

DS4004 HIL Digital I/O 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:	<a href="mailto:info@dspace.de">info@dspace.de</a>
Web:	<a href="http://www.dspace.com">http://www.dspace.com</a>

## 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](mailto: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 Functions and Macros	7
Macros.....	8
Base Address of the I/O Board.....	8
Initialization Functions.....	9
ds4004_init.....	9
ds4004_digout_mode_set.....	10
Run-Time Functions.....	13
ds4004_vbat_status_get.....	13
Digital I/O Functions	15
Code Examples.....	16
Example of Digital Input Functionality.....	16
Example of Digital Output Functionality.....	17
Initialization Functions.....	19
ds4004_digin_init.....	19
ds4004_digout_init.....	21
Run-Time Functions.....	24
ds4004_bit_in.....	24
ds4004_bit_in32.....	26
ds4004_bit_out.....	27
ds4004_bit_out32.....	29
Timing I/O Functions	31
Code Examples.....	32
Example of PWM Signal Measurement (PWM2D).....	32
Example of PWM Signal Generation (D2PWM).....	33
Example of Frequency Measurement (F2D).....	34
Example of Square-Wave Signal Generation (D2F).....	36
Initialization Functions.....	38
ds4004_pwm2d_init.....	38
ds4004_d2pwm_init.....	41

ds4004_f2d_init.....	44
ds4004_d2f_init.....	47
Run-Time Functions.....	51
ds4004_pwm2d.....	51
ds4004_d2pwm.....	53
ds4004_f2d.....	55
ds4004_d2f.....	57
 Interrupt Functions.....	 59
Code Examples.....	60
Example of Interrupt Generation.....	60
Initialization Functions.....	62
ds4004_int_mode_set.....	62
 Function Execution Times.....	 65
Information on the Test Environment.....	65
Measured Execution Times.....	66
 Index.....	 69





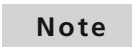



# About This Reference

## Contents

This RTLib reference (Real-Time Library) gives detailed descriptions of the C functions needed to program a DS4004 HIL Digital I/O Board. The C functions can be used to program RTI-specific Simulink S-functions, or to implement your real-time 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](http://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 Functions and Macros

Where to go from here

Information in this section

<a href="#">Macros.....</a>	<a href="#">8</a>
To get information on general macros.	
<a href="#">Initialization Functions.....</a>	<a href="#">9</a>
To perform global initialization of the board and of its output ports.	
<a href="#">Run-Time Functions.....</a>	<a href="#">13</a>
To check a port's VBAT status.	

# Macros



## Introduction

To get information on general macros.

## 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	<code>DS2001_1_BASE</code>
DS2002	20H	<code>DS2002_1_BASE</code>
DS2101	80H	<code>DS2101_1_BASE</code>
DS2001	90H	<code>DS2001_2_BASE</code>
DS2101	A0H	<code>DS2101_2_BASE</code>



# Initialization Functions

**Introduction** To perform global initialization of the board and of its output ports.

<b>Where to go from here</b>	<b>Information in this section</b>
	<a href="#">ds4004_init..... 9</a> To perform the basic initialization of the DS4004.
	<a href="#">ds4004_digout_mode_set..... 10</a> To enable or disable the digital output circuits for specified port(s).

## ds4004\_init

**Syntax** `void ds4004_init (phs_addr_t base)`

**Include file** `ds4004.h`

**Purpose** To perform the basic initialization of the DS4004.

**Description** This is the basic initialization function for the DS4004. The parameter structure is allocated. All setup parameters are set to the following default values for every channel (1 ... 32) of every port (1 ... 3):

- Digital output mode is disabled.
- Digital outputs are set to low (0).
- PWM generation duty is set to 0% (output low).
- All channels are in input mode.
- All interrupts are disabled.
- The input threshold of all channels is set to 2.5 V.
- The mode of all channels is set to Digital Input.
- All switches are disabled.

**Parameters** **base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

<b>Return value</b>	None
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 65.
<b>Example</b>	<pre>ds4004_init(DS4004_1_BASE);</pre>
<b>Related topics</b>	<b>References</b> <a href="#">Base Address of the I/O Board</a> ..... 8

## ds4004\_digout\_mode\_set

<b>Syntax</b>	<pre>void ds4004_digout_mode_set (     phs_addr_t base,     UInt32 port_mask,     UInt32 mode)</pre>
<b>Include file</b>	ds4004.h
<b>Purpose</b>	To enable or disable the digital output circuits for specified port(s).
<b>Description</b>	<p>This function writes the specified mode for the digital output circuits to the specified port(s). The setting refers to the digital I/O unit and the timing I/O unit. The input circuits are not affected. For information on the digital outputs, refer to <a href="#">Digital Outputs (PHS Bus System Hardware Reference)</a>.</p> <p>To set an external voltage (VBAT1 or VBAT2) for a digital output, you have to perform the following steps:</p> <ul style="list-style-type: none"> <li>▪ Call one of the I/O Initialization functions (<a href="#">ds4004_digout_init</a>, <a href="#">ds4004_d2pwm_init</a>, or <a href="#">ds4004_d2f_init</a>) to specify the functionality and select the low-side and high-side switches.</li> <li>▪ Call <a href="#">ds4004_digout_mode_set</a> to enable the digital outputs.</li> </ul>
<b>Parameters</b>	<b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 8.

**port\_mask** Specifies the port bitmask (0x0 ... 0x7) for the selected DS4004 board.

The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_MASK_PORT1	0x1	Port 1
DS4004_MASK_PORT2	0x2	Port 2
DS4004_MASK_PORT3	0x4	Port 3

Use a logical OR operation to select more than one port channel-wise.

**mode** Enables or disables the digital output circuits for the selected port(s).

The following symbols are predefined:

Predefined Symbol	Meaning
DS4004_DIGOUT_ENABLE	Enables the digital output circuits.
DS4004_DIGOUT_DISABLE	Disables the digital output circuits.

#### Note

- If the digital output circuits are disabled, they are in high impedance state (high-Z), so that writing to the output channels has no effect. The input circuits are not affected by this setting.
- When you set more than one port at the same time, the ports are not set synchronously.

---

**Return value** None

---

**Execution times** For information, refer to [Function Execution Times](#) on page 65.

---

#### Example

```
// Enable digital output drivers of port 1 and port 2
ds4004_digout_mode_set(
    DS4004_1_BASE,                /* select first DS4004 */
    DS4004_MASK_PORT1 | DS4004_MASK_PORT2, /* set port 1 and 2 */
    DS4004_DIGOUT_ENABLE);        /* enable digital output drivers */
```

For more detailed examples, refer to


- [Example of Digital Output Functionality](#) on page 17
- [Example of PWM Signal Generation \(D2PWM\)](#) on page 33
- [Example of Square-Wave Signal Generation \(D2F\)](#) on page 36

Related topics

Basics

- [Basics on Standard I/O \(DS4004 Features !\[\]\(633dd45d48d71eb51a85c6dd83ee51e9\_img.jpg\)\)](#)
- [Writing to a Digital Output via RTLib Functions \(DS4004 Features !\[\]\(bdddf9191a284aa0945448444083c5b0\_img.jpg\)\)](#)

References

Base Address of the I/O Board.....	8
Digital Outputs (PHS Bus System Hardware Reference  )	
ds4004_bit_out.....	27
ds4004_bit_out32.....	29
ds4004_digout_init.....	21

# Run-Time Functions

**Introduction** To check a port's VBAT status.

## ds4004\_vbat\_status\_get

**Syntax**

```
__INLINE UInt32 ds4004_vbat_status_get (  
    phs_addr_t base,  
    UInt32 port,  
    UInt32 rail)
```

**Include file** ds4004.h

**Purpose** To read the status of a VBAT rail of a specified port.

**Description**

This function reads the current status of the selected supply voltage rail on the specified port. The return value indicates an overvoltage or an undervoltage on the VBAT rails.

- An undervoltage disables high-side switches of the affected rail.
- An overvoltage disables all switches of the affected rail.

Due to hysteresis, switches are not enabled at precisely the border of the permitted voltage range.

**Parameters**

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**rail** Selects the rail whose status is returned. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_VBAT1	1	VBAT 1 rail
DS4004_VBAT2	2	VBAT 2 rail

**Return value**

Indicates the status of the selected VBAT rail. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_VBAT_OK	0x0	VBATx voltage is in the permitted range ( $5\text{ V} \leq \text{VBATx} \leq 60\text{ V}$ )
DS4004_UNDERVOLTAGE	0x1	Indicates an undervoltage ( $\text{VBATx} < 5\text{ V}$ )
DS4004_OVERVOLTAGE	0x2	Indicates an overvoltage ( $\text{VBATx} > 60\text{ V}$ )

**Execution times**

For information, refer to [Function Execution Times](#) on page 65.

**Example**

```
UInt32 status;
// Read VBAT1 status of port 1
status = ds4004_vbat_status_get(
    DS4004_1_BASE,          /* select first DS4004 */
    1,                      /* select port 1 */
    DS4004_VBAT1);          /* get status of VBAT1 */
```

**Related topics****References**

[Base Address of the I/O Board..... 8](#)

# Digital I/O Functions

**Introduction** To access the board's digital I/O unit.

**Where to go from here**

**Information in this section**

<a href="#">Code Examples.....</a>	<a href="#">16</a>
To show how to access the board's digital I/O channels.	
<a href="#">Initialization Functions.....</a>	<a href="#">19</a>
To initialize the board's digital I/O unit for read or write access.	
<a href="#">Run-Time Functions.....</a>	<a href="#">24</a>
To perform read or write access to digital I/O channels.	

# Code Examples

## Introduction

To show how to access the board's digital I/O channels.

## Where to go from here

## Information in this section

### [Example of Digital Input Functionality..... 16](#)

This example demonstrates how to read from the board's digital I/O channels.

### [Example of Digital Output Functionality..... 17](#)

This example demonstrates how to write to the board's digital I/O channels.

## Example of Digital Input Functionality

## Introduction

This example demonstrates how to read from the board's digital I/O channels.

## Demo

```
#include <brtENV.h> /* basic real-time environment */
#include <ds4004.h> /* DS4004 library */

int main()
{
    UInt32 val; /* store digital value */

    init(); /* initialize hardware system */
    ds4004_init(DS4004_1_BASE); /* initialize DS4004 board */

    msg_info_set(MSG_SM_RTLIB, 0, "System started.");

    /* initialize digital input port 1 using all 32 bit of port 1 */
    ds4004_digin_init(DS4004_1_BASE, /* base address of first DS4004 */
        1, /* select port 1 */
        0xFFFFFFFF, /* select all 32 channels */
        2.5); /* set input threshold to 2.5V */

    /* read complete port 1 */
    val = ds4004_bit_in32(DS4004_1_BASE, 1);
    msg_info_printf(MSG_SM_RTLIB, 0, "Port1: 0x%X", val);

    /* read channel 2 and channel 3 of port 1 */
    val = ds4004_bit_in(DS4004_1_BASE, 1, DS4004_MASK_CH2 | DS4004_MASK_CH3);
    msg_info_printf(MSG_SM_RTLIB, 0, "Read Value: 0x%X", val);
}
```



```

/* Check if channel 8 of port 1 is set */
if(ds4004_bit_in(DS4004_1_BASE, 1, DS4004_MASK_CH8))
{
    msg_info_printf(MSG_SM_RTLIB, 0, "Bit8 is set");
}
else
{
    msg_info_printf(MSG_SM_RTLIB, 0, "Bit8 is clear");
}

while(1) /* background process */
{
    RTLIB_BACKGROUND_SERVICE();
}

} /* main() */

```

## Related topics

### Basics

Basics on Standard I/O (DS4004 Features )  
 Reading a Digital Input via RTLlib Functions (DS4004 Features )

### References

ds4004_bit_in.....	24
ds4004_bit_in32.....	26
ds4004_digin_init.....	19
ds4004_init.....	9

## Example of Digital Output Functionality

### Introduction

This example demonstrates how to write to the board's digital I/O channels.

### Demo

```

#include <brtenv.h> /* basic real-time environment */
#include <ds4004.h> /* DS4004 Library */

int main()
{
    init(); /* initialize hardware system */
    ds4004_init(DS4004_1_BASE); /* initialize DS4004 board */

    msg_info_set(MSG_SM_RTLIB, 0, "System started.");

    /* initialize digital output port 1 using all 32 bit of the port */
    ds4004_digout_init(DS4004_1_BASE, /* base address of first DS4004 */
                      1, /* select port 1 */
                      0xFFFFFFFF, /* select all 32 channels */
                      DS4004_LS_ENABLE, /* enable low-side switch */
                      DS4004_HS_VBAT1_ENABLE); /* enable high-side sw. VBAT1 */

```

```

/* set initial value on port 1 */
ds4004_bit_out32(DS4004_1_BASE, 1, 0x00000000); /* set all bits to 0 */

/* global enable of port 1 */
ds4004_digout_mode_set(DS4004_1_BASE,
                      DS4004_MASK_PORT1,
                      DS4004_DIGOUT_ENABLE);

RTLIB_TIC_DELAY(1.0); /* wait for 1 second */

/* write to single bits of port 1: set bit 1 and bit 32 to 1 */
ds4004_bit_out(DS4004_1_BASE,
              1,
              DS4004_MASK_CH1 | DS4004_MASK_CH32,
              0x80000001);

RTLIB_TIC_DELAY(1.0); /* wait for 1 second */

/* write to all bits of port 1: set port value to 0xAAAA5555 */
ds4004_bit_out32(DS4004_1_BASE,
                1,
                0xAAAA5555);

while(1) /* background process */
{
    RTLIB_BACKGROUND_SERVICE();
}

} /* main() */

```

## Related topics

### Basics

[Basics on Standard I/O \(DS4004 Features !\[\]\(0f848bbd71cef6b345273b16f905912a\_img.jpg\)\)](#)

[Writing to a Digital Output via RTLib Functions \(DS4004 Features !\[\]\(339a16584d5da0f0a3ca4e9ec17bf6a1\_img.jpg\)\)](#)

### References

<a href="#">ds4004_bit_out.....</a>	<a href="#">27</a>
<a href="#">ds4004_bit_out32.....</a>	<a href="#">29</a>
<a href="#">ds4004_digout_init.....</a>	<a href="#">21</a>
<a href="#">ds4004_digout_mode_set.....</a>	<a href="#">10</a>
<a href="#">ds4004_init.....</a>	<a href="#">9</a>

# Initialization Functions

Introduction	To initialize the board's digital I/O unit for read or write access.
--------------	--

Where to go from here	Information in this section
	<div><a href="#">ds4004_digin_init</a>..... 19 To set up the digital input mode of the selected channels on a specified port.</div> <div><a href="#">ds4004_digout_init</a>..... 21 To setup the digital output mode of the selected channels on a specified port.</div>

## ds4004\_digin\_init

Syntax	<pre>void ds4004_digin_init (     phs_addr_t base,     UInt32 port,     UInt32 ch_mask,     dsfloat threshold)</pre>
--------	--

Include file	ds4004.h
--------------	----------

Purpose	To set up the digital input mode of the selected channels on a specified port.
---------	--

Description	<p>This function sets the I/O functionality of the selected channels (parameter <b>ch_mask</b>) of the specified port (parameter <b>port</b>) to digital input mode. It also sets the corresponding input threshold level.</p> <p>If more than one channel is selected by using a logical OR operator for the channel mask, all the selected channels are set up identically.</p> <p>If you want to set parameters individually, call <b>ds4004_digin_init</b> for each channel individually. This function overwrites the existing initializations of the selected channels.</p>
-------------	---

Parameters	<b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 8.
------------	---

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**ch\_mask** Specifies the channel bitmask (0x00000000 ... 0xFFFFFFFF) for the selected port. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_MASK_CH1	0x00000001	Channel 1
DS4004_MASK_CH2	0x00000002	Channel 2
DS4004_MASK_CH3	0x00000004	Channel 3
DS4004_MASK_CH4	0x00000008	Channel 4
DS4004_MASK_CH5	0x00000010	Channel 5
DS4004_MASK_CH6	0x00000020	Channel 6
DS4004_MASK_CH7	0x00000040	Channel 7
DS4004_MASK_CH8	0x00000080	Channel 8
...	...	...
DS4004_MASK_CH31	0x40000000	Channel 31
DS4004_MASK_CH32	0x80000000	Channel 32

Use a logical OR operation to select more than one channel channel-wise.

**threshold** Specifies the input threshold level (1 V ... 23.8 V) used for the selected channels. The resolution is 0.1 V.

---

**Return value** None

---

**Execution times** For information, refer to [Function Execution Times](#) on page 65.

---

#### Example

```
// Set channels 1 and 2 of port 1 to digital input with 10 V threshold
ds4004_digin_init(
    DS4004_1_BASE,           /* select first DS4004 */
    1,                       /* select port 1 */
    DS4004_MASK_CH1 | DS4004_MASK_CH2, /* select channels 1 and 2 */
    10.0);                   /* set threshold to 10.0 volt */
```


For a more detailed example, refer to [Example of Digital Input Functionality](#) on page 16.

Related topics

Basics

- [Basics on Standard I/O \(DS4004 Features !\[\]\(86b7331e04fe40a56bcff2e9c065738b\_img.jpg\)\)](#)
- [Reading a Digital Input via RTLib Functions \(DS4004 Features !\[\]\(92f87f30b7499b35d0173f4346c498d6\_img.jpg\)\)](#)

References

Base Address of the I/O Board.....	8
ds4004_bit_in.....	24
ds4004_bit_in32.....	26
Function Execution Times.....	65
I/O Circuits and Electrical Characteristics (PHS Bus System Hardware Reference  )	

ds4004\_digout\_init

Syntax

```
void ds4004_digout_init (
    phs_addr_t base,
    UInt32 port,
    UInt32 ch_mask,
    UInt32 ls_config,
    UInt32 hs_config)
```

Include file

ds4004.h

Purpose

To setup the digital output mode of the selected channels on a specified port.

Description

This function sets the I/O functionality of the selected channels (parameter **ch\_mask**) of the specified port (parameter **port**) to digital output mode. The configurations of high-side and low-side switches are set.

If you select more than one channel by using the logical OR operator for the channel mask, all the selected channels are setup identically. If you want to set parameters individually, call **ds4004\_digout\_init** for every channel individually. This function overwrites the existing initializations of the selected channels.

The output state of each I/O channel depends on its individual settings for the low-side switch and the high-side switches. You can use the supply rails L (GND), H1 (VBAT1), and H2 (VBAT2).

- If you set the low-side switch L (GND), the digital output channel is set to low-side switch mode.
- If you set the high-side switch H1 (VBAT1) or H2 (VBAT2), the digital output channel is actively driven in high-side switch mode.

- If you set the high-side switches H1 (VBAT1) *and* H2 (VBAT2), the digital output channel is also actively driven in high-side switch mode, but the output voltage is driven to the highest supply voltage (VBAT1 or VBAT2).
- If you set low-side switch L (GND) and the high-side switches H1 (VBAT1) *and/or* H2 (VBAT2), the digital output channel is actively driven in push-pull mode. Push-pull driver mode means that the output source is actively driven to both high and low level.

## Parameters

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**ch\_mask** Specifies the channel bitmask (0x00000000 ... 0xFFFFFFFF) for the selected port. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_MASK_CH1	0x00000001	Channel 1
DS4004_MASK_CH2	0x00000002	Channel 2
DS4004_MASK_CH3	0x00000004	Channel 3
DS4004_MASK_CH4	0x00000008	Channel 4
DS4004_MASK_CH5	0x00000010	Channel 5
DS4004_MASK_CH6	0x00000020	Channel 6
DS4004_MASK_CH7	0x00000040	Channel 7
DS4004_MASK_CH8	0x00000080	Channel 8
...	...	...
DS4004_MASK_CH31	0x40000000	Channel 31
DS4004_MASK_CH32	0x80000000	Channel 32

Use a logical OR operation to select more than one channel channel-wise.

**ls\_config** Specifies the low-side switch setting for the selected channels. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_LS_DISABLE	0	Low-side switch disabled
DS4004_LS_ENABLE	1	Low-side switch enabled

**hs\_config** Specifies the high-side switch settings for the selected channels. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_HS_DISABLE	0	High-side switches disabled
DS4004_HS_VBAT1_ENABLE	1	High-side switch to VBAT1
DS4004_HS_VBAT2_ENABLE	2	High-side switch to VBAT2
DS4004_HS_VBAT12_ENABLE	3	High-side switch to VBAT1 and VBAT2

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

Execution times	For information, refer to <a href="#">Function Execution Times</a> on page 65.
-----------------	--

Example

```
// Set channel 1 and channel 2 of port 1 to digital output with
// Low-side and high-side1 switches enabled
ds4004_digout_init(
    DS4004_1_BASE,                /* select first DS4004 */
    1,                            /* select port 1 */
    DS4004_MASK_CH1 | DS4004_MASK_CH2, /* select channels 1 and 2 */
    DS4004_LS_ENABLE,             /* use Low-side switch */
    DS4004_HS_VBAT1_ENABLE);      /* use high-side switch VBAT1 */
```


For a more detailed example, refer to [Example of Digital Output Functionality](#) on page 17.

Related topics

Basics

[Basics on Standard I/O \(DS4004 Features !\[\]\(5dc7c9690f4dcd34cd856af73dafd0a6\_img.jpg\)](#))  
[Writing to a Digital Output via RTLib Functions \(DS4004 Features !\[\]\(95f36e31cdb1a43835207b277a635db7\_img.jpg\)](#))

References

Base Address of the I/O Board.....	8
ds4004_bit_out.....	27
ds4004_bit_out32.....	29
ds4004_digout_mode_set.....	10
I/O Circuits and Electrical Characteristics (PHS Bus System Hardware Reference 	

## Run-Time Functions

### Introduction

To perform read or write access to digital I/O channels.

### Where to go from here

### Information in this section

<a href="#">ds4004_bit_in.....</a>	<a href="#">24</a>
To read selected bits of the 32-bit digital I/O line of a specified port.	
<a href="#">ds4004_bit_in32.....</a>	<a href="#">26</a>
To read all 32 bits of the 32-bit digital I/O line of a selected port.	
<a href="#">ds4004_bit_out.....</a>	<a href="#">27</a>
To write data to selected bits of the 32-bit digital I/O line of a specified port.	
<a href="#">ds4004_bit_out32.....</a>	<a href="#">29</a>
To write data to all 32 bits of the 32-bit digital I/O line of a specified port.	

## ds4004\_bit\_in

### Syntax

```
__INLINE UInt32 ds4004_bit_in (
    phs_addr_t base,
    UInt32 port,
    UInt32 ch_mask)
```

### Include file

ds4004.h

### Purpose

To read selected bits of the 32-bit digital I/O line of a specified port.

### Description

The 32-bit digital I/O lines of the specified port are read and the bits specified by the `ch_mask` parameter are returned via the return value.

To use the digital input functionality, the channels must be initialized as digital inputs by calling `ds4004_digin_init`. The current output state of channels programmed as outputs is read back.

### Parameters

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.



**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**ch\_mask** Specifies the channel bitmask (0x00000000 ... 0xFFFFFFFF) for the selected port. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_MASK_CH1	0x00000001	Channel 1
DS4004_MASK_CH2	0x00000002	Channel 2
DS4004_MASK_CH3	0x00000004	Channel 3
DS4004_MASK_CH4	0x00000008	Channel 4
DS4004_MASK_CH5	0x00000010	Channel 5
DS4004_MASK_CH6	0x00000020	Channel 6
DS4004_MASK_CH7	0x00000040	Channel 7
DS4004_MASK_CH8	0x00000080	Channel 8
...	...	...
DS4004_MASK_CH31	0x40000000	Channel 31
DS4004_MASK_CH32	0x80000000	Channel 32

Use a logical OR operation to select more than one channel channel-wise.

---

**Return value** Returns the value of the logical AND operation between the `ch_mask` parameter and the input value of the specified port.

---

**Execution times** For information, refer to [Function Execution Times](#) on page 65.

---

### Example

```
UInt32 value;
// Read channel 1 and channel 2 of port 1
value = ds4004_bit_in(
    DS4004_1_BASE,           /* select first DS4004 */
    1,                       /* select port 1 */
    DS4004_MASK_CH1 | DS4004_MASK_CH2); /* select channels 1 and 2 */
```

For a more detailed example, refer to [Example of Digital Input Functionality](#) on page 16.

## Related topics

## Basics

[Basics on Standard I/O \(DS4004 Features !\[\]\(99f58673407353e96a019fbca558fd72\_img.jpg\)\)](#)  
[Reading a Digital Input via RTLib Functions \(DS4004 Features !\[\]\(2113e5cba4d11862fa536c379e9b61cd\_img.jpg\)\)](#)

## References

[Base Address of the I/O Board..... 8](#)  
[ds4004\\_bit\\_in32..... 26](#)  
[ds4004\\_digin\\_init..... 19](#)  
[I/O Circuits and Electrical Characteristics \(PHS Bus System Hardware Reference !\[\]\(339a16584d5da0f0a3ca4e9ec17bf6a1\_img.jpg\)\)](#)

## ds4004\_bit\_in32

## Syntax

```
__INLINE UInt32 ds4004_bit_in32 (
    phs_addr_t base,
    UInt32 port)
```

## Include file

ds4004.h

## Purpose

To read all 32 bits of the 32-bit digital I/O line of a selected port.

## Description

The 32-bit digital I/O lines of the specified port are read and returned through the return value.

To use the digital input functionality, the channels must be initialized as digital inputs by calling `ds4004_digin_init`. The current output state of channels programmed as outputs is read back.

## Parameters

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

## Return value

Returns the input value of the specified port.

## Execution times

For information, refer to [Function Execution Times](#) on page 65.

**Example**

```

UInt32 value;
// Read complete port 1
value = ds4004_bit_in32(
    DS4004_1_BASE,                /* select first DS4004 */
    1);                          /* select port 1 */


```

For a more detailed example, refer to [Example of Digital Input Functionality](#) on page 16.

**Related topics****Basics**

[Basics on Standard I/O \(DS4004 Features !\[\]\(cbe2492b119e39e02a1dab2af4a4b296\_img.jpg\)\)](#)  
[Reading a Digital Input via RTLib Functions \(DS4004 Features !\[\]\(2f36c159ea3670f7a62f64a4f1cf5c05\_img.jpg\)\)](#)

**References**

Base Address of the I/O Board..... 8  
 ds4004\_bit\_in..... 24  
 ds4004\_digin\_init..... 19  
 I/O Circuits and Electrical Characteristics (PHS Bus System Hardware Reference )

## ds4004\_bit\_out

**Syntax**

```

__INLINE void ds4004_bit_out (
    phs_addr_t base,
    UInt32 port,
    UInt32 ch_mask,
    UInt32 data)

```

**Include file**

ds4004.h

**Purpose**

To write data to selected bits of the 32-bit digital I/O line of a specified port.

**Description**

The data bits specified by the ch\_mask parameter are written to the 32-bit digital I/O line of a specified port. All other bits remain unchanged. The written value will appear at the corresponding output if its mode is set to digital output. The value of channels initialized as inputs or timing outputs will be ignored.

**Note**

- All digital outputs are high impedance after reset. Outputs are enabled by the `ds4004_digout_mode_set` function.
- Before the digital output functionality is used, the corresponding channels must be initialized as digital output by calling `ds4004_digout_init`.

**Parameters**

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**ch\_mask** Specifies the channel bitmask (0x00000000 ... 0xFFFFFFFF) for the selected port. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_MASK_CH1	0x00000001	Channel 1
DS4004_MASK_CH2	0x00000002	Channel 2
DS4004_MASK_CH3	0x00000004	Channel 3
DS4004_MASK_CH4	0x00000008	Channel 4
DS4004_MASK_CH5	0x00000010	Channel 5
DS4004_MASK_CH6	0x00000020	Channel 6
DS4004_MASK_CH7	0x00000040	Channel 7
DS4004_MASK_CH8	0x00000080	Channel 8
...	...	...
DS4004_MASK_CH31	0x40000000	Channel 31
DS4004_MASK_CH32	0x80000000	Channel 32

Use a logical OR operation to select more than one channel channel-wise.

**data** Specifies the data to be written.

The single bits of this value correspond to the related channels, i.e., the LSB corresponds to channel 1 and the MSB corresponds to channel 32.

**Return value**

None

**Execution times**

For information, refer to [Function Execution Times](#) on page 65.

**Example**

```
// Set channel 1, clear channel 2 of port 1, all other channels are unaffected
ds4004_bit_out(
    DS4004_1_BASE,                /* select first DS4004 */
    1,                             /* select port 1 */
    DS4004_MASK_CH1 | DS4004_MASK_CH2, /* select channels 1 and 2 */
    0x00000001);                  /* set channel 1, clear ch.2 */
```

For a more detailed example, refer to [Example of Digital Output Functionality](#) on page 17.

## Related topics

### Basics

[Basics on Standard I/O \(DS4004 Features !\[\]\(e2376d476d06eb31946dc01a69a4403a\_img.jpg\)\)](#)  
[Writing to a Digital Output via RTLib Functions \(DS4004 Features !\[\]\(bbb3388d591ef640dd8a8c4262f2866a\_img.jpg\)\)](#)

### References

<a href="#">Base Address of the I/O Board.....</a>	<a href="#">8</a>
<a href="#">ds4004_bit_out32.....</a>	<a href="#">29</a>
<a href="#">ds4004_digout_init.....</a>	<a href="#">21</a>
<a href="#">ds4004_digout_mode_set.....</a>	<a href="#">10</a>

## ds4004\_bit\_out32

### Syntax

```
__INLINE void ds4004_bit_out32 (
    phs_addr_t base,
    UInt32 port,
    UInt32 data)
```

### Include file

ds4004.h

### Purpose

To write data to all 32 bits of the 32-bit digital I/O line of a specified port.

### Description

The value data is written to the digital I/O line of a specified port. The written value will appear at the corresponding output, if its mode is set to digital output. The value of channels initialized as inputs or timing outputs will be ignored.

#### Note

- All digital outputs are high impedance after reset. Outputs are enabled by the `ds4004_digout_mode_set` function.
- Before the digital output functionality is used, the corresponding channels must be initialized as digital output by calling `ds4004_digout_init`.

### Parameters

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**data** Specifies the data to be written.

The single bits of this value correspond to the related channels, i.e., the LSB corresponds to channel 1 and the MSB corresponds to channel 32.

**Return value** None

**Execution times** For information, refer to [Function Execution Times](#) on page 65.

**Example**

```
// Set complete port 1 to 0xAAAA5555
ds4004_bit_out32(
    DS4004_1_BASE,           /* select first DS4004 */
    1,                       /* select port 1 */
    0xAAAA5555);             /* set 32 bit port to 0xAAAA5555 */
```

For a more detailed example, refer to [Example of Digital Output Functionality](#) on page 17.

**Related topics**

**Basics**

- [Basics on Standard I/O \(DS4004 Features !\[\]\(c580b67c7cd5c9e9e19f04ff6d5093e0\_img.jpg\)\)](#)
- [Writing to a Digital Output via RTLib Functions \(DS4004 Features !\[\]\(81f7c93ea32d9f7160f5d63859611838\_img.jpg\)\)](#)

**References**

<a href="#">Base Address of the I/O Board.....</a>	<a href="#">8</a>
<a href="#">ds4004_bit_out.....</a>	<a href="#">27</a>
<a href="#">ds4004_digout_init.....</a>	<a href="#">21</a>
<a href="#">ds4004_digout_mode_set.....</a>	<a href="#">10</a>

# Timing I/O Functions

**Introduction** To access the board's timing I/O unit.

**Where to go from here**

**Information in this section**

<a href="#">Code Examples.....</a>	<a href="#">32</a>
To show how to access the board's timing I/O unit.	
<a href="#">Initialization Functions.....</a>	<a href="#">38</a>
To initialize the board's timing I/O unit.	
<a href="#">Run-Time Functions.....</a>	<a href="#">51</a>
To use the board's timing I/O features.	

## Code Examples

### Introduction

To show how to access the board's timing I/O unit.

### Where to go from here

### Information in this section

<a href="#">Example of PWM Signal Measurement (PWM2D).....</a>	<a href="#">32</a>
This example demonstrates how to measure a PWM signal.	
<a href="#">Example of PWM Signal Generation (D2PWM).....</a>	<a href="#">33</a>
This example demonstrates how to generate a PWM signal.	
<a href="#">Example of Frequency Measurement (F2D).....</a>	<a href="#">34</a>
This example demonstrates how to measure the frequency of a square-wave signal.	
<a href="#">Example of Square-Wave Signal Generation (D2F).....</a>	<a href="#">36</a>
This example demonstrates how to generate a square-wave signal.	

## Example of PWM Signal Measurement (PWM2D)

### Introduction

This example demonstrates how to measure a PWM signal.

### Demo

```
#include <brtenv.h>                /* basic real-time environment */
#include <ds4004.h>                /* DS4004 Library */

int main()
{
    dsfloat period, duty;

    init;                          /* initialize hardware system */
    ds4004_init(DS4004_1_BASE);    /* initialize DS4004 board */

    msg_info_set(MSG_SM_RTLIB, 0, "System started.");

    /* initialize PWM input port 1 channel 1 */
    ds4004_pwm2d_init(DS4004_1_BASE, /* base address of first DS4004 */
                      1,             /* select port 1 */
                      DS4004_MASK_CH1, /* select channel 1 */
                      2.5,           /* set input threshold to 2.5V */
                      DS4004_TIMING_RANGE1, /* use timing range 1 */
                      DS4004_PWM2D); /* use normal D2PWM mode */
}
```



```

while(1)
{
    /* get period and duty cycle of port 1 channel 1 */
    ds4004_pwm2d(DS4004_1_BASE,          /* baseaddress of first DS4004 */
                1,                      /* select port 1 */
                1,                      /* select channel 1 */
                &period,                /* address where to write period */
                &duty);                 /* address where to write duty */

    msg_info_printf(MSG_SM_RTLIB, 0, "Read: Period %f, %f", period, duty);

    RTLIB_TIC_DELAY(100e-3);            /* delay 100 ms */

    RTLIB_BACKGROUND_SERVICE();
}
} /* main() */

```

## Related topics

### Basics

[Basics on PWM Signal Measurement \(PWM2D\) \(DS4004 Features !\[\]\(c3d993ca47bfe2a953c700506ce31fa0\_img.jpg\)\)](#)  
[Measuring a PWM Signal via RTLib Functions \(DS4004 Features !\[\]\(c468cde8f04e2e2a6ba3c2a373e05c45\_img.jpg\)\)](#)

### References

[ds4004\\_init.....9](#)  
[ds4004\\_pwm2d.....51](#)  
[ds4004\\_pwm2d\\_init.....38](#)

## Example of PWM Signal Generation (D2PWM)

### Introduction

This example demonstrates how to generate a PWM signal.

### Demo

```

#include <brtENV.h>                      /* basic real-time environment */
#include <ds4004.h>                      /* DS4004 Library */

int main()
{
    init;                               /* initialize hardware system */
    ds4004_init(DS4004_1_BASE);         /* initialize DS4004 board */

    msg_info_set(MSG_SM_RTLIB, 0, "System started.");
}

```

```

/* initialize PWM output on port 1 channel 1 */
ds4004_d2pwm(DS4004_1_BASE,          /* base address of first DS4004 */
             1,                      /* select port 1 */
             DS4004_MASK_CH1,        /* select channel 1 */
             DS4004_LS_ENABLE,        /* enable low-side switch */
             DS4004_HS_VBAT1_ENABLE, /* enable high-side switch VBAT1 */
             DS4004_TIMING_RANGE1,    /* use timing range 1 */
             DS4004_D2PWM);           /* normal PWM generation */

/* global enable of port 1 */
ds4004_digout_mode_set(DS4004_1_BASE,
                      DS4004_MASK_PORT1,
                      DS4004_DIGOUT_ENABLE);

/* set period and duty cycle for port 1 channel 1 */
ds4004_d2pwm(DS4004_1_BASE,          /* base address of first DS4004 */
             1,                      /* select port 1 */
             1,                      /* select channel 1 */
             1e-3,                   /* Period = 1 ms */
             0.2);                   /* duty cycle = 20 % */



while(1)                             /* background process */
{
    RTLIB_BACKGROUND_SERVICE();
}

} /* main() */

```

## Related topics

### Basics

Basics on PWM Signal Generation (D2PWM) (DS4004 Features )  
 Generating a PWM Signal via RTLib Functions (DS4004 Features )

### References

ds4004_d2pwm.....	53
ds4004_d2pwm_init.....	41
ds4004_digout_mode_set.....	10
ds4004_init.....	9

## Example of Frequency Measurement (F2D)

### Introduction

This example demonstrates how to measure the frequency of a square-wave signal.

**Demo**

This demo application shows the usage of the frequency measurement functionality of the DS4004.

```
#include <brtenv.h> /* basic real-time environment */
#include <ds4004.h> /* DS4004 Library */

int main()
{
    dsfloat frequency;

    init; /* initialize hardware system */
    ds4004_init(DS4004_1_BASE); /* initialize DS4004 board */

    msg_info_set(MSG_SM_RTLIB, 0, "System started.");

    /* initialize frequency input port 1 channel 1 */
    ds4004_f2d_init(DS4004_1_BASE, /* base address of first DS4004 */
                    1, /* select Port 1 */
                    DS4004_MASK_CH1, /* select Channel 1 */
                    2.5, /* set input threshold to 2.5V */
                    DS4004_TIMING_RANGE1); /* use Timing Range 1 */

    while(1)
    {
        /* frequency of port 1 channel 1 */
        frequency = ds4004_f2d(DS4004_1_BASE, /* base address of first DS4004 */
                                1, /* select port 1 */
                                1); /* select channel 1 */

        msg_info_printf(MSG_SM_RTLIB, 0, "Read: Frequency %f", frequency);

        RTLIB_TIC_DELAY(100e-3); /* delay 100 ms */

        RTLIB_BACKGROUND_SERVICE();
    }
} /* main() */
```

**Related topics****Basics**

Basics on Frequency Measurement (F2D) (DS4004 Features )  
 Measuring the Signal Frequency via RTLlib Functions (DS4004 Features )

**References**

ds4004_f2d.....	55
ds4004_f2d_init.....	44
ds4004_init.....	9

## Example of Square-Wave Signal Generation (D2F)

### Introduction

This example demonstrates how to generate a square-wave signal.

### Demo

This demo application shows the usage of the frequency generation functionality of the DS4004.

```
#include <brtenv.h> /* basic real-time environment */
#include <ds4004.h> /* DS4004 Library */

int main()
{
    init; /* initialize hardware system */
    ds4004_init(DS4004_1_BASE); /* initialize DS4004 board */

    msg_info_set(MSG_SM_RTLIB, 0, "System started.");

    /* initialize frequency generation on port 1 channel 1 */
    ds4004_d2f(DS4004_1_BASE, /* base address of first DS4004 */
               1, /* select port 1 */
               DS4004_MASK_CH1, /* select channel 1 */
               DS4004_LS_ENABLE, /* enable low-side switch */
               DS4004_HS_VBAT1_ENABLE, /* enable high-side switch to VBAT1 */
               DS4004_TIMING_RANGE1, /* use timing range 1 */
               DS4004_D2F_LOW); /* set output to low for too small
                                frequencies */

    /* global enable of Port 1 */
    ds4004_digout_mode_set(DS4004_1_BASE,
                           DS4004_MASK_PORT1,
                           DS4004_DIGOUT_ENABLE);

    /* set frequency for port 1 channel 1 */
    ds4004_d2f(DS4004_1_BASE, /* base address of first DS4004 */
               1, /* select port 1 */
               1, /* select channel 1 */
               1000); /* frequency = 1 kHz */

    while(1) /* background process */
    {
        RTLIB_BACKGROUND_SERVICE();
    }
} /* main() */
```

Related topics

Basics

- [Basics on Square-Wave Signal Generation \(D2F\) \(DS4004 Features !\[\]\(86b7331e04fe40a56bcff2e9c065738b\_img.jpg\)](#))
- [Generating a Square-Wave Signal via RTLib Functions \(DS4004 Features !\[\]\(92f87f30b7499b35d0173f4346c498d6\_img.jpg\)](#))

References

<a href="#">ds4004_d2f.....</a>	<a href="#">57</a>
<a href="#">ds4004_d2f_init.....</a>	<a href="#">47</a>
<a href="#">ds4004_digout_mode_set.....</a>	<a href="#">10</a>
<a href="#">ds4004_init.....</a>	<a href="#">9</a>

# Initialization Functions

## Introduction

To initialize the board's timing I/O unit.

## Where to go from here

## Information in this section

<a href="#">ds4004_pwm2d_init</a> .....	38
To set up the PWM input mode for the selected channels of a specified port.	
<a href="#">ds4004_d2pwm_init</a> .....	41
To set up the PWM output mode for the selected channels of a specified port.	
<a href="#">ds4004_f2d_init</a> .....	44
To setup the frequency measurement mode for the selected channels of a specified port.	
<a href="#">ds4004_d2f_init</a> .....	47
To set up the frequency output mode for the selected channels of a specified port.	

## ds4004\_pwm2d\_init

## Syntax

```
void ds4004_pwm2d_init (
    phs_addr_t base,
    UInt32 port,
    UInt32 ch_mask,
    dsfloat threshold,
    UInt32 range,
    UInt32 mode)
```

## Include file

ds4004.h

## Purpose

To set up the PWM input mode for the selected channels of a specified port.

## Description

This function sets the I/O functionality for the selected channels (parameter **ch\_mask**) of the specified port (parameter **port**) to PWM Input mode. It also sets the corresponding input threshold (parameter **threshold**), and the range (parameter **range**) and update mode (parameter **mode**) of the timing I/O unit. If more than one channel is selected by using a logical OR operator for the channel

mask, all the selected channels are identically. If you want to set parameters individually, call `ds4004_pwm2d_init` for every channel individually. This function overwrites the existing initializations of the selected channels.

## Parameters

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**ch\_mask** Specifies the channel bitmask (0x00000000 ... 0xFFFFFFFF) for the selected port. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_MASK_CH1	0x00000001	Channel 1
DS4004_MASK_CH2	0x00000002	Channel 2
DS4004_MASK_CH3	0x00000004	Channel 3
DS4004_MASK_CH4	0x00000008	Channel 4
DS4004_MASK_CH5	0x00000010	Channel 5
DS4004_MASK_CH6	0x00000020	Channel 6
DS4004_MASK_CH7	0x00000040	Channel 7
DS4004_MASK_CH8	0x00000080	Channel 8
...	...	...
DS4004_MASK_CH31	0x40000000	Channel 31
DS4004_MASK_CH32	0x80000000	Channel 32

Use a logical OR operation to select more than one channel channel-wise.

**threshold** Specifies the input threshold level (1 V ... 23.8 V) used for the selected channels. The resolution is 0.1 V.

**range** Specifies the period range of the timing I/O unit in the range 1 ... 16. The setting applies to all the selected channels. The following symbols are predefined:

Predefined Symbol	Value	Meaning			
		Minimum Period		Maximum Period	Resolution
		Theoretical	Practical		
DS4004_TIMING_RANGE1	1	200 ns	6.7 $\mu$ s	3.27 ms	50 ns
DS4004_TIMING_RANGE2	2	400 ns	6.7 $\mu$ s	6.55 ms	100 ns
DS4004_TIMING_RANGE3	3	800 ns	6.7 $\mu$ s	13.1 ms	200 ns
DS4004_TIMING_RANGE4	4	1.6 $\mu$ s	6.7 $\mu$ s	26.2 ms	400 ns
DS4004_TIMING_RANGE5	5	3.2 $\mu$ s	6.7 $\mu$ s	52.4 ms	800 ns
DS4004_TIMING_RANGE6	6	6.4 $\mu$ s	6.7 $\mu$ s	104 ms	1.6 $\mu$ s
DS4004_TIMING_RANGE7	7	12.8 $\mu$ s	12.8 $\mu$ s	209 ms	3.2 $\mu$ s
DS4004_TIMING_RANGE8	8	25.6 $\mu$ s	25.6 $\mu$ s	419 ms	6.4 $\mu$ s
DS4004_TIMING_RANGE9	9	51.2 $\mu$ s	51.2 $\mu$ s	838 ms	12.8 $\mu$ s
DS4004_TIMING_RANGE10	10	103 $\mu$ s	103 $\mu$ s	1.67 s	25.6 $\mu$ s

Predefined Symbol	Value	Meaning			
		Minimum Period		Maximum Period	Resolution
		Theoretical	Practical		
DS4004_TIMING_RANGE11	11	205 $\mu$ s	205 $\mu$ s	3.35 s	51.2 $\mu$ s
DS4004_TIMING_RANGE12	12	410 $\mu$ s	410 $\mu$ s	6.71 s	103 $\mu$ s
DS4004_TIMING_RANGE13	13	820 $\mu$ s	820 $\mu$ s	13.4 s	205 $\mu$ s
DS4004_TIMING_RANGE14	14	1.64 ms	1.64 ms	26.8 s	410 $\mu$ s
DS4004_TIMING_RANGE15	15	3.28 ms	3.28 ms	53.6 s	820 $\mu$ s
DS4004_TIMING_RANGE16	16	6.56 ms	6.56 ms	107.3 s	1.64 ms

**Note****Signal periods and resolution**

Each high period and each low period of the measured signal must be longer (not equal) than the resolution to avoid missing pulses.

**mode** Update mode of the PWM measurement. The setting applies to all the selected channels. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_PWM2D	0	The measured values are updated at the end of each $T_{\text{high}}$ and $T_{\text{low}}$ period of the PWM signal. The update is asynchronous to the period.
DS4004_PWM2D_SYNC_UPDATE	1	The measured values are updated at the end of each $T_{\text{low}}$ period of the PWM signal only. The update is synchronous to the period.

**Return value** None

**Execution times** For information, refer to [Function Execution Times](#) on page 65.

**Example**

```
// Set channel 1 and channel 2 of port 1 to PWM input with
// threshold 10 V using timing range 1 and mode PWM2D for
// all selected channels
ds4004_pwm2d_init(
    DS4004_1_BASE,                /* select first DS4004 */
    1,                            /* select port 1 */
    DS4004_MASK_CH1 | DS4004_MASK_CH2, /* select channels 1 and 2 */
    10.0,                        /* set threshold to 10.0 volt */
    DS4004_TIMING_RANGE1,        /* use timing range 1 */
    DS4004_PWM2D);               /* use asynchronous pwm mode */
```



For a more detailed example, refer to [Example of PWM Signal Measurement \(PWM2D\)](#) on page 32.

Related topics

Basics

- [Basics on PWM Signal Measurement \(PWM2D\) \(DS4004 Features !\[\]\(f15d3c54be60b4fd0ce1da9fb3f67256\_img.jpg\)\)](#)
- [Measuring a PWM Signal via RTLib Functions \(DS4004 Features !\[\]\(7bf135d42c40a6430c927b2fd03d7659\_img.jpg\)\)](#)

References

<a href="#">Base Address of the I/O Board.....</a>	<a href="#">8</a>
<a href="#">ds4004_pwm2d.....</a>	<a href="#">51</a>

ds4004\_d2pwm\_init

Syntax

```
void ds4004_d2pwm_init (
    phs_addr_t base,
    UInt32 port,
    UInt32 ch_mask,
    UInt32 ls_config,
    UInt32 hs_config,
    UInt32 range,
    UInt32 mode)
```

Include file

ds4004.h

Purpose

To set up the PWM output mode for the selected channels of a specified port.

Description

This function sets the I/O functionality for the selected channels (parameter **ch\_mask**) of the specified port (parameter **port**) to PWM output mode. It also sets the range (parameter **range**) and update mode (parameter **mode**) of the timing I/O unit, and the configurations of high-side and low-side switches (parameters **ls\_config** and **hs\_config**).

If more than one channel is selected by using a logical OR operator for the channel mask, all the selected channels are identically. If you want to set parameters individually, call **ds4004\_d2pwm\_init** for every channel individually. This function overwrites the existing initializations of the selected channels.

The output state of each I/O channel depends on its individual settings for the low-side switch and the high-side switches. You can use the supply rails L (GND), H1 (VBAT1), and H2 (VBAT2).

- If you set the low-side switch L (GND), the digital output channel is set to low-side switch mode.
- If you set the high-side switch H1 (VBAT1) *or* H2 (VBAT2), the digital output channel is actively driven in high-side switch mode.
- If you set the high-side switches H1 (VBAT1) *and* H2 (VBAT2), the digital output channel is also actively driven in high-side switch mode, but the output voltage is driven to the highest supply voltage (VBAT1 or VBAT2).
- If you set low-side switch L (GND) and the high-side switches H1 (VBAT1) *and/or* H2 (VBAT2), the digital output channel is actively driven in push-pull mode. Push-pull driver mode means that the output source is actively driven to both high and low level.

#### Parameters

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**ch\_mask** Specifies the channel bitmask (0x00000000 ... 0xFFFFFFFF) for the selected port. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_MASK_CH1	0x00000001	Channel 1
DS4004_MASK_CH2	0x00000002	Channel 2
DS4004_MASK_CH3	0x00000004	Channel 3
DS4004_MASK_CH4	0x00000008	Channel 4
DS4004_MASK_CH5	0x00000010	Channel 5
DS4004_MASK_CH6	0x00000020	Channel 6
DS4004_MASK_CH7	0x00000040	Channel 7
DS4004_MASK_CH8	0x00000080	Channel 8
...	...	...
DS4004_MASK_CH31	0x40000000	Channel 31
DS4004_MASK_CH32	0x80000000	Channel 32

Use a logical OR operation to select more than one channel channel-wise.

**ls\_config** Specifies the low-side switch setting for the selected channels. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_LS_DISABLE	0	Low-side switch disabled
DS4004_LS_ENABLE	1	Low-side switch enabled

**hs\_config** Specifies the high-side switch settings for the selected channels.

The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_HS_DISABLE	0	High-side switches disabled
DS4004_HS_VBAT1_ENABLE	1	High-side switch to VBAT1
DS4004_HS_VBAT2_ENABLE	2	High-side switch to VBAT2
DS4004_HS_VBAT12_ENABLE	3	High-side switch to VBAT1 and VBAT2

**range** Specifies the period range of the timing I/O unit in the range 1 ... 16. The setting applies to all the selected channels.

The following symbols are predefined:

Predefined Symbol	Value	Meaning			
		Minimum Period		Maximum Period	Resolution
		Theoretical	Practical		
DS4004_TIMING_RANGE1	1	100 ns	6.7 $\mu$ s	3.27 ms	50 ns
DS4004_TIMING_RANGE2	2	200 ns	6.7 $\mu$ s	6.55 ms	100 ns
DS4004_TIMING_RANGE3	3	400 ns	6.7 $\mu$ s	13.1 ms	200 ns
DS4004_TIMING_RANGE4	4	800 ns	6.7 $\mu$ s	26.2 ms	400 ns
DS4004_TIMING_RANGE5	5	1.6 $\mu$ s	6.7 $\mu$ s	52.4 ms	800 ns
DS4004_TIMING_RANGE6	6	3.2 $\mu$ s	6.7 $\mu$ s	104 ms	1.6 $\mu$ s
DS4004_TIMING_RANGE7	7	6.4 $\mu$ s	6.7 $\mu$ s	209 ms	3.2 $\mu$ s
DS4004_TIMING_RANGE8	8	12.8 $\mu$ s	12.8 $\mu$ s	419 ms	6.4 $\mu$ s
DS4004_TIMING_RANGE9	9	25.6 $\mu$ s	25.6 $\mu$ s	838 ms	12.8 $\mu$ s
DS4004_TIMING_RANGE10	10	51.2 $\mu$ s	51.2 $\mu$ s	1.67 s	25.6 $\mu$ s
DS4004_TIMING_RANGE11	11	103 $\mu$ s	103 $\mu$ s	3.35 s	51.2 $\mu$ s
DS4004_TIMING_RANGE12	12	205 $\mu$ s	205 $\mu$ s	6.71 s	103 $\mu$ s
DS4004_TIMING_RANGE13	13	410 $\mu$ s	410 $\mu$ s	13.4 s	205 $\mu$ s
DS4004_TIMING_RANGE14	14	820 $\mu$ s	820 $\mu$ s	26.8 s	410 $\mu$ s
DS4004_TIMING_RANGE15	15	1.64 ms	1.64 ms	53.6 s	820 $\mu$ s
DS4004_TIMING_RANGE16	16	3.28 ms	3.28 ms	107.3 s	1.64 ms

**mode** Specifies the update mode of the PWM generation unit. The setting applies to all the selected channels.

The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_D2PWM	0	New values for the period and/or the duty cycle are updated immediately.
DS4004_D2PWM_SYNC_UPDATE	4	New values for the period and/or the duty cycle are updated at the next rising edge of the PWM output signal. The update is synchronous for constant period values only.

---

**Return value** None

---

**Execution times** For information, refer to [Function Execution Times](#) on page 65.

---

**Example**

```
// Set channel 1 and channel 2 of port 1 to PWM output with
// Low-side and high-side1 switches enabled. Using timing
// range 1 and mode DS4004_D2PWM
ds4004_d2pwm_init(
    DS4004_1_BASE,,           /* select first DS4004 */
    1,                        /* select port 1 */
    DS4004_MASK_CH1 | DS4004_MASK_CH2, /* select channels 1 and 2 */
    DS4004_LS_ENABLE,         /* use Low-side switch */
    DS4004_HS_VBAT1_ENABLE,    /* use high-side switch VBAT1 */
    DS4004_TIMING_RANGE1,      /* use timing range 1 */
    DS4004_D2PWM);            /* use asynchronous pwm mode */
```

For a more detailed example, refer to [Example of PWM Signal Generation \(D2PWM\)](#) on page 33.

---

**Related topics**

**Basics**

[Basics on PWM Signal Generation \(D2PWM\) \(DS4004 Features !\[\]\(05be7c7a8995decd503647c99211f7c2\_img.jpg\)](#))  
[Generating a PWM Signal via RTLib Functions \(DS4004 Features !\[\]\(16cd6e1a39784ecf52b4db09f4865f40\_img.jpg\)](#))

**References**

[Base Address of the I/O Board.....8](#)  
[ds4004\\_d2pwm.....53](#)

## ds4004\_f2d\_init

**Syntax**

```
void ds4004_f2d_init (
    phs_addr_t base,
    UInt32 port,
    UInt32 ch_mask,
    dsfloat threshold,
    UInt32 range)
```

---

**Include file** ds4004.h

---

**Purpose** To setup the frequency measurement mode for the selected channels of a specified port.

**Description**

This function sets the I/O functionality for the selected channels (parameter **ch\_mask**) of the specified port (parameter **port**) to frequency measurement mode. It also sets the corresponding input threshold (parameter **threshold**) and the range (parameter **range**) of the timing I/O unit.

If more than one channel is selected by using a logical OR operator for the channel mask, all the selected channels are identically. This function overwrites the existing initializations of the selected channels.

**Parameters**

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**ch\_mask** Specifies the channel bitmask (0x00000000 ... 0xFFFFFFFF) for the selected port. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_MASK_CH1	0x00000001	Channel 1
DS4004_MASK_CH2	0x00000002	Channel 2
DS4004_MASK_CH3	0x00000004	Channel 3
DS4004_MASK_CH4	0x00000008	Channel 4
DS4004_MASK_CH5	0x00000010	Channel 5
DS4004_MASK_CH6	0x00000020	Channel 6
DS4004_MASK_CH7	0x00000040	Channel 7
DS4004_MASK_CH8	0x00000080	Channel 8
...	...	...
DS4004_MASK_CH31	0x40000000	Channel 31
DS4004_MASK_CH32	0x80000000	Channel 32

Use a logical OR operation to select more than one channel channel-wise.

**threshold** Specifies the input threshold level (1 V ... 23.8 V) used for the selected channels. The resolution is 0.1 V.

**range** Specifies the frequency range of the timing I/O unit in the range 1 ... 16. The setting applies to all the selected channels.

The following symbols are predefined:

Predefined Symbol	Value	Meaning		
		Minimum Frequency	Maximum Frequency	Resolution
DS4004_TIMING_RANGE1	1	9.54 Hz	150 kHz	50 ns
DS4004_TIMING_RANGE2	2	4.77 Hz	150 kHz	100 ns
DS4004_TIMING_RANGE3	3	2.39 Hz	150 kHz	200 ns
DS4004_TIMING_RANGE4	4	1.20 Hz	150 kHz	400 ns
DS4004_TIMING_RANGE5	5	0.60 Hz	150 kHz	800 ns

Predefined Symbol	Value	Meaning		
		Minimum Frequency	Maximum Frequency	Resolution
DS4004_TIMING_RANGE6	6	0.30 Hz	150 kHz	1.6 $\mu$ s
DS4004_TIMING_RANGE7	7	0.15 Hz	150 kHz	3.2 $\mu$ s
DS4004_TIMING_RANGE8	8	75 mHz	78.12 kHz	6.4 $\mu$ s
DS4004_TIMING_RANGE9	9	38 mHz	39.06 kHz	12.8 $\mu$ s
DS4004_TIMING_RANGE10	10	19 mHz	19.53 kHz	25.6 $\mu$ s
DS4004_TIMING_RANGE11	11	10 mHz	9.76 kHz	51.2 $\mu$ s
DS4004_TIMING_RANGE12	12	5.0 mHz	4.88 kHz	103 $\mu$ s
DS4004_TIMING_RANGE13	13	2.5 mHz	2.44 kHz	205 $\mu$ s
DS4004_TIMING_RANGE14	14	1.2 mHz	1.22 kHz	410 $\mu$ s
DS4004_TIMING_RANGE15	15	0.6 mHz	610.35 Hz	820 $\mu$ s
DS4004_TIMING_RANGE16	16	0.3 mHz	305.17 Hz	1.64 ms

**Note****Signal periods and resolution**

Each high period and each low period of the measured signal must be longer (not equal) than the resolution to avoid missing pulses.

**Return value** None

**Execution times** For information, refer to [Function Execution Times](#) on page 65.

**Example**

```
// Set channel 1 and channel 2 of port 1 to frequency input
// with threshold 10V. Using timing range 1 for all selected
// channels.
ds4004_f2d_init(
    DS4004_1_BASE,                /* select first DS4004 */
    1,                            /* select port 1 */
    DS4004_MASK_CH1 | DS4004_MASK_CH2, /* select channels 1 and 2 */
    10.0,                         /* set threshold to 10.0 volt */
    DS4004_TIMING_RANGE1);        /* use timing range 1 */
```

For a more detailed example, refer to [Example of Frequency Measurement \(F2D\)](#) on page 34.

Related topics

Basics

- [Basics on Frequency Measurement \(F2D\) \(DS4004 Features !\[\]\(a88007b249b36c75dcbde101f514cec3\_img.jpg\)\)](#)
- [Measuring the Signal Frequency via RTLib Functions \(DS4004 Features !\[\]\(800628c068083563f747129d8b339031\_img.jpg\)\)](#)

References

Base Address of the I/O Board.....	8
ds4004_f2d.....	55

ds4004\_d2f\_init

Syntax

```
void ds4004_d2f_init (  
    phs_addr_t base,  
    UInt32 port,  
    UInt32 ch_mask,  
    UInt32 ls_config,  
    UInt32 hs_config,  
    UInt32 range,  
    UInt32 mode)
```

Include file

ds4004.h

Purpose

To set up the frequency output mode for the selected channels of a specified port.

Description

This function sets the I/O functionality for the selected channels (parameter **ch\_mask**) of the specified port (parameter **port**) to frequency output mode. It also sets the range (parameter **range**) and mode (parameter **mode**) of the timing I/O unit, and the configurations of high-side and low-side switches (parameters **ls\_config** and **hs\_config**).

If more than one channel is selected by using a logical OR operator for the channel mask, all the selected channels are identically. If you want to set parameters individually call **ds4004\_d2f\_init** for every channel individually. This function overwrites the existing initializations of the selected channels.

The output state of each I/O channel depends on its individual settings for the low-side switch and the high-side switches. You can use the supply rails L (GND), H1 (VBAT1), and H2 (VBAT2).

- If you set the low-side switch L (GND), the digital output channel is set to low-side switch mode.

- If you set the high-side switch H1 (VBAT1) or H2 (VBAT2), the digital output channel is actively driven in high-side switch mode.
- If you set the high-side switches H1 (VBAT1) *and* H2 (VBAT2), the digital output channel is also actively driven in high-side switch mode, but the output voltage is driven to the highest supply voltage (VBAT1 or VBAT2).
- If you set low-side switch L (GND) and the high-side switches H1 (VBAT1) *and/or* H2 (VBAT2), the digital output channel is actively driven in push-pull mode. Push-pull driver mode means that the output source is actively driven to both high and low level.

## Parameters

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**ch\_mask** Specifies the channel bitmask (0x00000000 ... 0xFFFFFFFF) for the selected port. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_MASK_CH1	0x00000001	Channel 1
DS4004_MASK_CH2	0x00000002	Channel 2
DS4004_MASK_CH3	0x00000004	Channel 3
DS4004_MASK_CH4	0x00000008	Channel 4
DS4004_MASK_CH5	0x00000010	Channel 5
DS4004_MASK_CH6	0x00000020	Channel 6
DS4004_MASK_CH7	0x00000040	Channel 7
DS4004_MASK_CH8	0x00000080	Channel 8
...	...	...
DS4004_MASK_CH31	0x40000000	Channel 31
DS4004_MASK_CH32	0x80000000	Channel 32

Use a logical OR operation to select more than one channel channel-wise.

**ls\_config** Specifies the low-side switch setting for the selected channels. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_LS_DISABLE	0	Low-side switch disabled
DS4004_LS_ENABLE	1	Low-side switch enabled

**hs\_config** Specifies the high-side switch settings for the selected channels. The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_HS_DISABLE	0	High-side switches disabled
DS4004_HS_VBAT1_ENABLE	1	High-side switch to VBAT1
DS4004_HS_VBAT2_ENABLE	2	High-side switch to VBAT2



Predefined Symbol	Value	Meaning
DS4004_HS_VBAT12_ENABLE	3	High-side switch to VBAT1 and VBAT2

**range** Specifies the frequency range of the timing I/O unit in the range 1 ... 16. The setting applies to all the selected channels. The following symbols are predefined:

Predefined Symbol	Value	Meaning		
		Minimum Frequency	Maximum Frequency	Resolution
DS4004_TIMING_RANGE1	1	9.54 Hz	150 kHz	100 ns
DS4004_TIMING_RANGE2	2	4.77 Hz	150 kHz	200 ns
DS4004_TIMING_RANGE3	3	2.39 Hz	150 kHz	400 ns
DS4004_TIMING_RANGE4	4	1.20 Hz	150 kHz	800 ns
DS4004_TIMING_RANGE5	5	0.60 Hz	150 kHz	1.6 $\mu$ s
DS4004_TIMING_RANGE6	6	0.30 Hz	150 kHz	3.2 $\mu$ s
DS4004_TIMING_RANGE7	7	0.15 Hz	150 kHz	6.4 $\mu$ s
DS4004_TIMING_RANGE8	8	75 mHz	78.12 kHz	12.8 $\mu$ s
DS4004_TIMING_RANGE9	9	38 mHz	39.06 kHz	25.6 $\mu$ s
DS4004_TIMING_RANGE10	10	19 mHz	19.53 kHz	51.2 $\mu$ s
DS4004_TIMING_RANGE11	11	10 mHz	9.76 kHz	103 $\mu$ s
DS4004_TIMING_RANGE12	12	5.0 mHz	4.88 kHz	205 $\mu$ s
DS4004_TIMING_RANGE13	13	2.5 mHz	2.44 kHz	410 $\mu$ s
DS4004_TIMING_RANGE14	14	1.2 mHz	1.22 kHz	820 $\mu$ s
DS4004_TIMING_RANGE15	15	0.6 mHz	610.35 Hz	1.64 ms
DS4004_TIMING_RANGE16	16	0.3 mHz	305.17 Hz	3.28 ms

**mode** Specifies the *zero frequency mode* of the timing generation unit. The setting applies to all the selected channels.

The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_D2F_LOW	0	Square-wave signal generation, the output is set to low level when the written frequency is smaller than the minimum frequency based on the frequency range setting.
DS4004_D2F_HIGH	1	Square-wave signal generation, the output is set to high level, when the written frequency is smaller than the minimum frequency based on the frequency range setting.
DS4004_D2F_HOLD	2	Square-wave signal generation, the output holds the last signal level (low or high), when the written frequency is smaller than the minimum frequency based on the frequency range setting.

Return value	None
Execution times	For information, refer to <a href="#">Function Execution Times</a> on page 65.

Example

```
// Set channel 1 and channel 2 of port 1 to frequency output
// with low-side and high-side 1 switches enabled. Using
// timing range 1 and mode DS4004_D2F_LOW.
ds4004_d2f_init(
    DS4004_1_BASE,           /* select first DS4004 */
    1,                       /* select port 1 */
    DS4004_MASK_CH1 | DS4004_MASK_CH2, /* select channels 1 and 2 */
    DS4004_LS_ENABLE,       /* use low-side switch */
    DS4004_HS_VBAT1_ENABLE, /* use high-side switch VBAT1 */
    DS4004_TIMING_RANGE1,   /* use timing range 1 */
    DS4004_D2F_LOW);       /* use D2F_LOW mode */
```

For a more detailed example, refer to [Example of Square-Wave Signal Generation \(D2F\)](#) on page 36.

Related topics

Basics

[Basics on Square-Wave Signal Generation \(D2F\) \(DS4004 Features !\[\]\(eb70ac3e793aea24b49d2cd9f7b0b269\_img.jpg\)\)](#)  
[Generating a Square-Wave Signal via RTLib Functions \(DS4004 Features !\[\]\(cf2d3def06148aad259395138469f3b5\_img.jpg\)\)](#)

References

[Base Address of the I/O Board.....8](#)  
[ds4004\\_d2f.....57](#)

# Run-Time Functions

## Introduction

To use the board's timing I/O features.

## Where to go from here

## Information in this section

<a href="#">ds4004_pwm2d</a> .....	51
To capture the PWM period and duty cycle of the specified PWM input channel of a selected port.	
<a href="#">ds4004_d2pwm</a> .....	53
To update the PWM period and duty cycle of the specified PWM output channel.	
<a href="#">ds4004_f2d</a> .....	55
To measure the frequency of a square-wave signal.	
<a href="#">ds4004_d2f</a> .....	57
To set the frequency of the square-wave signal on the specified channel of a selected port.	

## ds4004\_pwm2d

## Syntax

```
__INLINE void ds4004_pwm2d (
    phs_addr_t base,
    UInt32 port,
    UInt32 channel,
    dsfloat *period,
    dsfloat *duty)
```

## Include file

ds4004.h

## Purpose


To capture the PWM period and duty cycle of the specified PWM input channel of a selected port.

## Description

The function measures the PWM period and duty cycle of the specified input channel. For information on the available range and mode, refer to [Basics on PWM Signal Measurement \(PWM2D\) \(DS4004 Features !\[\]\(2bae76de5ebbd5c4d7d47162f1673734\_img.jpg\)](#)).

The frequency range can be set by calling `ds4004_pwm2d_init` during initialization.

#### Note

- Before the PWM input functionality is used, the corresponding channels must be initialized as PWM input by calling `ds4004_pwm2d_init`.
- To minimize the quantization effects on the frequency resolution and duty cycle, you should select the period range with the best possible resolution (resolution values as small as possible). For detailed information, refer to [Basics on PWM Signal Measurement \(PWM2D\)](#) (DS4004 Features .

#### Parameters

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**channel** Specifies the channel number (1 ... 32) for the specified port.

**period** Address where the measured period is written. The value is stated in seconds.

**duty** Address where the measured duty cycle is written. The duty cycle is scaled to the range 0 ... 1.0. The following table shows the relation to the duty cycle stated in percent:

Scaling	Relation
0.0	0%
...	...
0.5	50%
...	...
1.0	100%

#### Return value

None

#### Execution times

For information, refer to [Function Execution Times](#) on page 65.

#### Example

```
dsfloat period, duty;
// Measure PWM signal of port 1 channel 1
ds4004_pwm2d(
    DS4004_1_BASE,           /* select first DS4004 */
    1,                       /* select port 1 */
    1,                       /* select channel 1 */
    &period,                 /* address to store read period */
    &duty);                  /* address to store read duty cycle */
```

For a more detailed example, refer to [Example of PWM Signal Measurement \(PWM2D\)](#) on page 32.

## Related topics

### Basics

[Basics on PWM Signal Measurement \(PWM2D\) \(DS4004 Features !\[\]\(e2376d476d06eb31946dc01a69a4403a\_img.jpg\)\)](#)  
[Measuring a PWM Signal via RTLib Functions \(DS4004 Features !\[\]\(bbb3388d591ef640dd8a8c4262f2866a\_img.jpg\)\)](#)

### References

[Base Address of the I/O Board..... 8](#)  
[ds4004\\_pwm2d\\_init..... 38](#)

## ds4004\_d2pwm

### Syntax

```
__INLINE void ds4004_d2pwm (
    phs_addr_t base,
    UInt32 port,
    UInt32 channel,
    dsfloat period,
    dsfloat duty)
```

### Include file

ds4004.h

### Purpose

To update the PWM period and duty cycle of the specified PWM output channel.

### Description

The function sets the PWM period and duty cycle of the specified output channel. For information on the available range and mode, refer to [Basics on PWM Signal Generation \(D2PWM\) \(DS4004 Features !\[\]\(41aea2746216b27a6939d696d8e035da\_img.jpg\)\)](#).

You can specify the frequency range and the update mode for the period and/or the duty cycle by calling `ds4004_d2pwm_init` during initialization.

#### Note

- All digital outputs are high impedance after reset. Outputs are enabled using the `ds4004_digout_mode_set` function.
- Before the PWM output functionality is used, the corresponding channels must be initialized as PWM output by calling `ds4004_d2pwm_init`.
- To minimize the quantization effect on the frequency resolution and the duty cycle, you should select the period range with the best possible resolution (resolution values as small as possible). For detailed information, refer to [Basics on PWM Signal Generation \(D2PWM\) \(DS4004 Features !\[\]\(756219e9389f679d57027482aa5cf5fc\_img.jpg\)](#)).

#### Parameters

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**channel** Specifies the channel number (1 ... 32) for the specified port.

**period** Specifies the PWM period in seconds. Values should remain within the selected period range (refer to `ds4004_d2pwm_init` on page 41). For information on PWM signal generation and its restrictions, refer to [Basics on PWM Signal Generation \(D2PWM\) \(DS4004 Features !\[\]\(e1d6102fe77919492c04879c8450f1f5\_img.jpg\)](#)). The period is saturated to minimum/maximum value based on the timing range.

**duty** Specifies the PWM duty cycle in the range 0 ... 1.0. The following table shows the relation to the duty cycle given in percent:

Scaling	Relation
0.0	0%
...	...
0.5	50%
...	...
1.0	100%

The duty cycle is saturated to minimum/maximum value.

#### Return value

None

#### Execution times

For information, refer to [Function Execution Times](#) on page 65.

**Example**

```
// Set PWM signal of port 1 channel 1 with period = 1ms,
// dutycycle = 50%
ds4004_d2pwm(
    DS4004_1_BASE,           /* select first DS4004 */
    1,                       /* select port 1 */
    1,                       /* select channel 1 */
    1e-3,                    /* set period to 1 ms */
    0.5);                   /* set duty cycle to 50% */
```

For a more detailed example, refer to [Example of PWM Signal Generation \(D2PWM\)](#) on page 33.

**Related topics****Basics**

[Basics on PWM Signal Generation \(D2PWM\) \(DS4004 Features !\[\]\(8d0f0e0fe25b320c33272c52aec1fbca\_img.jpg\)](#))  
[Generating a PWM Signal via RTLib Functions \(DS4004 Features !\[\]\(c1e4487e48462435243c9e117557e045\_img.jpg\)](#))

**References**

[Base Address of the I/O Board..... 8](#)  
[ds4004\\_d2pwm\\_init..... 41](#)

## ds4004\_f2d

**Syntax**

```
__INLINE dsfloat ds4004_f2d (
    phs_addr_t base,
    UInt32 port,
    UInt32 channel)
```

**Include file**

ds4004.h

**Purpose**


To measure the frequency of a square-wave signal.

**Description**

The function measures the signal frequency of the specified input channel. The frequency value is scaled to Hz and returned by the return value. The resolution of the frequency signal is 21 bit and depends on the selected prescaler setting. For information on the available range, refer to [Basics on Frequency Measurement \(F2D\) \(DS4004 Features !\[\]\(683dba75afe26e28cd4de5730b776760\_img.jpg\)](#)).

The frequency ranges can be set by calling `ds4004_f2d_init` during initialization.

**Note**

- Before the F2D functionality is used, the corresponding channels must be initialized as frequency input by calling `ds4004_f2d_init`.
- To minimize the quantization effect on the frequency resolution, you should select the frequency range with the best possible resolution (resolution values as small as possible). For detailed information, refer to [Basics on Frequency Measurement \(F2D\)](#) (DS4004 Features .

Parameters

**base**

Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port**

Specifies the port number (1 ... 3) for the selected DS4004 board.

**channel**

Specifies the channel number (1 ... 32) for the specified port.

Return value

Measured frequency in Hz.

Execution times

For information, refer to [Function Execution Times](#) on page 65.

**Example**

```
dsfloat frequency;
// Measure frequency of port 1 channel 1
frequency = ds4004_f2d(
    DS4004_1_BASE,           /* select first DS4004 */
    1,                       /* select port 1 */
    1);                      /* select channel 1 */
```

For a more detailed example, refer to [Example of Frequency Measurement \(F2D\)](#) on page 34.

**Related topics**

Basics

Basics on Frequency Measurement (F2D) (DS4004 Features )

Measuring the Signal Frequency via RTLib Functions (DS4004 Features )

References

Base Address of the I/O Board..... 8

ds4004\_f2d\_init..... 44



## ds4004\_d2f

### Syntax

```
__INLINE void ds4004_d2f(
    phs_addr_t base,
    UInt32 port,
    UInt32 channel,
    dsfloat frequency)
```

### Include file

ds4004.h

### Purpose

To set the frequency of the square-wave signal on the specified channel of a selected port.

### Description

The function outputs a digital signal with the specified frequency on the appropriate output channel. The resolution of the frequency signal is 20 bit and depends on the selected prescaler setting. For information on the available range, refer to [Basics on Square-Wave Signal Generation \(D2F\) \(DS4004 Features !\[\]\(faf942dc3e59ce8eb64b4ac481eca7e0\_img.jpg\)](#)). The frequency ranges can be set during initialization by calling `ds4004_d2f_init`.

#### Note

- All digital outputs are high impedance after reset. Outputs are enabled using the `ds4004_digout_mode_set` function.
- Before the D2F functionality is used, the corresponding channels must be initialized as frequency output by calling `ds4004_d2f_init`.
- To minimize the quantization effect on the frequency resolution, you should select the frequency range with the best possible resolution (resolution values as small as possible). For detailed information, refer to [Basics on Square-Wave Signal Generation \(D2F\) \(DS4004 Features !\[\]\(564903337f30b845a5f6979939a95fe6\_img.jpg\)](#).

### Parameters

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**channel** Specifies the channel number (1 ... 32) for the specified port.

**frequency** Specifies the frequency of the generated signal in Hz. The value is saturated to minimum/maximum value depending on the selected timing range. New values for the frequency are updated immediately.

### Return value

None

Execution times

For information, refer to [Function Execution Times](#) on page 65.

Example

```
// Set frequency of port 1 channel 1 to 1 kHz
ds4004_d2f(
    DS4004_1_BASE,          /* select first DS4004 */
    1,                      /* select port 1 */
    1,                      /* select channel 1 */
    1000);                  /* set frequency to 1 kHz */
```

For a more detailed example, refer to [Example of Square-Wave Signal Generation \(D2F\)](#) on page 36.

Related topics

Basics

- [Basics on Square-Wave Signal Generation \(D2F\) \(DS4004 Features !\[\]\(97faa0168e491544be255cfcab218e9b\_img.jpg\)](#))
- [Generating a Square-Wave Signal via RTLib Functions \(DS4004 Features !\[\]\(b2166b76608b8499cffc130bf1b1fe60\_img.jpg\)](#))

References

<a href="#">Base Address of the I/O Board.....</a>	<a href="#">8</a>
<a href="#">ds4004_d2f_init.....</a>	<a href="#">47</a>

# Interrupt Functions

**Introduction** To use the board's configurable interrupts.

**Where to go from here**

**Information in this section**

<a href="#">Code Examples.....</a>	<a href="#">60</a>
To show how to use specific channels for interrupt control.	
<a href="#">Initialization Functions.....</a>	<a href="#">62</a>
To enable specific input channels for interrupt control.	

## Code Examples

### Introduction

To show how to use specific channels for interrupt control.

## Example of Interrupt Generation

### Introduction

This example demonstrates how to use different interrupts of the DS4004.

### Demo

```

#include <brtenv.h>                                /* basic real-time environment */
#include <ds4004.h>                                /* DS4004 library */

/*-----*/
/* ISR for Interrupt 0 of DS4004 */
void int0()
{
    msg_info_printf(0, 0, "Interrupt 0");
}

/*-----*/
/* ISR for Interrupt 3 of DS4004 */
void int3()
{
    msg_info_set(0, 0, "Interrupt 3");
}

/*-----*/
int main()
{
    init;                                           /* initialize hardware system */
    ds4004_init(DS4004_1_BASE);                    /* initialize DS4004 board */

    msg_info_set(MSG_SM_RTLIB, 0, "System started.");

    /* initialize channel as digital input for use as interrupt source */
    ds4004_digin_init(DS4004_1_BASE,              /* base address of first DS4004 */
                      1,                            /* select port 1 */
                      DS4004_MASK_CH1,             /* select channel 1 */
                      2.5);                        /* set input threshold to 2.5V */

    ds4004_digin_init(DS4004_1_BASE,              /* base address of first DS4004 */
                      2,                            /* select port 2 */
                      DS4004_MASK_CH2,             /* select channel 2 */
                      2.5);                        /* set input threshold to 2.5V */

    /**** enable interrupt for port 1 channel 1 on rising edge (interrupt 0) ****/
    ds4004_int_mode_set(DS4004_1_BASE,           /* base address of first DS4004 */
                        1,                          /* select port 1 */
                        DS4004_MASK_CH1,           /* select channel 1 */
                        DS4004_INT_RISING);        /* select rising edge */

```

```

install_phs_int_vector(DS4004_1_BASE, 0, int0); /* inst. ISR for interrupt 0 */

/** enable interrupt for port 2 channel 2 on Falling Edge (interrupt 3) */
ds4004_int_mode_set(DS4004_1_BASE,          /* base address of first DS4004 */
                   2,                      /* select port 1 */
                   DS4004_MASK_CH2,        /* select channel 1 */
                   DS4004_INT_FALLING);    /* select falling edge */

install_phs_int_vector(DS4004_1_BASE, 3, int3); /* inst. ISR for interrupt 3 */

RTLIB_INT_ENABLE();                          /* global enable of interrupts */

while(1)                                     /* background process */
{
    RTLIB_BACKGROUND_SERVICE();
}

} /* main() */

```

## Related topics

### Basics

[Interrupt Control \(DS4004 Features !\[\]\(e2376d476d06eb31946dc01a69a4403a\_img.jpg\)\)](#)

[Specifying Interrupt Control via RTLib Functions \(DS4004 Features !\[\]\(74d4806277d7e73349d8e8c0897931e9\_img.jpg\)\)](#)

### References

<a href="#">ds4004_digin_init</a> .....	19
<a href="#">ds4004_init</a> .....	9
<a href="#">ds4004_int_mode_set</a> .....	62

# Initialization Functions

**Introduction**

To enable specific input channels for interrupt control.

## ds4004\_int\_mode\_set

**Syntax**

```
void ds4004_int_mode_set (  
    phs_addr_t base,  
    UInt32 port,  
    UInt32 ch_mask,  
    UInt32 mode)
```

**Include file**

ds4004.h

**Purpose**

To initialize the interrupt generation of the DS4004.

**Description**

This function enables/disables the interrupts of the DS4004 board and selects the edge an interrupt is generated on. Interrupts can be generated on channel 1 and channel 2 of each port on the rising edge, on the falling edge or on both edges (rising and falling). The minimum pulse length to detect an interrupt is 2  $\mu$ s, if the threshold is set to half the signal level.

To specify the threshold level for the interrupt signal, refer to [ds4004\\_digin\\_init](#) on page 19, [ds4004\\_pwm2d\\_init](#) on page 38, or [ds4004\\_f2d\\_init](#) on page 44.

To use an interrupt, an interrupt service routine (ISR) must be registered by calling `install_phs_int_vector` and the corresponding channel must be configured as digital input by calling `ds4004_digin_init`, `ds4004_pwm2d_init`, or `ds4004_f2d_init`.

Interrupts are assigned to the channel of the slave interrupt control unit (ICU) as follows:

Port	Channel	Slave ICU Channel
1	1	0
1	2	1
2	1	2
2	2	3
3	1	4
3	2	5

**Parameters**

**base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 8.

**port** Specifies the port number (1 ... 3) for the selected DS4004 board.

**ch\_mask** Specifies the channel bitmask (0x00000000 ... 0x00000003) for the selected port.

The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_MASK_CH1	0x00000001	Channel 1
DS4004_MASK_CH2	0x00000002	Channel 2

Use a logical OR operation to select more than one channel channel-wise.

**mode** Specifies the edge the interrupt is triggered by. Interrupts can be generated on a rising edge, a falling edge and both edges.

The following symbols are predefined:

Predefined Symbol	Value	Meaning
DS4004_INT_DISABLE	0x0	Interrupts disabled
DS4004_INT_FALLING	0x1	Interrupt on falling edge
DS4004_INT_RISING	0x2	Interrupt on rising edge
DS4004_INT_BOTH	0x3	Interrupt on both edges

**Return value**

None

**Execution times**

For information, refer to [Function Execution Times](#) on page 65.

**Example**

```
// Enable interrupt on rising edge for channel 1 on port 1
ds4004_int_mode_set(
    DS4004_1_BASE,           /* select first DS4004 */
    1,                       /* select port 1 */
    DS4004_MASK_CH1,         /* select ch.1 by channel mask */
    DS4004_INT_RISING);      /* enable interrupt on rising edge */
```

For a more detailed example, refer to [Example of Interrupt Generation](#) on page 60.

Related topics

Basics

- [Basics on DS4004 Interrupts \(DS4004 Features !\[\]\(687b6c142f51ac6f390f8bd444e38d03\_img.jpg\)\)](#)
- [Specifying Interrupt Control via RTLib Functions \(DS4004 Features !\[\]\(861b7aaa71df51b93037a486c3b17630\_img.jpg\)\)](#)

HowTos

- [How to Specify Interrupt Generation via RTI \(DS4004 Features !\[\]\(448bd415caa8b52d2aeb4d58499267b2\_img.jpg\)\)](#)

References

- [Base Address of the I/O Board.....8](#)



# Function Execution Times

Introduction	To give you the mean function execution times and basic information on the test environment used.
--------------	---

Where to go from here	<div>Information in this section</div> <div><div>Information on the Test Environment.....65</div><div>To provide information on the test environment because the execution times of the C functions can vary, since they depend on different factors and they are influenced by the test environment used.</div><div>Measured Execution Times.....66</div><div>To get the mean execution times of the board's RTLib functions.</div></div>
-----------------------	--

## Information on the Test Environment

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.
--------------	--

Test environment	<div>The execution time of a function can vary, since it depends on different factors, for example:</div> <div><ul style="list-style-type: none"><li>▪ CPU clock and bus clock frequency of the processor board used</li><li>▪ Optimization level of the compiler and the usage of inlining</li><li>▪ Parameters used</li></ul></div> <div>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</div>
------------------	--

`downxxxx` tool (optimization and inlining). The execution times in the tables below are always the mean measurement values.

#### 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.

The properties of the processor boards used are:

	<b>DS1006</b>
CPU clock	2.6 GHz / 3.0 GHz
Bus clock	133 MHz

#### Related topics

#### References

[Measured Execution Times..... 66](#)

## Measured Execution Times

#### Introduction

The following tables provide the mean execution times of the board's RTLib functions.

#### Overall functions

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

Function	Execution Time	
	<b>DS1006 with 2.6 GHz</b>	<b>DS1006 with 3.0 GHz</b>
<code>ds4004_init</code>	323.30 $\mu$ s	318.70 $\mu$ s
<code>ds4004_int_mode_set</code>	0.77 $\mu$ s	0.78 $\mu$ s
<code>ds4004_digout_mode_set</code>	2.99 $\mu$ s	2.40 $\mu$ s
<code>ds4004_vbat_status_get</code>	0.62 $\mu$ s	0.60 $\mu$ s

#### Digital I/O functions

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

Function	Execution Time	
	<b>DS1006 with 3.0 GHz</b>	<b>DS1006 with 2.6 GHz</b>
<code>ds4004_digin_init</code>	57.07 $\mu$ s	57.63 $\mu$ s
<code>ds4004_digout_init</code>	4.18 $\mu$ s	4.19 $\mu$ s
<code>ds4004_bit_in</code>	0.61 $\mu$ s	0.62 $\mu$ s

Function	Execution Time	
	DS1006 with 3.0 GHz	DS1006 with 2.6 GHz
ds4004_bit_in32	0.61 $\mu$ s	0.62 $\mu$ s
ds4004_bit_out	0.79 $\mu$ s	0.78 $\mu$ s
ds4004_bit_out32	0.02 $\mu$ s	0.02 $\mu$ s

## Timing I/O functions

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

Function	Execution Time	
	DS1006 with 2.6 GHz	DS1006 with 3.0 GHz
ds4004_pwm2d_init	58.77 $\mu$ s	58.27 $\mu$ s
ds4004_d2pwm_init	12.71 $\mu$ s	12.08 $\mu$ s
ds4004_f2d_init	58.77 $\mu$ s	58.25 $\mu$ s
ds4004_d2f_init	12.72 $\mu$ s	12.08 $\mu$ s
ds4004_pwm2d	0.64 $\mu$ s	0.63 $\mu$ s
ds4004_d2pwm	0.08 $\mu$ s	0.07 $\mu$ s
ds4004_f2d	0.64 $\mu$ s	0.63 $\mu$ s
ds4004_d2f	0.06 $\mu$ s	0.05 $\mu$ s

## Related topics

### References

[Information on the Test Environment.....](#) 65



**B**

base address 8

**C**

Common Program Data folder 6

**D**

Documents folder 6

DS4004

    execution times 65

ds4004\_bit\_in 24

ds4004\_bit\_in32 26

ds4004\_bit\_out 27

ds4004\_bit\_out32 29

ds4004\_d2f 57

ds4004\_d2f\_init 47

ds4004\_d2pwm 53

ds4004\_d2pwm\_init 41

ds4004\_digin\_init 19

ds4004\_digout\_init 21

ds4004\_digout\_mode\_set 10

ds4004\_f2d 55

ds4004\_f2d\_init 44

ds4004\_init 9

ds4004\_int\_mode\_set 62

ds4004\_pwm2d 51

ds4004\_pwm2d\_init 38

ds4004\_vbat\_status\_get 13

DSxxxx\_n\_BASE 8

**E**

execution times

    DS4004 65

**L**

Local Program Data folder 6

