AN970
APPLICATION NOTE

SPI COMMUNICATION BETWEEN ST7 AND EEPROM

by 8-Bit Micro Application Team

INTRODUCTION

The goal of this application note is to present a practical example of communication using the SPI peripheral of the ST7.

It shows an easy way of communicating between a ST7 microcontroller and a M95xxx SPI EEPROM. The purpose is to perform, through SPI, a write in the memory, followed by a read of the written data.

1 ST7 / EEPROM SPI INTERFACE

This section summarizes the main features of the ST7/EEPROM SPI interface. Please refer to the ST7 datasheet for more details.

The Serial Peripheral Interface (SPI) allows full-duplex, synchronous, serial communication between devices. A SPI system may consist of a master and several slaves, or of a system in which devices may be either master or slave.

There is only one master at any one time.

The Bus signals are:

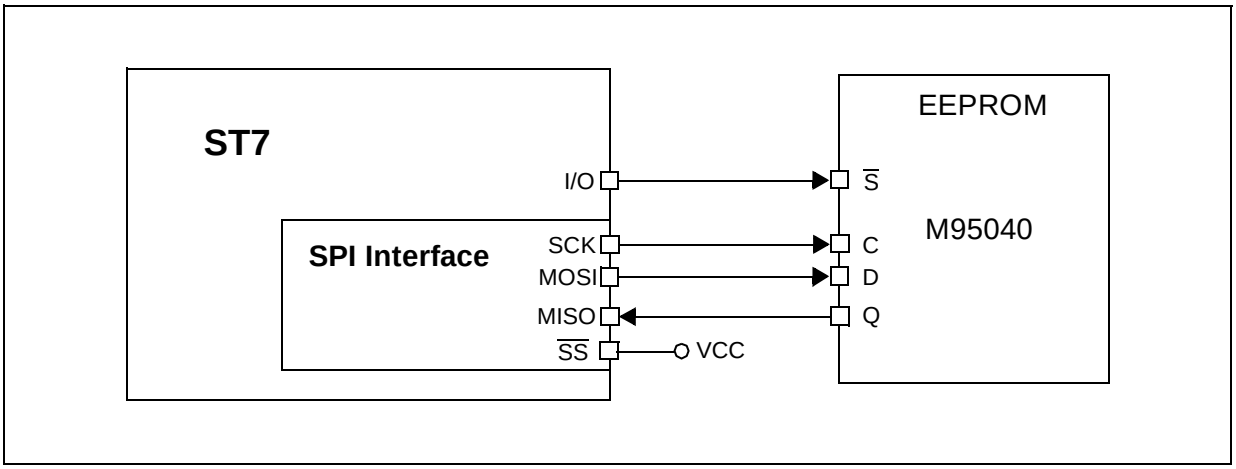
- The serial clock (SCK).
- The MOSI (Master Out Slave In).
- The MISO (Master In Slave Out).

One more pin, the $\overline{SS}$ pin (slave select), is needed to select the slave or the master mode for each device. this can be done in hardware or software mode.

In this application the ST7 is always used as master ($\overline{SS}$ pin = high level) and configures the EEPROM mode through an output. The ST7 and SPI interface set-up is shown Figure 1.

During SPI transfer, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). Data are transmitted MSB first. The serial clock is used to synchronize the data transfer during a sequence of eight clock pulses.

Figure 1. ST7 and SPI Interface Set-Up



2 ST72264 CONFIGURATION

2.1 I/O CONTROL

Four pins of the ST72264 are used:

- The 3 data and clock SPI pins (SCK, MOSI, MISO).
- An output pin to select and deselect the M95xxx.
- $\overline{SS}$ pin to select master or slave mode.

In our application, the output for selecting the M95xxx is pin 3 of Port B. It is configured as output push-pull (refer to the datasheet for details).

2.2 SPI PERIPHERAL

2.2.1 General

This peripheral is configured with the SPI Control Register.

7							0
SPIE	SPE	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0

The output must be enabled (SPE = 1).

If the $\overline{SS}$ pin is high (see SPICSR), the ST72264 can be declared as master by setting the MSTR bit.

The transmission speed, in master mode, is selected using the SPR0, SPR1 and SPR2 bits.

The CPOL and CPHA bits define the timing characteristics.

When the SPIE bit is set, SPI interrupts are enabled (not used in our case).

2.2.2 $\overline{SS}$ software configuration

The $\overline{SS}$ pin can be fixed by hardware, either connected to Vss (for a slave configuration) or to Vdd (for a master configuration). But it can also be software driven through the SPICSR register (SSM and SSI bits) :

7							0
SPIF	WCOL	OVR	MODF	-	SOD	SSM	SSI

To set the master configuration (as in our case) : SSM=1 and SSI=1

To set the slave configuration: SSM=1 and SSI=0

2.2.3 Baud rates

First the communication speed must be chosen. As the clock is given by the master, the user has to configure the baud rate of the ST7 SPI (SPR[2-0] bits in the SPICR register).

For each CPU frequency, six Baud Rates are available. If f_{CPU} = 8MHz, the baud rate range is 62.5kHz to 2MHz.

2.2.4 Clock phase and polarity

Then the clock polarity and the clock phase have to be chosen.

The clock polarity (CPOL bit of the SPCR register) controls the steady state value of the clock when no data is being transferred.

The clock phase (CPHA bit of the SPCR register) selects on which clock transition the bit capture is made and consequently on which clock transition data is latched.

The user must be careful of the fact that some devices do not allow all timing relationships. For instance, the ST95040 EEPROM device accepts only CPOL,CPHA = (0,0) or (1,1) configurations.

3 M95040 EEPROM MANAGEMENT AND CONFIGURATION

3.1 MAIN FEATURES

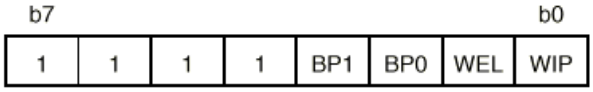
This is a 4K memory composed of two pages of 2K bytes.

All instructions, addresses and data are shifted in and out of the chip MSB first.

The write protect pin ($\overline{W}$) and the hold pin ($\overline{H}$) are not used in our application (both are high level).

3.2 STATUS REGISTER

This device has one status register.



BP1, BP0: Read and write bits
WEL, WIP: Read only bits.
b7 to b4: Read only bits.

The BP1 and BP0 bits in the status register can be used to write protect a block of memory. In this application, both bits are cleared, allowing write access to all the memory.

The WEL bit indicates the status of the write enable latch.

The WIP bit indicates whether the memory is busy with a write operation.

3.3 INSTRUCTION SET

Prior to any operation, the device must be selected ($\overline{SS}$ pin at low level), then an one-byte instruction code must be sent to the EEPROM. The device has a set of 6 instructions (see Table 1 Instruction Set).

Table 1. Instruction Set

Instruction	Description	Instruction format
WREN	Set Write Enable Latch	0000 0110
WRDI	Reset Write Enable Latch	0000 0100
RDSR	Read Status Register	0000 0101
WRSR	Write Status Register	0000 0001
READ	Read Data from Memory Array	0000 $A_8$011
WRITE	Write Data to Memory Array	0000 $A_8$010

Notes: A_8 = 1, Upper page selected.
A_8 = 0, Lower page selected.

4 ST7 / EEPROM SPI PROTOCOL

The use of the main functions are briefly described below. For a complete description of the protocol, please refer to the M95040 SPI EEPROM data sheet.

Write_enable: The memory contains a write enable latch. This latch must be set prior to any WRITE or WRSR operation.

Byte_write: This function will write up to 16 bytes of data in the EEPROM. After the WRITE instruction, the memory address must be specified before sending data.

Byte_read: This function reads the memory. After the READ instruction, the memory address is specified. To be able to send the data, the EEPROM must receive the clock from the master: the ST7. This is made by sending a dummy byte. This operation will generate the 8 clock bits needed. The dummy value will not be seen by the EEPROM.

Write_SR: Give the proper value to the Status Register prior to any communication.

Read_SR: This process is similar to the byte read process. It will be used to check if there is a write in progress (WIP bit of the status register).

5 SPI COMMUNICATION BETWEEN AN ST7 AND AN EEPROM

The software included with this application note is only the SPI driver. The complete software can be found in the software library in the ST internet website. It is of course only an example. It is up to the user to adapt it to his specific application.

5.1 GENERAL DESCRIPTION

The software is a polling SPI communication between the ST72264 and EEPROM.

The first part of the software performs to the initialization of the ST72264 (core and SPI peripheral) and of the EEPROM (value of status register for no protection).

Then it executes the write cycles. The «write_loop» writes in the EEPROM, the values 0 to 127 at addresses 255 down to 128.

The third part executes the read cycles. The «read_loop» reads the content of the EEPROM at addresses 255 down to 128.

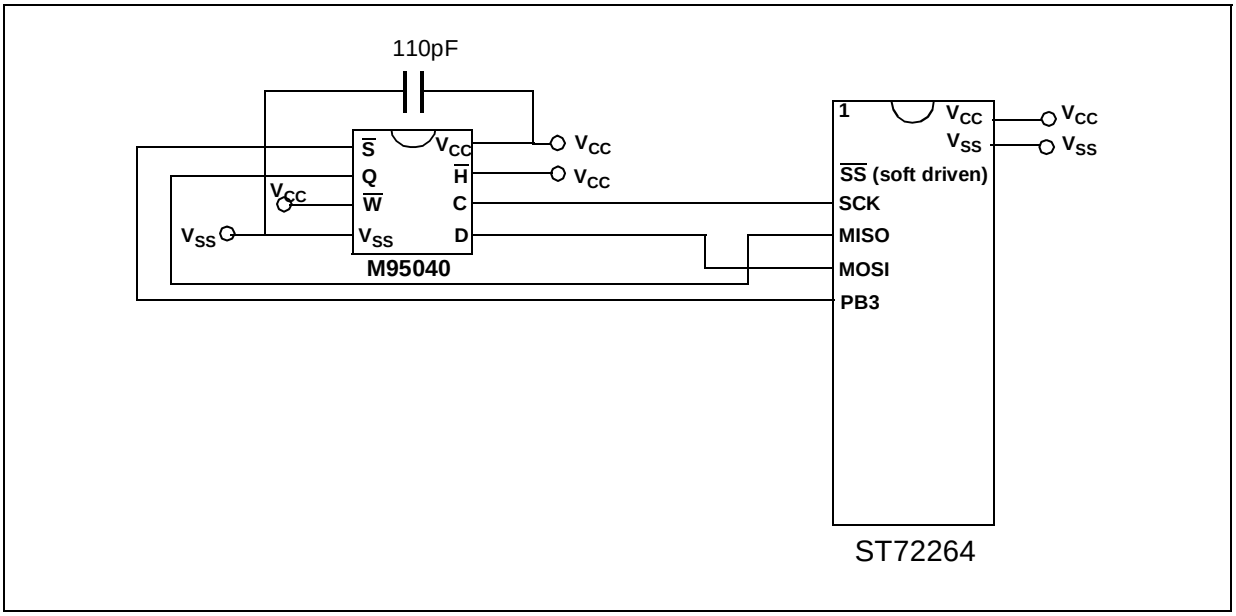
Finally, the software ends in an infinite loop.

5.2 HARDWARE DESCRIPTION

Two components were used for the application:

- 1 ST72264 MCU
- 1 M95040 EEPROM

Figure 2. Hardware description



$\overline{SS}$ pin of ST72264 is software driven (high state for a master device).

The SCK pin is connected to the serial clock pin (C) of the E²PROMs.

The MOSI pin to the serial input pins (D).

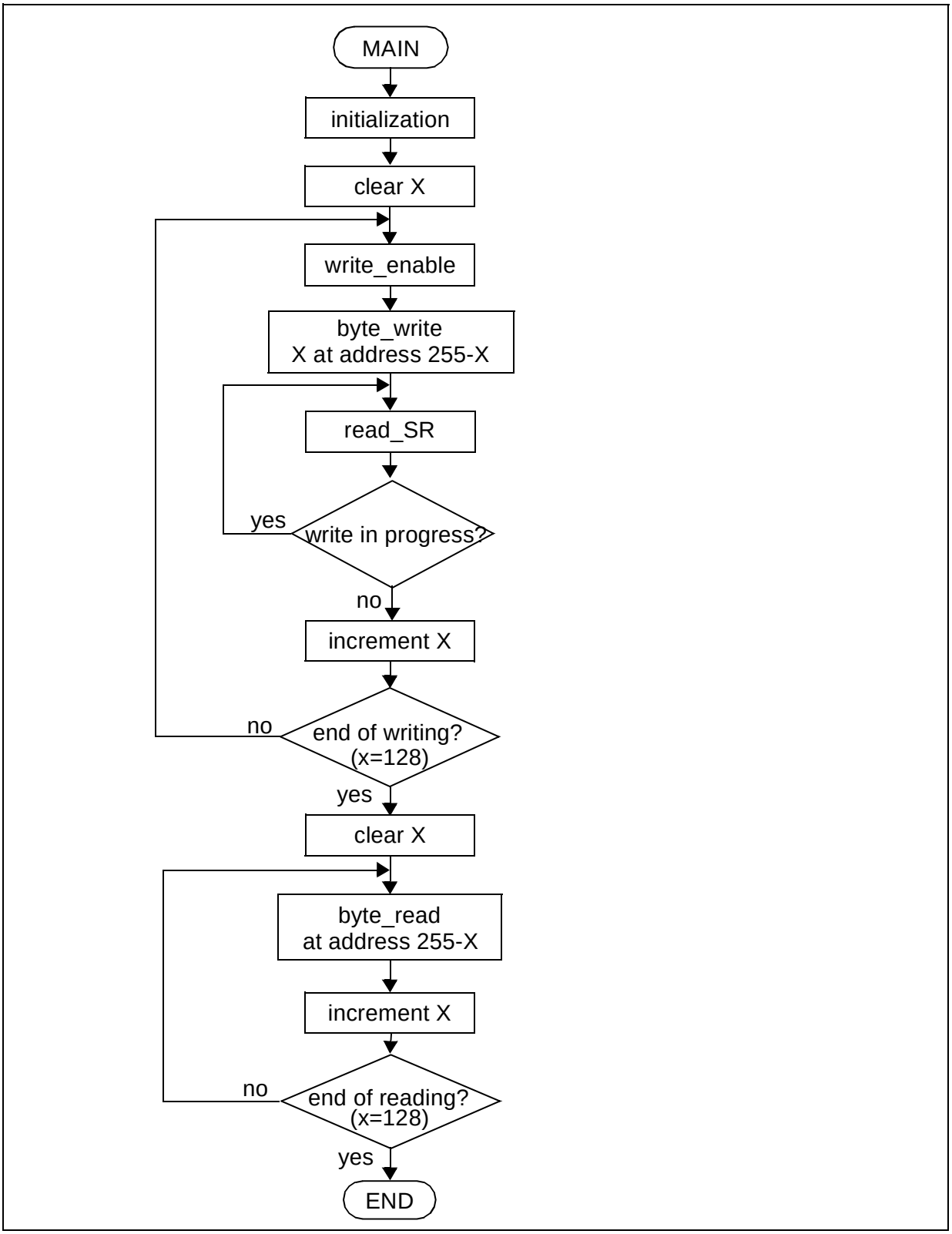
The MISO pin to the serial output pins (Q).

$\overline{S}$ pin of the EEPROM is connected to the PB3 output pin of ST72264.

5.3 FLOWCHARTS

The flowchart of the application is presented below (Figure 3.).

Figure 3. Flowchart: MAIN



5.4 USE OF SEVERAL EEPROMS

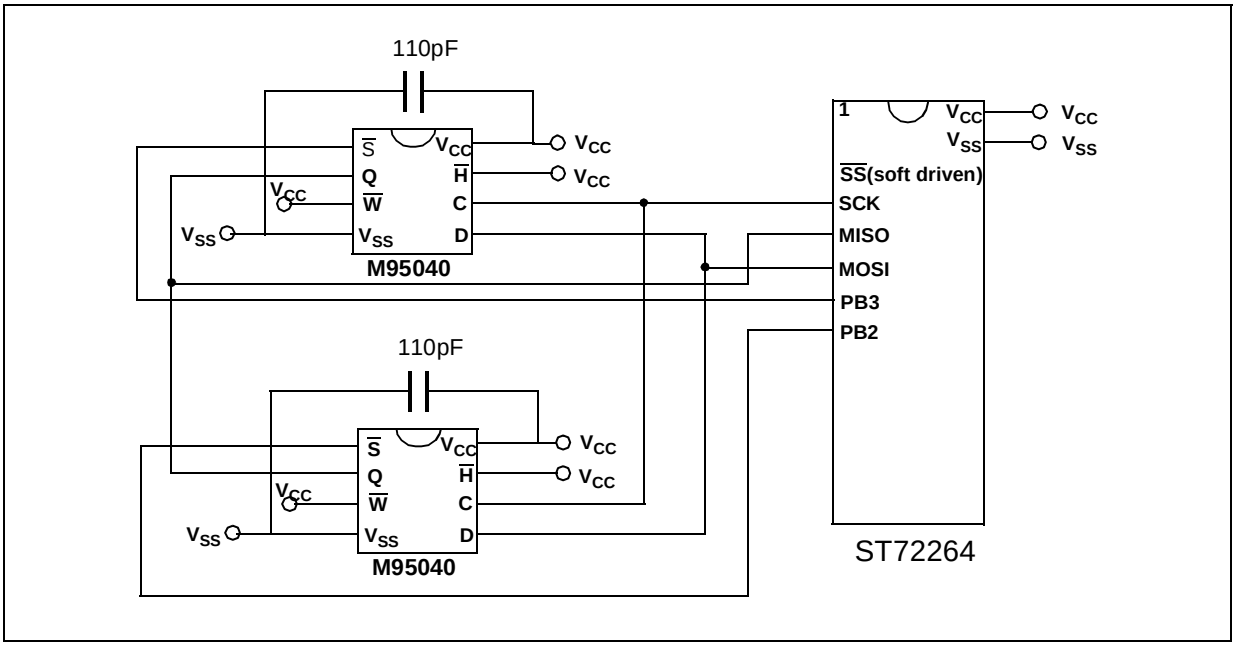
It is possible to communicate with several EEPROMs. Each one is selected in turn. Then the same protocol as described before is followed.

The main point is to **reserve one I/O pin to select each device.**

A hardware schematic using two devices is given below (Figure 4.).

The selection of the devices is made using the PB2 and PB3 pins.

Figure 4. Hardware schematic for two devices



5.5 SOFTWARE

All the source files in assembly code are given in the zip file with this application note.

The source files are for guidance only. STMicroelectronics shall not be held liable for any direct, indirect or consequential damages with respect to any claims arising from use of this software.