# AN1477
# APPLICATION NOTE

# EMULATED DATA EEPROM WITH XFLASH MEMORY

**by CMG/MCD Application Laboratory**

## INTRODUCTION

When the data EEPROM is not available in a ST7 device, it can be emulated by the XFlash memory with some restrictions. This Application Note describes how to emulate this feature with a ST72F264 device and the restrictions this emulation implies.

Data EEPROM can be emulated in all XFlash devices (all Lite, ST72F344, ....).

For more information concerning ST7 programming, including a .zip file with the complete corresponding code for the ST72F264, visit our web site at www.st.com.

## 1 RESTRICTIONS

1 To guarantee that the XFlash program memory is write protected when programming the Emulated Data EEPROM, the whole program memory MUST be located in sector 0 (refer to the datasheet). This implies that:
   – The maximum program memory size is 4 Kbytes (sector 0 set by option byte to the maximum size which allows sector 1 availability).
   – IAP is not available for program memory.

2 During emulated data EEPROM programming, the XFlash can not be executed. This implies that:
   – The software which programs the emulated data EEPROM must be located in RAM. This software needs at least 16 bytes of RAM as shown in following program example.
   – The interrupts cannot be served during programming so they have to be masked.

## 2 PROCEDURE

To program 1 byte in the emulated data EEPROM (located in sector 1) the following steps have to be done:

1 Enter the XFlash RASS key to unlock the access to the FCSR register (only once, after reset for example).

2 Download the programming driver into RAM (from 0083h to 008Fh for example).

3 Write the data and address to be programmed in a RAM buffer (at RAM address 0080h to 0082h for example).

4 Call the downloaded RAM driver to program the emulated data EEPROM.

## 3 ASSEMBLER PROGRAM EXAMPLE

The following program example describes a driver routine to be called to emulate data EEPROM with an XFlash ST7 device. This example assumes that all restrictions are taken into account.

In this example, the program memory size is 4 Kbytes and the data EEPROM is also 4 Kbytes, but these sizes can be adjusted through option bytes depending on the needs.

```
st7/
;******************************************************************************
; TITLE:    XdataE2Emul.asm
; AUTHOR:   CMG_MCD Application Team
; DESCRIPTION: Data EEPROM emulation with XFlash memory (ST72F264 example)
;******************************************************************************

 TITLE "XdataE2Emul.asm"

BYTES
```

```
FCSR  EQU $72                    ; XFLASH Control/Status register definition
 #DEFINE LAT  1
 #DEFINE PGM  0

E2DATA EQU $80                   ; 1 byte: Emul. EEPROM Data to be programmed
E2ADDR EQU $81                   ; 2 bytes: Emul. EEPROM Address to be programmed

WORDS

SEGMENT byte at E000-EFFF 'XFlash Sect1 -Emul Data EEPROM'

SEGMENT byte at F000-FFFF 'XFlash Sect0 -Program'

; < RESET >

       LD A,#$56                 ; Enter RASS keys to unlock FCSR register
       LD FCSR,A
       LD A,#$AE
       LD FCSR,A

; < USER APPLICATION PROGRAM >

 CALL XemulE2_ByteProg

; < USER APPLICATION PROGRAM >

; ------------------------------------------------------------------------
; ROUTINE: XemulE2_ByteProg
; DESCRIPTION: Emulated data EEPROM byte programming driver routine
; BEFORE: A  = data to be programmmed
;     X:Y = address where it has to be programmed [E000h..EFFFh]
; AFTER:  Interrupts are disabled
;     The requested data byte is programmed
; RESSOURCES:
;     Program size:  40 bytes in sector 0
;     Used RAM area: 16 bytes from 0080h to 008Fh.
; ------------------------------------------------------------------------

.XemulE2_ByteProg
       LD   E2DATA,A             ; Data to be programmed (0080h) is in A
       LD   {E2ADDR},X           ; High address to be programmed (0081h) in X
       LD   {E2ADDR+1},Y         ; Low address to be programmed (0082h) in Y

       LD   X,#$0C               ; Copy programming software driver
```

```
.RAM_Copy                       ; into RAM from address 0083h
       LD  A,(RAM_Driver,X)
       LD  ($83,X),A
       DEC  X
       JRPL RAM_Copy

       SIM                      ; Disable interrupts
       JP $83                   ; Call the programming driver located in RAM

.RAM_Driver
       BSET FCSR,#LAT           ; Enable Emul. EEPROM latches
       LD  A, E2DATA
       LD  [E2ADDR.w],A         ; Set address/data to be programmed
       BSET FC SR,#PGM          ; Launch the Emul. EEPROM programming
.EEPROM_Prog
       BTJT FCSR,#PGM,EEPROM_Prog; Wait for end of programming (~5ms)
       RET

;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
       END
```

"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 CONNEXION WITH THEIR PRODUCTS."

littlediode
COMPONENTS FOR ALL

Littlediode supplies new, hard to find or obsolete electronic components and semiconductors all over the world.

With over two million different components listed you are sure to find the part you need.

Feel free to visit us today at our online store:

**LittleDiode.com**

Looking forward to providing you with the best possible service.