

DS2101 D/A Board

RTLib 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.

© 2001 - 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
Macros	7
Base Address of the I/O Board.....	7
Board Initialization	9
ds2101_init.....	9
DAC Unit	11
ds2101_set_errmode.....	12
ds2101_set_range.....	13
ds2101_load.....	14
ds2101_load_uni.....	16
ds2101_out.....	17
ds2101_out_uni.....	18
ds2101_strobe.....	20
Function Execution Times	23
Information on the Test Environment.....	23
Measured Execution Times.....	24
Index	25

About This Reference

Content

This RTLib Reference (Real-Time Library) gives detailed descriptions of the C functions needed to program a DS2101 D/A 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.

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 reference 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_init \(DS1006 RTLib Reference\)](#) or [init \(DS1007 RTLib Reference\)](#).

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 different 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 DS2101, you have to perform the initialization process.

Note

The initialization function of the processor board must be called before the DS2101's initialization function.

ds2101_init

Syntax

```
void ds2101_init(phs_addr_t base)
```

Include file

ds2101.h

Purpose

To initialize the DS2101.

Description

All DS2101 registers are initialized to default values:

- -10 ... +10 V bipolar output voltage range
- All DAC data and latch registers are set to zero
- Zero output on I/O error

Note

This function must be called before any other DS2101 function can be used.

Parameters **base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 7.

Return value None

Messages The following messages are defined:

ID	Type	Message	Description
201	Error	ds2101_init(): Invalid PHS-bus base address 0x ????????	The value if the base parameter is not a valid PHS-bus address. This error may be caused if the PHS-bus connection of the I/O board is missing. Check the connection.
-146	Error	ds2101_init(0x??): Board not found!	No DS2101 board could be found at the specified PHS-bus address. Check if the DSxxxx_n_base macro corresponds to the I/O board used.
-53	Warning	ds2101_init(0x??): Jumper setting is not matching SW default initialization! STP register: 0x???????? instead of 0x????????	The value of the STP register could not be verified successfully. May be the DS2101 jumper setting is not correct. Remove all jumpers.

Execution times For information, refer to [Function Execution Times](#) on page 23.

Example The following example shows how to initialize a DS2101:

```
void main(void)
{
    init();
    ds2101_init(DS2101_1_BASE);
    ...
}
```

Related topics

References

[Base Address of the I/O Board](#)..... 7
[Macros](#)..... 7

DAC Unit

Introduction

The DS2101 D/A Board features 5 parallel D/A converter channels with 12-bit resolution.

Note

You have to initialize the DS2101 with the `ds2101_init` function before you can use one of these functions.

Where to go from here

Information in this section

To initialize the D/A converters	
ds2101_set_errmode.....	12
ds2101_set_range.....	13
To load an output value to the latched data register	
ds2101_load.....	14
ds2101_load_uni.....	16
To set the analog output	
ds2101_out.....	17
ds2101_out_uni.....	18
To strobe DS2101 channels	
ds2101_strobe.....	20

ds2101_set_errmode

Syntax

```
void ds2101_set_errmode(
    phs_addr_t base,
    int channel,
    int errmode)
```

Include file

ds2101.h

Purpose

To set the I/O error mode of the specified DAC channel.

Note

The `ds2101_init` function must be called before this function can be used.

I/O mapping

For details on the I/O mapping, refer to [DAC Unit \(DS2101 Features !\[\]\(758ebdf4629c903da74c2e079717ae32_img.jpg\)](#)).

Parameters

base Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 7.

channel Specifies the channel number within the range 1 ... 5. The `DS2101_CH_ALL` symbol selects all 5 channels.

errmode Specifies the I/O error mode. The following symbols are predefined:

Symbol	Description
<code>DS2101_ZERO</code>	Zero output on I/O error
<code>DS2101_KEEP</code>	Keeping output voltage on I/O error

Return value

None

Messages

The following message is defined:

ID	Type	Message	Description
-50	Error	ds2101_set_errmode(0x??): Board not initialized!	The DS2101 has not been initialized by a preceding call to the <code>ds2101_init</code> function.

Execution times

For information, refer to [Function Execution Times](#) on page 23.

Example

This example shows how to use the function:

```
ds2101_set_errmode(DS2101_1_BASE, 4, DS2101_KEEP);
```

Channel 4 of the DS2101 is set to keeping output voltage on I/O error.

Related topics**References**

Base Address of the I/O Board.....	7
ds2101_init.....	9
Macros.....	7

ds2101_set_range

Syntax

```
void ds2101_set_range(
    phs_addr_t base,
    int channel,
    int range)
```

Include file

ds2101.h

Purpose

To set the output voltage range of the specified DAC channel.

Note

The `ds2101_init` function must be called before this function can be used.

I/O mapping

For details on the I/O mapping, refer to [DAC Unit \(DS2101 Features !\[\]\(4436e6b00b9d5e62c2a161129eb3e4d0_img.jpg\)](#)).

Parameters

base Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 7.

channel Specifies the channel number within the range 1 ... 5. The `DS2101_CH_ALL` symbol selects all 5 channels.

range Specifies the input voltage range. The following symbols are predefined:

Symbol	Output Voltage Range
DS2101_RNG5	–5 ... +5 V (bipolar)
DS2101_RNG10	–10 ... +10 V (bipolar)
DS2101_RNG10U	0 ... +10 V (unipolar)

Return value None

Messages The following message is defined:

ID	Type	Message	Description
-50	Error	ds2101_set_range(0x??): Board not initialized!	The DS2101 has not been initialized by a preceding call to <code>ds2101_init</code> .

Execution times For information, refer to [Function Execution Times](#) on page 23.

Example This example shows how to use the function:

```
ds2101_set_range(DS2101_1_BASE, 1, DS2101_RNG5);
```

Channel 1 of the DS2101 is set to the output voltage range –5 ... +5 V.

Related topics


References

Base Address of the I/O Board.....	7
ds2101_init.....	9
ds2101_load_uni.....	16
ds2101_out_uni.....	18
Macros.....	7

ds2101_load

Syntax

```
void ds2101_load(
    phs_addr_t base,
    int channel,
    dsfloat value)
```

Include file	ds2101.h				
Purpose	To load an output value to the latched data register in bipolar mode.				
Description	<p>This function can be used for latched data output to a DAC channel of a DS2101. The output value, which must be within the range $-1.0 \dots +1.0$, is scaled to a 32-bit integer value by the factor 2^{31} and written to the respective DAC latched data register. The ds2101_strobe function must be used to update the DAC output.</p> <div> <p>Note</p> <ul style="list-style-type: none"> ▪ The ds2101_init function must be called before this function can be used. ▪ The DAC channel must have been initialized to a bipolar voltage range. </div>				
I/O mapping	For details on the I/O mapping, refer to DAC Unit (DS2101 Features)  .				
Parameters	<p>base Specifies the PHS-bus base address. Refer to Base Address of the I/O Board on page 7.</p> <p>channel Specifies the logical channel number within the range 1 ... 5.</p> <p>value Specifies the DAC output value within the range $-1.0 \dots +1.0$.</p>				
Return value	None				
Execution times	For information, refer to Function Execution Times on page 23.				
Example	<p>This example shows how to use the function:</p> <pre>ds2101_load(DS2101_1_BASE, 1, 0.5);</pre> <p>If channel 1 is initialized to the $-10 \dots +10$ V voltage range, channel 1 is loaded for 5 V output voltage.</p>				
Related topics	<p>References</p> <table> <tr> <td>Base Address of the I/O Board.....</td><td>7</td></tr> <tr> <td>ds2101_init.....</td><td>9</td></tr> </table>	Base Address of the I/O Board	7	ds2101_init	9
Base Address of the I/O Board	7				
ds2101_init	9				

ds2101_strobe	20
Macros	7

ds2101_load_uni

Syntax

```
void ds2101_load_uni(
    phs_addr_t base,
    int channel,
    dsfloat value)
```

Include file

ds2101.h

Purpose

To load an output value to the latched data register in unipolar mode.

Description

This function can be used in conjunction with the unipolar mode of the DS2101, if the output voltage range 0 ... +10 V is initialized by the **ds2101_set_range** function.

Note

- The **ds2101_init** function must be called before this function can be used.
- The **ds2101_strobe** function must be used to update the DAC output.
- The DAC channel must have been initialized to the unipolar voltage range.

I/O mapping

For details on the I/O mapping, refer to [DAC Unit \(DS2101 Features\)](#) .

Parameters

base Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 7.

channel Specifies the logical channel number within the range 1 ... 5.

value Specifies the DAC output value within the range -1.0 ... +1.0.

Return value

None

Execution times

For information, refer to [Function Execution Times](#) on page 23.

Example

This example shows how to use the function:

```
ds2101_load_uni(DS2101_1_BASE, 1, 0.5);
```

If channel 1 is initialized to the 0.0 ... +10 V voltage range, channel 1 is loaded for 5 V output voltage.

Related topics**References**

Base Address of the I/O Board.....	7
ds2101_init.....	9
ds2101_set_range.....	13
ds2101_strobe.....	20
Macros.....	7

ds2101_out

Syntax

```
void ds2101_out(
    phs_addr_t base,
    int channel,
    dsfloat value)
```

Include file

ds2101.h

Purpose

To set the analog output in bipolar mode.

Description

This function can be used for data output to a DAC channel of a DS2101. The output value, which must be within the range $-1.0 \dots +1.0$, is scaled to a 32-bit integer value by the factor 2^{31} and written to the respective DAC data register. Then the corresponding strobe bit in the data strobe register is set to update the DAC output register.

Note

- The `ds2101_init` function must be called before this function can be used.
- The DAC channel must have been initialized to a bipolar voltage range.

I/O mapping	For details on the I/O mapping, refer to DAC Unit (DS2101 Features ).
--------------------	---

Parameters	base Specifies the PHS-bus base address. Refer to Base Address of the I/O Board on page 7. channel Specifies the logical channel number within the range 1 ... 5. value Specifies the DAC output value within the range -1.0 ... +1.0.
-------------------	---

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

Execution times	For information, refer to Function Execution Times on page 23.
------------------------	--

Example	This example shows how to use the function:
----------------	---

```
ds2101_out(DS2101_1_BASE, 1, 0.5);
```

If channel 1 is initialized to the -10 ... +10 V voltage range, the output of channel 1 is set to 5 V voltage.

Related topics	References
-----------------------	-------------------


Base Address of the I/O Board	7
ds2101_init	9
Macros	7

ds2101_out_uni

Syntax	<pre>void ds2101_out_uni(phs_addr_t base, int channel, dsfloat value)</pre>
---------------	--

Include file	ds2101.h
---------------------	----------

Purpose	To set the analog output of the D/A converters in unipolar mode.
----------------	--

Description	<p>This function can be used in conjunction with the unipolar mode of the DS2101, if the output voltage range 0.0 ... +10 V is initialized by the <code>ds2101_set_range</code> function.</p> <div> Note <ul style="list-style-type: none"> ▪ The <code>ds2101_init</code> function must be called before this function can be used. ▪ The DAC channel must have been initialized to the unipolar voltage range. </div>								
I/O mapping	For details on the I/O mapping, refer to DAC Unit (DS2101 Features ).								
Parameters	<p>base Specifies the PHS-bus base address. Refer to Base Address of the I/O Board on page 7.</p> <p>channel Specifies the logical channel number within the range 1 ... 5.</p> <p>value Specifies the DAC output value within the range 0.0 ... +1.0.</p>								
Return value	None								
Execution times	For information, refer to Function Execution Times on page 23.								
Example	<p>This example shows how to use the function:</p> <pre>ds2101_out_uni(DS2101_1_BASE, 1, 0.5);</pre> <p>If channel 1 is initialized to the 0.0 ... +10 V voltage range the output of channel 1 is set to 5 V voltage.</p>								
Related topics	<p>References</p> <table> <tr> <td>Base Address of the I/O Board.....</td> <td>7</td> </tr> <tr> <td>ds2101_init.....</td> <td>9</td> </tr> <tr> <td>ds2101_set_range.....</td> <td>13</td> </tr> <tr> <td>Macros.....</td> <td>7</td> </tr> </table>	Base Address of the I/O Board.....	7	ds2101_init.....	9	ds2101_set_range.....	13	Macros.....	7
Base Address of the I/O Board.....	7								
ds2101_init.....	9								
ds2101_set_range.....	13								
Macros.....	7								

ds2101_strobe

Syntax

```
void ds2101_strobe(  
    phs_addr_t base,  
    long mask)
```

Include file

ds2101.h

Purpose

To strobe the DAC channels.

Description

The outputs of the channels specified in the **mask** parameter are updated simultaneously.

Before this function is called, the data must be copied to the latched data register with `ds2101_load` or `ds2101_load_uni`. The `ds2101_strobe` function copies all contents of the latched data registers simultaneously to the data registers to update the output.

Note

The `ds2101_init` function must be called before this function can be used.

Parameters

base Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 7.

mask Specifies the value written to the LDS register to output latched data. The following symbols are predefined:

Symbol	Channel
DS2101_CH_1	1
DS2101_CH_2	2
DS2101_CH_3	3
DS2101_CH_4	4
DS2101_CH_5	5

The symbols can be combined using the logical OR operation.

Return value

None

Execution times

For information, refer to [Function Execution Times](#) on page 23.

Example

This example shows how to use the function:

```
void sub_fct()
{
    ...
    ds2101_load(DS2101_1_BASE, 1, -0.4);
    ds2101_load(DS2101_1_BASE, 2, 0.9);
    ds2101_strobe(DS2101_1_BASE, DS2101_CH_1 | DS2101_CH_2);
    ...
}
```

Channel 1 is loaded for -4 V and channel 2 for +9 V output voltage. The outputs of channels 1 and 2 are updated simultaneously.

Related topics

References

Base Address of the I/O Board.....	7
ds2101_init.....	9
ds2101_load.....	14
ds2101_load_uni.....	16
Macros.....	7

Function Execution Times

Objective 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	Information in this section
	Information on the Test Environment..... 23
	Measured Execution Times..... 24

Information on the Test Environment

Test environment

The execution time of a function can vary, since it depends on different factors, for example:

- CPU clock and bus clock frequency of the processor board used
- Optimization level of the compiler
- Use of inlining parameters

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 **down<xxx>** tool (optimization and inlining). The execution times in the tables below are always the mean measurement values.

The properties of the processor boards used are:

	DS1006
CPU clock	2.6 GHz / 3.0 GHz
Bus clock	133 MHz

Measured Execution Times

Execution times are available for the following RTLib units:

- Initialization
- DAC unit

Note

The following execution times contain mean values for a sequence of I/O accesses. The execution time of a single call might be lower because of buffered I/O access.

Initialization

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

Function	Mean Execution Time	
	DS1006 with 2.6 GHz	DS1006 with 3.0 GHz
ds2101_init	46.85 μ s	58.88 μ s

DAC unit

The following execution times have been measured for the DAC unit:

Function	Mean Execution Time	
	DS1006 with 2.6 GHz	DS1006 with 3.0 GHz
ds2101_set_range	1.40 μ s	1.39 μ s
ds2101_set_errmode	1.39 μ s	1.38 μ s
ds2101_out	0.03 μ s	0.02 μ s
ds2101_out_uni	0.05 μ s	0.03 μ s
ds2101_load	0.03 μ s	0.02 μ s
ds2101_load_uni	0.05 μ s	0.03 μ s
ds2101_strobe	0.02 μ s	0.01 μ s

B

base address 7

C

Common Program Data folder 6

D

Documents folder 6

DS2101

function execution times 23

ds2101_init 9

ds2101_load 14

ds2101_load_uni 16

ds2101_out 17

ds2101_out_uni 18

ds2101_set_errmode 12

ds2101_set_range 13

ds2101_strobe 20

DSxxxx_n_BASE 7

F

function execution times

DS2101 23

L

Local Program Data folder 6

