



### I2C timing configuration tool for STM32F3xxxx and STM32F0xxxx microcontrollers

## Introduction

This application note presents the I2C timing configuration tool for the STM32F3xxxx and STM32F0xxxx microcontroller families.

The STM32F0xxxx and STM32F3xxxx devices embed a high-speed Inter Circuit Communication peripheral (I2C) supporting standard mode (100 KHz), fast mode (400 KHz) and fast mode plus (1 MHz). The I2C implements a new clock scheme allowing the peripheral to be used as a wake-up source from low-power mode on address match.

The purpose of this tool is to help the user configure the I2C timings, taking into consideration the I2C bus specification.

The configuration tool is implemented in the Microsoft Excel “*I2C\_Timing\_Config\_Tool\_Vx.y.z.xls*” file which can be downloaded from [www.st.com](http://www.st.com).

For Vx.y.z, please refer to the tool version, for example: V1.0.0.

Before using the clock tool, it is essential to read the STM32 microcontroller reference manuals (RM0313 for STM32F37xxx products, RM0316 for STM32F30xxx products and RM0091 for STM32F0xxxx products). This application note is not a substitute for the reference manuals.

This document applies to the products listed in the following table. They will be referred to as “STM32” throughout the document.

**Table 1. Applicable products**

Type	Part numbers
Microcontrollers	STM32F051xx STM32F050xx STM32F302xx STM32F303xx STM32F313xx STM32F372xx STM32F373xx STM32F383xx

# Contents

1      **Glossary** ..... **5**

2      **Getting started** ..... **6**

      2.1    Software requirements ..... 6

      2.2    Hardware requirements ..... 7

          2.2.1    Introduction ..... 7

          2.2.2    I2C timing specification ..... 7

          2.2.3    I2C clock scheme ..... 9

          2.2.4    I2C timing register ..... 10

3      **Tutorials** ..... **13**

4      **Conclusion** ..... **16**

5      **Revision history** ..... **17**



List of tables

Table 1.   Applicable products ..... 1

Table 2.   Definition of terms ..... 5

Table 3.   I2C timings specification (see I2C specification, rev.03, June 2007) ..... 8

Table 4.   Document revision history ..... 17

## List of figures

Figure 1.	I2C bus timing (see I2C specification, rev.03, June 2007) .....	7
Figure 2.	I2C clock scheme .....	9
Figure 3.	Timing register .....	10
Figure 4.	Data setup time generation from SCLDEL .....	10
Figure 5.	Data hold time generation from SDADEL .....	11
Figure 6.	High and low period generation from SCLH and SCLL .....	11
Figure 7.	I2C timing configuration tool user interface .....	13
Figure 8.	Calculation is completed .....	14
Figure 9.	Error message .....	15

# 1 Glossary

**Table 2. Definition of terms**

Term	Description
AF	Analog filter
DNF	Digital noise filter
HSI	High-speed internal clock
I2C	Inter circuit communication
I2CCLK	I2C kernel clock
PCLK	APBx clock
PRESC	Prescaler
SCL	Serial clock line
SDA	Serial data line
SYSCLK	System clock

## 2 Getting started

This section describes the requirements and procedures needed to start using the timing configuration tool.

### 2.1 Software requirements

To use the timing configuration tool with Windows operating system, a recent version of Windows, such as Windows XP, Vista or Windows 7, must be installed on the PC with at least 256 Mbytes of RAM.

Before starting to use the timing configuration tool, make sure that Microsoft Office is installed on your machine and then follow these steps:

- Download the latest version of the **I2C timing configuration tool** for the STM32 devices from [www.st.com](http://www.st.com).
- Enable macros and ActiveX controls as shown below:

#### Excel 1997-2003 version

1. Click **Tools** in the menu bar.
2. Click **Macro**.
3. Click **Security**.
4. Click **Low (not recommended)**.

*Note:* If ActiveX controls are not enabled, a warning message is displayed asking you to enable ActiveX. In this case, you should click "OK" to enable it.

#### Excel 2007-2010 version

1. Click the **Microsoft Office** button and then click **Excel options**.
2. Click **Trust Center**, click **Trust center settings**, and then click **Macro settings**.
3. Click **Enable all macros (not recommended, potentially dangerous code can run)**.
4. Click **Trust Center**, click **Trust center settings**, and then click **ActiveX settings**.
5. Click **Enable all controls without restrictions and without prompting (not recommended; potentially dangerous controls can run)**.
6. Click **OK**.

*Note:* For more information about how to enable macros and ActiveX controls, refer to the Microsoft Office website.

## 2.2 Hardware requirements

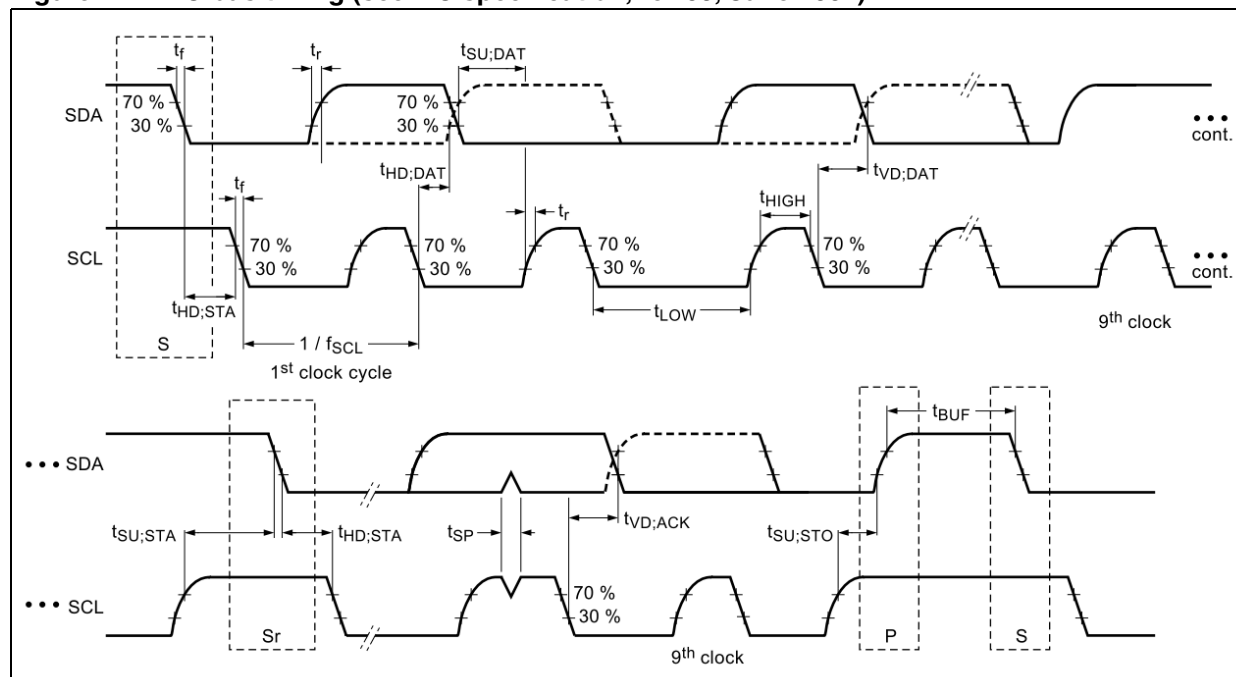
### 2.2.1 Introduction

The I2C timing configuration tool is designed to help the end-user easily configure the timing settings for the I2C peripheral and guarantee its operation as specified in the I2C timing specification.

### 2.2.2 I2C timing specification

The I2C timings should be configured with values that are compliant with the I2C bus specification:

**Figure 1. I2C bus timing (see I2C specification, rev.03, June 2007)**



The table below shows the value range of these timings:

**Table 3. I2C timings specification (see I2C specification, rev.03, June 2007)**

Symbol	Parameter	Standard		Fast mode		Fast mode +		Unit
		Min	Max	Min	Max	Min	Max	
$f_{SCL}$	SCL clock frequency	0	100	0	400	0	1000	KHz
$t_{LOW}$	Low period of the SCL clock	4.7	-	1.3	-	0.5	-	$\mu s$
$t_{HIGH}$	High Period of the SCL clock	4		0.6		0.26	-	$\mu s$
$t_r$	Rise time of both SDA and SCL signals	-	1000	$20 + 0.1C_b^{(1)}$	300	-	120	ns
$t_f$	Fall time of both SDA and SCL signals	-	300	$20 + 0.1C_b^{(1)}$	300	-	120	ns
$t_{HD;DAT}$	Data hold time	0	-	0	-	0	-	$\mu s$
$t_{VD;DAT}$	Data valid time	-	$3.45^{(2)}$	-	$0.9^{(2)}$	-	$0.45^{(2)}$	$\mu s$
$t_{VD;ACK}$	Data valid acknowledge time	-	$3.45^{(2)}$	-	$0.9^{(2)}$	-	$0.45^{(2)}$	$\mu s$
$t_{SU;DAT}$	Data setup time	250	-	100	-	50	-	ns
$t_{HD;STA}$	Hold time (repeated) START condition	4.0	-	0.6	-	0.26	-	$\mu s$
$t_{SU;STA}$	Set-up time for a repeated START condition	4.7	-	0.6	-	0.26		$\mu s$
$t_{SU;STO}$	Set-up time for STOP condition	4.0	-	0.6	-	0.26	-	$\mu s$
$t_{BUF}$	Bus free time between a STOP and START condition	4.7	-	1.3	-	0.5	-	$\mu s$

1.  $C_b$  = total capacitance of one bus line in pF.

2. The maximum  $t_{HD;DAT}$  could be 3.45  $\mu s$ , 0.9  $\mu s$  and 0.45  $\mu s$  for standard mode, fast mode and fast mode plus, but must be less than the maximum of  $t_{VD;DAT}$  or  $t_{VD;ACK}$  by a transition time.

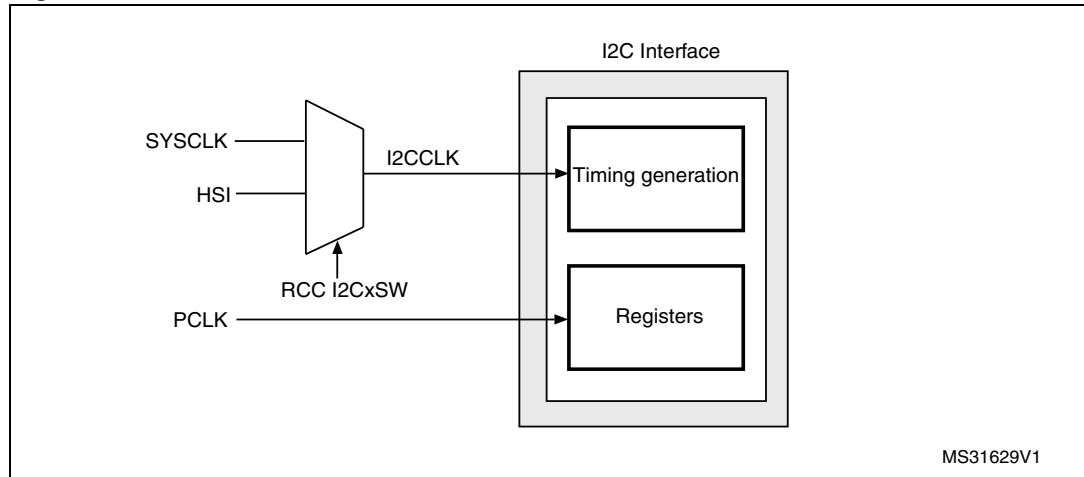


### 2.2.3 I2C clock scheme

The I2C kernel is clocked by an independent clock source. The clock source can be:

- HSI (default source)
- SYSCLK

**Figure 2. I2C clock scheme**



These two clocks allow I2C to operate independently from the PCLK frequency.

Setting HSI as I2C clock source frequency allows the use of wake-up from STOP mode capability at address match.

The I2CCLK period  $t_{I2CCLK}$  must respect the following conditions:

$$t_{I2CCLK} < (t_{LOW} - t_{filters}) / 4 \text{ and } t_{I2CCLK} < t_{HIGH}$$

$t_{filters}$ : when enabled, sum of the delays brought by the analog filter and the digital filter.

Analog filter delay is maximum 260 ns and digital filter delay is  $DNF \times t_{I2CCLK}$ .

The PCLK clock period  $t_{PCLK}$  must respect the following condition:

$$t_{PCLK} < 4/3 t_{SCL}$$

Please refer to the RCC section in STM32 product reference manual for more details about the selection of the I2C clock source.

## 2.2.4 I2C timing register

The I2C timing register is defined as the following figure shows:

**Figure 3. Timing register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL[3:0]				SDADEL[3:0]			
rw								rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
rw								rw							

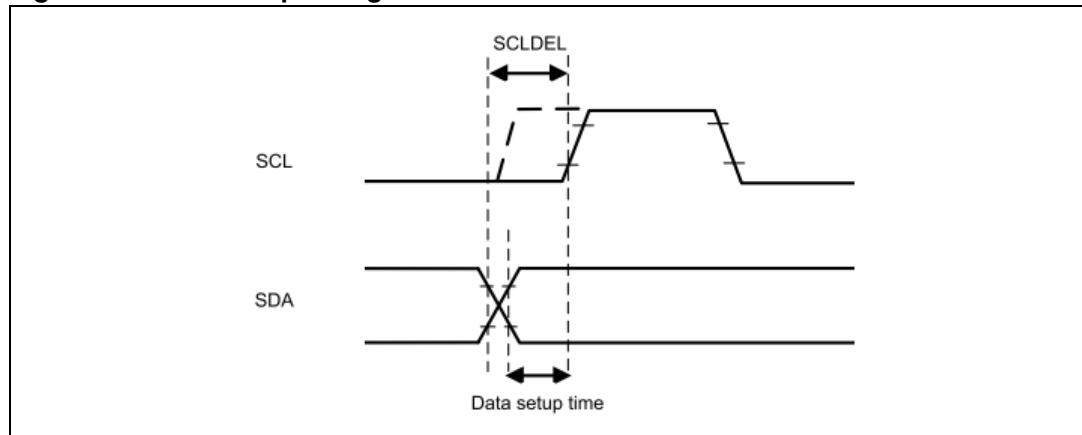
PRESC[3:0] is used to prescale I2C clock source (I2CCLK); it allows the generation of a divided clock. The period of this divided clock  $t_{\text{PRESC}}$  is defined by:

$$t_{\text{PRESC}} = (\text{PRESC} + 1) \times t_{\text{I2CCLK}}$$

The time unit  $t_{\text{PRESC}}$  is used for the generation of other I2C timings.

SCLDEL[3:0] is used to program the data setup time ( $t_{\text{SU;DAT}}$ ) as shown in the following figure:

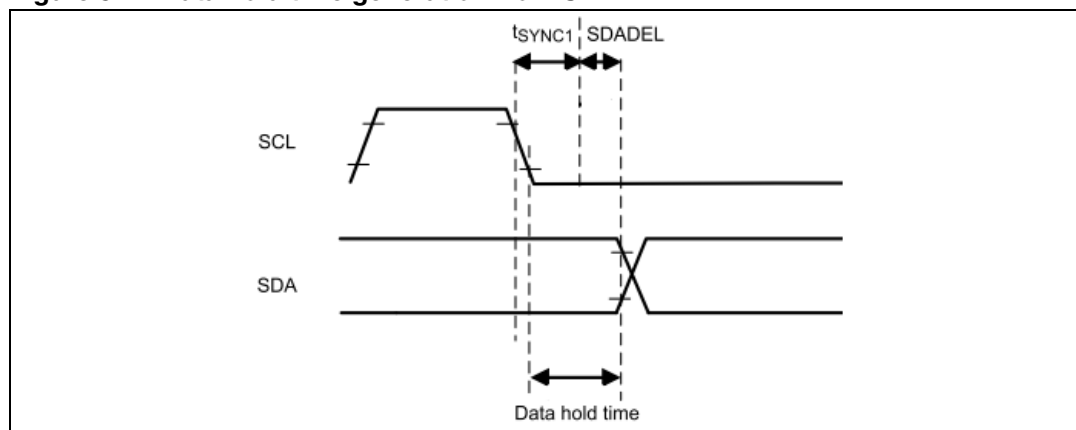
**Figure 4. Data setup time generation from SCLDEL**



SCLDEL is defined as follows:

$$\{[t_{\text{r(max)}} + t_{\text{SU;DAT(min)}}] / [t_{\text{PRESC}}]\} - 1 \leq \text{SCLDEL}$$

SDADEL[3:0] is used to program the data hold time ( $t_{\text{HD;DAT}}$ ) as shown in the following figure:

**Figure 5. Data hold time generation from SDADEL**

$t_{\text{SYNC1}}$  duration depends on these parameters:

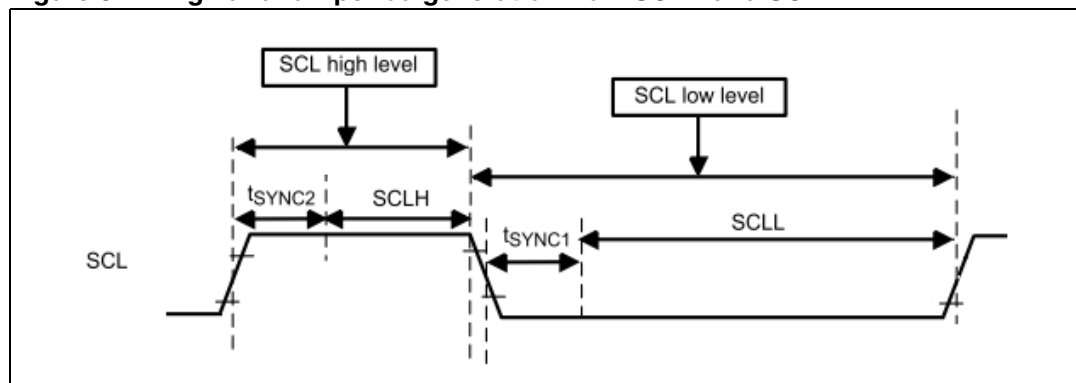
- SCL falling time
- When enabled, input delay brought by the analog filter:  $0.05 \mu\text{s} < t_{\text{AF}} < 0.26 \mu\text{s}$
- When enabled, input delay brought by the digital filter:  $t_{\text{DNF}} = \text{DNF} \times t_{\text{I2CCLK}}$
- Delay due to SCL synchronization to I2CCLK clock (2 to 3 I2CCLK periods)

SDADEL is defined by:

$$\text{SDADEL} \geq \{t_{\text{f(max)}} + t_{\text{HD;DAT(min)}} - t_{\text{AF(min)}} - t_{\text{DNF}} - [2 \times t_{\text{I2CCLK}}]\} / \{t_{\text{PRESC}}\}$$

$$\text{SDADEL} \leq \{t_{\text{VD;DAT(max)}} - t_{\text{r(max)}} - t_{\text{AF(max)}} - t_{\text{DNF}} - [3 \times t_{\text{I2CCLK}}]\} / \{t_{\text{PRESC}}\}$$

SCLH[7:0] and SCLL[7:0] are used to configure I2C speed frequency when master mode is selected. SCLH generates the high period of the SCL clock ( $t_{\text{HIGH}}$ ) and SCLL generates the low period of the SCL clock ( $t_{\text{LOW}}$ ). The figure below shows how these timings are deduced:

**Figure 6. High and low period generation from SCLH and SCLL**

$t_{\text{SYNC2}}$  duration depends on these parameters:

- SCL rising time
- When enabled, input delay brought by the analog filter:  $0.05 \mu\text{s} < t_{\text{AF}} < 0.26 \mu\text{s}$
- When enabled, input delay brought by the digital filter:  $t_{\text{DNF}} = \text{DNF} \times t_{\text{I2CCLK}}$
- Delay due to SCL synchronization to I2CCLK clock (2 to 3 I2CCLK periods)

SCL clock period ( $t_{SCL}$ ) which defines I2C speed frequency ( $f_{SCL} = 1/t_{SCL}$ ) is defined by:

$$t_{SCL} = t_f + t_{LOW} + t_r + t_{HIGH}$$

SCLH and SCLL are defined as follows:

$$t_{HIGH(min)} \leq t_{AF(min)} + t_{DNF} + 2 \times t_{I2CCLK} + [(SCLH+1) \times t_{PRESC}]$$

$$t_{LOW(min)} \leq t_{AF(min)} + t_{DNF} + 2 \times t_{I2CCLK} + [(SCLL+1) \times t_{PRESC}]$$

*Note: SCLH and SCLL values depend on the rise and fall time.*

*The rise time is defined by:*

$$t_r = R_p \times C_b \times 0.8473 \text{ (} R_p \text{ is the pull\_up resistor and } C_b \text{ is the bus capacitance)}$$

*The fall time depends on the software configuration of the I/O. Please refer to "I/O AC characteristics" table in STM32 products datasheets to get the value of fall time.*

### 3 Tutorials

This section describes how to use the I2C timing configuration tool.

**Figure 7. I2C timing configuration tool user interface**

**I2C Timing Configuration Tool for STM32F3xx and STM32F0xx devices V1.0.0** ©COPYRIGHT STMicroelectronics MCD Application Team

**Please enter the input parameters :**

Device Mode: Master

I2C Speed Mode: Standard Mode

I2C Speed Frequency (KHz): 100

I2C Clock Source Frequency (KHz): 8000

For STM32F3xx devices, max clock frequency is 72 MHz  
For STM32F0xx devices, max clock frequency is 48 MHz

Analog Filter Delay: ON

Coefficient of Digital Filter: 0

Rise Time (ns): 100

Fall Time (ns): 10

**Output Result (Timing register):**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC				Reserved				SCLDEL				SDADEL			
0								0				0			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH								SCLL							
0								0							

TIMINGR register Value : 0x00000000 (Double Click to copy)

Error (%) : 0,0000 %

Run Reset

**!!! This program work only if "macros" are enabled in "EXCEL" !!!**

2003 version :To enable "macros":>>> TOOLS>> MACRO>> SECURITY>> we recommend to set it to "medium" (you will be asked for macros to be executed)

2007-2010 version :To enable "macros":>>> Developer>> Macro Security>> MacroSettings>>Choose "Enable all macros "

Please active the "Developer" tab menu if is not active :>>> File >> Options>> Customize Ribbon>> Enable Developer tab menu

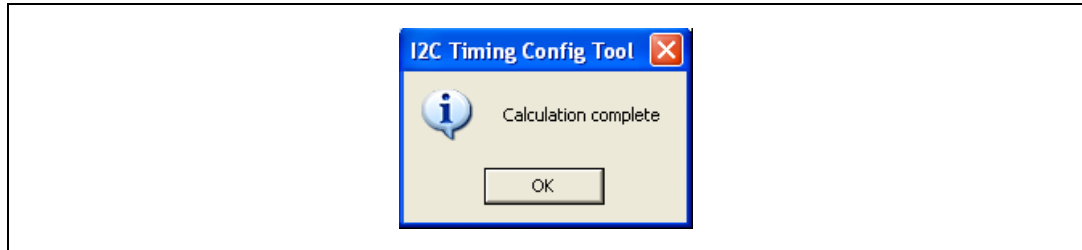
**Note:** The **"Reset"** button resets the input parameters to their default configuration.

To get the value of the timing register, follow these steps:

1. Select device mode by choosing **"Master"** or **"Slave"** in the list box.
2. Configure the speed mode by selecting one of the following modes in the list box:
  - Standard mode: maximum frequency is 100 KHz.
  - Fast mode: maximum frequency is 400 KHz.
  - Fast mode Plus: maximum frequency is 1000 KHz.
3. Set the desired I2C speed frequency (master clock).
4. Set the value of I2C clock source frequency.
5. Specify if analog noise filter is enabled or not.
6. Specify if digital noise filter is used or not by setting the filter coefficient (this coefficient should be an integer from 0 to 15).
7. Set the value of rise time.
8. Set the value of fall time.

9. Click the RUN button:
  - a) If the calculation of the timing register is completed, the following message is displayed:

**Figure 8. Calculation is completed**



In this case, you can copy the generated value from TIMINGR register value test-box and use it to configure the I2C timing (a double click in the result box copies the value).

Here is an example showing how to use the generated value to initialize the I2C timing register using the standard peripheral library of STM32 products.

We suppose that:

- Master mode is selected and desired I2C speed frequency is 100 KHz in fast mode.
- The I2C clock source frequency is 48 MHz with SYSCLK as source.
- Analog and digital noise filters are disabled.
- Rise time value is 65 ns and the fall time value is 5 ns.

The generated value for this configuration is **0xA0120227**.

Follow this steps to configure the I2C peripheral:

1. Declare the I2C initialization structure:
 

```
I2C_InitTypeDef I2C_InitStructure;
```
2. Initialize structure parameters:
 

```
I2C_InitStructure.I2C_Mode = I2C_Mode_I2C;
I2C_InitStructure.I2C_AnalogFilter = I2C_AnalogFilter_Disable;
I2C_InitStructure.I2C_DigitalFilter = 0x00;
I2C_InitStructure.I2C_OwnAddress1 = 0x00;
I2C_InitStructure.I2C_Ack = I2C_Ack_Enable;
I2C_InitStructure.I2C_AcknowledgedAddress = I2C_AcknowledgedAddress_7bit;
I2C_InitStructure.I2C_Timing = 0xA0120227;
```
3. Call I2C\_Init() function:
 

```
I2C_Init(I2C1, &I2C_InitStructure);
```

  - b) If the user configuration does not provide a result compliant with the I2C timing specification, this message is displayed:

**Figure 9. Error message**

In this case, you should try to generate the timing register value with a different configuration.

## 4 Conclusion

This application note provides a brief description of the I2C timing register configuration and explains how to use the I2C timing configuration tool with the STM32 microcontroller devices.



## 5 Revision history

**Table 4. Document revision history**

<b>Date</b>	<b>Revision</b>	<b>Changes</b>
24-Jan-2013	1	Initial release

**Please Read Carefully:**

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

**UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.**

**UNLESS EXPRESSLY APPROVED IN WRITING BY TWO AUTHORIZED ST REPRESENTATIVES, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.**

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2013 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

[www.st.com](http://www.st.com)