, which means that no single value can be accessed by the FPGA application until the last value of the group is written.

Parameters

base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to [Base Address of the I/O Board](#) on page 13).

grp_nr Specifies the number of the data group in the range 1 ... 63.

grp_size Specifies the group size. The number of values is in the range 1 ... 128.

reg_nr Specifies the pointer to the variable containing the number of each group register in the range 1 ... 128. The variable itself is an array of **grp_size** size.

Note

For best performance, the registers in the group should be in ascending order without gaps.

data Specifies the pointer to the variable containing the register values as an array of **grp_size** size.

scaling Specifies the pointer to the variable containing the scaling factors as an array of **grp_size** size. Use the predefined symbols.

Symbol	Meaning
DS5203_WRITE_SCALE_BIN_PT_0	Scaling factor: $2^0=1$
DS5203_WRITE_SCALE_BIN_PT_1	Scaling factor: $2^1=2$

Symbol	Meaning
DS5203_WRITE_SCALE_BIN_PT_2	Scaling factor: $2^2=4$
DS5203_WRITE_SCALE_BIN_PT_3	Scaling factor: $2^3=8$
...	...
DS5203_WRITE_SCALE_BIN_PT_32	Scaling factor: $2^{32}=4,294,967,296$

mode Specifies the pointer to the variable containing the data modes as an array of **grp_size** size. Use the predefined symbols.

Symbol	Meaning
DS5203_DATA_MODE_SIGNED	The data is processed as a signed data type.
DS5203_DATA_MODE_UNSIGNED	The data is processed as an unsigned data type.

Return value None

Execution Times For information on the execution times, refer to [Function Execution Times](#) on page 53.

Example This example shows you how to write data in signed mode to data group 1. It contains 1 value that is scaled by 2^{16} :

```
dsfloat scaling_factor = DS5203_WRITE_SCALE_BIN_PT_16;
UInt32 mode = DS5203_DATA_MODE_SIGNED;
dsfloat data = 1.3;
...
ds5203_write_reg_grp(
    DS5203_1_BASE,
    1,
    1,
    &reg_nr,
    &data,
    &scaling_factor,
    &mode);
```

Related topics

References

ds5203_init	15
ds5203_read_buf	43
ds5203_read_reg	35
ds5203_read_reg_grp	39
ds5203_write_buf	45
ds5203_write_reg	37

ds5203_read_buf

Syntax

```
__INLINE void ds5203_read_buf(
    phs_addr_t base,
    UInt32 buf_nr,
    UInt32* data_length,
    dsfloat* data,
    dsfloat* scaling,
    UInt32 mode,
    UInt32* status)
```

Include file

ds5203.h

Purpose

To read data from a buffer of the DS5203 FPGA Board.

Description

This function reads a specified number of values from a buffer and scales the data by a specified factor. You can configure the scaling and the mode parameters only for the entire buffer and not separately for each buffer value. The buffer state is also returned.

Parameters

base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to [Base Address of the I/O Board](#) on page 13).

buf_nr Specifies the number of the buffer in the range 1 ... 32.

data_length Specifies the pointer to the variable containing the data length as the number of buffer elements in the range 1 ... **buffer_size**. The maximum buffer size depends on the buffer definition in the FPGA framework used.

data Specifies the pointer to the variable containing the buffer values as an array of **data_length** size.

scaling Specifies the pointer to the variable containing the scaling factor for reading. Use the predefined symbols.

Symbol	Meaning
DS5203_READ_SCALE_BIN_PT_0	Scaling factor: $1/2^0 = 1.00$
DS5203_READ_SCALE_BIN_PT_1	Scaling factor: $1/2^1 = 0.50$
DS5203_READ_SCALE_BIN_PT_2	Scaling factor: $1/2^2 = 0.25$
DS5203_READ_SCALE_BIN_PT_3	Scaling factor: $1/2^3 = 0.125$
...	...
DS5203_READ_SCALE_BIN_PT_32	Scaling factor: $1/2^{32} = 2.328e^{-10}$

mode Specifies the data processing mode. Use one of the predefined symbols.

Symbol	Meaning
DS5203_DATA_MODE_SIGNED	The data is processed as a signed data type.
DS5203_DATA_MODE_UNSIGNED	The data is processed as an unsigned data type.

status Specifies the pointer to the variable containing the buffer status. To check the buffer status, use the predefined symbols.

Symbol	Meaning
DS5203_BUF_OVFL	Data loss due to buffer overflow.
DS5203_BUF_NEW	Data is new, buffer was not read before.

Return value None

Execution Times For information on the execution times, refer to [Function Execution Times](#) on page 53.

Example This example shows you how to read data in signed mode from a buffer. The values are scaled by $1/2^{16}$:

```
dsfloat data[1024];
UInt32 data_length;
dsfloat scaling_factor = DS5203_READ_SCALE_BIN_PT_16;
UInt32 status;
...
ds5203_read_buf(
    DS5203_1_BASE,
    1,
    &data_length,
    data,
    &scaling_factor,
    DS5203_DATA_MODE_SIGNED,
    &status);
```

Related topics

References

ds5203_init	15
ds5203_read_reg	35
ds5203_read_reg_grp	39
ds5203_write_buf	45
ds5203_write_reg	37
ds5203_write_reg_grp	41

ds5203_write_buf

Syntax

```
__INLINE void ds5203_write_buf(
    phs_addr_t base,
    UInt32 buf_nr,
    UInt32 data_length,
    dsfloat* data,
    dsfloat* scaling,
    UInt32 mode)
```

Include file

ds5203.h

Purpose

To write data to a buffer of the DS5203 FPGA Board.

Description

This function writes a specified number of values to a buffer and scales the data by a specified factor.

Parameters

base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to [Base Address of the I/O Board](#) on page 13).

buf_nr Specifies the number of the buffer in the range 1 ... 32.

data_length Specifies the data length as the number of buffer elements in the range 1 ... **buffer_size**. The maximum buffer size depends on the buffer definition in the FPGA framework used.

data Specifies the pointer to the variable containing the buffer values as an array of **data_length** size.

scaling Specifies the pointer to the variable containing the scaling factor for writing. Use the predefined symbols.

Symbol	Meaning
DS5203_WRITE_SCALE_BIN_PT_0	Scaling factor: $2^0=1$
DS5203_WRITE_SCALE_BIN_PT_1	Scaling factor: $2^1=2$
DS5203_WRITE_SCALE_BIN_PT_2	Scaling factor: $2^2=4$
DS5203_WRITE_SCALE_BIN_PT_3	Scaling factor: $2^3=8$
...	...
DS5203_WRITE_SCALE_BIN_PT_32	Scaling factor: $2^{32}=4,294,967,296$

mode Specifies the data processing mode. Use one of the predefined symbols.

Symbol	Meaning
DS5203_DATA_MODE_SIGNED	The data is processed as a signed data type.
DS5203_DATA_MODE_UNSIGNED	The data is processed as an unsigned data type.

Return value None

Execution Times For information on the execution times, refer to [Function Execution Times](#) on page 53.

Example This example shows you how to write 128 values in signed mode to a buffer. The values are scaled by 2^{16} :

```
dsfloat data[128] = {1.3};
dsfloat scaling_factor = DS5203_WRITE_SCALE_BIN_PT_16;
...
ds5203_write_buf(
    DS5203_1_BASE,
    1,
    128,
    &data,
    &scaling_factor,
    DS5203_DATA_MODE_SIGNED);
```

Related topics

References

ds5203_init	15
ds5203_read_buf	43
ds5203_read_reg	35
ds5203_read_reg_grp	39
ds5203_write_reg	37
ds5203_write_reg_grp	41

FPGA Programming Functions

Introduction

Function to download the FPGA application to the FPGA of the DS5203 FPGA Board.

Note

- The processor board must be initialized before you can program the FPGA.
- The FPGA must be programmed before the initialization of the DS5203 FPGA Board.

ds5203_program

Syntax

```
void ds5203_program(
    phs_addr_t base,
    ds5203_prgr_dat *data)
```

Include file

ds5203.h

Purpose

To download the FPGA application to the FPGA of the DS5203 FPGA Board.

Description

This function programs the FPGA of the DS5203 FPGA Board. The **mode** parameter in the **ds5203_prgr_dat** data structure is used to specify loading the program into the SRAM (FPGA) or the flash memory of the board. This function can also decompress (dscompress) bitstreams before programming.

The data structure is contained in the generated model INI file.

If you call this function, the VCM module of your processor application providing information on versions and configuration data is updated.

For detailed information on the data structure used, refer to [ds5203_prgr_dat](#) on page 9.

Parameters

base Specifies the PHS-bus base address of the board, see DSxxxx_n_BASE macros (refer to [Base Address of the I/O Board](#) on page 13).

data Specifies the pointer to the `ds5203_prgr_dat` data structure containing the specified programming data, for example, the programming mode or version information.

Return value

None

Messages

The following messages are defined:

ID	Type	Message	Description
201	Error	ds5203_program(0x??): invalid PHS-bus base address 0x??!	The PHS-bus base address is not valid. An invalid PHS-bus base address of 0x00 can occur if the I/O board is not connected to the PHS bus.
300	Error	ds5203_program(0x??): Board not found!	No DS5203 board could be found at the specified PHS-bus address.
301	Error	ds5203_program(0x??): Board is not responding! Hardware reset failed.	The DS5203 board has not booted after power-up.
308	Error	ds5203_program(0x??): Programming of FPGA failed!	Programming of FPGA failed for any purpose. One reason may be a timeout after programming FPGA (no ready signal after programming). Can occur with other error messages.
309	Error	ds5203_program(0x??): Error in transmission of Bitstream!	A CRC error occurred in transmission of bitstream
311	Error	ds5203_program(0x??): Timeout error! Programming aborted!	After writing PiggyBack ID to FPGA, CPLD did not release BUSY-signal. This indicates a wrong ID/bitstream.
312	Info	ds5203_program(0x??): Compress: ?; SRAM: ?; FLASH: ?	Indicates by truth value which modes of operation are activated (1 = activated).
313	Info	ds5203_program(0x??): DeCompressing Bitstream ...	Decompressing of bitstream was started.
314	Info	ds5203_program(0x??): DeCompressing of Bitstream successful!	Status message indicating that the bitstream was decompressed successfully.
315	Info	ds5203_program(0x??): No DeCompression of Bitstream selected!	Decompression of FPGA was left out because this was selected by mode parameter.
316	Info	ds5203_program(0x??): No programming of FPGA selected!	Programming of FPGA was left out because this was selected by mode parameter.
317	Info	ds5203_program(0x??): Programming FPGA ...	Programming of FPGA was started.

ID	Type	Message	Description
318	Info	?? percent finished	Status message, which indicates the progress of programming the FPGA.
319	Info	ds5203_program(0x??): Programming of FPGA successful!	Status message, which indicates that programming of the FPGA was successful.
323	Info	ds5203_program(0x??): Erasing flash memory ...	Status message which indicates that erasing of flash memory has started.
324	Info	ds5203_program(0x??): Erasing of flash memory finished!	Debug message which indicates that erasing of flash memory finished successfully.
325	Info	ds5203_program(0x??): Programming flash memory ...	Status message which indicates that programming of flash memory has started.
326	Info	ds5203_program(0x??): Programming word ?? of ??.	Debug message, which shows which word was already programmed.
327	Info	ds5203_program(0x??): Programming of Flash Succeeded!	Status message which indicates that programming of flash memory finished successfully.
328	Info	ds5203_program(0x??): Programming of Flash not executed!	Debug message which indicates that programming of flash memory was left out.
329	Error	ds5203_program(0x??): Invalid Size of decompressed Bitstream entered!	This message indicates that the size of the compressed bitstream is significantly greater than the size of the decompressed bitstream. This is not allowed.
330	Error	ds5203_program(0x??): FPGA is in wrong Mode (Configuration). Programming failed!	This indicates that the FPGA is in the configuration mode after reset.
331	Error	ds5203_program(0x??): FPGA is in wrong Mode (not-Configuration). Perhaps you tried to download a bitstream, which was not built for the hardware platform you use. Programming failed.	This indicates that the FPGA is not in the configuration mode after writing a framework ID to the FPGA. This error occurs, when an invalid FW_ID is transmitted to the FPGA.
332	Error	ds5203_program(0x??): The model INI file you use is incompatible to your hardware!	The model INI file contains a list of board revisions (inherited from framework). The board used must exist in this list.
333	Error	ds5203_program(0x??): The model INI file you use is incompatible to the RTLib version you use.	The model INI file contains the FPGA blockset version it was generated with. If you want to program the FPGA application using a host PC with an incompatible RTLib version installed, this error occurs.

Execution Times

For information on the execution times, refer to [Function Execution Times](#) on page 53.

Example

This example shows how to use the function generally:

```
extern void *fpga_prgrm_data_ptr_<AppID>;
ds5203_prgrn_dat *datastruct = (ds5203_prgrn_dat*)fpga_prgrm_data_ptr_<AppID>;
ds5203_program(base, datastruct);
```

Assuming that the created model INI file contains DS5203_MAIN_84303E38761313.c, the program function and the data structure look like the following example.

Extract from the above-mentioned C file:

```
#include <dstypes.h>
UInt8 bitstream_84303E38761313[];
UInt32 compat_boards_maj_84303E38761313[];
UInt32 compat_boards_min_84303E38761313[];
struct
{
    UInt32 model_blkst_major;
    UInt32 model_blkst_minor;
    UInt32 model_blkst_maintenance;
    UInt8 *bitstream_pt;
    UInt32 length;
    UInt32 length_unpckd;
    UInt32 *compatible_boards_list_maj_pt;
    UInt32 *compatible_boards_list_min_pt;
    UInt32 compatible_boards_list_length;
    UInt32 fw_id;
    UInt32 mode;
} prgr_dat_84303E38761313 =
{
    1, /* blockset major version */
    1, /* blockset minor version */
    0, /* blockset maintenance version */
    (UInt8*) &bitstream_84303E38761313, /* pointer to bitstream */
    0x3199e, /* length compressed (= 203,166 bytes) */
    0x441f80, /* length decompressed (= 4,464,512 bytes) */
    (UInt32*) &compat_boards_maj_84303E38761313, /* pointer to comp boards major */
    (UInt32*) &compat_boards_min_84303E38761313, /* pointer to comp boards minor */
    1, /* number of compatible boards */
    1, /* framework identifier */
    (0x02 | 0x04 | 0x08)
    /* programming mode: DS5203_PRGRM_SRM | DS5203_PRGRM_COM | DS5203_PRGRM_DBGOUT
    */
};
void *fpga_prgrm_data_ptr_84303E38761313 = (void*) &prgr_dat_84303E38761313;
UInt32 compat_boards_maj_84303E38761313[] =
{
    2
};
UInt32 compat_boards_min_84303E38761313[] =
{
    0
};
UInt8 bitstream_84303E38761313[] =
{
    249,255,255,255,255,249,4,4,
    249,8,8,249,16,16,0,0,
    0,187,17,34,0,68,249,8,
    ...
}
...
```

Extract from the main program:

```
extern void *fpga_prgrm_data_ptr_84303E38761313;  
ds5203_prgr_dat *datastruct = (ds5203_prgr_dat*)  
fpga_prgrm_data_ptr_84303E38761313;  
ds5203_program(base, datastruct);
```

Related topics

References

ds5203_init.....	15
ds5203_prgr_dat.....	9
Overview of the DS5203 Real-Time Library.....	7

Function Execution Times

Introduction	The execution times of the C functions can vary, since they depend on different factors. The measured execution times are influenced by the test environment used. This section gives you basic information on the test environment and contains the mean function execution times.
--------------	---

Where to go from here	<div>Information in this section</div> <div><div>Information on the Test Environment..... 53</div><div>This topic contains basic information on the measurement test environment of the execution times.</div><div>Measured Execution Times..... 54</div><div>This topic contains the measured execution times.</div></div>
-----------------------	---

Information on the Test Environment

Test environment	<p>The execution time of a function can vary, since it depends on different factors, for example:</p> <ul style="list-style-type: none">▪ CPU clock and bus clock frequency of the processor board used▪ Optimization level of the compiler and the usage of inlining▪ Parameters used <p>The test programs that are used to measure the execution time of the functions listed below have been generated and compiled with the default settings of the <code>downxxx</code> tool (optimization and inlining). The execution times in the tables below are always the mean measurement values.</p>
------------------	--

The properties of the processor boards used are:

	DS1006	DS1006 Multicore	DS1007
CPU clock	2.6 GHz	2.8 GHz	2 GHz
Bus clock	133 MHz	133 MHz	133 MHz

Measured Execution Times

Introduction

Mean Execution times are available for the following functional units:

- Initialization
- Identification functions
- Interrupt handling
- Data exchange functions
- FPGA programming

Initialization

The following execution time has been measured for the initialization function:

Function	Mean Execution Time		
	DS1006	DS1006 Multicore	DS1007
ds5203_init	310 μ s	326 μ s	242 μ s

Identification functions

The following execution times have been measured for the identification functions:

Function	Mean Execution Time		
	DS1006	DS1006 Multicore	DS1007
ds5203_get_board_rev	0.58 μ s	0.58 μ s	0.33 μ s
ds5203_get_fw_id	0.60 μ s	0.60 μ s	0.35 μ s
ds5203_get_pb_id	0.58 μ s	0.58 μ s	0.33 μ s
ds5203_get_appl_id	1.11 μ s	1.12 μ s	0.63 μ s
ds5203_get_appl_id_string	33.07 μ s	33.73 μ s	18.20 μ s

Interrupt handling

The following execution times have been measured for the interrupt functions:

Function	Mean Execution Time		
	DS1006	DS1006 Multicore	DS1007
ds5203_enable_int	0.81 μ s	0.81 μ s	0.57 μ s
ds5203_disable_int	0.81 μ s	0.81 μ s	0.57 μ s
ds5203_ack_int	0.18 μ s	0.18 μ s	0.20 μ s
ds5203_pending_int	0.59 μ s	0.59 μ s	0.35 μ s
ds5203_read_int	0.59 μ s	0.59 μ s	0.35 μ s

Data exchange functions

The following execution times have been measured for the data exchange functions:

Function	Mean Execution Time		
	DS1006	DS1006 Multicore	DS1007
ds5203_read_reg	0.60 μ s	0.60 μ s	0.43 μ s
ds5203_write_reg	0.20 μ s	0.20 μ s	0.21 μ s
ds5203_read_reg_grp ¹⁾	5.67 μ s	5.66 μ s	3.75 μ s
ds5203_write_reg_grp ¹⁾	1.69 μ s	1.69 μ s	1.77 μ s
ds5203_read_buf ²⁾	0.54 ms	0.56 ms	0.36 ms
ds5203_write_buf ²⁾	0.15 ms	0.15 ms	0.15 ms

¹⁾ Group size = 10

²⁾ Buffer size = 1024

FPGA programming function

The following execution times have been measured for the FPGA programming functions:

Function		Mean Execution Time		
		DS1006	DS1006 Multicore	DS1007
ds5203_program	... into SRAM	224.3 ms	223.7 ms	239.7 ms
	... into flash memory	55.46 s	55.61 s	54.95 s

B

base address 13
board initialization 15

C

Common Program Data folder 6

D

data exchange functions 35
Documents folder 6
DS5203
 execution times 54
DS5203 Real-Time Library 7
 overview 7
ds5203_ack_int 30
ds5203_disable_int 29
ds5203_enable_int 27
ds5203_get_appl_id 23
ds5203_get_appl_id_string 24
ds5203_get_board_rev 19
ds5203_get_fw_id 20
ds5203_get_pb_id 22
ds5203_init 15
ds5203_pending_int 31
ds5203_prgr_dat 9
ds5203_program 47
ds5203_read_buf 43
ds5203_read_int 33
ds5203_read_reg 35
ds5203_read_reg_grp 39
ds5203_write_buf 45
ds5203_write_reg 37
ds5203_write_reg_grp 41

E

execution times
 DS5203 54

F

FPGA programming functions 47
function execution times 53

I

identifying DS5203 components 19
information on the test environment 53
interrupt functions 27

L

Local Program Data folder 6

M

macros 13

