AN2576
Application note

STM32F10xxx SPI application examples

Introduction

This application note is intended to provide practical application examples of the STM32F10xxx SPI 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.*

Contents

1 STM32F10xxx full-duplex SPI-SPI communication with software NSS management

1.1 Overview

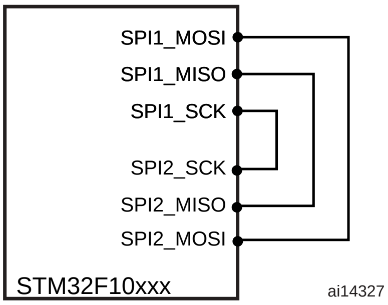
This section describes how to set an SPI-SPI full-duplex communication. The NSS pin software management allows master-to-slave and slave-to-master permutation without any hardware modification.

1.2 Hardware description

Figure 1 shows a typical connection in full-duplex mode between the STM32F10xxx SPI1 and SPI2.

The SPI1 and SPI2 data inputs (MISO) are connected together. The SPI1 and SPI2 data outputs (MOSI) are connected together. And the SPI1 and SPI2 clock inputs (SCK) are also connected together. The SPI1 and SPI2 NSS pins are kept unconnected thanks to the software NSS management.

Figure 1. STM32F10xxx full-duplex SPI-SPI communication



1.3 Firmware description

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

The NSS pins are configured by software to set SPI1 as the master and, SPI2 as the slave. First, a data buffer is sent from the master (SPI1) to the slave (SPI2) and at the same time, the salve sends another data buffer to the master.
In the second part of the example, using software and without any hardware modification, SPI1 is re-configured as the slave and SPI2, as the master. A new full-duplex communication is established and two data buffers are exchanged between the master and the slave. All the transmitted and received buffers are then compared to check that all the data have been correctly exchanged.

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

1.4 Conclusion

The NSS software management is given to remove the need for any hardware modification in a given application where a switch from master to slave or slave to master is required. Owing to the software management of the NSS pins, the user has no hardware constraint in this type of application.

2 STM32F10xxx simplex SPI-SPI communication using interrupts

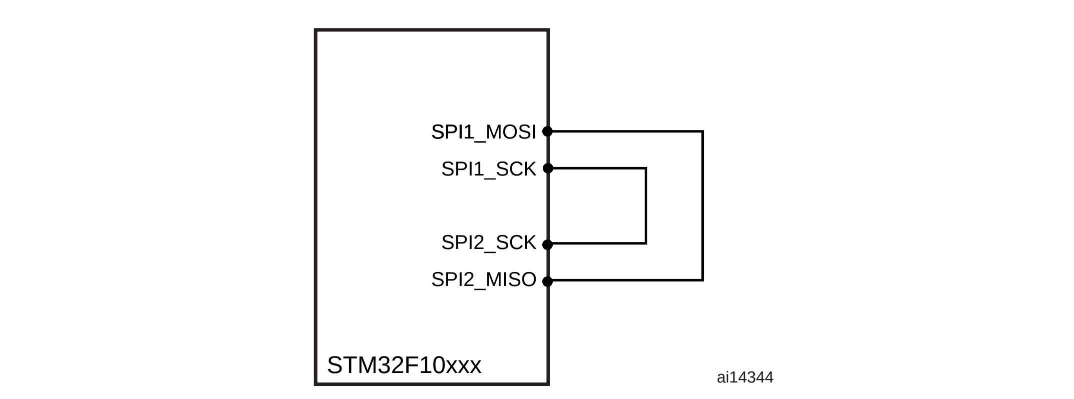
2.1 Overview

This section describes how to set an SPI-SPI simplex communication using TxE and RxNE interrupts.

2.2 Hardware description

Figure 2 shows a typical connection in simplex mode between the STM32F10xxx SPI1 and SPI2. The SPI1 data output (MOSI) is connected to the SPI2 data input (MISO) and the SPI1 and SPI2 clock (SCK) lines are connected together. The SPI1 and SPI2 NSS pins are kept unconnected thanks to the software NSS management.

Figure 2. STM32F10xxx simplex SPI-SPI communication



2.3 Firmware description

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

The NSS pins are configured by software to set SPI1 as the master and, SPI2 as the slave. A data buffer is sent from the master (SPI1) using the TxE interrupt, to the slave (SPI2) that receives the data using the RxNE interrupt. The transmitted buffer and the received buffer are compared to check that all data have been correctly exchanged.

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

2.4 Conclusion

The use of the TxE and RxNE interrupts within a communication reduces the code density and makes it easy for the user to transmit and receive data. In simplex communication mode, the user only needs to use the master MOSI pin and the slave MISO pin with the CLK pins. In this way, the other SPI pins can be used for other purposes.

3 STM32F10xxx full-duplex SPI-SPI communication with CRC transfer

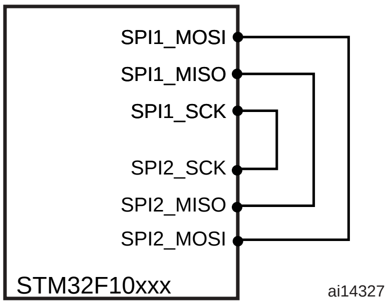
3.1 Overview

This section describes how to set an SPI-SPI full-duplex communication followed by a CRC transmission.

3.2 Hardware description

Figure 3 shows a typical connection between the STM32F10xxx SPI1 and SPI2. The SPI1 and SPI2 data inputs (MISO) are connected together. The SPI1 and SPI2 data outputs (MOSI) are connected together. And the SPI1 and SPI2 clock inputs (SCK) are also connected together. The SPI1 and SPI2 NSS pins are kept unconnected thanks to software NSS management.

Figure 3. STM32F10xxx full-duplex SPI-SPI communication



3.3 Firmware description

The provided firmware includes the SPI driver that supports all SPI communications through a set of functions.
The NSS pins are configured by software to set SPI1 as the master and, SPI2 as the slave. First, a data buffer is sent from the master (SPI1) to the slave (SPI2) and at the same time, the salve sends another data buffer to the master. The CRC calculation is already enabled. At the end of both buffer transmission, CRC transfer is enabled for both master and slave. The CRC values received by SPI1 and SPI2 are stored into two variables and the user may use them to check whether the communications completed successfully. The transmitted buffer and the received buffer are compared to check that all data have been correctly exchanged. If the received CRC value is erroneous, the CRCERR flag is set in the SPI_SR; if the received CRC value is correct, the CRCERR flag remains cleared.

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

3.4 Conclusion

By making it possible to check the transmitted or received data using CRC, the STM32F10xxx brings a greater reliability to SPI communications than most peripherals.

4 Simplex SPI-SPI communication with DMA and hardware NSS management

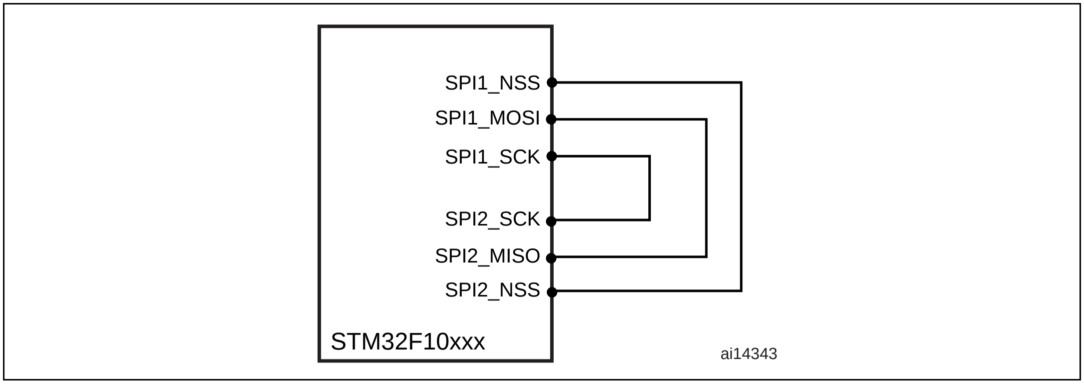
4.1 Overview

This section describes how to set an SPI-SPI simplex communication using DMA. The NSS pin is managed by hardware.

4.2 Hardware description

Figure 4 shows a typical connection in simplex mode, between the STM32F10xxx SPI1 and SPI2. SPI2 data output (MOSI) is connected to SPI1 data input (MISO) and the SPI1 and SPI2 clock (SCK) lines are connected together. The SPI1 and SPI2 NSS pins are also connected together.

Figure 4. STM32F10xxx simplex SPI-SPI communication



4.3 Firmware description

The provided firmware includes the SPI driver that supports all SPI communications through a set of functions.
The NSS pins are configured by hardware to set SPI2 as the master and SPI1 as the slave: by enabling SS output (that is by setting the SPI_CR2 SSOE bit), the SPI2 NSS pin resets the SPI1 NSS pin and configures SPI1 as the slave. The master (SPI2) sends a data buffer to the slave (SPI1) by polling the TxE flag. SPI1 receives the data through DMA channel2. At the end of the data transfer, the transmitted and the received buffers are compared to check whether all data have been correctly exchanged.

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

4.4 Conclusion

The use of DMA within a communication further reduces the code density and execution time. It is an easy way of transmitting and receiving data. The SSOE bit makes it possible to choose a unique master when all devices on the bus have their NSS pins connected together.

5 STM32F10xxx SPI and M25P64 Flash memory communication

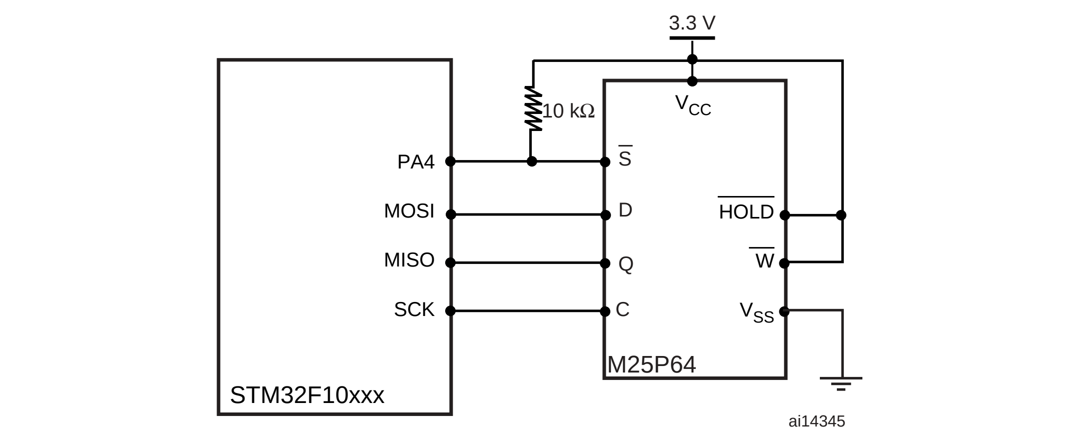
5.1 Overview

This section describes how to use the SPI firmware library with an associated SPI Flash memory driver to communicate with an M25P64 Flash memory. An example is given that uses most of the SPI Flash memory driver functionality: Write, Read, Erase, Get Flash ID, etc.

5.2 Hardware description

Figure 5 shows a typical interface between the STM32F10xxx and an M25P64 SPI Flash memory. The STM32F10xxx data input (MISO), data output (MOSI) and clock (SCK) input connected to the M25P64 data input (Q), data output (D) and clock (C) input, respectively. Another STM32F10xxx output pin (PA4) is connected to the M25P64 Chip Select ($\overline{S}$) input so that the STM32F10xxx can identify a unique SPI Flash memory out of all the ones being used on the basis of the incoming operation itself.

Figure 5. STM32F10xxx and M25P64 SPI Flash memory interface



5.3 Firmware description

The provided firmware includes the SPI Flash memory driver that supports all erase, write and read operations through a set of functions. An example of use for most of these functions is provided.

First, the SPI Flash memory device identification (ID) is read. This ID is then compared with the expected value. The sector to be accessed is then erased and the buffer write and read operations are performed. The written and the read buffers are then compared. When this is done, a new Sector Erase operation is performed to clear the data already written in the selected SPI Flash memory sector. Finally, a read operation is carried out to check that all data have been correctly erased.

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

5.4 Conclusion

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

6 Revision history

Table 1. Document revision history

Date	Revision	Changes
28-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.

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