Application Note 28

Title: **How to use EM6x80 in Smart Power Mgmt Applications**

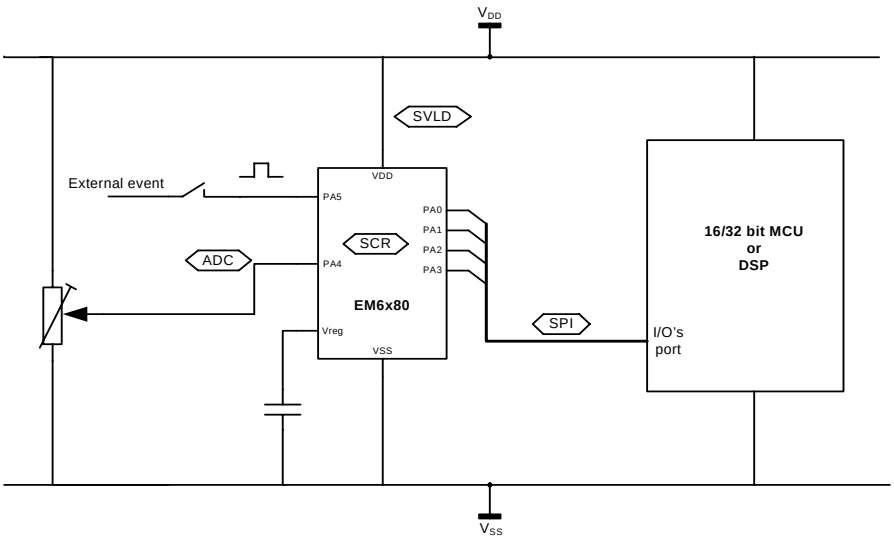
Product Family: **4-bit Microcontroller**

Part Number: EM6x80
Keywords: 4-bit microcontroller
Date: February 25, 2005

Introduction:

One of the multitude of applications for which the EM6x80 is eminently suited is as an intelligent power supply supervisor and power manager for another larger micro-controller. For this type of application the EM6x80 provides a one-component solution for power supply monitoring and management for the complete application.

Schematic example:



Application description:

During standard operation the EM6x80 is used as a voltage supervisor by using the SVLD function in continuous mode. When the voltage level falls below the minimum defined level the EM6x80 sends a sleep command to the application microcontroller. As soon as the voltage level recovers and becomes higher than the minimum defined level, the EM6x80 sends a wake-up command to restart the system.

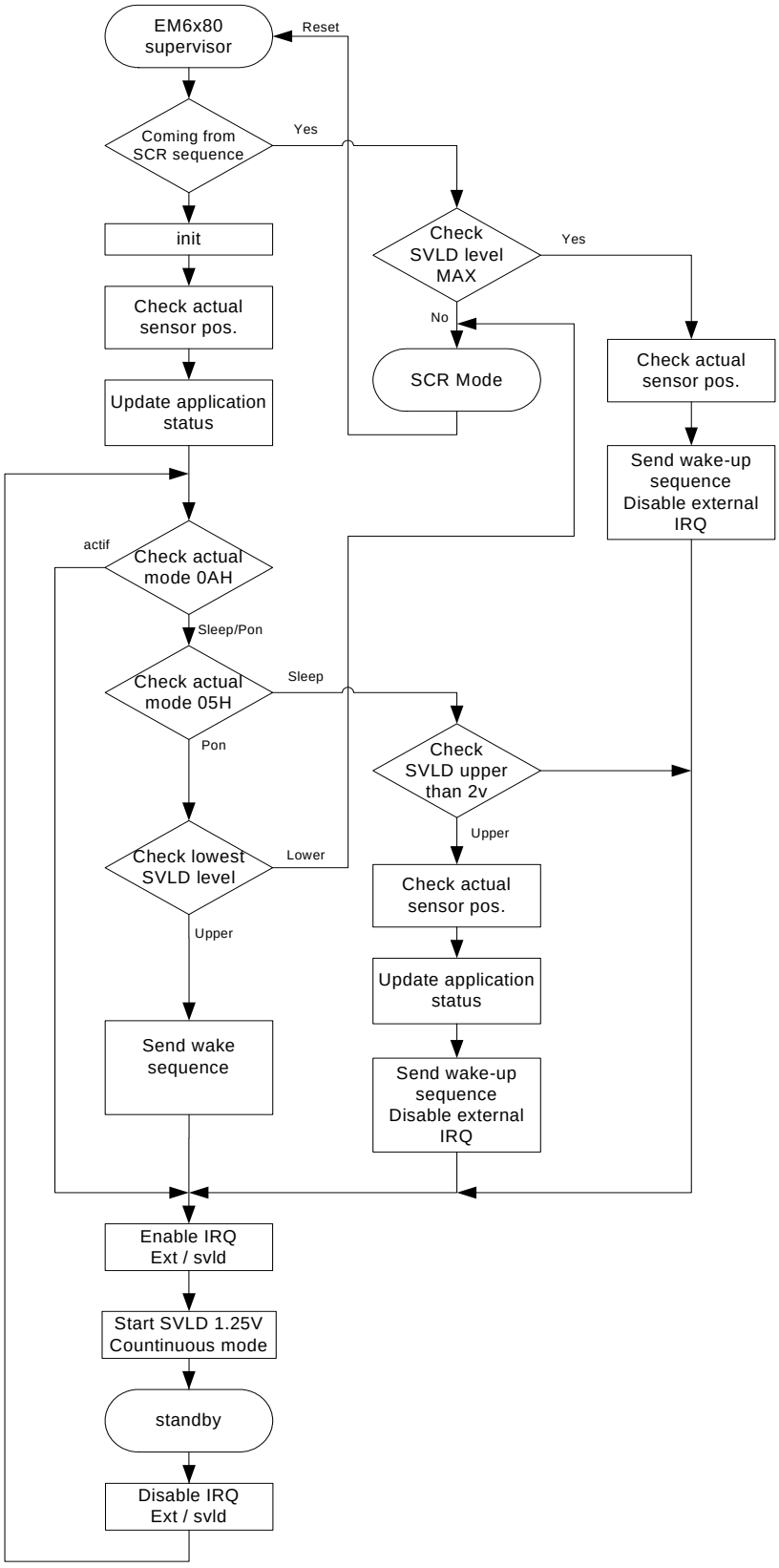
For applications where the activity is not continuous it is possible to put the application micro-controller and the EM6x80 in sleep mode for maximum power savings. As soon as activity is detected on the pin the EM6x80 is reactivated. Before activating the full system, the EM6x80 tests the system voltage and waits until it returns to an acceptable value if necessary.

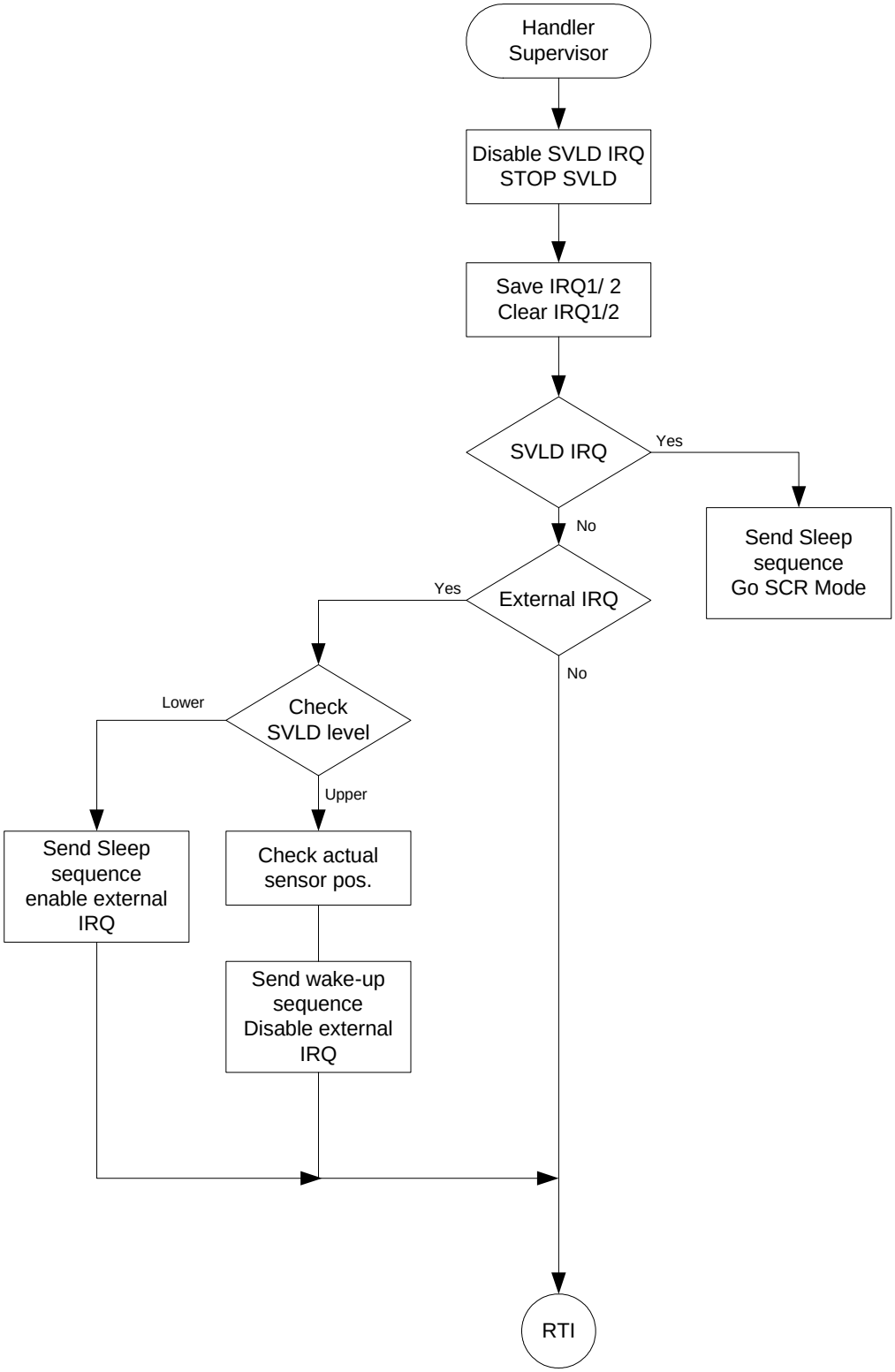
In this application we are using the standard serial bus for communication, but it is also possible to use a Manchester protocol or only 1 wire of the serial port leaving the other pins free for other uses.

The Sleep Counter Reset (SCR) is an EM patented feature available on the EM6x80, which can be used to save power when the battery supply is weak. The SCR enables a periodic wake-up of the EM6x80 even though it is in Sleep Mode and all the clocks are stopped. The SCR allows, for example, automatic turn-on of an application along with absolute minimum power consumption. When activated, the SCR provides a delay, which is programmable from 14ms to 8.14 seconds after which the EM6x80 is automatically reset. It is possible to determine if the reset was a power-on reset or a sleep counter reset by examination of the sleep bit in register RegSysCntl2.

In this application if the reset comes from the sleep counter, the SVLD is used to determine if the supply voltage level is now above the minimum defined level (for example 2V) before restarting the full system.

Software flowchart:





Software Code:

```
;-------------------------------------------------------------------------------
        INCLUDE V6680REG.ASM
;-------------------------------------------------------------------------------
        ORG 0
        JMP MAIN
;-------------------------------------------------------------------------------
;       Variables
;-------------------------------------------------------------------------------
        INCLUDE 6680_RAM_MAP.asm
;-------------------------------------------------------------------------------
; Interrupt Handler:
;-------------------------------------------------------------------------------
Handler:
        STA     STACK0          ; Save ACCU
        LDI     0DH
        AND     RegIRQMask2
        STA     RegIRQMask2     ; disable SVLD IRQ
        STI     RegVLDCntl, 01H; Disable SVLD countinuous mode
        LDR     RegIRQ1                 ; Save IRQ flags
        STA     V_IRQ_1                 ; Save IRQ flags
        LDR     RegIRQ2                 ; Save IRQ flags
        STA     V_IRQ_2                 ; Save IRQ flags
SVLD_Check:
        LDI     02H
        XOR     V_IRQ_2
        JPNZ    External_IRQ
        STI     RegPACntl3, 0CH
        STI     RegPACntl4, 02H
        STI     RegPA0OE, 06H           ; Set PA2,1 as output
        STI     RegSCntl1, 0CH          ; master Mode
        LDR     V_ADC_LEVEL
        STA     RegSDataL
        STI     RegSDataH, 05H          ; Sleep CMD
        STI     RegIRQMask2,00H                 ; disable SPI IRQ
        STI     RegSysCNTL1, 09H        ; Enable General IRQ and Remove Test Mode
        STI     RegSCntl2, 08H          ; Start SPI interface in slave mode
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        STI     RegSysCntl2,04H         ;Enable sleep mode
        LDI     03H
        OR      RegSleepCR
        STA     RegSleepCR              ; Select maximum length of sleep time
        STI     RegSysCntl1, 05H        ; Go in Sleep
External_IRQ:
        SHRR    V_IRQ_1
        JPC     SVLD_TEST_HANDLER

HandEnd:
        LDR     STACK0                  ; reload ACCU
        RTI
```

```
SVLD_TEST_HANDLER:
        STI     RegSVLDLev, 04H                 ; SELECT 1.40V SVLD level
Start_SVLD_HANDLER:
        STI     RegVLDCNTL, 04H         ; Start one SVLD mesure
check_busy_bit_SVLD_HANDLER:
        NOP
        LDI     04H
        AND     RegVldCntl
        JPNZ    check_busy_bit_SVLD_HANDLER
        SHLR    RegVldCntl
        JPC     SVLD_UPPER_HANDLER
SVLD_LOWER_HANDLER:
        STI     V_Master_CMD, 00H
        STI     RegPACntl3, 0CH
        STI     RegPACntl4, 02H
        STI     RegPA0OE, 06H           ; Set PA2,1 as output
        STI     RegSCntl1, 0CH          ; master Mode
        LDR     V_ADC_LEVEL
        STA     RegSDataL
        STI     RegSDataH, 05H          ; Sleep CMD is 05H
        STI     RegIRQMask2,00H                 ; enable SPI IRQ
        STI     RegSysCNTL1, 09H        ; Enable General IRQ and Remove Test Mode
        STI     RegSCntl2, 08H          ; Start SPI interface in slave mode
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        JMP     HandEnd

SVLD_UPPER_HANDLER:
        STI     RegSleepCR, 08H                 ; Remove Pull on PA4 pin
        STI     RegSVLDLev, 0FH                 ; SELECT UPPER ADC level
Start_ADC_HANDLER:
        STI     RegVLDCNTL, 0DH         ; Start ADC mesure on PA4
check_busy_bit_HANDLER:
        NOP
        LDI     04H
        AND     RegVldCntl
        JPNZ    check_busy_bit_HANDLER
End_ADC_measure_HANDLER:
        LDI     08H
        AND     RegVldCntl
        JPNZ    END_ADC_HANDLER
DEC_ADC_LEVEL_HANDLER:
        DEC     RegSVLDLev
        JPZ     SCR_mode
        STA     RegSVLDLEV
        JMP     Start_ADC_HANDLER


END_ADC_HANDLER:
        LDR     RegSVLDLev
        STA     V_ADC_LEVEL             ;information for the master MCU
        STI     V_Master_CMD, 00H
        STI     RegPACntl3, 0CH
        STI     RegPACntl4, 02H
```

```
        STI     RegPA0OE, 06H       ; Set PA2,1 as output
        STI     RegSCntl1, 0CH      ; master Mode
        LDR     V_ADC_LEVEL
        STA     RegSDataL
        STI     RegSDataH, 0AH      ; Wake-up CMD is 0AH
        STI     RegIRQMask2,00H                 ; enable SPI IRQ
        STI     RegSysCNTL1, 09H    ; Enable General IRQ and Remove Test Mode
        STI     RegSCntl2, 08H      ; Start SPI interface in slave mode
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        JMP     HandEnd
MAIN:



        LDI     04H
        AND     RegSysCntl2
        JPNZ    SCR_return
init:
        STI     RegPACntl3, 00H
        STI     RegPACntl4, 00H
ADC_measure:
        STI     RegSleepCR, 08H             ; Remove Pull on PA4 pin
        STI     RegSVLDLev, 0FH             ; SELECT UPPER ADC level
Start_ADC:
        STI     RegVLDCNTL, 0DH     ; Start ADC mesure on PA4
check_busy_bit:
        NOP
        LDI     04H
        AND     RegVldCntl
        JPNZ    check_busy_bit
End_ADC_measure:
        LDI     08H
        AND     RegVldCntl
        JPNZ    END_ADC
DEC_ADC_LEVEL:
        DEC     RegSVLDLev
        JPZ     SCR_mode
        STA     RegSVLDLEV
        JMP     Start_ADC
END_ADC:
        LDR     RegSVLDLev
        STA     V_ADC_LEVEL         ;information for the master MCU
        STI     RegPACntl3, 0CH
        STI     RegPACntl4, 02H
        STI     RegPA0OE, 06H       ; Set PA2,1 as output
        STI     RegSCntl1, 0CH      ; master Mode
        LDR     V_ADC_LEVEL
        STA     RegSDataL
        STI     RegSDataH,00H       ; check CMD only
        LDI     01H
        OR      RegIRQMask2
```

```
        STA     RegIRQMask2                     ; enable SPI IRQ
        STI     RegSysCNTL1, 09H        ; Enable General IRQ and Remove Test Mode
        STI     RegSCntl2, 08H          ; Start SPI interface in slave mode
        halt
Test_Actual_mode:
        LDI     0AH
        XOR     RegSDataH
        JPZ     Standby
PON:
SVLD_measure:
        STI     RegSVLDLev, 0AH                 ; SELECT 2V SVLD level
Start_SVLD:
        STI     RegVLDCNTL, 05H         ; Start one SVLD mesure
check_busy_bit_SVLD:
        NOP
        LDI     04H
        AND     RegVldCntl
        JPNZ    check_busy_bit_SVLD
End_SVLD_measure:
        LDI     08H
        AND     RegVldCntl
        JPNZ    Send_Wake_up
Ckeck_lowest_SVLD:
        STI     RegSVLDLev, 05H                 ; SELECT 1.25V SVLD level
Start_SVLD_lower:
        STI     RegVLDCNTL, 05H         ; Start one SVLD mesure
check_busy_bit_SVLD_lowest:
        NOP
        LDI     04H
        AND     RegVldCntl
        JPNZ    check_busy_bit_SVLD_lowest
End_SVLD_Lowest_measure:
        LDI     08H
        AND     RegVldCntl
        JPNZ    SCR_MODE


        Call    Sleep_sequence
Standby:


Enable_IRQ:
        STI     RegIRQMask2, 02H        ; Enable SVLD IRQ
        STI     RegPACntl2, 01H
        STI     RegIRQMask1, 01H        ; Enable PA5 IRQ for External Event
        STI     RegSVLDLev, 05H                 ; SELECT 1.25V SVLD level
        STI     RegVLDCntl, 07H         ; Enable SVLD countinuous mode
        Halt
        STI     RegVLDCntl, 00H         ; Disable SVLD countinuous mode
        STI     RegIRQMask2, 00H        ; Disable SVLD IRQ
        STI     RegIRQMask1, 00H        ; Disable PA5 IRQ for External Event
        JMP     Test_Actual_mode
SCR_mode:
        STI     RegPACntl3, 0CH
        STI     RegPACntl4, 02H
        STI     RegPA0OE, 06H           ; Set PA2,1 as output
        STI     RegSCntl1, 0CH          ; master Mode
        LDR     V_ADC_LEVEL
        STA     RegSDataL
        STI     RegSDataH, 05H          ; Sleep CMD
        LDI     01H
        OR      RegIRQMask2
        STA     RegIRQMask2                     ; enable SPI IRQ
        STI     RegSysCNTL1, 09H        ; Enable General IRQ and Remove Test Mode
        STI     RegSCntl2, 08H          ; Start SPI interface in slave mode
        halt
        STI     RegSysCntl2,04H         ;Enable sleep mode
        LDI     03H
```

```
        OR      RegSleepCR
        STA     RegSleepCR          ; Select maximum length of sleep time
        STI     RegSysCntl1, 05H    ; Go in Sleep

Send_Wake_up:
        STI     V_Master_CMD, 00H
        STI     RegPACntl3, 0CH
        STI     RegPACntl4, 02H
        STI     RegPA0OE, 06H       ; Set PA2,1 as output
        STI     RegSCntl1, 0CH      ; master Mode
        LDR     V_ADC_LEVEL
        STA     RegSDataL
        STI     RegSDataH, 0AH      ; Wake-up CMD is 0AH
        LDI     01H
        OR      RegIRQMask2
        STA     RegIRQMask2         ; enable SPI IRQ
        STI     RegSysCNTL1, 09H    ; Enable General IRQ and Remove Test Mode
        STI     RegSCntl2, 08H      ; Start SPI interface in slave mode
        halt
        JMP     Standby

Sleep_sequence:
        STI     V_Master_CMD, 00H
        STI     RegPACntl3, 0CH
        STI     RegPACntl4, 02H
        STI     RegPA0OE, 06H       ; Set PA2,1 as output
        STI     RegSCntl1, 0CH      ; master Mode
        LDR     V_ADC_LEVEL
        STA     RegSDataL
        STI     RegSDataH, 05H      ; Sleep CMD is 05H
        LDI     01H
        OR      RegIRQMask2
        STA     RegIRQMask2                 ; enable SPI IRQ
        STI     RegSysCNTL1, 09H    ; Enable General IRQ and Remove Test Mode
        STI     RegSCntl2, 08H      ; Start SPI interface in slave mode
        halt
        RET
Wake_up_sequence:
        STI     V_Master_CMD, 00H
        STI     RegPACntl3, 0CH
        STI     RegPACntl4, 02H
        STI     RegPA0OE, 06H       ; Set PA2,1 as output
        STI     RegSCntl1, 0CH      ; master Mode
        LDR     V_ADC_LEVEL
        STA     RegSDataL
        STI     RegSDataH, 0AH      ; Wake-up CMD is 0AH
        LDI     01H
        OR      RegIRQMask2
        STA     RegIRQMask2                 ; enable SPI IRQ
        STI     RegSysCNTL1, 09H    ; Enable General IRQ and Remove Test Mode
        STI     RegSCntl2, 08H      ; Start SPI interface in slave mode
        halt
        STI     RegIRQMask2,00H

        RET


SCR_return:
        STI     RegSVLDLev, 0FH             ; SELECT 2.75V SVLD level maximum
Start_SVLD_max:
        STI     RegVLDCNTL, 05H     ; Start one SVLD mesure
check_busy_bit_max:
        NOP
        LDI     04H
        AND     RegVldCntl
        JPNZ    check_busy_bit_max
```

```
End_SVLD_max_measure:
        LDI     08H
        AND     RegVldCntl
        JPNZ    Active_mode
        JMP     SCR_mode
Active_mode:
        STI     RegSleepCR, 08H                 ; Remove Pull on PA4 pin
        STI     RegSVLDLev, 0FH                 ; SELECT UPPER ADC level
Start_ADC_SCR:
        STI     RegVLDCNTL, 0DH         ; Start ADC mesure on PA4
check_busy_bit_SCR:
        NOP
        LDI     04H
        AND     RegVldCntl
        JPNZ    check_busy_bit_SCR
End_ADC_measure_SCR:
        LDI     08H
        AND     RegVldCntl
        JPNZ    END_ADC_SCR
DEC_ADC_LEVEL_SCR:
        DEC     RegSVLDLev
        JPZ     SCR_mode
        STA     RegSVLDLEV
        JMP     Start_ADC_SCR
END_ADC_SCR:
        LDR     RegSVLDLev
        STA     V_ADC_LEVEL             ;information for the master MCU
        STI     RegVLDCNTL, 01H         ; Stop ADC mesure on PA4
        CALL    Wake_up_sequence
        JMP     Standby
```

Include File V6680REG.ASM :

```
;-------------------------------------------------------------------------------------------
;       EM6680 RAM Map File definition
;       28/01/05
;       Main Program AN28.asm

;       All RAM address are define on this file.
;-------------------------------------------------------------------------------------------
Stack0          EQU     00H
ADDR            EQU     01H     ; selection value on all program
V_SVLD_LEVEL    EQU     02H
V_ADC_LEVEL     EQU     03H     ; Used to send the ADC value to the master
V_Master_CMD    EQU     04H     ; Used to send commande to the master

V_IRQ_1         EQU     1BH     ; Used in 6680_2.asm
V_IRQ_2                 EQU     1CH     ; Used in 6680_2.asm
```