

DS5101 Digital Waveform Output 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.

© 2001 - 2021 by:  
dSPACE GmbH  
Rathenaustraße 26  
33102 Paderborn  
Germany

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

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

# Contents

About This Reference	5
Macros	7
Base Address of the I/O Board.....	7
Board Initialization	9
ds5101_init.....	9
ds5101_set_clock.....	10
ds5101_set_pullup.....	11
Timing I/O Unit	13
1-Phase PWM Signal Generation.....	14
ds5101_pwm_load.....	14
ds5101_pwm_parm.....	15
ds5101_pwm_update.....	16
AP5101_PWM_MASK.....	17
3-Phase PWM Signal Generation.....	19
ds5101_pwm3_intshift_load.....	19
ds5101_pwm3_intshift_parm.....	21
ds5101_pwm3_intshift_update.....	22
AP5101_PWM3_MASK.....	23
3-Phase PWM Signal Generation with Inverted and Non-Inverted Outputs.....	25
ds5101_pwm6_intshift_load.....	25
ds5101_pwm6_intshift_parm.....	27
ds5101_pwm6_intshift_update.....	28
AP5101_PWM6_MASK.....	30
Incremental Encoder Simulation.....	31
ds5101_inc_load.....	31
ds5101_inc_parm.....	32
ds5101_inc_update.....	33
AP5101_INC_MASK.....	34
Monoflop Signal Generation.....	36
ds5101_mono_load.....	36

ds5101_mono_parm.....	37
ds5101_mono_start.....	38
ds5101_mono_update.....	39
ds5101_delayed_mono_load.....	40
ds5101_delayed_mono_parm.....	41
ds5101_delayed_mono_update.....	43
AP5101_MONO_MASK.....	44
Arbitrary Signal Generation.....	45
Basics on Arbitrary Signal Generation.....	45
ds5101_clr_flags.....	46
ds5101_flag.....	47
ds5101_set_flags.....	48
ds5101_update_flags.....	49
ds5101_start.....	50
ds5101_stop.....	51
ds5101_state.....	52
ds5101_trigger.....	53
DS5101_FLAG_MASK.....	54
DS5101_TRG_MASK.....	55
<b>Interrupts</b>	<b>57</b>
ds5101_int_ack.....	57
ds5101_int_clear.....	58
ds5101_int_enable.....	59
ds5101_int_read.....	60
<b>Function Execution Times</b>	<b>63</b>
Information on the Test Environment.....	63
Measured Execution Times.....	64
<b>Index</b>	<b>67</b>

# About This Reference

## Contents





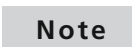



This RTLib Reference (Real-Time Library) gives detailed descriptions of the C functions needed to program a DS5101 Digital Waveform Output Board. The C functions can be used to program RTI-specific Simulink S-functions, or to implement your real-time models manually using C programs.

## Demo examples

There are examples for some features included in this documentation. You will find the relevant files after the installation of your dSPACE software in `<RCP_HIL_InstallationPath>\Demos\Ds100x\IOBoards\Ds5101`. Use ControlDesk to load and start the real-time application on your processor board.

## 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



# Board Initialization

**Introduction** Before you can use the DS5101, you have to perform the initialization process.

Where to go from here	Information in this section
	<a href="#">ds5101_init..... 9</a> To initialize the DS5101.
	<a href="#">ds5101_set_clock..... 10</a> To select the clock to be used.
	<a href="#">ds5101_set_pullup..... 11</a> To select the mode of the pull-up resistors.

## ds5101\_init

Syntax	<code>void ds5101_init (phs_addr_t base)</code>
Include file	<code>ds5101.h</code>
Purpose	To initialize the DS5101.
Description	This function performs a basic initialization of the DS5101. All channels are put into reset state. The on-board clock is selected and pull-up resistors are connected to V <sub>CC</sub> .

**Note**

- This function must be called before any other DS5101 function can be executed.
- The initialization function of the processor board must be called before the DS5101's initialization function.

**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	ds5101_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.
-186	Error	ds5101_init(0x??): Board not found!	No DS5101 could be found at the specified PHS-bus board address. Check if the DSxxxx_n_BASE macro corresponds to the I/O board used.
-818	Error	ds5101_init(0x??): Board is not responding! FPGA boot failed!	The DS5101 is not responding after a FPGA reset.

**Related topics****References**

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

## ds5101\_set\_clock

**Syntax**

```
void ds5101_set_clock (
    phs_addr_t base,
    int clock)
```

**Include file**                      ds5101.h

<

## ds5101\_set\_pullup


<b>Syntax</b>	<pre>void ds5101_set_pullup (     phs_addr_t base,     int pullup)</pre>
<b>Include file</b>	ds5101.h

---

**Purpose** To select the mode of the pull-up resistors.

---

**Description** The mode of the pull-up resistors is selected. The pull-up resistors could be connected to ground or to  $V_{CC}$ . After `ds5101_init` the pull-up resistors are connected to  $V_{CC}$ . This setting is lost, when you turn off and on the board.

For boards as of revision DS5101-04, it is possible to set the pull-up behavior channel-wise using pull-up jumpers on the board. In that case, the hardware setting has precedence and overrides any setting done with the `ds5101_set_pullup` function. For more information on setting the pull-up jumpers, refer to [Board Overview \(as of Revision DS5101-04\)](#) (PHS Bus System Hardware Reference ).

---

**Parameters**

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

**pullup** Specifies the mode of pull-up resistors. The following symbols are predefined:

Symbol	Meaning
DS5101_GND	For connecting resistors to ground
DS5101_VCC	For connecting resistors to $V_{CC}$

---

**Return value** None

---

**Messages** The following message is defined:

ID	Type	Message	Description
-50	Error	ds5101_set_pullup(0x??): Board not initialized!	The DS5101 has not been initialized by a preceding call to the <code>ds5101_init</code> function.

---

**Execution times** For information, refer to [Function Execution Times](#) on page 63.

---

**Related topics**

References

<a href="#">Base Address of the I/O Board</a> .....	7
<a href="#">ds5101_init</a> .....	9
<a href="#">Macros</a> .....	7

# Timing I/O Unit

**Introduction** To access standard and custom DWO applications.

## Where to go from here

## Information in this section

<a href="#">1-Phase PWM Signal Generation.....</a>	<a href="#">14</a>
To generate 1-phase PWM signals.	
<a href="#">3-Phase PWM Signal Generation.....</a>	<a href="#">19</a>
To generate a 3-phase pulse width modulation (PWM) signal.	
<a href="#">3-Phase PWM Signal Generation with Inverted and Non-Inverted Outputs.....</a>	<a href="#">25</a>
To generate a 3-phase pulse width modulation (PWM) signal with inverted and non-inverted outputs.	
<a href="#">Incremental Encoder Simulation.....</a>	<a href="#">31</a>
To simulate an incremental encoder.	
<a href="#">Monoflop Signal Generation.....</a>	<a href="#">36</a>
To generate a monoflop signal.	
<a href="#">Arbitrary Signal Generation.....</a>	<a href="#">45</a>
To access a custom DWO program that is executed on the DS5101 board.	

## Information in other sections

[Basics of the Timing I/O Unit \(DS5101 Features !\[\]\(830769b31eeeaca920791081939ff8ba\_img.jpg\)\)](#)  
Basics on generating signals with the DS5101 timing I/O unit.

# 1-Phase PWM Signal Generation

## Introduction

You can use the following functions to generate a 1-phase pulse width modulation (PWM) signal.

### Note

You have to initialize the DS5101 with `ds5101_init` function before you can use one of these functions.

## Where to go from here

## Information in this section

<a href="#">ds5101_pwm_load.....</a>	<a href="#">14</a>
To load the PWM DWO application.	
<a href="#">ds5101_pwm_parm.....</a>	<a href="#">15</a>
To get the current PWM period and duty cycle.	
<a href="#">ds5101_pwm_update.....</a>	<a href="#">16</a>
To update the period and duty cycle of the specified PWM channel.	
<a href="#">AP5101_PWM_MASK.....</a>	<a href="#">17</a>
To obtain a PWM channel mask.	

## ds5101\_pwm\_load

### Syntax

```
void ds5101_pwm_load (
    phs_addr_t base,
    long mask)
```

### Include file


`ap5101.h`

### Purpose

To load the PWM DWO application.

### Description


The PWM DWO application is loaded to the DS5101. Each PWM signal uses a single output channel and thus up to 16 PWM generators selected by the `mask` parameter can be loaded. On the rising edge of the PWM-signal an interrupt request is generated.

<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals (PHS Bus System Hardware Reference </a> ).												
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>mask</b> Specifies the load mask (0x0001 ... 0xFFFF). Use the <code>AP5101_PWM_MASK</code> macro to obtain the mask for an individual PWM channel.</p>												
<b>Return value</b>	None												
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.												
<b>Related topics</b>	<p>References</p> <table border="0"> <tr><td><a href="#">AP5101_PWM_MASK.....</a></td><td><a href="#">17</a></td></tr> <tr><td><a href="#">Base Address of the I/O Board.....</a></td><td><a href="#">7</a></td></tr> <tr><td><a href="#">ds5101_init.....</a></td><td><a href="#">9</a></td></tr> <tr><td><a href="#">ds5101_pwm_parm.....</a></td><td><a href="#">15</a></td></tr> <tr><td><a href="#">ds5101_pwm_update.....</a></td><td><a href="#">16</a></td></tr> <tr><td><a href="#">Macros.....</a></td><td><a href="#">7</a></td></tr> </table>	<a href="#">AP5101_PWM_MASK.....</a>	<a href="#">17</a>	<a href="#">Base Address of the I/O Board.....</a>	<a href="#">7</a>	<a href="#">ds5101_init.....</a>	<a href="#">9</a>	<a href="#">ds5101_pwm_parm.....</a>	<a href="#">15</a>	<a href="#">ds5101_pwm_update.....</a>	<a href="#">16</a>	<a href="#">Macros.....</a>	<a href="#">7</a>
<a href="#">AP5101_PWM_MASK.....</a>	<a href="#">17</a>												
<a href="#">Base Address of the I/O Board.....</a>	<a href="#">7</a>												
<a href="#">ds5101_init.....</a>	<a href="#">9</a>												
<a href="#">ds5101_pwm_parm.....</a>	<a href="#">15</a>												
<a href="#">ds5101_pwm_update.....</a>	<a href="#">16</a>												
<a href="#">Macros.....</a>	<a href="#">7</a>												

## ds5101\_pwm\_parm

<b>Syntax</b>	<pre>void ds5101_pwm_parm (     phs_addr_t base,     long channel,     dsfloat *tp,     dsfloat *duty)</pre>
<b>Include file</b>	<code>ap5101.h</code>
<b>Purpose</b>	To get the current PWM period and duty cycle.
<b>Description</b>	The current PWM period and duty cycle of the specified PWM channel are returned through the <code>tp</code> and <code>duty</code> parameters. The channel must have been loaded by using the <code>ds5101_pwm_load</code> function.

---

<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals (PHS Bus System Hardware Reference </a> ).
--------------------	---

---

<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>channel</b> Specifies the channel number within the range 1 ... 16.</p> <p><b>tp</b> Returns the current PWM period in seconds.</p> <p><b>duty</b> Returns the current PWM duty cycle.</p>
-------------------	---

---

<b>Return value</b>	None
---------------------	------

---

<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.
------------------------	--

---

**Related topics****References**

<a href="#">Base Address of the I/O Board.....</a>	<a href="#">7</a>
<a href="#">ds5101_init.....</a>	<a href="#">9</a>
<a href="#">ds5101_pwm_load.....</a>	<a href="#">14</a>
<a href="#">ds5101_pwm_update.....</a>	<a href="#">16</a>
<a href="#">Macros.....</a>	<a href="#">7</a>

## ds5101\_pwm\_update

---

**Syntax**

```
void ds5101_pwm_update (  
    phs_addr_t base,  
    long channel,  
    dsfloat tp,  
    dsfloat duty)
```


---

<b>Include file</b>	ap5101.h
---------------------	----------

---

<b>Purpose</b>	To update the period and duty cycle of the specified PWM channel.
----------------	---



<b>Description</b>	The period and duty cycle of the specified PWM channel are updated to the new values <code>tp</code> and <code>duty</code> after a complete PWM-signal period has been finished. The specified channel must have been loaded by using the <code>ds5101_pwm_load</code> function.
<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals (PHS Bus System Hardware Reference </a> ).
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>channel</b> Specifies the channel number within the range 1 ... 16.</p> <p><b>tp</b> Specifies the PWM period in seconds.</p> <p><b>duty</b> Specifies the PWM duty cycle within the range 0.0 ... 1.0.</p>
<b>Return value</b>	None
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.
<b>Related topics</b>	<p>References</p> <div> <a href="#">Base Address of the I/O Board</a>..... 7  <a href="#">ds5101_init</a>..... 9  <a href="#">ds5101_pwm_load</a>..... 14  <a href="#">ds5101_pwm_parm</a>..... 15  <a href="#">Macros</a>..... 7 </div>

## AP5101\_PWM\_MASK

<b>Syntax</b>	<code>long AP5101_PWM_MASK (int ch)</code>
<b>Include file</b>	<code>ap5101.h</code>
<b>Purpose</b>	To obtain a PWM channel mask.

<b>Description</b>	This macro can be used to obtain the channel mask for an individual PWM channel.
<b>Parameters</b>	<b>ch</b> Specifies the channel number within the range 1 ... 16.
<b>Return value</b>	This function returns the PWM channel mask.
<b>Related topics</b>	<b>References</b> <div>ds5101_pwm_load..... 14</div>

## 3-Phase PWM Signal Generation

### Introduction

You can use the following functions to generate a 3-phase pulse width modulation (PWM) signal.

#### Note

You have to initialize the DS5101 with `ds5101_init` function before you can use one of these functions.

### Where to go from here

#### Information in this section

<a href="#">ds5101_pwm3_intshift_load.....</a>	<a href="#">19</a>
To load the 3-phase PWM application with variable interrupt shift.	
<a href="#">ds5101_pwm3_intshift_parm.....</a>	<a href="#">21</a>
To get the current PWM period, duty cycle and interrupt shift.	
<a href="#">ds5101_pwm3_intshift_update.....</a>	<a href="#">22</a>
To update the period, duty cycle and interrupt shift of the specified PWM block.	
<a href="#">AP5101_PWM3_MASK.....</a>	<a href="#">23</a>
To obtain a 3-phase PWM channel mask.	

## ds5101\_pwm3\_intshift\_load

### Syntax

```
void ds5101_pwm3_intshift_load (
    phs_addr_t base,
    long mask)
```

### Include file

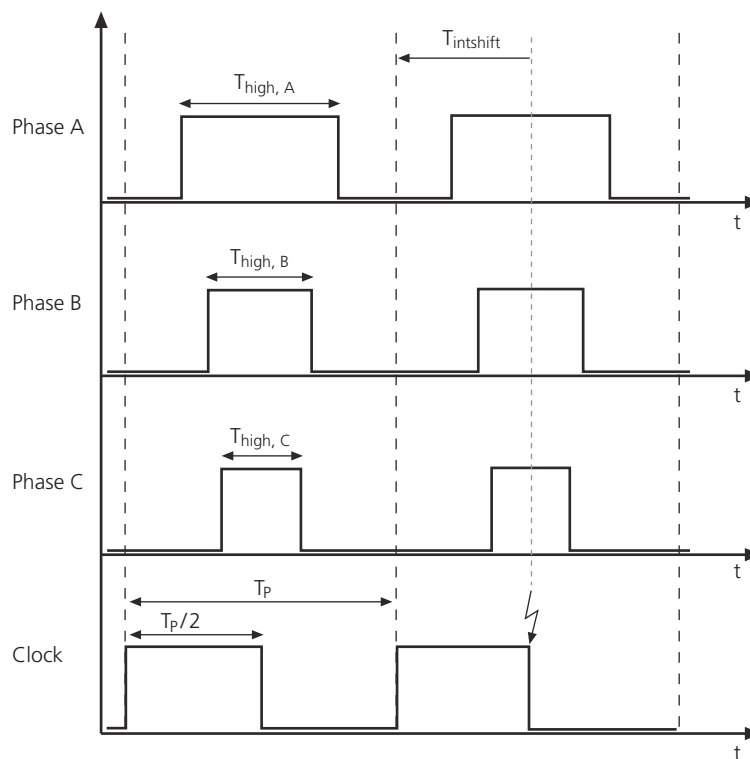
`ap5101.h`

### Purpose

To load the 3-phase PWM application with variable interrupt shift.

### Description

The 3-phase PWM application with interrupt shift is loaded to the DS5101. Each PWM block generates 4 output signals (A, B, C, clock) on subsequent channels. The clock channel generates an interrupt request at  $(tp/2 - \text{intshift})$ .



Up to 4 PWM blocks selected by the `mask` parameter can be loaded.

For detailed information about the channel assignment and channel masks, refer to [Mapping of I/O Signals \(PHS Bus System Hardware Reference\)](#).

#### Parameters

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

**mask** Specifies the load mask (0x000F ... 0xFFFF). Use the `AP5101_PWM3_MASK` macro to obtain the mask for an individual PWM block.

#### Return value


None

#### Execution times

For information, refer to [Function Execution Times](#) on page 63.

## Related topics

## References

AP5101_PWM3_MASK.....	23
Base Address of the I/O Board.....	7
ds5101_init.....	9
ds5101_pwm3_intshift_parm.....	21
ds5101_pwm3_intshift_update.....	22
Macros.....	7
Mapping of I/O Signals (PHS Bus System Hardware Reference  )	

## ds5101\_pwm3\_intshift\_parm

## Syntax

```
void ds5101_pwm3_intshift_parm (
    phs_addr_t base,
    long block,
    dsfloat *tp,
    dsfloat *da,
    dsfloat *db,
    dsfloat *dc,
    dsfloat *intshift)
```

## Include file

ap5101.h

## Purpose

To get the current PWM period, duty cycle and interrupt shift.

## Description

The current PWM period, duty cycles and interrupt shift value of the specified PWM block are returned through the **tp** and **da**, **db**, **dc** and **intshift** parameters. The block must have been loaded by using the **ds5101\_pwm3\_intshift\_load** function.

## I/O mapping

For details on the I/O mapping, refer to [Mapping of I/O Signals \(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.

**block** Specifies the block number within the range 1 ... 4.

**tp** Returns the current PWM period in seconds.

**da** Returns the current duty cycle of PWM phase A within the range 0.0 ... 1.0.

**db** Returns the current duty cycle of PWM phase B within the range 0.0 ... 1.0.

**dc** Returns the current duty cycle of PWM phase C within the range 0.0 ... 1.0.

**intshift** Returns the interrupt shift in seconds within the range 0.25e-6 ...  $tp/2 - 0.25e-6$ .

---

**Return value** None

---

**Execution times** For information, refer to [Function Execution Times](#) on page 63.

---

**Related topics**

**References**

Base Address of the I/O Board.....	7
ds5101_init.....	9
ds5101_pwm3_intshift_load.....	19
ds5101_pwm3_intshift_update.....	22
Macros.....	7

## ds5101\_pwm3\_intshift\_update

---

**Syntax**

```
void ds5101_pwm3_intshift_update (
    phs_addr_t base,
    long block,
    dsfloat tp,
    dsfloat da,
    dsfloat db,
    dsfloat dc,
    dsfloat intshift)
```

---


**Include file** ap5101.h

---

**Purpose** To update the period, duty cycle and interrupt shift of the specified PWM block.

---

**Description** The period, duty cycles and interrupt shift of the specified PWM block are updated to the new values **tp** and **da**, **db**, **dc** and **intshift**. The specified block must have been loaded by using the **ds5101\_pwm3\_intshift\_load** function.

<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals (PHS Bus System Hardware Reference </a> ).										
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>block</b> Specifies the block number within the range 1 ... 4.</p> <p><b>tp</b> Specifies the PWM period in seconds.</p> <p><b>da</b> Specifies the PWM phase A duty cycle within the range 0.0 ... 1.0.</p> <p><b>db</b> Specifies the PWM phase B duty cycle within the range 0.0 ... 1.0.</p> <p><b>dc</b> Specifies the PWM phase C duty cycle within the range 0.0 ... 1.0.</p> <p><b>intshift</b> Specifies the interrupt shift in seconds within the range 0.25e-6 ... <math>tp/2 - 0.25e-6</math>.</p>										
<b>Return value</b>	None										
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.										
<b>Related topics</b>	<p>References</p> <table> <tr> <td><a href="#">Base Address of the I/O Board</a></td><td>7</td></tr> <tr> <td><a href="#">ds5101_init</a></td><td>9</td></tr> <tr> <td><a href="#">ds5101_pwm3_intshift_load</a></td><td>19</td></tr> <tr> <td><a href="#">ds5101_pwm3_intshift_parm</a></td><td>21</td></tr> <tr> <td><a href="#">Macros</a></td><td>7</td></tr> </table>	<a href="#">Base Address of the I/O Board</a>	7	<a href="#">ds5101_init</a>	9	<a href="#">ds5101_pwm3_intshift_load</a>	19	<a href="#">ds5101_pwm3_intshift_parm</a>	21	<a href="#">Macros</a>	7
<a href="#">Base Address of the I/O Board</a>	7										
<a href="#">ds5101_init</a>	9										
<a href="#">ds5101_pwm3_intshift_load</a>	19										
<a href="#">ds5101_pwm3_intshift_parm</a>	21										
<a href="#">Macros</a>	7										

## AP5101\_PWM3\_MASK

<b>Syntax</b>	<code>long AP5101_PWM3_MASK (int blk)</code>
<b>Include file</b>	<code>ap5101.h</code>
<b>Purpose</b>	To obtain a 3-phase PWM channel mask.

<b>Description</b>	This macro can be used to obtain the channel mask for an individual 3-phase PWM block.
<b>Parameters</b>	<b>blk</b> Specifies the block number within the range 1 ... 4.
<b>Return value</b>	This function returns the 3-phase PWM channel mask.
<b>Related topics</b>	<b>References</b> <div><a href="#">ds5101_pwm3_intshift_load.....</a> 19</div>



# 3-Phase PWM Signal Generation with Inverted and Non-Inverted Outputs

**Introduction** You can use the following functions to generate a 3-phase pulse width modulation (PWM) signal with inverted and non-inverted outputs.

**Note** You have to initialize the DS5101 with `ds5101_init` function before you can use one of these functions.

Where to go from here	Information in this section
	<a href="#">ds5101_pwm6_intshift_load</a> ..... 25 To load the 3-phase/6-channel PWM application (with intshift).
	<a href="#">ds5101_pwm6_intshift_parm</a> ..... 27 To get the current PWM period, switch delay, duty cycles and interrupt shift of the specified PWM block.
	<a href="#">ds5101_pwm6_intshift_update</a> ..... 28 To update the period, switch delay, duty cycles and interrupt shift of the specified PWM block.
	<a href="#">AP5101_PWM6_MASK</a> ..... 30 To obtain a 3-phase/ 6-channel PWM channel mask.

## ds5101\_pwm6\_intshift\_load

**Syntax**

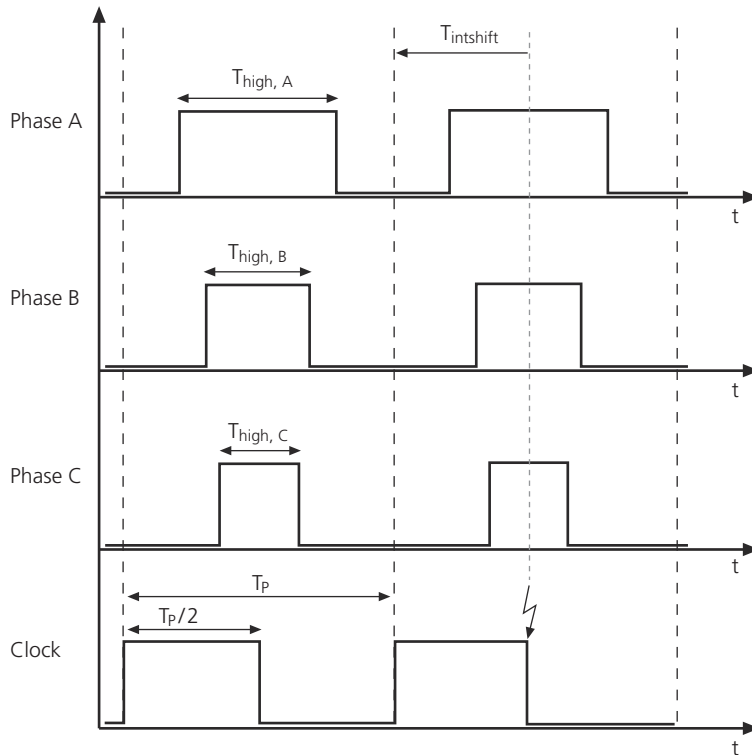
```
void ds5101_pwm6_intshift_load (  
    phs_addr_t base,  
    long mask)
```

**Include file** `ap5101.h`

**Purpose** To load the 3-phase/6-channel PWM application (with intshift).

**Description**

The 3-phase/6-channel PWM application is loaded to the DS5101. Each PWM block generates 7 output signals (A, /A, B, /B, C, /C, clock) on subsequent channels. The clock channel generates an interrupt request at  $(t_p/2 - \text{intshift})$ .



Up to 2 PWM blocks selected by the **mask** parameter can be loaded. Channels 8 and 16 are not used by this application.

**I/O mapping**

For details on the I/O mapping, refer to [Mapping of I/O Signals \(PHS Bus System Hardware Reference\)](#).

**Parameters**

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

**mask** Specifies the load mask (0x007F ... 0x3FFF). Use the `AP5101_PWM6_MASK` macro to obtain the mask for an individual PWM block.

**Return value**

None

**Execution times**

For information, refer to [Function Execution Times](#) on page 63.

**Related topics****References**

AP5101_PWM6_MASK.....	30
Base Address of the I/O Board.....	7
ds5101_init.....	9
ds5101_pwm6_intshift_parm.....	27
ds5101_pwm6_intshift_update.....	28
Macros.....	7

## ds5101\_pwm6\_intshift\_parm

**Syntax**

```
void ds5101_pwm6_intshift _parm (
    phs_addr_t base,
    long block,
    dsfloat *tp,
    dsfloat *td,
    dsfloat *da,
    dsfloat *db,
    dsfloat *dc,
    dsfloat *intshift)
```

**Include file**

ap5101.h

**Purpose**

To get the current PWM period, switch delay, duty cycles and interrupt shift of the specified PWM block.

**Description**

The current PWM period, switch delay, duty cycles and interrupt shift value of the specified PWM block are returned through the `tp`, `td`, and `da`, `db`, `dc` and `intshift` parameters. The block must have been loaded by using the `ds5101_pwm6_intshift_load` function.

**I/O mapping**

For details on the I/O mapping, refer to [Mapping of I/O Signals \(PHS Bus System Hardware Reference\)](#).

<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>block</b> Specifies the block number (1, 2).</p> <p><b>tp</b> Returns the current PWM period in seconds.</p> <p><b>td</b> Returns the current switch delay in seconds.</p> <p><b>da</b> Returns the current PWM phase A duty cycle within the range 0.0 ... 1.0.</p> <p><b>db</b> Returns the current PWM phase B duty cycle within the range 0.0 ... 1.0.</p> <p><b>dc</b> Returns the current PWM phase C duty cycle within the range 0.0 ... 1.0.</p> <p><b>intshift</b> Specifies the interrupt shift in seconds within the range 0.25e-6 ... <math>tp/2 - 0.25e-6</math>.</p>										
<b>Return value</b>	None										
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.										
<b>Related topics</b>	<p><b>References</b></p> <table border="0"> <tr> <td><a href="#">Base Address of the I/O Board</a>.....</td> <td>7</td> </tr> <tr> <td><a href="#">ds5101_init</a>.....</td> <td>9</td> </tr> <tr> <td><a href="#">ds5101_pwm6_intshift_load</a>.....</td> <td>25</td> </tr> <tr> <td><a href="#">ds5101_pwm6_intshift_update</a>.....</td> <td>28</td> </tr> <tr> <td><a href="#">Macros</a>.....</td> <td>7</td> </tr> </table>	<a href="#">Base Address of the I/O Board</a> .....	7	<a href="#">ds5101_init</a> .....	9	<a href="#">ds5101_pwm6_intshift_load</a> .....	25	<a href="#">ds5101_pwm6_intshift_update</a> .....	28	<a href="#">Macros</a> .....	7
<a href="#">Base Address of the I/O Board</a> .....	7										
<a href="#">ds5101_init</a> .....	9										
<a href="#">ds5101_pwm6_intshift_load</a> .....	25										
<a href="#">ds5101_pwm6_intshift_update</a> .....	28										
<a href="#">Macros</a> .....	7										


## ds5101\_pwm6\_intshift\_update

### Syntax

```
void ds5101_pwm6_intshift_update (
    phs_addr_t base,
    long block,
    dsfloat tp,
    dsfloat td,
    dsfloat da,
    dsfloat db,
    dsfloat dc,
    dsfloat intshift)
```

### Include file

ap5101.h

<b>Purpose</b>	To update the period, switch delay, duty cycles and interrupt shift of the specified PWM block.										
<b>Description</b>	The period, duty cycles and the interrupt shift value of the specified PWM block are updated to the new values <b>tp</b> and <b>da</b> , <b>db</b> , <b>dc</b> and <b>intshift</b> . The switch delay between the edges of the non-inverted phases and the related inverted phases is given by the <b>td</b> parameter. The specified block must have been loaded by using the <code>ds5101_pwm6_intshift_load</code> function.										
<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals (PHS Bus System Hardware Reference </a> ).										
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>block</b> Specifies the block number (1, 2).</p> <p><b>tp</b> Specifies the PWM period in seconds.</p> <p><b>td</b> Specifies the switch delay in seconds.</p> <p><b>da</b> Specifies the PWM phase A duty cycle within the range 0.0 ... 1.0.</p> <p><b>db</b> Specifies the PWM phase B duty cycle within the range 0.0 ... 1.0.</p> <p><b>dc</b> Specifies the PWM phase C duty cycle within the range 0.0 ... 1.0.</p> <p><b>intshift</b> Specifies the interrupt shift in seconds within the range 0.25e-6 ... <math>tp/2 - 0.25e-6</math>.</p>										
<b>Return value</b>	None										
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.										
<b>Related topics</b>	<p>References</p> <table> <tr> <td><a href="#">Base Address of the I/O Board</a></td><td>7</td></tr> <tr> <td><a href="#">ds5101_init</a></td><td>9</td></tr> <tr> <td><a href="#">ds5101_pwm6_intshift_load</a></td><td>25</td></tr> <tr> <td><a href="#">ds5101_pwm6_intshift_parm</a></td><td>27</td></tr> <tr> <td><a href="#">Macros</a></td><td>7</td></tr> </table>	<a href="#">Base Address of the I/O Board</a>	7	<a href="#">ds5101_init</a>	9	<a href="#">ds5101_pwm6_intshift_load</a>	25	<a href="#">ds5101_pwm6_intshift_parm</a>	27	<a href="#">Macros</a>	7
<a href="#">Base Address of the I/O Board</a>	7										
<a href="#">ds5101_init</a>	9										
<a href="#">ds5101_pwm6_intshift_load</a>	25										
<a href="#">ds5101_pwm6_intshift_parm</a>	27										
<a href="#">Macros</a>	7										

## AP5101\_PWM6\_MASK

<b>Syntax</b>	<code>long AP5101_PWM6_MASK (int blk)</code>
<b>Include file</b>	<code>ap5101.h</code>
<b>Purpose</b>	To obtain a 3-phase/ 6-channel PWM channel mask.
<b>Description</b>	This macro can be used to obtain the channel mask for an individual 3-phase/ 6-channel PWM block.
<b>Parameters</b>	<b>blk</b> Specifies the block number (1 or 2).
<b>Return value</b>	This function returns the 3-phase/6-channel PWM channel mask.
<b>Related topics</b>	<b>References</b> <a href="#">ds5101_pwm6_intshift_load.....25</a>

# Incremental Encoder Simulation

## Introduction

You can use the following functions to simulate an incremental encoder.

### Note

You have to initialize the DS5101 with `ds5101_init` function before you can use one of these functions.

## Where to go from here

## Information in this section

<a href="#">ds5101_inc_load</a> .....	31
To load the incremental encoder simulation DWO application.	
<a href="#">ds5101_inc_parm</a> .....	32
To get the current velocity of an incremental encoder simulation block.	
<a href="#">ds5101_inc_update</a> .....	33
To update the velocity of an incremental encoder simulation block.	
<a href="#">AP5101_INC_MASK</a> .....	34
To obtain an incremental encoder channel mask.	

## ds5101\_inc\_load

### Syntax

```
void ds5101_inc_load (
    phs_addr_t base,
    long mask)
```

### Include file

`ap5101.h`

### Purpose

To load the incremental encoder simulation DWO application.

### Description

The incremental encoder simulation DWO application is loaded to the DS5101. Each incremental encoder generates 2 output signals (PHI0, PHI90) on subsequent channels. The two output signals (PHI0, PHI90) form one incremental encoder block. Up to 7 incremental encoder blocks selected by the `mask` parameter can be loaded. The number of blocks is limited to 7 because only 7

flags are available for controlling the direction on DS5101 boards up to revision DS5101-02. Channels 15 and 16 are not used by this application.

### I/O mapping

For details on the I/O mapping, refer to [Mapping of I/O Signals \(PHS Bus System Hardware Reference\)](#).

### Parameters

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

**mask** Specifies the load mask (0x0003 ... 0x3FFF). Use the `AP5101_INC_MASK` macro to obtain the mask for an individual incremental encoder block.

### Return value

None

### Execution times

For information, refer to [Function Execution Times](#) on page 63.

### Related topics

#### References

<a href="#">AP5101_INC_MASK</a>	34
<a href="#">Base Address of the I/O Board</a>	7
<a href="#">ds5101_inc_parm</a>	32
<a href="#">ds5101_inc_update</a>	33
<a href="#">ds5101_init</a>	9
<a href="#">Macros</a>	7

## ds5101\_inc\_parm

### Syntax

```
void ds5101_inc_parm (
    phs_addr_t base,
    long block,
    dsfloat *vel)
```


### Include file

`ap5101.h`

### Purpose


To get the current velocity of an incremental encoder simulation block.



<b>Description</b>	The current velocity of the specified incremental encoder simulation block is returned through the <b>vel</b> parameter. The block must have been loaded by using the <b>ds5101_inc_load</b> function.
<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals (PHS Bus System Hardware Reference </a> ).
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>block</b> Specifies the block number within the range 1 ... 7.</p> <p><b>vel</b> Returns the current velocity.</p>
<b>Return value</b>	None
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.
<b>Related topics</b>	<p>References</p> <div> <a href="#">Base Address of the I/O Board.....7</a>  <a href="#">ds5101_inc_load.....31</a>  <a href="#">ds5101_inc_update.....33</a>  <a href="#">ds5101_init.....9</a>  <a href="#">Macros.....7</a> </div>

## ds5101\_inc\_update

<b>Syntax</b>	<pre>void ds5101_inc_update (     phs_addr_t base,     long block,     dsfloat vel)</pre>
<b>Include file</b>	ap5101.h
<b>Purpose</b>	To update the velocity of an incremental encoder simulation block.

<b>Description</b>	The velocity parameter of the specified incremental encoder simulation block is updated to the new value <b>vel</b> . The block must have been loaded by using the <b>ds5101_inc_load</b> function. Each block features two outputs (phi0, phi90). The number of blocks is limited to 7 because only 7 flags are available for controlling the direction on DS5101 boards up to revision DS5101-02. The channel numbers 1, 2 / 3, 4 / ... 13, 14 are available.										
<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals (PHS Bus System Hardware Reference </a> ).										
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>block</b> Specifies the block number within the range 1 ... 7.</p> <p><b>vel</b> Specifies the new velocity (in number of periods / second) within the range -1.0e6 ... +1.0e6.</p>										
<b>Return value</b>	None										
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.										
<b>Related topics</b>	<p><b>References</b></p> <table border="0"> <tr><td><a href="#">Base Address of the I/O Board</a></td><td>7</td></tr> <tr><td><a href="#">ds5101_inc_load</a></td><td>31</td></tr> <tr><td><a href="#">ds5101_inc_parm</a></td><td>32</td></tr> <tr><td><a href="#">ds5101_init</a></td><td>9</td></tr> <tr><td><a href="#">Macros</a></td><td>7</td></tr> </table>	<a href="#">Base Address of the I/O Board</a>	7	<a href="#">ds5101_inc_load</a>	31	<a href="#">ds5101_inc_parm</a>	32	<a href="#">ds5101_init</a>	9	<a href="#">Macros</a>	7
<a href="#">Base Address of the I/O Board</a>	7										
<a href="#">ds5101_inc_load</a>	31										
<a href="#">ds5101_inc_parm</a>	32										
<a href="#">ds5101_init</a>	9										
<a href="#">Macros</a>	7										

## AP5101\_INC\_MASK

<b>Syntax</b>	<code>long AP5101_INC_MASK (int blk)</code>
<b>Include file</b>	<code>ap5101.h</code>
<b>Purpose</b>	To obtain an incremental encoder channel mask.

Description	This macro can be used to obtain the channel mask for an individual incremental encoder block.
Parameters	<b>blk</b> Specifies the block number within the range 1 ... 7.
Return value	This function returns the incremental encoder channel mask.
Related topics	<div>References<div>ds5101_inc_load.....31</div></div>

# Monoflop Signal Generation

## Introduction

You can use the following functions to generate a monoflop signal.

### Note

You have to initialize the DS5101 with `ds5101_init` function before you can use one of these functions.

## Where to go from here

## Information in this section

<a href="#">ds5101_mono_load</a> .....	36
To load the monoflop signal DWO application.	
<a href="#">ds5101_mono_parm</a> .....	37
To get the current pulse width of a monoflop signal.	
<a href="#">ds5101_mono_start</a> .....	38
To start the specified monoflop signal.	
<a href="#">ds5101_mono_update</a> .....	39
To update the pulse width of a monoflop signal.	
<a href="#">ds5101_delayed_mono_load</a> .....	40
To load the delayed monoflop signal application.	
<a href="#">ds5101_delayed_mono_parm</a> .....	41
To read the current delay time and pulse width of a delayed monoflop signal.	
<a href="#">ds5101_delayed_mono_update</a> .....	43
To update the current delay time and pulse width of a monoflop signal.	
<a href="#">AP5101_MONO_MASK</a> .....	44
To obtain a monoflop channel mask.	


## ds5101\_mono\_load

### Syntax

```
void ds5101_mono_load (
    phs_addr_t base,
    long mask)
```

### Include file

`ap5101.h`

<b>Purpose</b>	To load the monoflop signal DWO application.														
<b>Description</b>	The monoflop signal DWO application is loaded to the DS5101. Each monoflop signal generates a high-active (MONO) and a low-active (/MONO) output signal on subsequent channels. Up to 8 monoflop signal blocks selected by the <b>mask</b> parameter can be loaded. On the falling edge of the high-active channel an interrupt request is generated.														
<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals (PHS Bus System Hardware Reference </a> ).														
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>mask</b> Specifies the load mask (0x0003 ... 0xFFFF). Use the <code>AP5101_MONO_MASK</code> macro to obtain the mask for an individual monoflop signal block.</p>														
<b>Return value</b>	None														
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.														
<b>Related topics</b>	<p><b>References</b></p> <table> <tr><td><a href="#">AP5101_MONO_MASK</a></td><td>44</td></tr> <tr><td><a href="#">Base Address of the I/O Board</a></td><td>7</td></tr> <tr><td><a href="#">ds5101_delayed_mono_parm</a></td><td>41</td></tr> <tr><td><a href="#">ds5101_init</a></td><td>9</td></tr> <tr><td><a href="#">ds5101_mono_start</a></td><td>38</td></tr> <tr><td><a href="#">ds5101_mono_update</a></td><td>39</td></tr> <tr><td><a href="#">Macros</a></td><td>7</td></tr> </table>	<a href="#">AP5101_MONO_MASK</a>	44	<a href="#">Base Address of the I/O Board</a>	7	<a href="#">ds5101_delayed_mono_parm</a>	41	<a href="#">ds5101_init</a>	9	<a href="#">ds5101_mono_start</a>	38	<a href="#">ds5101_mono_update</a>	39	<a href="#">Macros</a>	7
<a href="#">AP5101_MONO_MASK</a>	44														
<a href="#">Base Address of the I/O Board</a>	7														
<a href="#">ds5101_delayed_mono_parm</a>	41														
<a href="#">ds5101_init</a>	9														
<a href="#">ds5101_mono_start</a>	38														
<a href="#">ds5101_mono_update</a>	39														
<a href="#">Macros</a>	7														

## ds5101\_mono\_parm

### Syntax

```
void ds5101_mono_parm (
    phs_addr_t base,
    long block,
    dsfloat *tm)
```

<b>Include file</b>	<code>ap5101.h</code>												
<b>Purpose</b>	To get the current pulse width of a monoflop signal.												
<b>Description</b>	The current pulse width of the specified monoflop signal is returned through the <code>tm</code> parameter. The block must have been loaded by using the <code>ds5101_mono_load</code> function.												
<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals (PHS Bus System Hardware Reference)</a> .												
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>block</b> Specifies the block number within the range 1 ... 8.</p> <p><b>tm</b> Returns the current pulse width.</p>												
<b>Return value</b>	None												
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.												
<b>Related topics</b>	<p>References</p> <table> <tr> <td><a href="#">Base Address of the I/O Board</a></td> <td>7</td> </tr> <tr> <td><a href="#">ds5101_init</a></td> <td>9</td> </tr> <tr> <td><a href="#">ds5101_mono_load</a></td> <td>36</td> </tr> <tr> <td><a href="#">ds5101_mono_start</a></td> <td>38</td> </tr> <tr> <td><a href="#">ds5101_mono_update</a></td> <td>39</td> </tr> <tr> <td><a href="#">Macros</a></td> <td>7</td> </tr> </table>	<a href="#">Base Address of the I/O Board</a>	7	<a href="#">ds5101_init</a>	9	<a href="#">ds5101_mono_load</a>	36	<a href="#">ds5101_mono_start</a>	38	<a href="#">ds5101_mono_update</a>	39	<a href="#">Macros</a>	7
<a href="#">Base Address of the I/O Board</a>	7												
<a href="#">ds5101_init</a>	9												
<a href="#">ds5101_mono_load</a>	36												
<a href="#">ds5101_mono_start</a>	38												
<a href="#">ds5101_mono_update</a>	39												
<a href="#">Macros</a>	7												

## ds5101\_mono\_start

### Syntax


```
void ds5101_mono_start (
    phs_addr_t base,
    long block)
```

<b>Include file</b>	<code>ap5101.h</code>
<b>Purpose</b>	To start the specified monoflop signal.
<b>Description</b>	The specified monoflop signal is started by issuing a software trigger from the DSP. The block must have been loaded by using the <code>ds5101_mono_load</code> or <code>ds5101_delayed_mono_load</code> function.
<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals (PHS Bus System Hardware Reference)</a> .
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>block</b> Specifies the block number within the range 1 ... 8.</p>
<b>Return value</b>	None
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.
<b>Related topics</b>	<p>References</p> <ul style="list-style-type: none"> <li><a href="#">Base Address of the I/O Board</a>..... 7</li> <li><a href="#">ds5101_delayed_mono_load</a>..... 40</li> <li><a href="#">ds5101_delayed_mono_parm</a>..... 41</li> <li><a href="#">ds5101_delayed_mono_update</a>..... 43</li> <li><a href="#">ds5101_init</a>..... 9</li> <li><a href="#">ds5101_mono_load</a>..... 36</li> <li><a href="#">ds5101_mono_parm</a>..... 37</li> <li><a href="#">ds5101_mono_update</a>..... 39</li> <li><a href="#">Macros</a>..... 7</li> </ul>

## ds5101\_mono\_update

### Syntax


```
void ds5101_mono_update (
    phs_addr_t base,
    long block,
    dsfloat tm)
```

<b>Include file</b>	<code>ap5101.h</code>
<b>Purpose</b>	To update the pulse width of a monoflop signal.
<b>Description</b>	The pulse width of the specified monoflop signal is updated to the new value <code>tm</code> . The block must have been loaded by using the <code>ds5101_mono_load</code> function.
<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals (PHS Bus System Hardware Reference </a> ).
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>block</b> Specifies the block number within the range 1 ... 8.</p> <p><b>tm</b> Specifies the pulse width in seconds.</p>
<b>Return value</b>	None
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.
<b>Related topics</b>	<p>References</p> <div> <ul style="list-style-type: none"> <li><a href="#">Base Address of the I/O Board.....</a> 7</li> <li><a href="#">ds5101_init.....</a> 9</li> <li><a href="#">ds5101_mono_load.....</a> 36</li> <li><a href="#">ds5101_mono_parm.....</a> 37</li> <li><a href="#">ds5101_mono_start.....</a> 38</li> <li><a href="#">Macros.....</a> 7</li> </ul> </div>

## ds5101\_delayed\_mono\_load

<b>Syntax</b>	<pre>void ds5101_delayed_mono_load (     phs_addr_t base,     long mask)</pre>
<b>Include file</b>	<code>ap5101.h</code>




<b>Purpose</b>	To load the delayed monoflop signal application.														
<b>Description</b>	The delayed monoflop signal DWO application is loaded to the DS5101. Each delayed monoflop signal generates a high-active (MONO) and a low-active (/MONO) output signal on subsequent channels. Up to 8 monoflop signal blocks selected by the <b>mask</b> parameter can be loaded. On the rising edge of the high-active channel and on the rising edge of the low-active channel an interrupt request is generated.														
<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals (PHS Bus System Hardware Reference </a> ).														
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>mask</b> Specifies the load mask (0x0003 ... 0xFFFF). Use the <code>AP5101_MONO_MASK</code> macro to obtain the mask for an individual monoflop signal block.</p>														
<b>Return value</b>	None														
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.														
<b>Related topics</b>	<p>References</p> <table> <tr> <td><a href="#">AP5101_MONO_MASK</a></td><td>44</td></tr> <tr> <td><a href="#">Base Address of the I/O Board</a></td><td>7</td></tr> <tr> <td><a href="#">ds5101_delayed_mono_parm</a></td><td>41</td></tr> <tr> <td><a href="#">ds5101_delayed_mono_update</a></td><td>43</td></tr> <tr> <td><a href="#">ds5101_init</a></td><td>9</td></tr> <tr> <td><a href="#">ds5101_mono_start</a></td><td>38</td></tr> <tr> <td><a href="#">Macros</a></td><td>7</td></tr> </table>	<a href="#">AP5101_MONO_MASK</a>	44	<a href="#">Base Address of the I/O Board</a>	7	<a href="#">ds5101_delayed_mono_parm</a>	41	<a href="#">ds5101_delayed_mono_update</a>	43	<a href="#">ds5101_init</a>	9	<a href="#">ds5101_mono_start</a>	38	<a href="#">Macros</a>	7
<a href="#">AP5101_MONO_MASK</a>	44														
<a href="#">Base Address of the I/O Board</a>	7														
<a href="#">ds5101_delayed_mono_parm</a>	41														
<a href="#">ds5101_delayed_mono_update</a>	43														
<a href="#">ds5101_init</a>	9														
<a href="#">ds5101_mono_start</a>	38														
<a href="#">Macros</a>	7														

## ds5101\_delayed\_mono\_parm

### Syntax

```
void ds5101_delayed_mono_parm (
    phs_addr_t base,
    long block,
    dsfloat *td,
    dsfloat *tm)
```

<b>Include file</b>	<code>ap5101.h</code>
<b>Purpose</b>	To read the current delay time and pulse width of a delayed monoflop signal.
<b>Description</b>	The current pulse width of the specified monoflop signal is returned through the <code>tm</code> parameter. The current delay time is returned through the <code>td</code> parameter. The block must have been loaded by using the <code>ds5101_mono_load</code> function.
<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals (PHS Bus System Hardware Reference </a> ).
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>block</b> Specifies the block number within the range 1 ... 8.</p> <p><b>td</b> Returns the delay time (in seconds) before monoflop pulse is generated.</p> <p><b>tm</b> Returns current monoflop pulse width in seconds.</p>
<b>Return value</b>	None
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.
<b>Related topics</b>	<p>References</p> <ul style="list-style-type: none"> <li><a href="#">Base Address of the I/O Board</a>..... 7</li> <li><a href="#">ds5101_delayed_mono_load</a>..... 40</li> <li><a href="#">ds5101_delayed_mono_update</a>..... 43</li> <li><a href="#">ds5101_init</a>..... 9</li> <li><a href="#">ds5101_mono_load</a>..... 36</li> <li><a href="#">ds5101_mono_start</a>..... 38</li> <li><a href="#">Macros</a>..... 7</li> </ul>

## ds5101\_delayed\_mono\_update

### Syntax

```
void ds5101_delayed_mono_update (
    phs_addr_t base,
    long block,
    dsfloat td,
    dsfloat tm)
```

### Include file

ap5101.h

### Purpose

To update the current delay time and pulse width of a monoflop signal.

### Description

The delay time of the specified monoflop signal is updated to the new value **td**. The pulse width of the specified monoflop signal is updated to the new value **tm**. The block must have been loaded by using the **ds5101\_mono\_load** function.

### I/O mapping

For details on the I/O mapping, refer to [Mapping of I/O Signals \(PHS Bus System Hardware Reference\)](#).

### Parameters

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

**block** Specifies the block number within the range 1 ... 8.

**td** Specifies the delay time before pulse in seconds.

**tm** Specifies the pulse width in seconds.

### Return value

None

### Execution times

For information, refer to [Function Execution Times](#) on page 63.

### Related topics

#### References

Base Address of the I/O Board.....	7
ds5101_delayed_mono_load.....	40
ds5101_delayed_mono_parm.....	41
ds5101_init.....	9
ds5101_mono_load.....	36
ds5101_mono_start.....	38
Macros.....	7

## AP5101\_MONO\_MASK

<b>Syntax</b>	<code>long AP5101_MONO_MASK (int blk)</code>
<b>Include file</b>	<code>ap5101.h</code>
<b>Purpose</b>	To obtain a monoflop channel mask.
<b>Description</b>	This macro can be used to obtain the channel mask for an individual monoflop block.
<b>Parameters</b>	<b>blk</b> Specifies the block number within the range 1 ... 8.
<b>Return value</b>	This function returns the monoflop channel mask.
<b>Related topics</b>	<b>References</b> <div><a href="#">ds5101_delayed_mono_load.....</a> 40 <a href="#">ds5101_mono_load.....</a> 36</div>

# Arbitrary Signal Generation

**Introduction** When generating arbitrary pulse patterns, you can use the RTLib functions to access your custom DWO program that is executed on the DS5101 board.

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

<a href="#">Basics on Arbitrary Signal Generation.....</a>	<a href="#">45</a>
When generating arbitrary pulse patterns, you can use the RTLib functions to access your custom DWO program that is executed on the DS5101 board.	
<a href="#">ds5101_clr_flags.....</a>	<a href="#">46</a>
To clear control flags.	
<a href="#">ds5101_flag.....</a>	<a href="#">47</a>
To get the current state of a control flag.	
<a href="#">ds5101_set_flags.....</a>	<a href="#">48</a>
To set control flags.	
<a href="#">ds5101_update_flags.....</a>	<a href="#">49</a>
To write all control flags at once.	
<a href="#">ds5101_start.....</a>	<a href="#">50</a>
To start signal generation.	
<a href="#">ds5101_stop.....</a>	<a href="#">51</a>
To stop signal generation.	
<a href="#">ds5101_state.....</a>	<a href="#">52</a>
To read the current output state.	
<a href="#">ds5101_trigger.....</a>	<a href="#">53</a>
To send a trigger event to the specified channels.	
<a href="#">DS5101_FLAG_MASK.....</a>	<a href="#">54</a>
To obtain a flag mask for the control flag accessing functions.	
<a href="#">DS5101_TRG_MASK.....</a>	<a href="#">55</a>
To obtain a trigger channel mask.	

## Basics on Arbitrary Signal Generation

**Introduction** When generating arbitrary pulse patterns, you can use the RTLib functions to access your custom DWO program that is executed on the DS5101 board.

<b>Initialization</b>	You have to initialize the DS5101 with <code>ds5101_init</code> function before you can use one of these functions.
<b>DWO program</b>	For information on the DWO software environment and the DWO compiler, refer to the <i>DS5101 Board Reference</i> (C:\Program Files <x86>\Common Files\dSPACE\Help <ReleaseVersion>\ Print\DS5101BoardReference.pdf).
<b>Related topics</b>	<p>References</p> <p><a href="#">ds5101_init.....9</a></p>

## ds5101\_clr\_flags

<b>Syntax</b>	<pre>void ds5101_clr_flags (     phs_addr_t base,     long mask)</pre>
<b>Include file</b>	<code>ds5101.h</code>
<b>Purpose</b>	To clear control flags.
<b>Description</b>	<p>One or more control flags as specified by a bitmask are cleared. Setting the bit value to 1 clears the corresponding flag. All flags that are not masked remain unaffected. The maximum number of flags is 31 for DS5101 as of revision DS5101-04 or 7 for DS5101 up to revision DS5101-02.</p> <div> <p><b>Note</b></p> <p>The control flags 8 ... 31 are only available on DS5101 boards as of revision DS5101-04.</p> </div>
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>mask</b> Specifies the control flags to be cleared. DS5101 boards up to revision DS5101-02 use an 8-bit right-aligned bitmask (0x00 ... 0x7F). DS5101 boards as of revision DS5101-04 use a 32-bit right-aligned bitmask (0x00 ... 0x7FFFFFFF).</p>

To obtain the mask for an individual channel, use the `DS5101_FLAG_MASK` macro.

Return value	None
Execution times	For information, refer to <a href="#">Function Execution Times</a> on page 63.
Example	<pre>// clear the control flags 15 and 20 ds5101_clr_flags(DS5101_1_BASE, DS5101_FLAG_MASK(15)                   DS5101_FLAG_MASK(20) );</pre>

Related topics	References
	<div>Base Address of the I/O Board.....7 ds5101_flag.....47 ds5101_init.....9 ds5101_set_flags.....48 ds5101_update_flags.....49 Macros.....7</div>

## ds5101\_flag

Syntax	<pre>int ds5101_flag (     phs_addr_t base,     int flag)</pre>
Include file	ds5101.h
Purpose	To get the current state of a control flag.
Description	<p>The function returns the current state of the control flag specified by the <code>flag</code> parameter. The maximum number of flags is 7 (boards up to revision DS5101-02) or 31 (boards as of revision DS5101-04).</p> <div><b>Note</b> The control flags 8 ... 31 are only available on DS5101 boards as of revision DS5101-04.</div>

<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>flag</b> Specifies the control flag number within the range 1 ... 7 (boards up to revision DS5101-02) or 1 ... 31 (boards as of revision DS5101-04).</p>												
<b>Return value</b>	This function returns the state of the specified control flag (0, 1).												
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.												
<b>Example</b>	<pre>// get the status of flag 3 state = ds5101_flag(DS5101_1_BASE, 3);</pre>												
<b>Related topics</b>	<p>References</p> <table> <tr> <td>Base Address of the I/O Board.....</td> <td>7</td> </tr> <tr> <td>ds5101_clr_flags.....</td> <td>46</td> </tr> <tr> <td>ds5101_init.....</td> <td>9</td> </tr> <tr> <td>ds5101_set_flags.....</td> <td>48</td> </tr> <tr> <td>ds5101_update_flags.....</td> <td>49</td> </tr> <tr> <td>Macros.....</td> <td>7</td> </tr> </table>	Base Address of the I/O Board.....	7	ds5101_clr_flags.....	46	ds5101_init.....	9	ds5101_set_flags.....	48	ds5101_update_flags.....	49	Macros.....	7
Base Address of the I/O Board.....	7												
ds5101_clr_flags.....	46												
ds5101_init.....	9												
ds5101_set_flags.....	48												
ds5101_update_flags.....	49												
Macros.....	7												

## ds5101\_set\_flags

<b>Syntax</b>	<pre>void ds5101_set_flags (     phs_addr_t base,     long mask)</pre>
<b>Include file</b>	ds5101.h
<b>Purpose</b>	To set control flags.
<b>Description</b>	One or multiple control flags as specified by a bitmask are set to 1. All flags not masked remain unaffected. The maximum number of flags is 7 (boards up to revision DS5101-02) or 31 (boards as of revision DS5101-04).



Note

The control flags 8 ... 31 are only available on DS5101 boards as of revision DS5101-04.

**Parameters**

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

**mask** Specifies the control flags to be set. DS5101 boards up to revision DS5101-02 use an 8-bit right-aligned bitmask (0x00 ... 0x7F). DS5101 boards as of revision DS5101-04 use a 32-bit right-aligned bitmask (0x00 ... 0x7FFFFFFF). To obtain the mask for an individual channel, use the `DS5101_FLAG_MASK` macro.

**Return value** None

**Execution times** For information, refer to [Function Execution Times](#) on page 63.

**Example**

```
// set the control flags 2 and 7
ds5101_set_flags(DS5101_1_BASE, DS5101_FLAG_MASK(2) |
                DS5101_FLAG_MASK(7) );
```

Related topics

References	
<a href="#">Base Address of the I/O Board</a>	7
<a href="#">ds5101_clr_flags</a>	46
<a href="#">ds5101_flag</a>	47
<a href="#">DS5101_FLAG_MASK</a>	54
<a href="#">ds5101_init</a>	9
<a href="#">ds5101_update_flags</a>	49

# ds5101\_update\_flags

**Syntax**

```
void ds5101_update_flags (
    phs_addr_t base,
    long mask)
```

**Include file** ds5101.h

<b>Purpose</b>	To write all control flags at once.												
<b>Description</b>	All control flags are set to their corresponding bit value specified by the bitmask. This mean, the bitmask is directly written into the flag register. The maximum number of flags is 7 (boards up to revision DS5101-02) or 31 (boards as of revision DS5101-04).												
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>mask</b> Specifies the bit values of all control flags. DS5101 boards up to revision DS5101-02 use an 8-bit right-aligned bitmask (0x00 ... 0x7F). DS5101 boards as of revision DS5101-04 use a 32-bit right-aligned bitmask (0x00 ... 0xFFFFFFFF). To obtain the mask for an individual channel, use the <code>DS5101_FLAG_MASK</code> macro.</p>												
<b>Return value</b>	None												
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.												
<b>Example</b>	<pre>// set flag 1 and 15, all other flags are cleared ds5101_update_flags(DS5101_1_BASE, DS5101_FLAG_MASK(1)                      DS5101_FLAG_MASK(15) );</pre>												
<b>Related topics</b>	<p>References</p> <table> <tr> <td>Base Address of the I/O Board.....</td> <td>7</td> </tr> <tr> <td>ds5101_clr_flags.....</td> <td>46</td> </tr> <tr> <td>ds5101_flag.....</td> <td>47</td> </tr> <tr> <td>DS5101_FLAG_MASK.....</td> <td>54</td> </tr> <tr> <td>ds5101_init.....</td> <td>9</td> </tr> <tr> <td>ds5101_set_flags.....</td> <td>48</td> </tr> </table>	Base Address of the I/O Board.....	7	ds5101_clr_flags.....	46	ds5101_flag.....	47	DS5101_FLAG_MASK.....	54	ds5101_init.....	9	ds5101_set_flags.....	48
Base Address of the I/O Board.....	7												
ds5101_clr_flags.....	46												
ds5101_flag.....	47												
DS5101_FLAG_MASK.....	54												
ds5101_init.....	9												
ds5101_set_flags.....	48												

## ds5101\_start

### Syntax

```
void ds5101_start (
    phs_addr_t base,
    long mask)
```

Include file	ds5101.h
Purpose	To start signal generation.
Description	Signal generation is started on one or multiple channels as specified by a bitmask. Setting the bit value to 1 starts the signal generation.
Parameters	<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>mask</b> Specifies the channels to be started using a 16-bit right-aligned bitmask (0x0000 ... 0xFFFF).
Return value	None
Execution times	For information, refer to <a href="#">Function Execution Times</a> on page 63.
Example	<pre>// start channel 1 and 7 ds5101_start(DS5101_1_BASE, 0x0041);</pre>
Related topics	<div>References</div> <div><a href="#">Base Address of the I/O Board.....7</a> <a href="#">ds5101_init.....9</a> <a href="#">ds5101_stop.....51</a> <a href="#">Macros.....7</a></div>


## ds5101\_stop

Syntax	<pre>void ds5101_stop (     phs_addr_t base,     long mask)</pre>
Include file	ds5101.h

<b>Purpose</b>	To stop signal generation.						
<b>Description</b>	Signal generation is stopped on one or multiple channels as specified by a bitmask. Setting the bit value to 1 stops the signal generation.						
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>mask</b> Specifies the channels to be stopped using a 16-bit right-aligned bitmask (0x0000 ... 0xFFFF).</p>						
<b>Return value</b>	None						
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.						
<b>Example</b>	<pre>// stop channel 5 and 16 ds5101_stop(DS5101_1_BASE, 0x00008010);</pre>						
<b>Related topics</b>	<p>References</p> <table> <tr> <td><a href="#">Base Address of the I/O Board</a>.....</td> <td>7</td> </tr> <tr> <td><a href="#">ds5101_init</a>.....</td> <td>9</td> </tr> <tr> <td><a href="#">ds5101_start</a>.....</td> <td>50</td> </tr> </table>	<a href="#">Base Address of the I/O Board</a> .....	7	<a href="#">ds5101_init</a> .....	9	<a href="#">ds5101_start</a> .....	50
<a href="#">Base Address of the I/O Board</a> .....	7						
<a href="#">ds5101_init</a> .....	9						
<a href="#">ds5101_start</a> .....	50						

## ds5101\_state

<b>Syntax</b>	<pre>UInt8 ds5101_state (     phs_addr_t base,     int channel)</pre>
<b>Include file</b>	ds5101.h
<b>Purpose</b>	To read the current output state.
<b>Description</b>	This function reads the current output state of an individual channel.

<b>I/O mapping</b>	For details on the I/O mapping, refer to <a href="#">Mapping of I/O Signals</a> (PHS Bus System Hardware Reference  )
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>channel</b> Specifies the channel number within the range 1 ... 16.</p>
<b>Return value</b>	This function returns the current state of the specified channel (0, 1).
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.
<b>Related topics</b>	<p>References</p> <div> <a href="#">Base Address of the I/O Board</a>.....7  <a href="#">ds5101_init</a>.....9 </div>

## ds5101\_trigger

<b>Syntax</b>	<pre>void ds5101_trigger (     phs_addr_t base,     long mask)</pre>
<b>Include file</b>	ds5101.h
<b>Purpose</b>	To send a trigger event to the specified channels.
<b>Description</b>	A trigger event is sent to one or more channels as specified by the <b>mask</b> parameter. To trigger the channel, set the corresponding bit value to 1.
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>mask</b> Specifies the channels to be triggered using a 16-bit right-aligned bitmask (0x0000 ... 0xFFFF). To obtain the mask for an individual channel, you can use the DS5101_TRG_MASK macro.</p>

<b>Return value</b>	None
---------------------	------

**Example**

```
// trigger channel 1 and 8
ds5101_trigger(DS5101_1_BASE, DS5101_TRG_MASK(1) |
               DS5101_TRG_MASK(8) );
```

**Execution times**

For information, refer to [Function Execution Times](#) on page 63.

**Related topics****References**

Base Address of the I/O Board.....	7
ds5101_init.....	9
DS5101_TRG_MASK.....	55

## DS5101\_FLAG\_MASK

**Syntax**

```
long DS5101_FLAG_MASK (flag)
```

**Include file**

ds5101.h

**Purpose**

To obtain a flag mask for the control flag accessing functions.

**Description**

You can use this macro to obtain the flag mask for an individual control flag. It can be used with the `ds5101_update_flags`, `ds5101_set_flags` and `ds5101_clr_flags` functions.

**Parameters**

**flag** Specifies the control flag number within the range 1 ... 7 (boards up to revision DS5101-02) or 1 ... 31 (boards as of revision DS5101-04).

**Return value**

The 32-bit control flag mask

Related topics

References

<a href="#">ds5101_clr_flags.....</a>	<a href="#">46</a>
<a href="#">ds5101_set_flags.....</a>	<a href="#">48</a>
<a href="#">ds5101_update_flags.....</a>	<a href="#">49</a>

DS5101\_TRG\_MASK

Syntax

```
long DS5101_TRG_MASK (ch)
```

Include file

ds5101.h

Purpose

To obtain a trigger channel mask.

Description

You can use this macro to obtain the trigger mask for an individual channel.

Parameters

**ch** Specifies the channel number within the range 1 ... 16

Return value

The trigger channel mask

Related topics

References

<a href="#">ds5101_trigger.....</a>	<a href="#">53</a>
-------------------------------------	--------------------





# Interrupts

## Introduction

You can use the following functions for programming interrupts for one or several channels of a DS5101.

### Note

You have to initialize the DS5101 with `ds5101_init` function before you can use one of these functions.

## Where to go from here

## Information in this section

<a href="#">ds5101_int_ack.....</a>	<a href="#">57</a>
To acknowledge the interrupt of the specified channel.	
<a href="#">ds5101_int_clear.....</a>	<a href="#">58</a>
To acknowledge the interrupt of the masked channels.	
<a href="#">ds5101_int_enable.....</a>	<a href="#">59</a>
To enable interrupt generation on the specified channel.	
<a href="#">ds5101_int_read.....</a>	<a href="#">60</a>
To enable interrupt generation on the specified channel.	

## ds5101\_int\_ack

### Syntax

```
void ds5101_int_ack (
    phs_addr_t base,
    long channel)
```

<b>Include file</b>	<code>ds5101.h</code>										
<b>Purpose</b>	To acknowledge the interrupt of the specified channel.										
<b>Description</b>	The interrupt of the specified channel is acknowledged.										
<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>channel</b> Specifies the logical channel number within the range 1 ... 16. To get the number of the channel which has raised the interrupt, you can use the <code>ds5101_int_read</code> function.</p>										
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.										
<b>Related topics</b>	<p><b>References</b></p> <table> <tr> <td><a href="#">Base Address of the I/O Board</a>.....</td><td>7</td></tr> <tr> <td><a href="#">ds5101_init</a>.....</td><td>9</td></tr> <tr> <td><a href="#">ds5101_int_clear</a>.....</td><td>58</td></tr> <tr> <td><a href="#">ds5101_int_enable</a>.....</td><td>59</td></tr> <tr> <td><a href="#">ds5101_int_read</a>.....</td><td>60</td></tr> </table>	<a href="#">Base Address of the I/O Board</a> .....	7	<a href="#">ds5101_init</a> .....	9	<a href="#">ds5101_int_clear</a> .....	58	<a href="#">ds5101_int_enable</a> .....	59	<a href="#">ds5101_int_read</a> .....	60
<a href="#">Base Address of the I/O Board</a> .....	7										
<a href="#">ds5101_init</a> .....	9										
<a href="#">ds5101_int_clear</a> .....	58										
<a href="#">ds5101_int_enable</a> .....	59										
<a href="#">ds5101_int_read</a> .....	60										

## ds5101\_int\_clear

<b>Syntax</b>	<pre>void ds5101_int_clear (     phs_addr_t base,     long mask)</pre>
<b>Include file</b>	<code>ds5101.h</code>
<b>Purpose</b>	To acknowledge the interrupt of the masked channels.
<b>Description</b>	The interrupts of the masked channels are acknowledged.

<b>Parameters</b>	<p><b>base</b> Specifies the PHS-bus base address. Refer to <a href="#">Base Address of the I/O Board</a> on page 7.</p> <p><b>mask</b> Specifies the channel mask (0x0001 ... 0xFFFF).</p>												
<b>Execution times</b>	For information, refer to <a href="#">Function Execution Times</a> on page 63.												
<b>Related topics</b>	<p>References</p> <table> <tr> <td><a href="#">Base Address of the I/O Board</a>.....</td><td>7</td></tr> <tr> <td><a href="#">ds5101_init</a>.....</td><td>9</td></tr> <tr> <td><a href="#">ds5101_int_ack</a>.....</td><td>57</td></tr> <tr> <td><a href="#">ds5101_int_enable</a>.....</td><td>59</td></tr> <tr> <td><a href="#">ds5101_int_read</a>.....</td><td>60</td></tr> <tr> <td><a href="#">Macros</a>.....</td><td>7</td></tr> </table>	<a href="#">Base Address of the I/O Board</a> .....	7	<a href="#">ds5101_init</a> .....	9	<a href="#">ds5101_int_ack</a> .....	57	<a href="#">ds5101_int_enable</a> .....	59	<a href="#">ds5101_int_read</a> .....	60	<a href="#">Macros</a> .....	7
<a href="#">Base Address of the I/O Board</a> .....	7												
<a href="#">ds5101_init</a> .....	9												
<a href="#">ds5101_int_ack</a> .....	57												
<a href="#">ds5101_int_enable</a> .....	59												
<a href="#">ds5101_int_read</a> .....	60												
<a href="#">Macros</a> .....	7												

## ds5101\_int\_enable

<b>Syntax</b>	<pre>void ds5101_int_enable (     phs_addr_t base,     long channel,     long mode)</pre>
<b>Include file</b>	ds5101.h
<b>Purpose</b>	To enable interrupt generation on the specified channel.
<b>Description</b>	<p>The interrupt generation of the specified channel is enabled.</p> <p>To read a PHS interrupt in the real-time program, you must acknowledge it using the <code>ds5101_int_ack</code> function.</p>

**Parameters**

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

**channel** Specifies the channel number within the range 1 ... 16.

**mode** Specifies the mode (enable or disable). The following symbols are predefined:

Symbol	Meaning
DS5101_ENABLE	Enables the interrupt generation
DS5101_DISABLE	Disables the interrupt generation

**Execution times**

For information, refer to [Function Execution Times](#) on page 63.

**Related topics****References**

<a href="#">Base Address of the I/O Board</a> .....	7
<a href="#">ds5101_init</a> .....	9
<a href="#">ds5101_int_ack</a> .....	57
<a href="#">ds5101_int_clear</a> .....	58
<a href="#">ds5101_int_read</a> .....	60

## ds5101\_int\_read

**Syntax**

```
long ds5101_int_read (phs_addr_t base)
```

**Include file**

ds5101.h

**Purpose**

To read the interrupt requests.

**Description**

The interrupt requests from the modules are read from the IR register.

**Parameters**

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

**Return value** Bitmask with interrupt request from the modules (contents of IR register). The range is within 0x0000 ... 0xFFFF. The first bit is set if channel 1 is requested, the second bit is set if channel 2 is requested, and so on.

**Execution times** For information, refer to [Function Execution Times](#) on page 63.

**Related topics**

**References**

Base Address of the I/O Board.....	7
ds5101_init.....	9
ds5101_int_ack.....	57
ds5101_int_clear.....	58
ds5101_int_enable.....	59
Macros.....	7



# 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.....63</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.....64</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</li><li>▪ Use of inlining parameters</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>
------------------	---

`down<xxxx>` 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](#)..... 64

# Measured Execution Times

## Execution times

Execution times are available for the following RTLib units:

- [Initialization](#) on page 64
- [Timing I/O unit](#) on page 65
- [Interrupt functions](#) on page 66

### 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
<code>ds5101_init</code>	43.26 $\mu$ s	45.29 $\mu$ s



**Timing I/O unit**

The following execution times have been measured for the timing I/O functions used with DWO applications.

Function	Mean Execution Time	
	DS1006 with 2.6 GHz	DS1006 with 3.0 GHz
1-phase PWM signal generation		
ds5101_pwm_load	0.85 ms (1 channel) 1.06 ms (16 channels)	849.56 $\mu$ s (0 channels) 862.63 $\mu$ s (16 channels)
ds5101_pwm_update	2.20 $\mu$ s	2.21 $\mu$ s
ds5101_pwm_parm	1.47 $\mu$ s	1.65 $\mu$ s
3-phase PWM signal generation		
ds5101_pwm3_intshift_load	0.85 ms	849.56 $\mu$ s
ds5101_pwm3_intshift_update	5.67 $\mu$ s	5.64 $\mu$ s
ds5101_pwm3_intshift_parm	4.66 $\mu$ s	4.69 $\mu$ s
3-phase/6-channel PWM signal generation		
ds5101_pwm6_intshift_load	0.75 ms	743.50 $\mu$ s
ds5101_pwm6_intshift_update	4.49 $\mu$ s	4.36 $\mu$ s
ds5101_pwm6_intshift_parm	5.47 $\mu$ s	5.50 $\mu$ s
Monoflop signal generation		
ds5101_mono_load	0.85 ms	849.54 $\mu$ s
ds5101_mono_update	1.02 $\mu$ s	0.99 $\mu$ s
ds5101_mono_parm	0.85 $\mu$ s	0.95 $\mu$ s
ds5101_mono_start	0.14 $\mu$ s	0.029 $\mu$ s
ds5101_delayed_mono_load	0.88 ms	881.53 $\mu$ s
ds5101_delayed_mono_update	1.21 $\mu$ s	1.18 $\mu$ s
ds5101_delayed_mono_parm	1.43 $\mu$ s	1.51 $\mu$ s
Incremental encoder simulation		
ds5101_inc_load	0.78 ms	777.08 $\mu$ s
ds5101_inc_update	1.07 $\mu$ s	1.22 $\mu$ s
ds5101_inc_parm	1.69 $\mu$ s	1.74 $\mu$ s
Arbitrary signal generation		
ds5101_set_clock	1.50 $\mu$ s	1.66 $\mu$ s
ds5101_set_pullup	1.55 $\mu$ s	1.52 $\mu$ s
ds5101_set_flags	0.71 $\mu$ s	0.60 $\mu$ s
ds5101_update_flags	0.12 $\mu$ s	0.029 $\mu$ s
ds5101_clr_flags	0.62 $\mu$ s	0.60 $\mu$ s
ds5101_flag	0.62 $\mu$ s	0.59 $\mu$ s
ds5101_state	0.62 $\mu$ s	0.59 $\mu$ s
ds5101_trigger	0.04 $\mu$ s	0.029 $\mu$ s
ds5101_start	0.62 $\mu$ s	0.60 $\mu$ s
ds5101_stop	0.62 $\mu$ s	0.63 $\mu$ s

**Interrupt functions**

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

Function	Mean Execution Time	
	DS1006 with 2.6 GHz	DS1006 with 3.0 GHz
ds5101_int_enable	0.64 $\mu$ s	0.61 $\mu$ s
ds5101_int_ack	0.04 $\mu$ s	0.029 $\mu$ s
ds5101_int_clear	0.04 $\mu$ s	0.029 $\mu$ s
ds5101_int_read	0.63 $\mu$ s	0.60 $\mu$ s

**Related topics****Basics**

[Information on the Test Environment.....](#) 63

**A**

AP5101\_INC\_MASK 34  
 AP5101\_MONO\_MASK 44  
 AP5101\_PWM\_MASK 17  
 AP5101\_PWM3\_MASK 23  
 AP5101\_PWM6\_MASK 30

**B**

base address 7

**C**

Common Program Data folder 6

**D**

demo models  
     DS5101 5  
 Documents folder 6  
 DS5101  
     function execution times 64  
 ds5101\_clr\_flags 46  
 ds5101\_delayed\_mono\_load 40  
 ds5101\_delayed\_mono\_parm 41  
 ds5101\_delayed\_mono\_update 43  
 ds5101\_flag 47  
 DS5101\_FLAG\_MASK 54  
 ds5101\_inc\_load 31  
 ds5101\_inc\_parm 32  
 ds5101\_inc\_update 33  
 ds5101\_init 9  
 ds5101\_int\_ack 57  
 ds5101\_int\_clear 58  
 ds5101\_int\_enable 59  
 ds5101\_int\_read 60  
 ds5101\_mono\_load 36  
 ds5101\_mono\_parm 37  
 ds5101\_mono\_start 38  
 ds5101\_mono\_update 39  
 ds5101\_pwm\_load 14  
 ds5101\_pwm\_parm 15  
 ds5101\_pwm\_update 16  
 ds5101\_pwm3\_intshift\_load 19  
 ds5101\_pwm3\_intshift\_parm 21  
 ds5101\_pwm3\_intshift\_update 22  
 ds5101\_pwm6\_intshift\_load 25  
 ds5101\_pwm6\_intshift\_parm 27  
 ds5101\_pwm6\_intshift\_update 28  
 ds5101\_set\_clock 10  
 ds5101\_set\_flags 48  
 ds5101\_set\_pullup 11  
 ds5101\_start 50  
 ds5101\_state 52  
 ds5101\_stop 51  
 DS5101\_TRG\_MASK 55  
 ds5101\_trigger 53  
 ds5101\_update\_flags 49  
 DSxxx\_n\_BASE 7  
 DWO commands 46

**E**

examples  
     DS5101 5

**F**

function execution times  
     DS5101 64

**L**

Local Program Data folder 6

