

DS1305 library for Atmel AVR  
0.1.0

Generated by Doxygen 1.8.2

Sun Sep 23 2012 15:13:33

## Contents

<b>1</b>	<b>Main Page</b>	<b>1</b>
1.1	Information . . . . .	1
1.2	Specifications . . . . .	2
1.2.1	Language . . . . .	2
1.2.2	Compiler . . . . .	2
1.2.3	Target . . . . .	2
1.3	License . . . . .	2
<b>2</b>	<b>Todo List</b>	<b>3</b>
<b>3</b>	<b>Bug List</b>	<b>3</b>
<b>4</b>	<b>Data Structure Documentation</b>	<b>3</b>
4.1	ds1305_ram Struct Reference . . . . .	3
4.1.1	Detailed Description . . . . .	4
4.2	ds1305_time Struct Reference . . . . .	4
4.2.1	Detailed Description . . . . .	4
<b>5</b>	<b>File Documentation</b>	<b>4</b>
5.1	ds1305.h File Reference . . . . .	4
5.1.1	Detailed Description . . . . .	6
5.1.2	Function Documentation . . . . .	7
	<b>Index</b>	<b>13</b>

## 1 Main Page

Maxim Serial Alarm Real-Time Clock DS1305 functions for Atmel AVR

### 1.1 Information

#### Author

Patrice Nadeau

email : [patricen@telwarwick.net](mailto:patricen@telwarwick.net)

www : <http://nadeaup.homeip.net:8080/>

**Note**

For the moment, only the the SPI protocol will be developed  
Minimum year is 2000, maximum year is 2099 but it will be able to roll over to 2100 (hardware limitation of the ds1305)

**Todo** Trickle charging functions

3-Wire mode

**Precondition**

Voltage required :

- $V_{cc1} > V_{bat} + 0.2V$

## 1.2 Specifications

### 1.2.1 Language

C (c99)

### 1.2.2 Compiler

avr-gcc

### 1.2.3 Target

uC tested are in **bold**

- ATmega48
- ATmega88
- **ATmega168**
- **ATmega328**

**Datasheet**

<http://datasheets.maxim-ic.com/en/ds/DS1305.pdf>

## 1.3 License

BSD 3-Clause License

Copyright (c) 2010, 2011,2012 Patrice Nadeau

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Patrice Nadeau nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL Patrice Nadeau BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 2 Todo List

### File [ds1305.h](#)

use same option for ReadTimeAscii and WriteTimeAscii should probably have

- WriteDateAscii
- WriteTimeAscii
- WriteAlarmAscii

### Global [ds1305\\_ValidateAscii](#) (char \*string, uint8\_t format)

Add a check for alarms

### Global [ds1305\\_WriteAlarm](#) (struct [ds1305\\_time](#) time, uint8\_t alarm, uint8\_t interval)

Documentation could be a lot more clear

### page [Main Page](#)

Trickle charging functions

3-Wire mode

## 3 Bug List

### Global [ds1305\\_WriteDateAscii](#) (const char \*string)

(internal) time.year must be initialized once before being written

### Global [ds1305\\_WriteTime](#) (struct [ds1305\\_time](#) time)

see *\_WriteSecond*

### Global [ds1305\\_WriteTimeAscii](#) (const char \*string)

see *\_WriteSecond*

## 4 Data Structure Documentation

### 4.1 ds1305\_ram Struct Reference

Structure for the 96 bytes of RAM.

### Data Fields

- uint8\_t [byte](#) [96]  
*1 byte, from 0 to 95*

#### 4.1.1 Detailed Description

Structure for the 96 bytes of RAM.

## 4.2 ds1305\_time Struct Reference

Structure for time and alarms.

### Data Fields

- uint8\_t [year](#)  
*00 - 99*
- uint8\_t [month](#)  
*01 - 12*
- uint8\_t [date](#)  
*01 - 31*
- uint8\_t [day](#)  
*1 - 7*
- uint8\_t [hours](#)  
*00 - 23*
- uint8\_t [minutes](#)  
*00 - 59*
- uint8\_t [seconds](#)  
*00 - 59*

#### 4.2.1 Detailed Description

Structure for time and alarms.

### Note

Alarms don't use the following fields :

- **year**
- **month**
- **date**

## 5 File Documentation

### 5.1 ds1305.h File Reference

#### Data Structures

- struct [ds1305\\_time](#)

*Structure for time and alarms.*

- struct [ds1305\\_ram](#)

*Structure for the 96 bytes of RAM.*

## Macros

### To specify the functions

- #define [ds1305\\_ALARM0](#) 0
- #define [ds1305\\_ALARM1](#) 1
- #define [ds1305\\_TIMER](#) 2

### To specify the alarms intervals

- #define [ds1305\\_ALARM\\_ONCE](#) 15  
*Every seconds.*
- #define [ds1305\\_ALARM\\_SECOND](#) 7  
*When seconds match.*
- #define [ds1305\\_ALARM\\_MINUTE](#) 3  
*When minutes & seconds match.*
- #define [ds1305\\_ALARM\\_HOUR](#) 1  
*When hours, minutes & seconds match.*
- #define [ds1305\\_ALARM\\_DAY](#) 0  
*When days, hours, minutes & seconds match.*

### ASCII format and size in ISO 8601 extended

- #define [ds1305\\_ASCII\\_FMT\\_FULL\\_EXT](#) 19  
*YYYY-MM-DD HH:MM:SS.*
- #define [ds1305\\_ASCII\\_FMT\\_TIME\\_EXT](#) 8  
*Format HH:MM:SS.*
- #define [ds1305\\_ASCII\\_FMT\\_DATE\\_EXT](#) 10  
*Format YYYY-MM-DD.*
- #define [ds1305\\_ASCII\\_FMT\\_ALARM](#) 4  
*Format : Day, HH:MM:SS AM/PM.*

### ASCII errors

- #define [ds1305\\_ASCII\\_ERR\\_EMPTY](#) 10  
*Error code string is empty.*
- #define [ds1305\\_ASCII\\_ERR\\_LONG](#) 11  
*Error code string is too long.*
- #define [ds1305\\_ASCII\\_ERR\\_SHORT](#) 12  
*Error code string is too short.*
- #define [ds1305\\_ASCII\\_ERR\\_INVALID](#) 13  
*Error code string is invalid.*
- #define [ds1305\\_ASCII\\_ERR\\_DATE\\_FORMAT](#) 14  
*Error code date is wrong format.*
- #define [ds1305\\_ASCII\\_ERR\\_TIME\\_FORMAT](#) 15  
*Error code time is wrong format.*
- #define [ds1305\\_ASCII\\_ERR\\_DATE\\_RANGE](#) 16  
*Error code date is out of range.*
- #define [ds1305\\_ASCII\\_ERR\\_TIME\\_RANGE](#) 17  
*Error code time is out of range.*
- #define [ds1305\\_ASCII\\_UNDEFINED](#) 255  
*Error code unknown.*

## Functions

- void `ds1305_SPIInit` (void)  
*Initialize the DS1305 for the SPI protocol.*
- void `ds1305_ReadDate` (struct `ds1305_time` \*date)  
*Return the date in a struct.*
- void `ds1305_ReadTime` (struct `ds1305_time` \*time)  
*Return the time in a struct.*
- void `ds1305_WriteDate` (struct `ds1305_time` date)  
*Set the date.*
- void `ds1305_WriteTime` (struct `ds1305_time` time)  
*Set the time.*
- void `ds1305_WriteAlarm` (struct `ds1305_time` time, uint8\_t alarm, uint8\_t interval)  
*Write the time and interval into one of the alarms.*
- uint8\_t `ds1305_ReadAlarm` (struct `ds1305_time` \*time, uint8\_t alarm)  
*Read the time for alarm # alarm.*
- void `ds1305_WriteRAM` (struct `ds1305_ram` ram)  
*Write all the 96 Bytes to RAM.*
- void `ds1305_ReadRAM` (struct `ds1305_ram` \*ram)  
*Read all the 96 Bytes from RAM.*
- void `ds1305_TimeToStr` (struct `ds1305_time` time, char \*string, uint8\_t format)  
*Convert a structure of type ds1305\_time to a string.*
- void `ds1305_ReadDateAscii` (char \*string)  
*Read the current date in format ds1305\_ASCII\_FMT\_DATE\_EXT.*
- void `ds1305_ReadTimeAscii` (char \*string)  
*Read the current time in format ds1305\_ASCII\_FMT\_TIME\_EXT.*
- void `ds1305_ReadAlarmAscii` (char \*string)  
*Read the current alarm in format ds1305\_ASCII\_FMT\_ALARM.*
- uint8\_t `ds1305_ValidateAscii` (char \*string, uint8\_t format)  
*Validate a string.*
- void `ds1305_WriteDateAscii` (const char \*string)  
*Write an ASCII string representing date into the RTC.*
- void `ds1305_WriteTimeAscii` (const char \*string)  
*Write an ASCII string representing time into the RTC.*
- void `ds1305_EnableInt` (uint8\_t one\_pin, uint8\_t int0, uint8\_t int1)  
*Set the use of one pin (INT0) for INT0 and INT1, set INT0 and INT1.*
- void `ds1305_ClearInt` (uint8\_t nb)  
*Clear the interrupt flag for an alarm.*

### 5.1.1 Detailed Description

#### Author

Patrice Nadeau [patricen@telwarwick.net](mailto:patricen@telwarwick.net)

**Todo** use same option for ReadTimeAscii and WriteTimeAscii should probably have

- WriteDateAscii
- WriteTimeAscii
- WriteAlarmAscii

## 5.1.2 Function Documentation

## 5.1.2.1 void ds1305\_ClearInt ( uint8\_t nb )

Clear the interrupt flag for an alarm.

## Parameters

in	nb	Alarm number to clear <ul style="list-style-type: none"> <li>• ds1305_ALARM0</li> <li>• ds1305_ALARM1</li> </ul>
----	----	--

## 5.1.2.2 void ds1305.EnableInt ( uint8\_t one\_pin, uint8\_t int0, uint8\_t int1 )

Set the use of one pin (INT0) for INT0 and INT1, set INT0 and INT1.

## Parameters

in	one_pin	Use of one pin for both int <ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>
in	int0	Enable int0 <ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>
in	int1	Enable int1 <ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>

## Warning

Don't forget the external pull-up (resistor or AVR internal pull-up) !

## 5.1.2.3 uint8\_t ds1305\_ReadAlarm ( struct ds1305\_time \* time, uint8\_t alarm )

Read the *time* for alarm # *alarm*.

## Parameters

in	alarm	Alarm number to read <ul style="list-style-type: none"> <li>• ds1305_ALARM0</li> <li>• ds1305_ALARM1</li> </ul>
out	*time	A pointer to a structure of type <a href="#">ds1305_time</a>



**Returns**

Interval of the alarm

**Return values**

<code>ds1305_ALARM_ONCE</code>	One time
<code>ds1305_ALARM_SECOND</code>	Every seconds
<code>ds1305_ALARM_MINUTE</code>	Every minutes
<code>ds1305_ALARM_HOUR</code>	Every hours
<code>ds1305_ALARM_DAY</code>	Every days

**5.1.2.4 void ds1305.ReadAlarmAscii ( char \* string )**

Read the current alarm in format `ds1305_ASCII_FMT_ALARM`.

**Parameters**

out	<i>string</i>	A pointer to the output string
-----	---------------	--------------------------------

**5.1.2.5 void ds1305.ReadDate ( struct ds1305\_time \* date )**

Return the date in a struct.

**Parameters**

out	<i>*date</i>	A pointer to a structure of type <a href="#">ds1305_time</a>
-----	--------------	--

**5.1.2.6 void ds1305.ReadDateAscii ( char \* string )**

Read the current date in format `ds1305_ASCII_FMT_DATE_EXT`.

**Parameters**

out	<i>string</i>	A pointer to the output string
-----	---------------	--------------------------------

**5.1.2.7 void ds1305.ReadRAM ( struct ds1305\_ram \* ram )**

Read all the 96 Bytes from RAM.

**Parameters**

out	<i>*ram</i>	A pointer to a structure of type <a href="#">ds1305_ram</a>
-----	-------------	---

**5.1.2.8 void ds1305.ReadTime ( struct ds1305\_time \* time )**

Return the time in a struct.

**Parameters**

out	<i>*time</i>	A pointer to a structure of type <a href="#">ds1305_time</a>
-----	--------------	--

5.1.2.9 void ds1305\_ReadTimeAscii ( char \* *string* )

Read the current time in format ds1305\_ASCII\_FMT\_TIME\_EXT.

## Parameters

out	<i>string</i>	A pointer to the output string
-----	---------------	--------------------------------

## 5.1.2.10 void ds1305\_SPIInit ( void )

Initialize the DS1305 for the SPI protocol.

## Postcondition

Write protect is enable  
OSC is enable

## Note

SPI : MSB, Mode3, clock/16, interrupt disable  
DS1305 max clock is : 0.6MHz @ Vcc=2V, 2MHz @ Vcc=5V,  
Using spi\_CLOCK\_16 should permit an AVR speed up to 20MHz

## Warning

The SERMODE pin must be connected to Vcc to enable SPI mode

5.1.2.11 void ds1305\_TimeToStr ( struct ds1305\_time *time*, char \* *string*, uint8\_t *format* )

Convert a structure of type [ds1305\\_time](#) to a string.

## Parameters

in	<i>time</i>	Structure of type <a href="#">ds1305_time</a>
out	<i>string</i>	Pointer to a string
in	<i>format</i>	One of the ASCII format <ul style="list-style-type: none"> <li>• ds1305_ASCII_FMT_TIME</li> <li>• ds1305_ASCII_FMT_DATE</li> <li>• ds1305_ASCII_FMT_ALARM</li> </ul>

5.1.2.12 uint8\_t ds1305\_ValidateAscii ( char \* *string*, uint8\_t *format* )

Validate a string.

## Parameters

in	<i>string</i>	A pointer to a string
in	<i>format</i>	One of the following <ul style="list-style-type: none"> <li>• ds1305_ASCII_FMT_FULL_EXT</li> <li>• ds1305_ASCII_FMT_TIME_EXT</li> <li>• ds1305_ASCII_FMT_DATE_EXT</li> <li>• ds1305_ASCII_FMT_ALARM</li> </ul>

## Returns

Error code

## Return values

0	If valid
<i>ds1305_ASCII_ERR_XX</i>	Any other error code

## Example :

```

...
switch (ds1305_ValidateAscii(str, ds1305_ASCII_FMT_TIME_EXT
)) {
case ds1305_ASCII_ERR_LONG :
    printf("String too long\n");
    break;
case ds1305_ASCII_ERR_SHORT :
    printf("String too short\n");
    break;
};
...

```

## Warning

Need more checking before being ready

**Todo** Add a check for alarms

5.1.2.13 void ds1305.WriteAlarm ( struct ds1305\_time time, uint8\_t alarm, uint8\_t interval )

Write the time and interval into one of the alarms.

## Parameters

in	<i>time</i>	Structure of type <a href="#">ds1305_time</a>
in	<i>alarm</i>	Alarm number to write <ul style="list-style-type: none"> <li>• ds1305_ALARM0</li> <li>• ds1305_ALARM1</li> </ul>

<i>in</i>	<i>interval</i>	Interval of the alarm <ul style="list-style-type: none"> <li>• ds1305_ALARM_ONCE Once every seconds</li> <li>• ds1305_ALARM_SECOND When seconds matches</li> <li>• ds1305_ALARM_MINUTE When seconds and minutes matches</li> <li>• ds1305_ALARM_HOUR When seconds, minutes and hours matches</li> <li>• ds1305_ALARM_DAY When seconds, minutes, hours and days matches</li> </ul>
-----------	-----------------	---

**Example**

Set Alarm0 every 10 seconds

```
struct ds1305_time time;
time.seconds = 10;
ds1305_WriteAlarm(time, ds1305_ALARM0, ds1305_ALARM_SECOND);
```

**Todo** Documentation could be a lot more clear

#### 5.1.2.14 void ds1305\_WriteDate ( struct ds1305\_time date )

Set the date.

**Parameters**

<i>in</i>	<i>date</i>	A structure of type <a href="#">ds1305_time</a>
-----------	-------------	---

#### 5.1.2.15 void ds1305\_WriteDateAscii ( const char \* string )

Write an ASCII string representing date into the RTC.

**Parameters**

<i>in</i>	<i>string</i>	A pointer to a string
-----------	---------------	-----------------------

**Returns**

Status of the operation

**Return values**

0	Successful
---	------------

**Note**

Format : YYYY-MM-DD

**Precondition**

The string must be validated before with *ds1305\_ValidateAscii*

**Example :**

```

...
if (ds1305_ValidateAscii(str, ds1305_ASCII_FMT_DATE_EXT
) == 0)
    ds1305_WriteDateAscii(str);
else
    error();
...

```

**Bug** (internal) time.year must be initialized once before being written

#### 5.1.2.16 void ds1305.WriteRAM ( struct ds1305\_ram ram )

Write all the 96 Bytes to RAM.

##### Parameters

in	ram	Structure of type <a href="#">ds1305_ram</a>
----	-----	--

#### 5.1.2.17 void ds1305.WriteTime ( struct ds1305\_time time )

Set the time.

##### Parameters

in	time	A structure of type <a href="#">ds1305_time</a>
----	------	---

**Bug** see `_WriteSecond`

#### 5.1.2.18 void ds1305.WriteTimeAscii ( const char \* string )

Write an ASCII string representing time into the RTC.

##### Parameters

in	string	A pointer to a string
----	--------	-----------------------

##### Returns

Status of the operation

##### Return values

0	Successful
---	------------

##### Note

Format : HH:MM:SS

##### Precondition

The string must be validated before with `ds1305_ValidateAscii`

##### Example :

```
...
if (ds1305_ValidateAscii(str) == 0)           // the string is OK
    ds1305_WriteTimeAscii(str);
else
    error();
...
```

**Bug** see *\_WriteSecond*

## Index

ds1305.h, [4](#)

- ds1305\_ClearInt, [6](#)
- ds1305\_EnableInt, [6](#)
- ds1305\_ReadAlarm, [7](#)
- ds1305\_ReadAlarmAscii, [7](#)
- ds1305\_ReadDate, [7](#)
- ds1305\_ReadDateAscii, [8](#)
- ds1305\_ReadRAM, [8](#)
- ds1305\_ReadTime, [8](#)
- ds1305\_ReadTimeAscii, [8](#)
- ds1305\_SPIInit, [8](#)
- ds1305\_TimeToStr, [8](#)
- ds1305\_ValidateAscii, [9](#)
- ds1305\_WriteAlarm, [10](#)
- ds1305\_WriteDate, [10](#)
- ds1305\_WriteDateAscii, [10](#)
- ds1305\_WriteRAM, [11](#)
- ds1305\_WriteTime, [11](#)
- ds1305\_WriteTimeAscii, [11](#)

ds1305\_ClearInt

- ds1305.h, [6](#)

ds1305\_EnableInt

- ds1305.h, [6](#)

ds1305\_ReadAlarm

- ds1305.h, [7](#)

ds1305\_ReadAlarmAscii

- ds1305.h, [7](#)

ds1305\_ReadDate

- ds1305.h, [7](#)

ds1305\_ReadDateAscii

- ds1305.h, [8](#)

ds1305\_ReadRAM

- ds1305.h, [8](#)

ds1305\_ReadTime

- ds1305.h, [8](#)

ds1305\_ReadTimeAscii

- ds1305.h, [8](#)

ds1305\_SPIInit

- ds1305.h, [8](#)

ds1305\_TimeToStr

- ds1305.h, [8](#)

ds1305\_ValidateAscii

- ds1305.h, [9](#)

ds1305\_WriteAlarm

- ds1305.h, [10](#)

ds1305\_WriteDate

- ds1305.h, [10](#)

ds1305\_WriteDateAscii

- ds1305.h, [10](#)

ds1305\_WriteRAM

- ds1305.h, [11](#)

ds1305\_WriteTime

- ds1305.h, [11](#)

ds1305\_WriteTimeAscii

- ds1305.h, [11](#)

ds1305\_ram, [3](#)

ds1305\_time, [3](#)