DS4001 Timing and Digital I/O Board

RTLib Reference

Release 2021-A – May 2021

dSPACE

How to Contact dSPACE

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

How to Contact dSPACE Support

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

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

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

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

Software Updates and Patches

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

Important Notice

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

Contents

About This Reference

Contents

This RTLib Reference (Real-Time Library) gives detailed descriptions of the C functions needed to program a DS4001 Timing and 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
⚠ DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
⚠ WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
⚠ CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a hazard that, if not avoided, could result in property damage.
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	Indicates tips that can make your work easier.
?	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
📖	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

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

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

Special folders

Some software products use the following special folders:

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

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

or

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

Documents folder A standard folder for user-specific documents.

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

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

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

Accessing dSPACE Help and PDF Files

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

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

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

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

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

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

Macros

Introduction

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

Base Address of the I/O Board

DSxxxx_n_BASE Macros

When using I/O board functions, you always need the board's base address as a parameter. This address can easily be obtained by using the `DSxxxx_n_BASE` macros, where `DSxxxx` is the board name (for example, DS2001) and `n` is an index which counts boards of the same type. The board with the lowest base address is given index 1. The other boards of the same type are given consecutive numbers in order of their base addresses.

The macros reference an internal data structure which holds the addresses of all I/O boards in the system. The initialization function of the processor board (named `init`) creates this data structure. Hence, when you change an I/O board base address, it is not necessary to recompile the code of your application. For more information on the processor board's initialization function, refer to ds1006_init (DS1006 RTLib Reference 🕮) or init (DS1007 RTLib Reference 🕮).

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

Example

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

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

Board Initialization

Introduction

Before you can use the DS4001 board, you have to perform the initialization process.

ds4001_init

Syntax

```
void ds4001_init(phs_addr_t base)
```

Include file

`ds4001.h`

Purpose

To initialize the DS4001 board to the default settings:

- All I/O line groups are set to direct input mode (non-strobed)
- The system timing controller (STC) is reset

Note

- The initialization function of the processor board must be called before the **`ds4001_init`** function.
- Th **`ds4001_init`** function must be called before any other DS4001 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	ds4001_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.
-172	Error	ds4001_init(0x??): Board not found!	No DS4001 board could be found at the specified PHS-bus address. Check if the `DSxxxx_n_BASE` macro corresponds to the I/O board used.
-53	Warning	ds4001_init(0x??): Jumper setting is not matching SW default initialization! STP register 0x???????? instead of 0x????????	The value of the STP register could not be verified successfully. May be the DS4001 jumper setting is not correct. Remove all jumpers.

Execution times

For information, refer to Function Execution Times on page 49.

Example

This example shows how to use this function:

```
void main(void)
{
   init();
   ds4001_init(DS4001_1_BASE);
   ...
}
```

The DS4001 board at address DS4001_1_BASE is initialized.

Related topics

References

Base Address of the I/O Board.. 7
ds4001_stc_init.. 28

Digital I/O Unit

Introduction

The DS4001 board provides 32 bidirectional TTL-compatible digital I/O lines. They are divided into 4 groups with 8 I/O lines each.

Where to go from here

Information in this section

Information in other sections

Specifying a Bitmask for the Digital I/O Unit... 12
A bitmask is used to configure digital I/O pins.

Digital I/O Unit (DS4001 Features 📖)
The DS4001 contains a digital I/O unit with 32 TTL-compatible digital I/O lines.

Specifying a Bitmask for the Digital I/O Unit

Introduction

A bitmask is used to configure digital I/O pins.

Bitmask

A 32-bit right-aligned bitmask is used to configure digital I/O pins. Each four-digit binary value can be represented by its hexadezimal value:

Binary Value	Hexadecimal Value
0000	0x0
0001	0x1
0010	0x2
0011	0x3
0100	0x4
0101	0x5
0110	0x6
0111	0x7
1000	0x8
1001	0x9
1010	0xA
1011	0xB
1100	0xC
1101	0xD
1110	0xE
1111	0xF

The `ds4001_pio_initialize` function needs a bitmask to configure the initial values of the output pins. A value of 1 in the bitmask sets the corresponding output pin to high level and a value of 0 clears it.

You can see the correlation of I/O pin and bitmask value in the example.

Example

If you want to set the output pins 0 … 3, 4, 5, 8, and 31 to high level, you have to specify the following bitmask:

I/O pin	*31*	*30*	*29*	*28*	*27*	*26*	*25*	*24*	*23*	*22*	*21*	*20*	*19*	*18*	*17*	*16*
Binary	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hexadecimal	0x8				0x0				0x0				0x0			

I/O pin	*15*	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Binary	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1
Hexadecimal	0x0				0x1				0x3				0xF			

You have to set the bitmask parameter in hexadecimal format as `0x8000013F`.

Related topics

References

ds4001_pio_initialize.. 15

ds4001_pio_init

Syntax

```
void ds4001_pio_init(
   phs_addr_t base,
   int dir,
   int mode)
```

Include file

`ds4001.h`

Purpose

To initialize the DS4001 digital I/O.

Description

This function initializes the I/O line groups as inputs or outputs. The input mode for each group can be specified as strobed or direct (non-strobed). The output pins are set to low level at initialization.

Note

- The function **`ds4001_init`** must be called before this function can be used.
- Either this function or the **`ds4001_pio_initialize`** function must be called before the digital I/O unit can be used for data input or output.

I/O mapping

For details on the I/O mapping, refer to Digital I/O Unit (DS4001 Features 📖).

Parameters

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

dir Specifies the digital I/O data direction. The predefined symbols can be combined by bitwise OR operators to select any combination of input and output groups:

Predefined Symbol	Meaning
`DS4001_IN_0`	For I/O line group 0 as input
`DS4001_IN_1`	For I/O line group 1 as input
`DS4001_IN_2`	For I/O line group 2 as input
`DS4001_IN_3`	For I/O line group 3 as input
`DS4001_OUT_0`	For I/O line group 0 as output
`DS4001_OUT_1`	For I/O line group 1 as output
`DS4001_OUT_2`	For I/O line group 2 as output
`DS4001_OUT_3`	For I/O line group 3 as output

The predefined symbols `DS4001_IN_0` ... `DS4001_IN_3` are zero and may be omitted.

mode Specifies the input mode for each I/O line group. The predefined symbols can be combined by bitwise OR operators to select any combination of strobed and direct (non-strobed) input mode.

Predefined Symbol	Meaning
`DS4001_NON_STRB_0`	For I/O line group 0 as direct input mode
`DS4001_NON_STRB_1`	For I/O line group 1 as direct input mode
`DS4001_NON_STRB_2`	For I/O line group 2 as direct input mode
`DS4001_NON_STRB_3`	For I/O line group 3 as direct input mode
`DS4001_STRB_0`	For I/O line group 0 as strobed input mode
`DS4001_STRB_1`	For I/O line group 1 as strobed input mode
`DS4001_STRB_2`	For I/O line group 2 as strobed input mode
`DS4001_STRB_3`	For I/O line group 3 as strobed input mode

The predefined symbols `DS4001_NON_STRB_0` ... `DS4001_NON_STRB_3` are zero and may be omitted.

Return value

None

Messages

The following messages are defined:

ID	Type	Message	Description
-50	Error	ds4001_pio_init(0x??): Board not initialized!	The DS4001 board has not been initialized by a preceding call to the function **`ds4001_init`**.
-54	Error	ds4001_pio_initialize(0x??): Specified value is not matching jumper setup!	The value of the STP register could not be verified successfully. May be the DS4001 jumper setting is not correct. Remove all jumpers.

Execution times

For information, refer to Function Execution Times on page 49.

Example

This example shows how to use this function:

```
void main(void)
{
   init();
   ds4001_init(DS4001_1_BASE);
   ds4001_pio_init( DS4001_1_BASE,
      DS4001_OUT_0 | DS4001_OUT_1 |
      DS4001_IN_2 | DS4001_IN_3,
      DS4001_NON_STRB_0 | DS4001_NON_STRB_1 |
      DS4001_STRB_2 | DS4001_STRB_3);
   ...
}
```

In the above example groups 0 and 1 are programmed for output. Groups 2 and 3 are initialized for strobed input.

Related topics

References

Base Address of the I/O Board.. 7
ds4001_init.. 9
ds4001_pio_initialize.. 15

ds4001_pio_initialize

Syntax

```
__INLINE void ds4001_pio_initialize (
   phs_addr_t base,
   int dir,
   int mode,
   UInt32 data);
```

Include file

`ds4001.h`

Purpose

To initialize the digital I/O line groups with initialization values.

Description

This function initializes the I/O line groups as inputs or outputs. The input mode for each group can be specified as strobed or direct (non-strobed). The specified initialization values are assigned to the output pins.

> **Note**
>
> - The function `ds4001_init` must be called before this function can be used.
> - Either this function or the `ds4001_pio_init` function must be called before the digital I/O unit can be used for data input or output.

I/O mapping

For details on the I/O mapping, refer to Digital I/O Unit (DS4001 Features 🕮).

Parameters

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

dir Specifies the digital I/O data direction. The predefined symbols can be combined by bitwise OR operators to select any combination of input and output groups:

Predefined Symbol	Meaning
`DS4001_IN_0`	For I/O line group 0 as input
`DS4001_IN_1`	For I/O line group 1 as input
`DS4001_IN_2`	For I/O line group 2 as input
`DS4001_IN_3`	For I/O line group 3 as input
`DS4001_OUT_0`	For I/O line group 0 as output
`DS4001_OUT_1`	For I/O line group 1 as output
`DS4001_OUT_2`	For I/O line group 2 as output
`DS4001_OUT_3`	For I/O line group 3 as output

The predefined symbols `DS4001_IN_0` … `DS4001_IN_3` are zero and may be omitted.

mode Specifies the input mode for each I/O line group. The predefined symbols can be combined by bitwise OR operators to select any combination of strobed and direct (non-strobed) input mode.

Predefined Symbol	Meaning
`DS4001_NON_STRB_0`	For I/O line group 0 as direct input mode
`DS4001_NON_STRB_1`	For I/O line group 1 as direct input mode
`DS4001_NON_STRB_2`	For I/O line group 2 as direct input mode
`DS4001_NON_STRB_3`	For I/O line group 3 as direct input mode

Predefined Symbol	Meaning
`DS4001_STRB_0`	For I/O line group 0 as strobed input mode
`DS4001_STRB_1`	For I/O line group 1 as strobed input mode
`DS4001_STRB_2`	For I/O line group 2 as strobed input mode
`DS4001_STRB_3`	For I/O line group 3 as strobed input mode

data Specifies the 32-bit right-aligned bitmask to set initial values of the output pins. The corresponding bit of the bitmask must be 1 for setting the pin to high level. For information on specifying a bitmask, refer to Specifying a Bitmask for the Digital I/O Unit on page 12.

Return value

None

Messages

The following messages are defined:

ID	Type	Message	Description
-50	Error	ds4001_pio_initialize(0x??): Board not initialized!	The DS4001 board has not been initialized by a preceding call to the `ds4001_init` function.
-54	Error	ds4001_pio_initialize(0x??): Specified value is not matching jumper setup!	The value of the STP register could not be verified successfully. May be the DS4001 jumper setting is not correct. Remove all jumpers.

Execution times

For information, refer to Function Execution Times on page 49.

Related topics

References

Base Address of the I/O Board	7
ds4001_bit_in	19
ds4001_bit_out	20
ds4001_init	9
ds4001_out32	23
ds4001_pio_init	13

ds4001_read_status

Syntax

```
long ds4001_read_status(phs_addr_t base)
```

Include file

`ds4001.h`

Purpose

To read latched and none latched I/O error flag of DS4001 board.

Description

This function returns the state of the latched and none latched I/O error flag of the DS4001 board. You can use the definition `DS4001_ERR` to mask out the none latched I/O error flag and `DS4001_LERR` to mask out the latched I/O error flag.

For further information, refer to I/O Error Requesting (DS4001 Features 📖).

> **Note**
>
> - The latched error flag LERR is cleared by any access of this function. Thus use a copy of the function return value to test multiple flags at the same time.
> - The `ds4001_init` function must be called before this function can be used.

Parameters

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

Return value

This function returns the state of the I/O error flags.

Execution times

For information, refer to Function Execution Times on page 49.

Example

This example shows how to use this function:

```
void sub_fct(void)
{
   long flags;
   flags = ds4001_read_status(DS4001_1_BASE);
   if(flags & DS4001_ERR)
      error_handler1();
   else if(flags & DS4001_LERR)
      error_handler2();
}
```

The I/O error flags of the DS4001 board are read. If the ERR flag is set, the user function `error_handler1` is called.

If the LERR flag is set the user function `error_handler2` is called.

Related topics

References

ds4001_bit_in

Syntax

```
UInt32 ds4001_bit_in(
   phs_addr_t base,
   long mask)
```

Include file

`ds4001.h`

Purpose

To get bitwise digital input.

Description

The I/O port is read and the bits specified by the `mask` parameter are returned by the function return parameter. The relevant I/O line groups must be programmed for input.

> **Note**
>
> The function **`ds4001_init`** must be called before this function can be used.

I/O mapping

For details on the I/O mapping, refer to Digital I/O Unit (DS4001 Features 🕮).

Parameters

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

mask Specifies the digital I/O bits to be read by using a 32-bit right-aligned bitmask. The corresponding bit of the bitmask must be 1. For information on specifying a bitmask, refer to Specifying a Bitmask for the Digital I/O Unit on page 12.

Return value

The function returns the state of the specified input bit(s).

Execution times

For information, refer to Function Execution Times on page 49.

Example

This example shows how to use this function:

```
void main(void)
{
   init();
   ds4001_init(DS4001_1_BASE);
   ds4001_pio_init( DS4001_1_BASE,
      DS4001_IN_0 | DS4001_IN_1 |
      DS4001_OUT_2 | DS4001_OUT_3,
      DS4001_NON_STRB_0 | DS4001_NON_STRB_1 |
      DS4001_NON_STRB_2 | DS4001_NON_STRB_3);
   ...
}
```

```
void sub_fct(void)
{
   UInt32 port;
   long mask = 0x000000FF
   ...
   port = ds4001_bit_in(DS4001_1_BASE, mask);
   ...
}
```

Group 0 (pins 0 … 7) of the DS4001 I/O port is read.

Related topics

References

- Base Address of the I/O Board 7
- ds4001_init 9
- ds4001_pio_init 13

ds4001_bit_out

Syntax

```
void ds4001_bit_out(
   phs_addr_t base,
   long mask,
   UInt32 data)
```

Include file

`ds4001.h`

Purpose

To write bitwise digital output.

Description

The bits of data specified by the `mask` parameter are written to the I/O port. All other bits remain unchanged.

The relevant I/O line groups must be programmed for output and direct (non-strobed) input mode. If the strobed mode is used, the function will not operate properly, because non-destructive output of individual bits requires readout of the data register, which is not possible in strobed mode.

> **Note**
>
> The function `ds4001_init` must be called before this function can be used.

I/O mapping

For details on the I/O mapping, refer to Digital I/O Unit (DS4001 Features 🕮).

Parameters

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

mask Specifies the digital I/O bits to be written by using a 32-bit right-aligned bitmask. The corresponding bit of the bitmask must be 1. For information on specifying a bitmask, refer to Specifying a Bitmask for the Digital I/O Unit on page 12.

data Specifies the output data value by using a 32-bit right-aligned bitmask. A value of 1 in the bitmask sets the corresponding output pin to high level and a value of 0 clears it.

Return value

None

Execution times

For information, refer to Function Execution Times on page 49.

Example

This example shows how to use this function:

```
void main(void)
{
   init();
   ds4001_init(DS4001_1_BASE);
   ds4001_pio_init( DS4001_1_BASE,
      DS4001_OUT_0 | DS4001_OUT_1 |
      DS4001_IN_2 | DS4001_IN_3,
      DS4001_NON_STRB_0 | DS4001_NON_STRB_1 |
      DS4001_STRB_2 | DS4001_STRB_3);
   ...
}
```

```
void sub_fct(void)
{
   ...
   ds4001_bit_out(DS4001_1_BASE, 0x000000FF, 0xF1);
   ...
}
```

Output lines 1 … 3 of group 0 are cleared. Output lines 0, 4 … 7 and the output lines of group 1 remain unchanged.

Related topics

References

Base Address of the I/O Board 7
ds4001_init 9
ds4001_pio_init 13

ds4001_in32

Syntax

```
UInt32 ds4001_in32(phs_addr_t base)
```

Include file

`ds4001.h`

Purpose

To read the I/O port (32-bit digital input).

Description

If the I/O port is programmed for direct (non-strobed) input mode as performed by the `ds4001_pio_init` function, the 32-bit data word returned by `ds4001_in32` reflects the current state of the input lines.

If the I/O port is programmed for strobed mode, the 32-bit data word contains the state of the input lines after the last strobe signal.

The I/O lines specified as outputs are represented by their actual output data value.

Note

- The `ds4001_init` function must be called before this function can be used.
- All relevant I/O line groups must be programmed for input.

Parameters

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

Return value

This function returns 32-bit input data.

Execution times

For information, refer to Function Execution Times on page 49.

Example

This example shows how to use this function:

```
void sub_fct(void)
{
   UInt32 port;
   ...
   port = ds4001_in32(DS4001_1_BASE);
   ...
}
```

The 32-bit I/O port of the DS4001 board is read.

Related topics

References

Base Address of the I/O Board	7
ds4001_init	9
ds4001_pio_init	13

ds4001_out32

Syntax

```
void ds4001_out32(
   phs_addr_t base,
   UInt32 data)
```

Include file

`ds4001.h`

Purpose

To write data to the I/O port (32-bit digital output).

Description

The value data is written to the I/O port.

> **Note**
>
> - The `ds4001_init` function must be called before this function can be used.
> - All relevant I/O line groups must be programmed for output.

Parameters

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

data Specifies the output values for the output lines by using a 32-bit right-aligned bitmask. A value of 1 in the bitmask sets the corresponding bit to high level and a 0 clears the bit. For information on specifying a bitmask, refer to Specifying a Bitmask for the Digital I/O Unit on page 12.

Return value

None

Execution times

For information, refer to Function Execution Times on page 49.

Example

This example shows how to use this function:

```
void main(void)
{
   init();
   ds4001_init(DS4001_1_BASE);
   ds4001_pio_init( DS4001_1_BASE,
      DS4001_OUT_0 | DS4001_OUT_1 |
      DS4001_OUT_2 | DS4001_OUT_3,
      DS4001_NON_STRB_0 | DS4001_NON_STRB_1 |
      DS4001_NON_STRB_2 | DS4001_NON_STRB_3);
   ...
}
void sub_fct(void)
{
   Int32 data = 0x80000001;
   ...
   ds4001_out32(DS4001_1_BASE, data);
   ...
}
```

Pins 0 and 31 of the DS4001 I/O port are set to high level.

Related topics

References

Base Address of the I/O Board........ 7
ds4001_init........ 9
ds4001_pio_init........ 13

Timing I/O Unit

Introduction

The timing I/O unit of the DS4001 provides standard timer functions, signal generation, and signal measurement.

Where to go from here

Information in this section

Timer Functions

Introduction

The DS4001 provides 5 separately programmable 16-bit timers with one timing I/O channel (TTL) each.

Where to go from here

Information in this section

ds4001_stc_init

Syntax

```
void ds4001_stc_init(phs_addr_t base)
```

Include file

`ds4001.h`

Purpose

To initialize the system timing controller (STC).

Description

This function performs a reset and basic initialization of the system timing controller (also called timer access controller) contained on the DS4001.

Note

- The `ds4001_init` function must be called before this function can be used.
- This function must be called only once before the individual channels can be initialized for signal generation (PWM, D2F) or signal measurement (F2D).

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 message is defined:

ID	Type	Message	Description
-50	Error	ds4001_stc_init(): board at offset 0x?? not initialized!	The DS4001 board has not been initialized by a preceding call to the `ds4001_init` function.

Execution times

For information, refer to Function Execution Times on page 49.

Example

This example shows how to use this function:

```
void main(void)
{
   init();
   ds4001_init();
   ds4001_stc_init(DS4001_1_BASE);
   ...
}
```

The system timing controller (STC) contained on the DS4001 is initialized.

Related topics

References

Base Address of the I/O Board 7
ds4001_d2f_init 40
ds4001_f2d_init 44
ds4001_init 9
ds4001_pwm_init 33

ds4001_set_int_input

Syntax

```
void ds4001_set_int_input(
   phs_addr_t base,
   int channel)
```

Include file

`ds4001.h`

Purpose

To initialize the timer output as external interrupt input.

Description

This function initializes the respective timer output TMROUT1 … TMROUT5 as external interrupt input.

For further information, refer to Interrupts Provided by the DS4001 (DS4001 Features 🕮).

> **Note**
>
> The function `ds4001_init` must be called before this function can be used.

I/O mapping

For details on the I/O mapping, refer to Timer Interrupts (DS4001 Features 🕮).

Parameters

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

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

Return value

None

Messages

The following message is defined:

ID	Type	Message	Description
-50	Error	ds4001_set_int_input(0x??): Board not initialized!	The DS4001 board has not been initialized by a preceding call to the `ds4001_init` function.

Execution times

For information, refer to Function Execution Times on page 49.

Example

This example shows how to use this function:

```
void main(void)
{
   ...
   init();
   ds4001_init(DS4001_1_BASE);
   ds4001_set_int_input(DS4001_1_BASE, 1);
   ...
}
```

The DS4001 timer output TMROUT1 is initialized as external interrupt input.

Related topics

References

Base Address of the I/O Board..7
ds4001_init...9
Interrupts Provided by the DS4001 (DS4001 Features 🕮)

ds4001_timer_stop

Syntax

```
void  ds4001_timer_stop(
   phs_addr_t base,
   int channel)
```

Include file

`ds4001.h`

Purpose

To stop the DS4001 timer.

Description

The specified timer is stopped and the output is set to inactive and low level.

> **Note**
>
> The specified timer must be initialized by the `ds4001_pwm_init`, `ds4001_d2f_init` or `ds4001_f2d_init` function before `ds4001_timer_stop` can be used.

I/O mapping

For details on the I/O mapping, refer to PWM Signal Generation (DS4001 Features 🕮).

Parameters

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

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

Return value

None

Execution times

For information, refer to Function Execution Times on page 49.

Example

This example shows how to use this function:

```
...
duty1 = 0.0;
ds4001_stc_init(DS4001_1_BASE);
ds4001_pwm_init(DS4001_1_BASE, 1, DS4001_F2, 0.001, duty1);
...
while (...)
{
   ds4001_pwm(DS4001_1_BASE, 1, duty1);
   if (duty1 < 1.0)
      duty1 += 0.001;
   else
      ds4001_timer_stop(DS4001_1_BASE, 1);
}
...
```

Channel 1 is initialized for PWM signal generation. The duty cycle of the PWM signal on channel 1 is increased from 0 to 1. After the duty cycle has reached 1.0 the timer of channel 1 is stopped.

Related topics

References

Base Address of the I/O Board 7
ds4001_d2f_init 40
ds4001_f2d_init 44
ds4001_init 9
ds4001_pwm_init 33

PWM Signal Generation

Introduction

The timing I/O unit of the DS4001 provides outputs for 1-phase PWM signal generation on up to 4 channels.

Where to go from here

Information in this section

Information in other sections

Conflicting I/O Features (DS4001 Features 📖)
Shows the I/O features of the DS4001 which conflict with other I/O features.

ds4001_pwm_init

Syntax

```
int ds4001_pwm_init(
   phs_addr_t base,
   int channel,
   int clk_source,
   dsfloat period,
   dsfloat duty)
```

Include file

`ds4001.h`

Purpose

To initialize and start the PWM signal generation.

Description

The specified channel starts the PWM signal generation with the initial values for the `period` and `duty` parameters. The PWM signal can be measured at the respective timer output (TMROUT1 … TMROUT4).

> **Note**
>
> Channel 5 is used to generate the PWM period common for all PWM channels. The output of channel 5 (TMROUT5) must be connected to the gate inputs (TMRGS1 … TMRGS4) of the timers to be used for PWM generation. Thus only timers 1 … 4 are available for PWM generation.

If multiple channels are initialized for PWM generation, the parameters `clk_source` and `period` must be initialized to uniform values for all PWM channels.

> **Note**
>
> - The `ds4001_init` and `ds4001_stc_init` functions must be called before this function can be used.
> - If the PWM generation shall be re-initialized, for example to change the PWM period, the STC must be reset by using the `ds4001_stc_init` function before `ds4001_pwm_init` is called.

I/O mapping

For details on the I/O mapping, refer to PWM Signal Generation (DS4001 Features 🕮).

Parameters

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

channel Specifies the logical channel number within the range 1 … 4.

clk_source Specifies the required prescaler output F1 … F5 as the timer clock source. Depending on the selected clock source the described PWM frequency ranges can be achieved:

Predefined Symbol	Frequency Range
`DS4001_F1`	76.3 Hz … 833.33 kHz
`DS4001_F2`	7.63 Hz … 83.33 kHz
`DS4001_F3`	763 mHz … 8.33 kHz
`DS4001_F4`	76.3 mHz … 833.33 Hz
`DS4001_F5`	7.63 mHz … 83.33 Hz

To optimize the resolution of the generated PWM signal, you should always choose the frequency range with the lowest possible range number.

period Specifies the PWM period in seconds.

duty Specifies the initial duty cycle within the range 0.0 … 1.0.

Return value

The following value is returned:

Value	Meaning
0	No error occurred during initialization

This return value is only kept for compatibility purposes. In case of an error this function will perform an exit.

Messages

The following messages are defined:

ID	Type	Message	Description
-50	Error	ds4001_pwm_init(0x??): Board not initialized!	The DS4001 board has not been initialized by a preceding call to the `ds4001_init` function.
-173	Error	ds4001_pwm_init(0x??): Memory allocation error!	The allocation of some dynamic memory for internal data storage has failed.

Execution times

For information, refer to Function Execution Times on page 49.

Example

For an example, refer to `ds4001_pwm` on page 38.

Related topics

References

Base Address of the I/O Board.......7
ds4001_init.......9
ds4001_pwm.......38
ds4001_stc_init.......28

ds4001_pwm_init_adv

Syntax

```
int ds4001_pwm_init_adv(
   phs_addr_t base,
   int channel,
   int clk_source,
   dsfloat period,
   dsfloat duty,
   int inverse)
```

Include file

`ds4001.h`

Purpose

To initialize the advanced PWM signal generation.

Description

The specified channel starts the PWM signal generation with the initial values for the `period` and `duty` parameters. The `inverse` parameter enables or disables an inverse PWM signal generation. The PWM signal can be measured at the respective timer output (TMROUT1 … TMROUT4).

> **Note**
>
> Channel 5 is used to generate the PWM period common for all PWM channels. The output of channel 5 (TMROUT5) must be connected to the gate inputs (TMRGS1 … TMRGS4) of the timers to be used for PWM generation. Thus only timers 1 … 4 are available for PWM generation.

If multiple channels are initialized for PWM generation, the parameters `clk_source` and `period` must be initialized to uniform values for all PWM channels.

> **Note**
>
> - The functions `ds4001_init` and `ds4001_stc_init` must be called before this function can be used.
> - If the PWM generation shall be re-initialized, for example to change the PWM period, the STC must be reset by using the function `ds4001_stc_init` before `ds4001_pwm_init_adv` is called.

I/O mapping

For details on the I/O mapping, refer to PWM Signal Generation (DS4001 Features 🕮).

Parameters

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

channel Specifies the logical channel number within the range 1 … 4.

clk_source Specifies the required prescaler output F1 … F5 as the timer clock source. Depending on the selected clock source the described PWM frequency ranges can be achieved:

Predefined Symbol	Frequency Range
`DS4001_F1`	76.3 Hz … 833.33 kHz
`DS4001_F2`	7.63 Hz … 83.33 kHz
`DS4001_F3`	763 mHz … 8.33 kHz
`DS4001_F4`	76.3 mHz … 833.33 Hz
`DS4001_F5`	7.63 mHz … 83.33 Hz

To optimize the resolution of the generated PWM signal, you should always choose the frequency range with the lowest possible range number.

period Specifies the PWM period in seconds.

duty Specifies the initial duty cycle within the range 0.0 … 1.0.

inverse Specifies the switch for inverting PWM signal output.

Value	Meaning
0	Normal PWM signal
1	Inverted PWM signal

Return value

The following value is returned:

Value	Meaning
0	No error occurred during initialization

This return value is only kept for compatibility purposes. In case of an error this function will perform an exit.

Messages

The following messages are defined:

ID	Type	Message	Description
-50	Error	ds4001_pwm_init_adv(0x??): Board not initialized!	The DS4001 board has not been initialized by a preceding call to the `ds4001_init` function.
-173	Error	ds4001_pwm_init_adv(0x??): Memory allocation error!	The allocation of some dynamic memory for internal data storage has failed.

Execution times

For information, refer to Function Execution Times on page 49.

Example

For an example, refer to ds4001_pwm on page 38.

Related topics

References

ds4001_pwm

Syntax

```
void ds4001_pwm(
   phs_addr_t base,
   int channel,
   dsfloat duty)
```

Include file

`ds4001.h`

Purpose

To set the duty cycle of the generated PWM signal.

Description

The duty cycle of the generated PWM signal is changed to the new value specified by the parameter `duty`. The new value becomes effective with the rising edge of the next PWM period.

> **Note**
>
> The `ds4001_init`, `ds4001_stc_init`, `ds4001_pwm_init` or `ds4001_pwm_init_adv` function must be called before this function can be used.

I/O mapping

For details on the I/O mapping, refer to PWM Signal Generation (DS4001 Features 🕮).

Parameters

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

channel Specifies the logical channel number within the range 1 … 4.

duty Specifies the duty cycle within the range 0.0 … 1.0.

Return value

None

Execution times

For information, refer to Function Execution Times on page 49.

Example

This example shows how to use this function:

```
...
duty1 = 0.0;
duty2 = 0.2;
ds4001_stc_init(DS4001_1_BASE);
ds4001_pwm_init(DS4001_1_BASE, 1, DS4001_F2, 0.001, duty1);
ds4001_pwm_init(DS4001_1_BASE, 2, DS4001_F2, 0.001, duty2);
...
while (...)
{
   ds4001_pwm(DS4001_1_BASE, 1, duty1);
   if (duty1 < 1.0)
      duty1 += 0.001;
   else
      duty1 = 0.0;
}
...
```

Channels 1 and 2 are initialized for PWM generation. The duty cycle of the PWM signal on channel 1 is periodically increased from 0 to 1 while the duty cycle of the second PWM signal on channel 2 remains unchanged.

Related topics

References

Base Address of the I/O Board 7
ds4001_init 9
ds4001_pwm_init 33
ds4001_pwm_init_adv 35
ds4001_stc_init 28

Square-Wave Signal Generation (D2F)

Introduction

The timing I/O unit of the DS4001 provides outputs for square-wave signal generation on up to 5 channels.

Where to go from here

Information in this section

Information in other sections

Conflicting I/O Features (DS4001 Features 🕮)
Shows the I/O features of the DS4001 which conflict with other I/O features.

ds4001_d2f_init

Syntax

```
int ds4001_d2f_init(
   phs_addr_t base,
   int channel,
   int clk_source,
   dsfloat period)
```

Include file

`ds4001.h`

Purpose

To initialize the period of the generated square-wave signal.

Description

The specified channel starts the square-wave signal generation with the initial value for the `period` parameter. The signal can be measured at the timer output pin (TMROUT1 … TMROUT5) of the specified channel.

Note

The functions `ds4001_init` and `ds4001_stc_init` must be called before this function can be used.

I/O mapping

For details on the I/O mapping, refer to Square-Wave Signal Generation (D2F) (DS4001 Features 🕮).

Parameters

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

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

clk_source Specifies the required prescaler output F1 … F5 as the timer clock source. Depending on the selected clock source the frequency ranges described below can be achieved:

Predefined Symbol	Frequency Range
`DS4001_F1`	38.15 Hz … 1.25 MHz
`DS4001_F2`	3.815 Hz … 125 kHz
`DS4001_F3`	381.5 mHz … 12.5 kHz
`DS4001_F4`	38.5 mHz … 1.25 kHz
`DS4001_F5`	3.815 mHz … 125 Hz

To optimize the resolution of the generated square-wave signal, you should always choose the frequency range with the lowest possible range number.

period Specifies the initial signal period in seconds. The square-wave signal generation will only start if this value is unequal 0, otherwise the output is set to inactive and low level.

Return value

The following value is returned:

Value	Meaning
0	No error occurred during initialization

This return value is only kept for compatibility purposes. In case of an error this function will perform an exit.

Messages

The following messages are defined:

ID	Type	Message	Description
-50	Error	ds4001_d2f_init(0x??): Board not initialized!	The DS4001 board has not been initialized by a preceding call to the `ds4001_init` function.

ID	Type	Message	Description
-173	Error	ds4001_d2f_init(0x??): Memory allocation error!	The allocation of some dynamic memory for internal data storage has failed.

Execution times

For information, refer to Function Execution Times on page 49.

Example

For an example, refer to `ds4001_d2f` on page 42.

Related topics

References

ds4001_d2f

Syntax

```
void ds4001_d2f(
   phs_addr_t base,
   int channel,
   dsfloat period)
```

Include file

`ds4001.h`

Purpose

To change the frequency of the generated square-wave signal.

Description

The frequency of the generated square-wave signal is changed to the new value specified by the parameter `period`.

Note

The `ds4001_init`, `ds4001_stc_init` and `ds4001_d2f_init` functions must be called before this function can be used.

I/O mapping

For details on the I/O mapping, refer to Square-Wave Signal Generation (D2F) (DS4001 Features 📖).

Parameters

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

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

period Specifies the required signal period in seconds. The square-wave signal generation will be disabled if this value is set to 0. In this case the output is set to inactive and low level. A value unequal 0 will restart the square-wave signal generation.

Return value

None

Execution times

For information, refer to Function Execution Times on page 49.

Example

This example shows how to use this function:

```
...
prd = 0.1;
ds4001_stc_init(DS4001_1_BASE);
ds4001_d2f_init(DS4001_1_BASE, 1, DS4001_F2, prd);
ds4001_d2f_init(DS4001_1_BASE, 2, DS4001_F1, 1.0e-6);
...
while (...)
{
   ds4001_d2f(DS4001_1_BASE, 1, prd);
   if (prd > 10.0e-6)
      prd -= 1.0e-6;
   else
      prd = 0.1;
}
```

The frequency of the square-wave signal generated on channel 1 is periodically increased from 10 Hz to 100 kHz while channel 2 generates a square-wave signal with a constant frequency of 1 MHz.

Related topics

References

Base Address of the I/O Board	7
ds4001_d2f_init	40
ds4001_init	9
ds4001_stc_init	28

Square-Wave Signal Measurement (F2D)

Introduction

The timing I/O unit of the DS4001 provides inputs to measure the frequency of a square-wave signal on up to 5 channels.

Where to go from here

Information in this section

Information in other sections

Conflicting I/O Features (DS4001 Features 📖)
Shows the I/O features of the DS4001 which conflict with other I/O features.

ds4001_f2d_init

Syntax

```
int ds4001_f2d_init(
   phs_addr_t base,
   int channel,
   int clk_source)
```

Include file

`ds4001.h`

Purpose

To initialize the specified channel for frequency measurement.

Description

The specified channel is initialized for square-wave signal measurement. The input signal must be connected to the appropriate timer gate input (TMRGS1 … TMRGS5).

Note

The `ds4001_init` and `ds4001_stc_init` functions must be called before this function can be used.

I/O mapping

For details on the I/O mapping, refer to Square-Wave Signal Measurement (F2D) (DS4001 Features 🕮).

Parameters

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

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

clk_source Specifies the required prescaler output F1…F5 as the timer clock source. Depending on the selected clock source the frequency ranges described below can be measured:

Predefined Symbol	Frequency Range
`DS4001_F1`	76.3 Hz … 1.67 MHz
`DS4001_F2`	7.63 Hz … 167 kHz
`DS4001_F3`	763 mHz … 16.7 kHz
`DS4001_F4`	76.3 mHz … 1.67 kHz
`DS4001_F5`	7.63 mHz … 167 Hz

To optimize the resolution of the frequency measurement, you should always choose the frequency range with the lowest possible range number.

Return value

The following value is returned:

Value	Meaning
0	No error occurred during initialization

This return value is only kept for compatibility purposes. In case of an error this function will perform an exit.

Messages

The following messages are defined:

ID	Type	Message	Description
-50	Error	ds4001_f2d_init(0x??): Board at offset not initialized!	The DS4001 board has not been initialized by a preceding call to the ds4001_init on page 9 function.
-173	Error	ds4001_f2d_init(0x??): Memory allocation error!	The allocation of some dynamic memory for internal data storage has failed.

Execution times

For information, refer to Function Execution Times on page 49.

Example

For an example, refer to `ds4001_d2f` on page 42.

Related topics

References

Base Address of the I/O Board ... 7
ds4001_f2d ... 46
ds4001_init ... 9
ds4001_stc_init ... 28

ds4001_f2d

Syntax

```
dsfloat ds4001_f2d(
   phs_addr_t base,
   int channel)
```

Include file

`ds4001.h`

Purpose

To measure the current signal period.

Description

The current signal period of the input signal at the timer gate input of the specified channel is measured and returned by the function return parameter.

> **Note**
>
> The `ds4001_init`, `ds4001_stc_init` and `ds4001_f2d_init` functions must be called before this function can be used.

I/O mapping

For details on the I/O mapping, refer to Square-Wave Signal Measurement (F2D) (DS4001 Features 🕮).

Parameters

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

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

Return value

Returns the signal period in seconds.

Execution times

For information, refer to Function Execution Times on page 49.

Example

This example shows how to use this function:

```
...
ds4001_stc_init(DS4001_1_BASE);
ds4001_f2d_init(DS4001_1_BASE, 1, DS4001_F2);
...
while (...)
{
   prd = ds4001_f2d(DS4001_1_BASE, 1);
}
...
```

The signal period of the square-wave signal connected to the gate input of timer1 is periodically measured.

Related topics

References

Base Address of the I/O Board.. 7
ds4001_f2d_init.. 44
ds4001_init.. 9
ds4001_stc_init.. 28

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

Information in this section

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

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

The properties of the processor boards used are:

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

Measured Execution Times

Introduction

The following tables provide the mean execution times of the board's RTLib 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
`ds4001_init`	56.80 µs	68.13 µs

Digital I/O unit

The following execution times have been measured for the digital I/O unit.

Function	Mean Execution Time	
	DS1006 with 2.6 GHz	DS1006 with 3.0 GHz
`ds4001_pio_init`	2.36 µs	2.50 µs
`ds4001_pio_initialize`	1.84 µs	1.81 µs
`ds4001_in32`	0.61 µs	0.59 µs
`ds4001_out32`	0.04 µs	0.029 µs
`ds4001_bit_in`	0.62 µs	0.58 µs
`ds4001_bit_out`	0.62 µs	0.60 µs
`ds4001_read_status`	0.74 µs	0.60 µs

Timing I/O unit

The following execution times have been measured for the timing I/O unit.

Function	Mean Execution Time	
	DS1006 with 2.6 GHz	DS1006 with 3.0 GHz
Standard timer functions		
`ds4001_stc_init`	14.13 µs	15.04 µs
`ds4001_set_int_input`	7.41 µs	8.59 µs
`ds4001_timer_stop`	10.90 µs	10.71 µs
PWM signal generation		
`ds4001_pwm_init`	42.06 µs	41.94 µs
`ds4001_pwm_init_adv`	39.74 µs	42.08 µs
`ds4001_pwm`	6.31 µs	6.29 µs
Square-wave signal generation		
`ds4001_d2f_init`	15.49 µs	16.16 µs
`ds4001_d2f`	14.32 µs	15.32 µs
Square-wave signal measurement		
`ds4001_f2d_init`	15.89 µs	15.63 µs
`ds4001_f2d`	5.93 µs	7.09 µs

Related topics

References

Information on the Test Environment.. 49