

XEMICS

TN8000.13 Technical Note

***How to interface an I2C EEPROM with
the XE8000 microcontrollers family***

Author: Miguel Luis

For further information, please contact:

XEMICS SA

E-mail: info@xemics.com

Web: <http://www.xemics.com/>

Table of Contents

1	INTRODUCTION	3
2	HOW DOES THE EXAMPLE WORK?	3
3	INSTRUCTIONS ON HOW TO USE THE EEPROM_IIC SOFTWARE.....	4

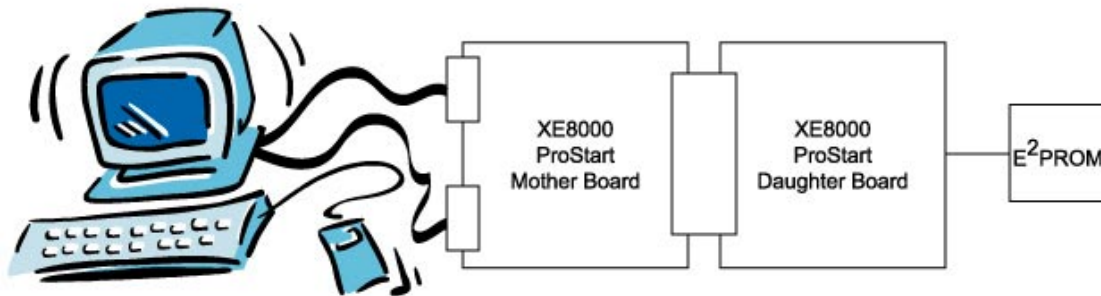
1 Introduction

This document shows how to connect and drive an I2C EEPROM with an XE8000 microcontroller.

The application reads information from an I2C EEPROM and sends it to the UART or gets data from the UART and sends it to the EEPROM.

2 How does the example work?

The software example implements an interface between a PC (Personal Computer) and an I2C EEPROM through any of the XE8000 family microcontrollers.



You can find the software example code in zip format on the XEMICS web site. The zip file is named eeProm_iic.zip.

In this example, the program starts by initialising two arrays with data. After this initialisation, it writes one of the arrays to the EEPROM. Then it reads the contents of the EEPROM to the other array and sends it to the PC. Once this initialisation is finished the software is waiting for a command from the PC.

The PC can send 3 types of commands to the microcontroller:

- Connection Test
- Write EEPROM
- Read EEPROM

The PC communicates with the microcontroller through the UART peripheral using a very simple protocol. Once the microcontroller has received all the data it sends them to the I2C EEPROM through the USRT peripheral.

The microcontroller software project is divided in two major parts.

1. PC communication handling.
 - The CommHandler.c file contains all the functions that are necessary to handle the protocol between the PC and the microcontroller.
 - The IRQHandler.c file contains the code that handles the interruptions from the UART (Rx IRQ and Tx IRQ).
 - The Globals.h file contains the variables type definitions and different other definitions for the protocol implementation.
2. I2C EEPROM communication handling
 - The eeProm_iic.c file contains the High-level functions for the I2C EEPROM protocol.
 - The LibUSRT816.s file contains the low-level functions for the I2C protocol using the USRT peripheral.
 - The LibUSRT816.h file is an interface between the C and the assembler.
 - The USRT.c file contains the high-level functions for the I2C EEPROM protocol. This high-level functions call the low-level functions defined in the LibUSRT816.s file.

- The USRTErr.c file contains the high-level protocol error handling.

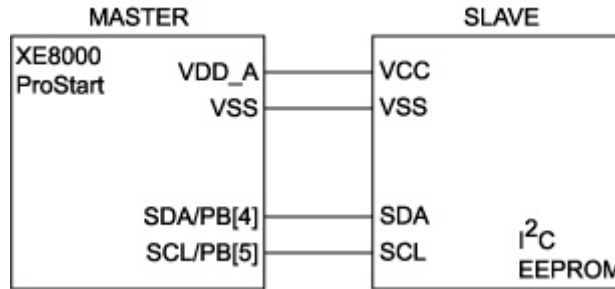
The Initialisation.c file is necessary to initialise the different microcontroller peripherals.

To communicate with the UART, one must set the RC Oscillator frequency with a precision better or equal to 5%. The DFLLDriver.c file does this.

The software starting point is located in the Main.c file.

3 Instructions on how to use the eeProm_iic software

1. Connect the I2C EEPROM to the ProStart according to the illustration.



2. Unzip the eeProm_iic.zip file to your hard disk.
3. Load the project eeProm_icc.crw in the CoolRIDE IDE.
4. Compile the Project
5. Upload the program to the ProStart.
6. **Attention:** You need to remove the jumpers for PortB [4] and PortB [5].
7. You need to have a RS232 Hexadecimal terminal. You can download a RS232 Hexadecimal terminal from the following address: <http://www.viddata.com/>. This program is a shareware.
8. Set the following parameters in you terminal:

Baud rate	: 19200
Bit length	: 8
Parity	: None
Stop bits	: 1
Handshaking	: none
9. You need to type the following commands (without spaces):


```

Connection Test : 24 43 10 04
uC Answer      : 24 61 10 04

EEPROM Read    : 24 52 10 04
uC Answer      : 24 XX XX XX XX XX XX XX XX XX XX 61 10 04

EEPROM Write   : 24 57 XX XX XX XX XX XX XX XX XX XX 10 04
uC Answer      : 24 61 10 04
      
```
10. Where XX are the contents of the EEPROM or the values that you want to write.

©XEMICS 2001

All rights are reserved. Reproduction whole or in part is prohibited without the prior written consent of the copyright owner. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.