



STM32F10xxx I²C application examples

Introduction

This application note is intended to provide practical application examples of the STM32F10xxx I²C peripheral use.

This document, its associated firmware, and other such application notes are written to accompany the STM32F10xxx firmware library. These are available for download from the STMicroelectronics website: www.st.com.

1 STM32F10xxx I²C-I²C communication using interrupts in 7-bit addressing mode

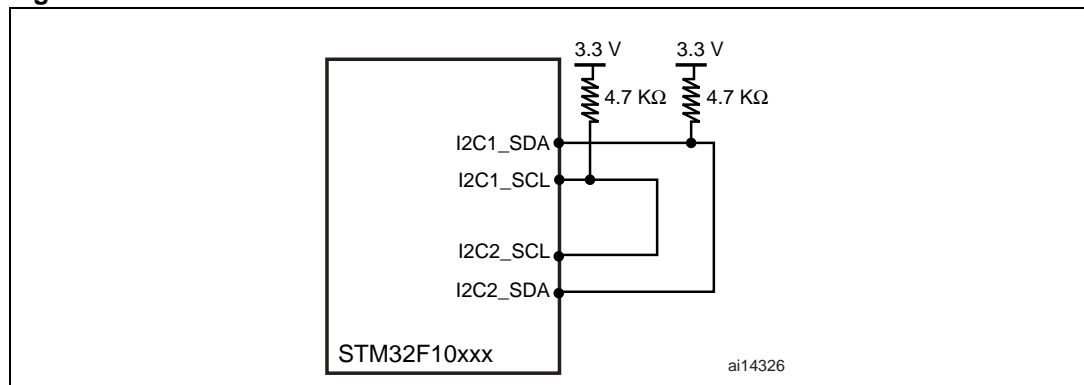
1.1 Overview

This section describes how to set an I²C-I²C communication from a master transmitter to a slave receiver, using interrupts in 7-bit addressing mode. The PEC is transmitted at the end of the transfer.

1.2 Hardware description

Figure 1 shows a typical connection between the STM32F10xxx I2C1 and I2C2. The I2C1 and I2C2 data (SDA) pins are connected together. The I2C1 and I2C2 clock (SCL) pins are also connected together. A pull-up resistor is connected to each line (SDA and SCL).

Figure 1. STM32F10xxx I²C-I²C communication



1.3 Firmware description

The provided firmware includes the I²C driver that supports all I²C communications through a set of functions.

After enabling the two I²C peripherals and the two event and buffer interrupts, the transfer in 7-bit addressing mode starts after the I2C1 start condition is generated. Each time an event occurs on the master or the slave, it is managed in the I2C1 or I2C2 interrupt routine, respectively. In this application, I2C1 Tx_Buffer is transmitted from the master (I2C1) to the slave (I2C2) and stored into I2C2 Rx_Buffer. At the end of the transfer, the PEC is transmitted from master to slave. It is then saved in the PEC_Value variable. All transmitted and received buffers are compared to check that all data have been correctly transferred.

This firmware is provided as *I2C example 1* in the STM32F10xxx firmware library, available from the STMicroelectronics microcontrollers site.

1.4 Conclusion

The use of interrupts in I²C communication makes it easier to check each generated event. PEC is another way of checking the reliability of the exchanged data.

2 STM32F10xxx I²C-I²C communication in dual addressing mode

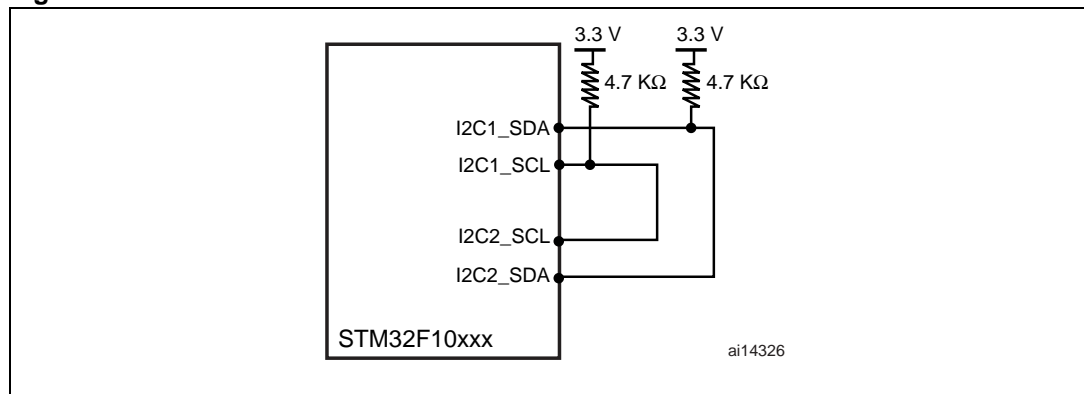
2.1 Overview

This section describes how to set an I²C-I²C communication by addressing the slave with its two addresses.

2.2 Hardware description

Figure 2 shows a typical connection between the STM32F10xxx I2C1 and I2C2. The I2C1 and I2C2 data (SDA) pins are connected together. The I2C1 and I2C2 clock (SCL) pins are also connected together. A pull-up resistor is connected to each line (SDA and SCL).

Figure 2. STM32F10xxx I²C-I²C communication



2.3 Firmware description

The provided firmware includes the I²C driver that supports all I²C communications through a set of functions.

Dual addressing implies two steps:

- First, the I2C1 master transmitter sends the I2C1 Tx_Buffer1 data buffer to the slave (I2C2) that saves the received data in I2C2 Rx_Buffer1. I2C2 is addressed by its first slave address I2C2_SLAVE1_ADDRESS7 programmed in the I2C2 OAR1 register. These transmitted and received buffers are compared to check that all data have been correctly transferred.
- Second, the I2C2 is now addressed by its second slave address I2C2_SLAVE2_ADDRESS7 programmed in the I2C2 OAR2 register. The I2C1 Tx_Buffer2 contents are transmitted by the master (I2C1) to the slave (I2C2) that stores them into I2C2 Rx_Buffer2. A second comparison takes place between the transmitted and received buffers to check that all data have been correctly transferred.

This firmware is provided as *I²C example 2* in the STM32F10xxx firmware library, available from the STMicroelectronics microcontrollers website.

2.4 Conclusion

The STM32F10xxx I²C peripheral may be addressed with one of its two configurable slave addresses.

3 STM32F10xxx I²C-I²C communication using ARP capability

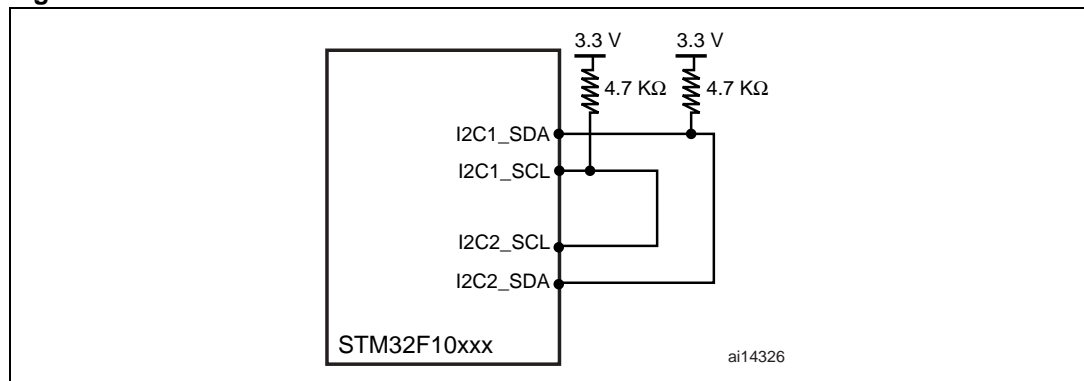
3.1 Overview

This section describes how to set an I²C-I²C communication using the ARP (address resolution protocol) capability with PEC transmission.

3.2 Hardware description

Figure 3 shows a typical connection between the STM32F10xxx I2C1 and I2C2. The I2C1 and I2C2 data (SDA) pins are connected together. The I2C1 and I2C2 clock (SCL) pins are also connected together. A pull-up resistor is connected to each line (SDA and SCL).

Figure 3. STM32F10xxx I²C-I²C communication



3.3 Firmware description

The provided firmware includes the I²C driver that supports all I²C communications through a set of functions.

After configuring the I2C1 and I2C2 as SMBus Host and Device, respectively, both I²Cs are enabled. The PEC calculation is enabled for both I²Cs. The ARP capability is enabled for slave I2C2. After the start condition, the master (I2C1) sends the SMBus default header and I2C2 responds by setting its SMBDEFAULT flag. The master (I2C1) then issues the Prepare to ARP command to the slave (I2C2). PEC transfer is then enabled for both I²Cs, and the PEC value received on I2C2 is stored into the PEC_Value variable.

This firmware is provided as *I²C example 3* in the STM32F10xxx firmware library, available from the STMicroelectronics microcontrollers website.

3.4 Conclusion

The STM32F10xxx I²C peripherals support the SMBus address resolution protocol (ARP).

4 STM32F10xxx I²C-to-I²C communication in 10-bit addressing mode

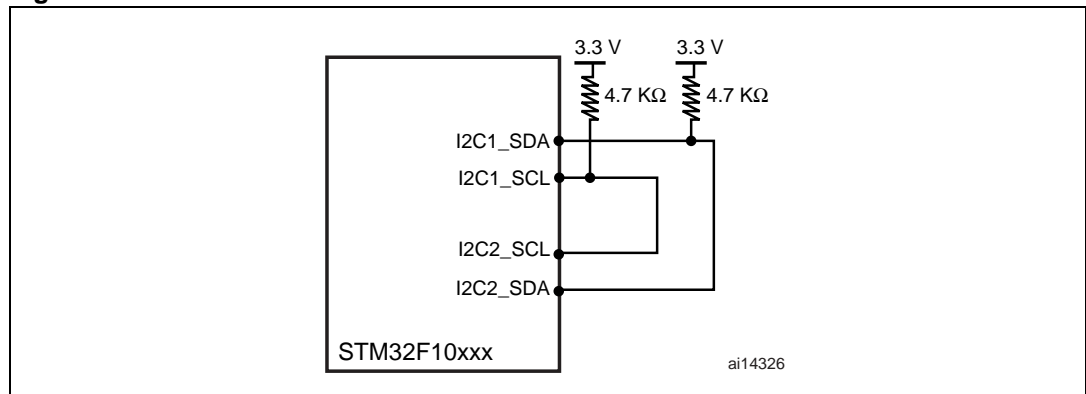
4.1 Overview

This section describes how to set an I²C-to-I²C communication in 10-bit addressing mode.

4.2 Hardware description

Figure 4 shows a typical connection between the STM32F10xxx I2C1 and I2C2. The I2C1 and I2C2 data (SDA) pins are connected together. The I2C1 and I2C2 clock (SCL) pins are also connected together. A pull-up resistor is connected to each line (SDA and SCL).

Figure 4. STM32F10xxx I²C-to-I²C communication



4.3 Firmware description

The provided firmware includes the I²C driver that supports all I²C communications through a set of functions.

After enabling the two I²C peripherals, the transfer in 10-bit addressing mode starts after the I2C1 start condition is generated. Each time an event occurs on the master or the slave, it is managed on I2C1 or I2C2, respectively. In this application, I2C1 Tx_Buffer is transmitted from the master (I2C1) to the slave (I2C2) and stored into I2C2 Rx_Buffer. All transmitted and received buffers are compared to check that all data have been correctly transferred.

This firmware is provided as *I²C example 4* in the STM32F10xxx firmware library, available from the STMicroelectronics microcontrollers website.

4.4 Conclusion

The STM32F10xxx I²C peripherals support standard 10-bit addressing mode.

5 STM32F10xxx I²C and M24C08 EEPROM communication

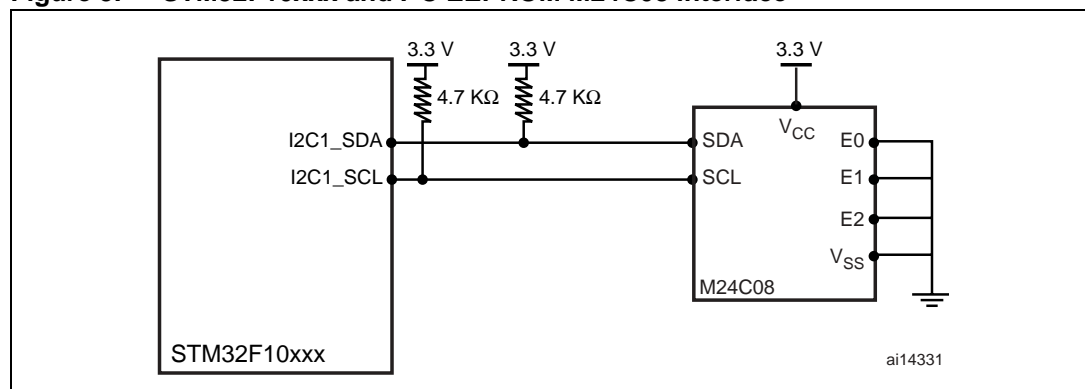
5.1 Overview

This section describes how to use the I²C firmware library and an associated I²C EEPROM driver to communicate with an M24C08 EEPROM. An example is given that uses most of the I²C EEPROM driver functionality: Buffer Write and Buffer Read, etc.

5.2 Hardware description

Figure 5 shows a typical connection between the STM32F10xxx I2C1 and an I²C EEPROM. The I2C1 and the EEPROM data pins (SDA) are connected together. The I2C1 and the EEPROM clock (SCL) pins are also connected together. A pull-up resistor must be connected on each line (SDA and SCL). The E2, E1 and E0 EEPROM pins must be tied to ground. The EEPROM V_{CC} pin must be connected to 3.3 V while the EEPROM V_{SS} pin must be connected to ground.

Figure 5. STM32F10xxx and I²C EEPROM M24C08 interface



5.3 Firmware description

The provided firmware includes the I²C EEPROM driver that supports all write and read operations through a set of functions. An example of use for most of these functions is provided.

First, the contents of Tx1_Buffer are written to the EEPROM_WriteAddress1 and the written data are read. The written and the read buffer data are then compared. After a time, the EEPROM reverts to its Standby state. Then, a second write operation is performed and this time, Tx2_Buffer is written to EEPROM_WriteAddress2, that is the address that comes just after the last written address. After completion of the operation, the written data are read. The contents of the written and the read buffers are compared. Once this is done, the two written buffers are read starting from EEPROM_WriteAddress1, and stored into a unique buffer. The contents of the unique buffer are then read and compared to the data in Tx1_Buffer and Tx2_Buffer.

This firmware is provided as *I²C example 5* in the STM32F10xxx firmware library, available from the STMicroelectronics microcontrollers website.

5.4 Conclusion

With a simple hardware connection and this I²C EEPROM driver example, the user is able to develop greater and more complex communication applications between the STM32F10xxx and any interfaced I²C EEPROM.

6 Revision history

Table 1. Document revision history

Date	Revision	Changes
26-Jun-2007	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 AN AUTHORIZED ST REPRESENTATIVE, 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.

© 2007 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 - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com