



AN971 APPLICATION NOTE

I²C COMMUNICATION BETWEEN ST7 AND M24Cxx EEPROM

by Microcontroller Division Applications

INTRODUCTION

The goal of this application note is to present an practical example of communication using the I²C peripheral of the ST7. It shows a basic single master communication between a ST7 microcontroller and an M24Cxx I²C bus EEPROM. The purpose is to implement, from the ST7 through the I²C interface, a write and a read to the external EEPROM without error management.

1 ST7 I²C INTERFACE

The ST7 I²C peripheral allows multi master and slave communication with bus error management. In this application, only single master mode is used without error management. As polling mode is the most difficult mode to implement, the application is based on this mode, but it can be easily adapted for interrupt management.

The I²C synchronous communication needs only two signals: SCL (Serial clock line) and SDA (Serial data line). The corresponding port pins have to be configured as floating inputs.

Please refer to the ST7 datasheet for more details.

1.1 COMMUNICATION SPEED

The ST7 I²C peripheral allows a large range of communication speeds. It is able to work in standard and fast I²C modes.

In master mode the communication speed is given by the Clock Control Register (CCR). An example is given in [Table 1](#).

Table 1. Example of Possible I²C Communication Speeds ($f_{CPU}=8$ MHz)

	Standard Mode					Fast Mode		
Speed [KHz]	15.5	25.00	50.00	70.00	100.00	167.00	190.00	333.00
CCR [hex]	EC	9E	4E	37	26	8E	8C	86

1.2 START, STOP CONDITION AND ACKNOWLEDGE GENERATION

In master mode, the Start and Stop conditions can be generated by setting the START and STOP bits in the Control Register (CR).

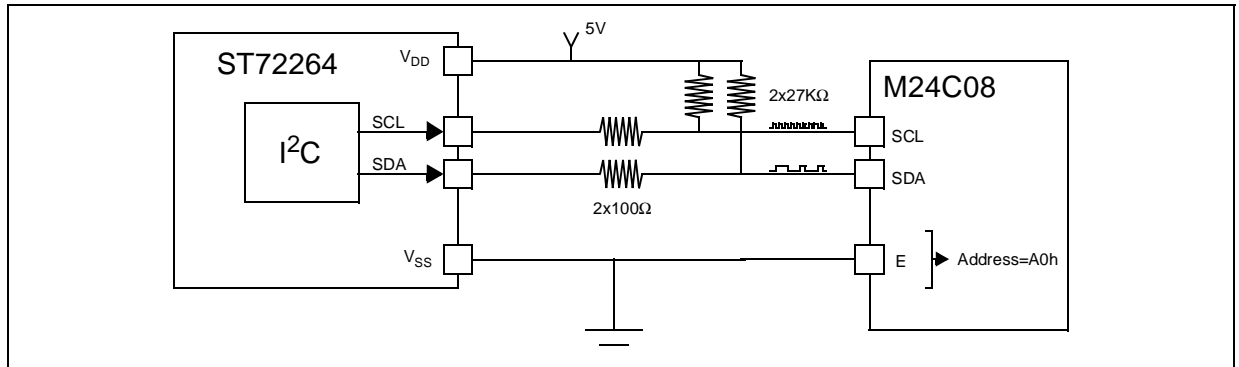
An Acknowledge is sent after an address or a data byte is received when the ACK bit is set in the Control Register (CR).

2 ST7 / M24CXX I²C COMMUNICATION APPLICATION

2.1 HARDWARE CONFIGURATION

The ST7 / M24Cxx I²C communication application hardware consists of a ST72264 microcontroller which communicates with an external M24C08 EEPROM through an I²C bus interface.

Figure 1. ST7 / EEPROM I²C Communications Application



2.2 ST7 I²C PERIPHERAL BASIC DRIVERS

In this chapter all registers refer to the ST7 I²C peripheral (unless otherwise specified).

2.3 INITIALIZE THE I²C PERIPHERAL

In this application the initialization of the ST7 I²C peripheral is done completely by software without taking into account the hardware reset status.

First the Control Register (CR) is cleared and the Data (DR) and Status (SR1,SR2) registers are touched to clear any pending events.

Then, the peripheral is enabled through the Control Register (CR). This action needs to write twice in the register due to the fact that the Control Register (CR) bits can be set only when the PE enable bit is already set. To allow the peripheral to acknowledge the received data the ACK bit of the Control Register (CR) is set.

As the M24C08 EEPROM is specified with a maximum I²C clock speed at 100KHz, the ST7 I²C peripheral is set to this speed (CCR=26h) in the application.

2.4 INITIATING A COMMUNICATION ON THE I²C BUS

To initiate an I²C communication, first a start condition has to be generated and then the selected slave address has to be sent, both by the master.

In the ST7 I²C peripheral this action is done by setting the START bit of the Control Register (CR) followed by writing the slave address in the Data Register (DR) with the least significant bit correctly set (0 = transmission, 1 = reception).

2.5 SENDING A DATA BYTE ON THE I²C BUS

To transmit a new data byte from the ST7 I²C peripheral on the I²C bus, the address or data byte previously transmitted has to be completed correctly. This previous byte transmission check is done by a polling loop on the BTF flag of the Status Register 1 (SR1). If during this loop an error is detected in the Status Registers (SR1,SR2) then the application goes into an infinite loop (no error management).

When the previous data transmission is over, the application writes the new data byte to be transmitted in the Data Register (DR).

Note: If the data byte to be transmitted is the first one after the slave address, a dummy write in the Control Register (CR) has to be performed to allow the setting of the BTF bit (see ST7 datasheet for more details). In this application, this dummy write is done by setting the PE bit for each data byte transmission (see description of EV6 in the Transfer Sequencing diagram in I²C datasheet description).

2.6 RECEIVING A DATA BYTE ON THE I²C BUS

To receive a new data byte in the ST7 I²C peripheral from the I²C bus, the data byte to receive has to be completed correctly. This byte reception check is done by a polling loop on the BTF flag of the Status Register 1 (SR1). If during this loop an error is detected in the Status Registers (SR1, SR2) then the application goes in an infinite loop (no error management).

When the data reception is finalized, the application reads the new data byte received in the Data Register (DR).

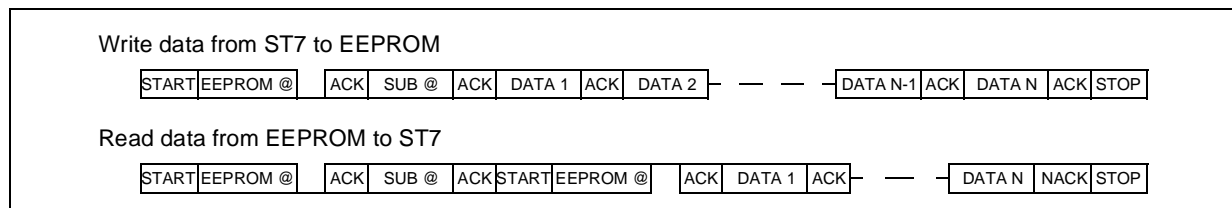
To close the communication: Before reading the last byte from the DR register, set the STOP bit to generate the Stop condition. The interface goes automatically back to slave mode (M/SL bit cleared). In order to generate the non-acknowledge pulse after the last received data byte, the ACK bit must be cleared just before reading the second last data byte.

Note: If the data byte to be received is the first one after the slave address, a dummy write in the Control Register (CR) has to be performed to allow the setting of the BTF bit (see ST7 datasheet for more details). In this application, this dummy write is done by setting the PE bit for each data byte reception (see description of EV6 in the Transfer Sequencing diagram in I²C datasheet description).

2.7 COMMUNICATING WITH THE I²C EEPROM

The communication protocol between the ST7 and the external M24Cxx EEPROM is given in Figure 2. For more details, please refer to the ST24C08 datasheet.

Figure 2. I²C Communication Protocol

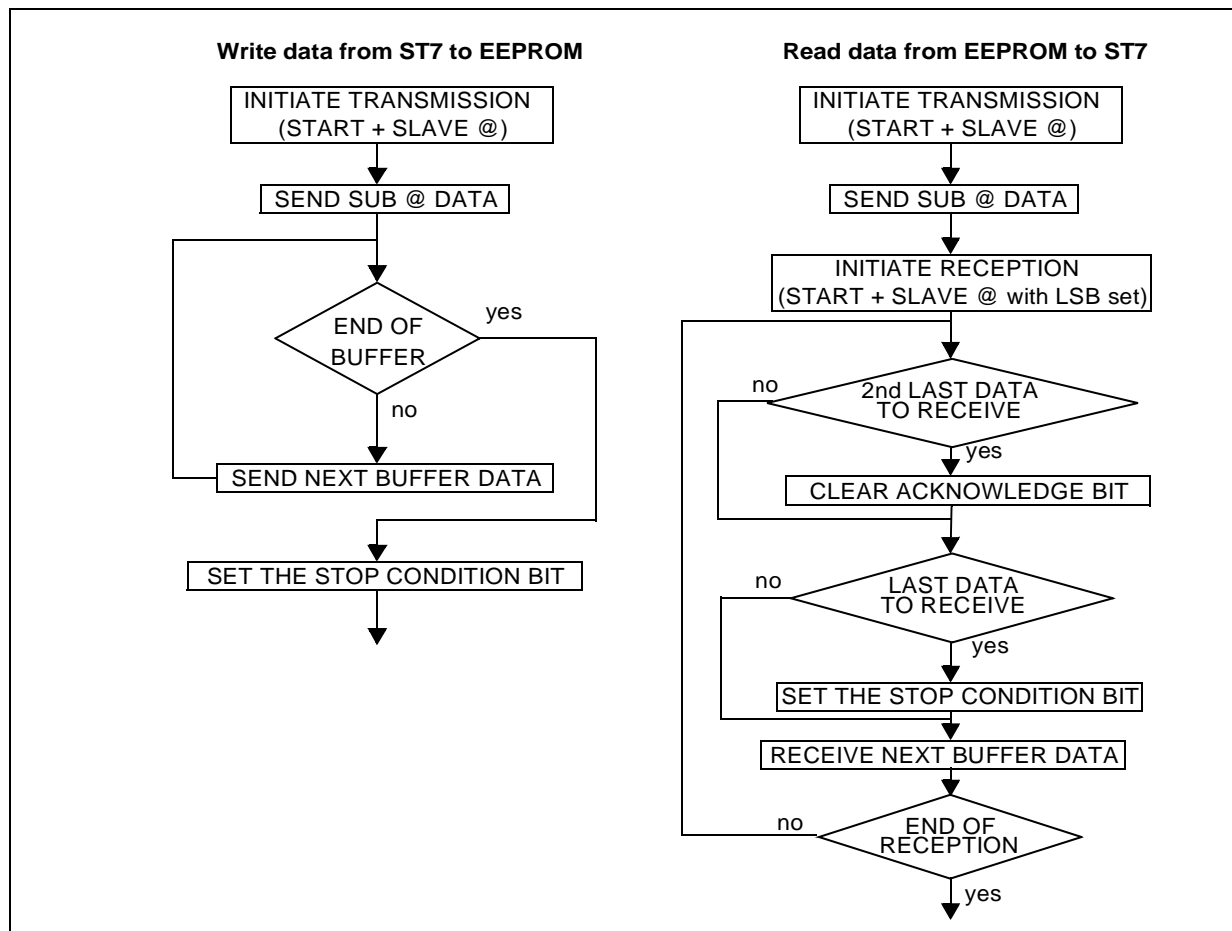


The ST7 / M24Cxx communication application is based on two steps:

- Write the content of an output buffer (from the ST7 ROM) in the M24Cxx EEPROM
- Read this written buffer from the M24Cxx EEPROM to the ST7 RAM.

Figure 3. shows the flowchart of these two steps.

Figure 3. Communication Application Flowchart



3 SOFTWARE

All the source files in assembly code are given in 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.

To adapt this polling software to interrupt management, replace the polling waiting loop by an interrupt event.

“THE PRESENT NOTE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS WITH INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM THE CONTENT OF SUCH A NOTE AND/OR THE USE MADE BY CUSTOMERS OF THE INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.”

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without the express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics

©2003 STMicroelectronics - All Rights Reserved.

Purchase of I²C Components by STMicroelectronics conveys a license under the Philips I²C Patent. Rights to use these components in an I²C system is granted provided that the system conforms to the I²C Standard Specification as defined by Philips.

STMicroelectronics Group of Companies

Australia - Brazil - Canada - China - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan
Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

<http://www.st.com>