

DS3001 Incremental Encoder Interface 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.

© 1999 - 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
Initialization Function	9
ds3001_init.....	9
I/O Functions	11
ds3001_clear_counter.....	12
ds3001_clear_lerr.....	13
ds3001_prepare_index_reset.....	14
ds3001_read_delta_position.....	16
ds3001_read_err.....	17
ds3001_read_index.....	18
ds3001_read_lerr.....	20
ds3001_read_line_count.....	21
ds3001_read_position.....	22
ds3001_set_counter_reset_mode.....	23
ds3001_set_filter_clock.....	25
ds3001_set_line_type.....	27
Function Execution Times	29
Information on the Test Environment.....	29
Measured Execution Times.....	30
Index	33



# About This Reference

## Content

This RTLib Reference (Real-Time Library) gives detailed descriptions of the C functions needed to program a DS3001 Incremental Encoder Interface 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](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.

# 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



# Initialization Function

**Objective**

Before you can use the DS3001 board, you have to perform the initialization process.

**Note**

The initialization function of the processor board must be called before the DS3001 board's initialization function.

## ds3001\_init

**Syntax**

```
void ds3001_init(phs_addr_t base)
```

**Include file**

ds3001.h

**Purpose**

To initialize the DS3001 board.

**Description**

The DS3001 is initialized to the following default settings:

- Single ended lines for all channels
- All position counters are set to zero
- Counter clearing once on index pulse
- 20 MHz filter clock

**Note**

This function must be called before any other DS3001 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	ds3001_init(): Invalid PHS-bus base address 0x???????	The value of 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.
-167	Error	ds3001_init(0x?): Board not found!	No DS3001 board could be found at the specified PHS-bus address. Check whether the DSxxx_n_BASE macro corresponds to the I/O board used.
-168	Error	ds3001_init(0x?): Memory allocation error!	The allocation of some dynamic memory for internal data storage has failed.
-169	Error	ds3001_init(0x?): I/O error detected!	This error occurs if an hardware error has been detected on the DS3001 board. This error can be caused by an cable break in the wiring of the encoder or by a sensor supply voltage failure.
-53	Warning	ds3001_init(0x?): Jumper setup is not matching SW default initialization! STP register: 0x???????? instead of 0x????????	The value of the STP register could not be verified because the jumper setting is not correct.

**Execution times**                      For information, refer to [Function Execution Times](#) on page 29.

**Example**                                  This example shows how to initialize a DS3001 at address DS3001\_1\_BASE:

```
void main(void)
{
    init();
    ds3001_init(DS3001_1_BASE);
    ...
}
```

## Related topics

## References

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

# I/O Functions

## Objective

The DS3001 Incremental Encoder Interface Board features 5 independent incremental encoder interface channels and captures digital position signals.

### Note

You have to initialize the DS3001 board with function `ds3001_init` before you can use one of these functions.

## Where to go from here

## Information in this section

<a href="#">ds3001_clear_counter.....</a>	<a href="#">12</a>
To clear the counter.	
<a href="#">ds3001_clear_lerr.....</a>	<a href="#">13</a>
To clear the latched I/O error flag (LERR).	
<a href="#">ds3001_prepare_index_reset.....</a>	<a href="#">14</a>
To setup the index reset.	
<a href="#">ds3001_read_delta_position.....</a>	<a href="#">16</a>
To read the delta position.	
<a href="#">ds3001_read_err.....</a>	<a href="#">17</a>
To read the I/O error flag (ERR).	
<a href="#">ds3001_read_index.....</a>	<a href="#">18</a>
To read the index pulse.	
<a href="#">ds3001_read_lerr.....</a>	<a href="#">20</a>
To read the latched I/O error flag (LERR).	
<a href="#">ds3001_read_line_count.....</a>	<a href="#">21</a>
To read the position counter as 32-bit word.	
<a href="#">ds3001_read_position.....</a>	<a href="#">22</a>
To read the position scaled to a floating-point value.	

<a href="#">ds3001_set_counter_reset_mode.....</a>	<a href="#">23</a>
To setup the index reset mode.	
<a href="#">ds3001_set_filter_clock.....</a>	<a href="#">25</a>
To setup the noise filter clock.	
<a href="#">ds3001_set_line_type.....</a>	<a href="#">27</a>
To setup the transmission line type of the specified channel.	

## ds3001\_clear\_counter

### Syntax

```
void ds3001_clear_counter(
    phs_addr_t base,
    int channel)
```

### Include file

ds3001.h

### Purpose

To clear the counter of an incremental encoder channel.

### Description

The position counter specified by the parameters **base** and **channel** is cleared by writing a 1 to the corresponding CNTCLR bit in the DS3001's counter reset register.

#### Note

The function `ds3001_init` must be called before this function can be used.

### I/O mapping

For details on the I/O mapping, refer to [Signal Mapping to I/O Pins \(PHS Bus System Hardware Reference !\[\]\(a8f9309f944226d1420f5fed22e2b6e6\_img.jpg\)](#)).

### Parameters

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

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

### Return value

None

**Execution times**

For information, refer to [Function Execution Times](#) on page 29.

**Example**

This example shows how to clear the encoder position counter of channel 1:

```
void sub_fct(void)
{
    ...
    ds3001_clear_counter(DS3001_1_BASE, 1);
    ...
}
```

**Related topics****References**

Base Address of the I/O Board.....	7
ds3001_init.....	9
Macros.....	7

## ds3001\_clear\_lerr

**Syntax**

```
void ds3001_clear_lerr(phis_addr_t base)
```

**Include file**

ds3001.h

**Purpose**

To clear the latched I/O error flag (LERR).

**Description**

This function clears the latched I/O error flag (LERR) by setting the LECLR bit in the counter reset register. The LERR is set if an I/O error has been detected on the DS3001 board. The LERR flag remains set also when the I/O error is no longer active.

**Note**

The function `ds3001_init` 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.

---

**Return value** None

---

**Execution times** For information, refer to [Function Execution Times](#) on page 29.

---

**Example** This example shows how to clear the latched I/O error flag:

```
void sub_fct(void)
{
    ...
    ds3001_clear_lerr(DS3001_1_BASE);
    ...
}
```

---

**Related topics**

**References**

```
Base Address of the I/O Board..... 7
ds3001_init..... 9
Macros..... 7
```

---

## ds3001\_prepare\_index\_reset

---

**Syntax**

```
void ds3001_prepare_index_reset(
    phs_addr_t base,
    int channel)
```

---

**Include file** ds3001.h

---

**Purpose** To prepare the index reset of the incremental encoder channels.

---

**Description**

This function prepares the index reset on the specified incremental encoder channel. After the index reset has been prepared using this function, the encoder counter will be reset after the next index signal. All following index signal will have no effect until the function `ds3001_prepare_index_reset` is called again to prepare next index reset.

**Note**

- The function `ds3001_init` must be called before this function can be used.
- The incremental encoder channel must have been initialized to 'reset once' index reset mode by using the function `ds3001_set_counter_reset_mode`.

**I/O mapping**

For details on the I/O mapping, refer to [Signal Mapping to I/O Pins \(PHS Bus System Hardware Reference !\[\]\(9dfdaff1d86ba3c1f8353b4d1b61b8c5\_img.jpg\)](#)).

**Parameters**

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

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

**Return value**

None

**Execution times**

For information, refer to [Function Execution Times](#) on page 29.

**Example**

This example shows how to use this function:

```
void main(void)
{
    ...
    init();
    ds3001_init();
    ds3001_set_counter_reset_mode(
        DS3001_1_BASE, 1, DS3001_ONCE);
    ds3001_prepare_index_reset(DS3001_1_BASE, 1);
    ...
}
```

The encoder channel 1 is set to 'reset once' index reset mode and the index reset is prepared. The encoder counter of channel 1 will be reset on the next index signal.

**Related topics****References**

Base Address of the I/O Board.....	7
ds3001_init.....	9
ds3001_set_counter_reset_mode.....	23
Macros.....	7

## ds3001\_read\_delta\_position

### Syntax

```
dsfloat ds3001_read_delta_position(
    phs_addr_t base,
    int channel)
```

### Include file

ds3001.h

### Purpose

To read the delta position.

### Description

In some cases the position difference between 2 simulation steps is required, for example for velocity computation. This function returns the position difference relative to the previous call to the function `ds3001_read_delta_position` scaled to the floating-point range  $-1.0 \dots +1.0$ .

#### Note

- The function `ds3001_init` must be called before this function can be used.
- If reset-on-index is set for the specified encoder channel, you have to regard the following situation: When an index has occurred before this function has been executed, the previously read position is set to 0. This causes a deviation between the real and the calculated delta position.

### I/O mapping

For details on the I/O mapping, refer to [Signal Mapping to I/O Pins \(PHS Bus System Hardware Reference !\[\]\(e9474ce1d70442456f8fe9c393ea149c\_img.jpg\)](#)).

### Parameters

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

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

### Return values

The following values are returned:

Value	Meaning
$-1.0 \dots +1.0$	Encoder delta position value

### Execution times

For information, refer to [Function Execution Times](#) on page 29.



**Example**

This example shows how to read the encoder delta position of channel 1 as floating-point value:

```
dsfloat enc_delta;
void sub_fct(void)
{
    ...
    enc_delta = ds3001_read_delta_position(DS3001_1_BASE, 1);
    ...
}
```

**Related topics****References**

Base Address of the I/O Board.....	7
ds3001_init.....	9
Macros.....	7

## ds3001\_read\_err

**Syntax**

```
int ds3001_read_err(phs_addr_t base)
```

**Include file**

ds3001.h

**Purpose**

To read the I/O error flag (ERR).

**Description**

This function reads the state of the I/O error flag (ERR) of the DS3001 board by reading the ERR bit in the ID-register. The ERR flag is active as long an I/O error is present. The ERR flag is cleared if the I/O error is no longer present.

**Note**

The function `ds3001_init` 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.

**Return values**

The following values are returned:

Value	Meaning
0	No I/O error present
1	I/O error condition present

**Execution times**

For information, refer to [Function Execution Times](#) on page 29.

**Example**

This example shows how to use this function:

```
void sub_fct(void)
{
    ...
    if( ds3001_read_err(DS3001_1_BASE) )
        error_handler();
    ...
}
```

If an I/O error on the DS3001 board is present, the user function `error_handler` is called.

**Related topics****References**

<a href="#">Base Address of the I/O Board.....</a>	<a href="#">7</a>
<a href="#">ds3001_init.....</a>	<a href="#">9</a>
<a href="#">Macros.....</a>	<a href="#">7</a>

## ds3001\_read\_index

**Syntax**

```
int ds3001_read_index(
    phs_addr_t base,
    int channel)
```

**Include file**

`ds3001.h`

**Purpose**

To read the index pulse of the incremental encoder channels.

**Note**

The function `ds3001_init` must be called before this function can be used.

The index reset is generated on the rising edge of the index pulse. This means that an index reset is generated when moving into the index position. An index reset is not generated if the index line is already active while an index reset is requested. An index reset interrupt, however, is generated on the falling edge of the index pulse.

### I/O mapping

For details on the I/O mapping, refer to [Signal Mapping to I/O Pins \(PHS Bus System Hardware Reference !\[\]\(4729e517bc6a7cd81c8025b9646574fb\_img.jpg\)](#)).

### Parameters

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

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

### Return values

The following values are returned:

Value	Meaning
0	No index pulse detected
1	Index pulse detected

### Execution times

For information, refer to [Function Execution Times](#) on page 29.

### Example

This example shows how to use this function:

```
void sub_fct(void)
{
    ...
    if( ds3001_read_index(DS3001_1_BASE, 1) )
        user_fct();
    ...
}
```

If an index pulse has been detected on channel 1 the function `user_fct` is called.

### Related topics

#### References

<a href="#">Base Address of the I/O Board</a> .....	7
<a href="#">ds3001_init</a> .....	9
<a href="#">Macros</a> .....	7

## ds3001\_read\_lerr

---

**Syntax**

```
int ds3001_read_lerr(phis_addr_t base)
```

---

**Include file**

```
ds3001.h
```

---

**Purpose**

To read the latched I/O error flag (LERR).

---

**Description**

This function reads the state of the latched I/O error flag (LERR) by reading the LERR bit in the ID-register. The LERR is set if an I/O error has been detected on the DS3001 board. The LERR flag remains set also when the I/O error is no longer active.

**Note**

The function `ds3001_init` 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.

---

**Return values**

The following values are returned:

Value	Meaning
0	No I/O error detected
1	I/O error detected

---

**Execution times**

For information, refer to [Function Execution Times](#) on page 29.

---

**Example**

This example shows how to use this function:

```
void sub_fct(void)
{
    ...
    if( ds3001_read_lerr(DS3001_1_BASE) )
        error_handler();
    ...
}
```

If an I/O error on the DS3001 board has been detected, the user function `error_handler` is called.

## Related topics

## References

Base Address of the I/O Board.....	7
ds3001_init.....	9
Macros.....	7

## ds3001\_read\_line\_count

## Syntax

```
UInt32 ds3001_read_line_count(
    phs_addr_t base,
    int channel)
```

## Include file

ds3001.h

## Purpose

To read the position counter.

**Note**

The function `ds3001_init` must be called before this function can be used.

## I/O mapping

For details on the I/O mapping, refer to [Signal Mapping to I/O Pins \(PHS Bus System Hardware Reference !\[\]\(799877f5c2f906134441300079881630\_img.jpg\)](#)).

## Parameters

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

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

## Return value

This function returns a 24-bit position counter value left-aligned in a 32-bit word.

## Execution times

For information, refer to [Function Execution Times](#) on page 29.

**Example**

This example shows how to read the encoder lines of channel 1 as 24-bit integer value:

```
UInt32 enc_val;
void sub_fct(void)
{
    ...
    enc_val = ds3001_read_line_count(DS3001_1_BASE, 1);
    ...
}
```

**Related topics****References**

Base Address of the I/O Board.....	7
ds3001_init.....	9
Macros.....	7

## ds3001\_read\_position

**Syntax**

```
dsfloat ds3001_read_position(
    phs_addr_t base,
    int channel)
```

**Include file**

ds3001.h

**Purpose**

To read the position counter values.

**Description**

The 24-bit position counter value is scaled to a floating-point value in the range  $-1.0 \dots +1.0$  by the factor  $2^{-31}$  because data is read via the PHS bus left aligned within a 32-bit word.

**Note**

The function `ds3001_init` must be called before this function can be used.

**I/O mapping**

For details on the I/O mapping, refer to [Signal Mapping to I/O Pins \(PHS Bus System Hardware Reference !\[\]\(f9f168a9979beed8b01f8750d577d508\_img.jpg\)](#)).

<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 7.
	<b>channel</b> Specifies the logical channel number within the range 1 ... 5.

**Return value** The following values are returned:

Value	Meaning
-1.0 ... +1.0	Encoder position value

**Execution times** For information, refer to [Function Execution Times](#) on page 29.

**Example** This example shows how to read the encoder position of channel 1 as floating-point value:

```
dsfloat enc_val;
void sub_fct(void)
{
    ...
    enc_val = ds3001_read_position(DS3001_1_BASE, 1);
    ...
}
```

## Related topics

### References

<a href="#">Base Address of the I/O Board</a> .....	7
<a href="#">ds3001_init</a> .....	9
<a href="#">Macros</a> .....	7

# ds3001\_set\_counter\_reset\_mode

**Syntax**

```
void ds3001_set_counter_reset_mode(
    phs_addr_t base,
    int channel,
    int mode)
```

**Include file** ds3001.h

**Purpose**

To select the index-reset mode of the DS3001 board channels.

**Note**

The function `ds3001_init` must be called before this function can be used.

**I/O mapping**

For details on the I/O mapping, refer to [Signal Mapping to I/O Pins \(PHS Bus System Hardware Reference !\[\]\(fa6f3af6bfa46c5d4a2d362681095beb\_img.jpg\)](#)).

**Parameters**

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

**channel** Specifies the logical channel number within the range 1 ... 5. For selecting all 5 channels, you can use the predefined symbol `DS3001_CH_ALL`.

**mode** Specifies the index-reset mode:

Symbol	Meaning
<code>DS3001_ONCE</code>	For resetting counter once on index
<code>DS3001_ALWAYS</code>	For resetting counter always on index

**Return value**

None

**Messages**

The following message is defined:

ID	Type	Message	Description
-50	Error	<code>ds3001_set_counter_reset_mode(0x??)</code> : Board not initialized!	The DS3001 board has not been initialized by a preceding call to the function <code>ds3001_init</code> .

**Execution times**

For information, refer to [Function Execution Times](#) on page 29.

**Example**

This example shows how to initialize channel 1 to reset always on index:

```
void main(void)
{
    init();
    ds3001_init(DS3001_1_BASE);
    ds3001_set_counter_reset_mode(
        DS3001_1_BASE, 1, DS3001_ALWAYS);
    ...
}
```



## Related topics

## References

Base Address of the I/O Board.....	7
ds3001_init.....	9
ds3001_prepare_index_reset.....	14
Macros.....	7

## ds3001\_set\_filter\_clock

## Syntax

```
void ds3001_set_filter_clock(
    phs_addr_t base,
    int channel,
    int clock)
```

## Include file

ds3001.h

## Purpose

To select the noise filter clock for the DS3001 board channel.

**Note**

The function `ds3001_init` must be called before this function can be used.

The required clock frequency depends on the minimum encoder state-width of the incremental sensor. This state-width depends on the encoder resolution, the maximum rotation speed of the shaft and the phase deviation of the encoder. The digital noise filter accepts an encoder state if it is stable for at least 4 clock periods and rejects noise pulses shorter than 3 clock periods. The following table shows the relationship of the clock frequency, the required minimum encoder state-width and the maximum noise pulse width. Note, that the minimum encoder state-width applies to a clean input signal. If noise is present the length of all noise pulses in an encoder state must be added to the minimum encoder state width to obtain the correct clock rate.

Clock	Min. Encoder State-Width	Max. Noise Pulse Width
312.5 kHz	12.8 $\mu$ s	9.6 $\mu$ s
625 kHz	6.4 $\mu$ s	4.8 $\mu$ s
1.25 MHz	3.2 $\mu$ s	2.4 $\mu$ s
2.5 MHz	1.6 $\mu$ s	1.2 $\mu$ s
5 MHz	800 ns	600 ns
10 MHz	400 ns	300 ns
20 MHz	200 ns	150 ns

**I/O mapping**

For details on the I/O mapping, refer to [Signal Mapping to I/O Pins \(PHS Bus System Hardware Reference !\[\]\(3dfb8d66e81160ad61421a3452093d1b\_img.jpg\)](#)).

**Parameters**

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

**channel** Specifies the logical channel number within the range 1 ... 5. For selecting all 5 channels, you can use the predefined symbol `DS3001_CH_ALL`.

**clock** Specifies the noise filter clock:

Symbol	Meaning
<code>DS3001_F1</code>	For 312.5 kHz
<code>DS3001_F2</code>	For 625 kHz
<code>DS3001_F3</code>	For 1.25 MHz
<code>DS3001_F4</code>	For 2.5 MHz
<code>DS3001_F5</code>	For 5 MHz
<code>DS3001_F6</code>	For 10 MHz
<code>DS3001_F7</code>	For 20 MHz

**Return value**

None

**Messages**

The following message is defined:

ID	Type	Message	Description
-50	Error	<code>ds3001_set_filter_clock(0x??)</code> : Board not initialized!	The DS3001 board has not been initialized by a preceding call to the function <code>ds3001_init</code> .

**Execution times**

For information, refer to [Function Execution Times](#) on page 29.

**Example**

This example shows how to set the filter clock of channel 1 to 625 kHz:

```
void main(void)
{
    init();
    ds3001_init(DS3001_1_BASE);
    ds3001_set_filter_clock(DS3001_1_BASE, 1, DS3001_F2);

    ...
}
```

Related topics

References

Base Address of the I/O Board.....	7
ds3001_init.....	9
Macros.....	7

ds3001\_set\_line\_type

Syntax

```
void ds3001_set_line_type(  
    phs_addr_t base,  
    int channel,  
    int type)
```

Include file

ds3001.h

Purpose

To select the transmission line type of the specified channel.

Note

The function `ds3001_init` must be called before this function can be used.

I/O mapping

For details on the I/O mapping, refer to [Signal Mapping to I/O Pins \(PHS Bus System Hardware Reference !\[\]\(b792654f2cef9719eabeb6c5be00811e\_img.jpg\)](#)).

Parameters

- base** Specifies the PHS-bus base address. Refer to [Base Address of the I/O Board](#) on page 7.
- channel** Specifies the logical channel number within the range 1 ... 5. For selecting all 5 channels, you can use the predefined symbol `DS3001_CH_ALL`.
- type** Specifies the type of transmission line:

Symbol	Meaning
DS3001_DIFF	For differential transmission line (RS422) This mode should be used in noisy environments or when long cables are used between the sensor and the DS3001. In addition to the improved noise rejection the differential line mode features a cable break detector which activates the I/O-Error line on the PHS-bus if both differential lines of a signal are carrying the same voltage level.

Symbol	Meaning
DS3001_SINGLE	For resetting single-ended transmission line (TTL) In this mode the inverting lines of an encoder input must be grounded. The cable break circuit is disabled in single-ended line mode. If an encoder input of the DS3001 is not used, the encoder signal type should be set to single-ended mode to avoid generation of I/O errors due to floating inputs.

**Return value** None

**Messages** The following message is defined:

ID	Type	Message	Description
-50	Error	ds3001_set_line_type(0x??): Board not initialized!	The DS3001 board has not been initialized by a preceding call to the function <code>ds3001_init</code> .

**Execution times** For information, refer to [Function Execution Times](#) on page 29.

**Example** This example shows how to initialize channel 2 to differential transmission line:

```
void main(void)
{
    init();
    ds3001_init(DS3001_1_BASE);
    ds3001_set_line_type(DS3001_1_BASE, 2, DS3001_DIFF);
    ...
}
```

**Related topics**

**References**

Base Address of the I/O Board.....	7
ds3001_init.....	9
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.

<b>Where to go from here</b>	<b>Information in this section</b>
	<a href="#">Information on the Test Environment.....</a> 29
	<a href="#">Measured Execution Times.....</a> 30

## 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:

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

**Related topics****References**[Measured Execution Times..... 30](#)

## Measured Execution Times

**Introduction**

Execution times are available for the following RTLib units:

- Initialization
- Setup functions
- Access functions
- Status functions

**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
ds3001_init	10.05 ms	10.04 ms

**Setup functions**

The following execution times have been measured for the setup functions of the incremental encoder interface:

Function	Mean Execution Time	
	DS1006 with 2.6 GHz	DS1006 with 3.0 GHz
ds3001_set_counter_reset_mode	1.41 $\mu$ s	1.39 $\mu$ s
ds3001_set_line_type	1.40 $\mu$ s	1.38 $\mu$ s
ds3001_set_filter_clock	1.40 $\mu$ s	1.39 $\mu$ s
ds3001_prepare_index_reset	0.03 $\mu$ s	0.02 $\mu$ s

**Access functions**

The following execution times have been measured for the register access functions of the incremental encoder interface:

Function	Mean Execution Time	
	DS1006 with 2.6 GHz	DS1006 with 3.0 GHz
ds3001_read_position	0.61 $\mu$ s	0.59 $\mu$ s
ds3001_read_delta_position	0.61 $\mu$ s	0.60 $\mu$ s
ds3001_read_line_count	0.60 $\mu$ s	0.59 $\mu$ s
ds3001_clear_counter	0.05 $\mu$ s	0.03 $\mu$ s

**Status functions**

The following execution times have been measured for the status functions of the incremental encoder interface:

Function	Mean Execution Time	
	DS1006 with 2.6 GHz	DS1006 with 3.0 GHz
ds3001_read_index	0.60 $\mu$ s	0.59 $\mu$ s
ds3001_clear_lerr	0.03 $\mu$ s	0.02 $\mu$ s
ds3001_read_lerr	0.60 $\mu$ s	0.59 $\mu$ s
ds3001_read_err	0.60 $\mu$ s	0.58 $\mu$ s

**Related topics****Basics**

[Information on the Test Environment.....](#) 29





**B**

base address 7

**C**

Common Program Data folder 6

**D**

Documents folder 6

DS3001

function execution times 29

DS3001\_ALWAYS 24

DS3001\_CH\_ALL 24

ds3001\_clear\_counter 12

ds3001\_clear\_lerr 13

DS3001\_DIFF 27

DS3001\_Fx 26

ds3001\_init 9

DS3001\_ONCE 24

ds3001\_prepare\_index\_reset 14

ds3001\_read\_delta\_position 16

ds3001\_read\_err 17

ds3001\_read\_index 18

ds3001\_read\_lerr 20

ds3001\_read\_line\_count 21

ds3001\_read\_position 22

ds3001\_set\_counter\_reset\_mode 23

ds3001\_set\_filter\_clock 25

ds3001\_set\_line\_type 27

DS3001\_SINGLE 27

DSxxx\_n\_BASE 7

**F**

function execution times

DS3001 29

**L**

Local Program Data folder 6

**N**

noise filter clock 25

