Application Note 18

Title: **Ohmmeter application with EM6X20**

Product Family: **4 bits Microcontroller**

Part Number: EM6620, EM6520
Keywords: 4 bits micro-controller, ROM, RAM, data table
Date: January 07, 2002 REV. B/416

This application note describes a method for implementing an ohmmeter using the EM662X series of micro-controllers. This demo software is created for the EM6X20 which have a LCD driver inside. The ohmmeter requires only two external components. It's software and hardware configurable for resistance measurement with resolutions until 16 bits. This method uses a software calibration technique that compensates for voltage, time, and temperature.

Theory of operation

This application uses a capacity charging circuit (figure 1) to convert resistance to time, which can be easily measured using a micro-controller.
The supply voltage is applied to a calibration resistor. The capacitor is charged up until the threshold on the chip input trips. This generates a software calibration value that is used to calibrate out most circuit errors changes in the input threshold voltage and temperature variations. After the capacitor is discharged, the reference voltage is applied to the resistance to be measured. The time to trip the threshold is then measured and compared to the calibration value to determine the actual resistance.

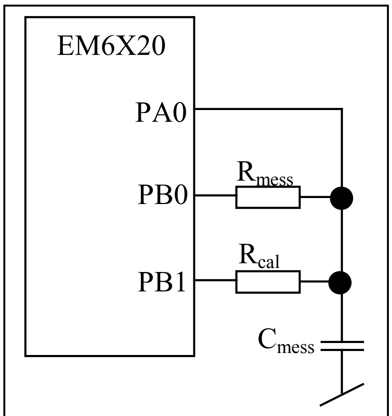

Figure 1.

Circuit configuration

The values of R_{cal} and C_{mess} are selected based upon the number of bits of resolution required. R_{cal} should be one half the largest value resistance to be measured:

$$C_{mess} = \frac{-T}{Ln\left(1-\frac{V_T}{V_{dd}}\right)\cdot R_{\max}} = \frac{-\left(65535\cdot\frac{1}{16384Hz}\right)}{Ln\left(1-\frac{1,5V}{3,0V}\right)\cdot 500k\Omega} = 11,54\mu F \Rightarrow 4,7\mu F$$

T = Time to do the number of bits resolution desired (resolution * frequency counter = FFFF * 16kHz).
V_{dd} = Supply voltage (3,0V).
V_T = Threshold voltage of the EM6X20 (1,5V).
R_{max} = Maximum resistance value to be measured (500kΩ).

Actual value for the capacitor should be smaller than calculated to ensure that the EM6x20 does not have a overflow during the measurement.

Example's program

The assembly code implementing for the ohmmeter example (figure 1) is at the Appendix A. This program measures time up to 16 bits (65535 cycles) and calculates the results using 16 bits multiply and divide modules.
The measured performance reach on the ohmmeter is ±1% between 1-500kΩ. If the measurement's zone is smaller, you can have better performance.

Flowchart

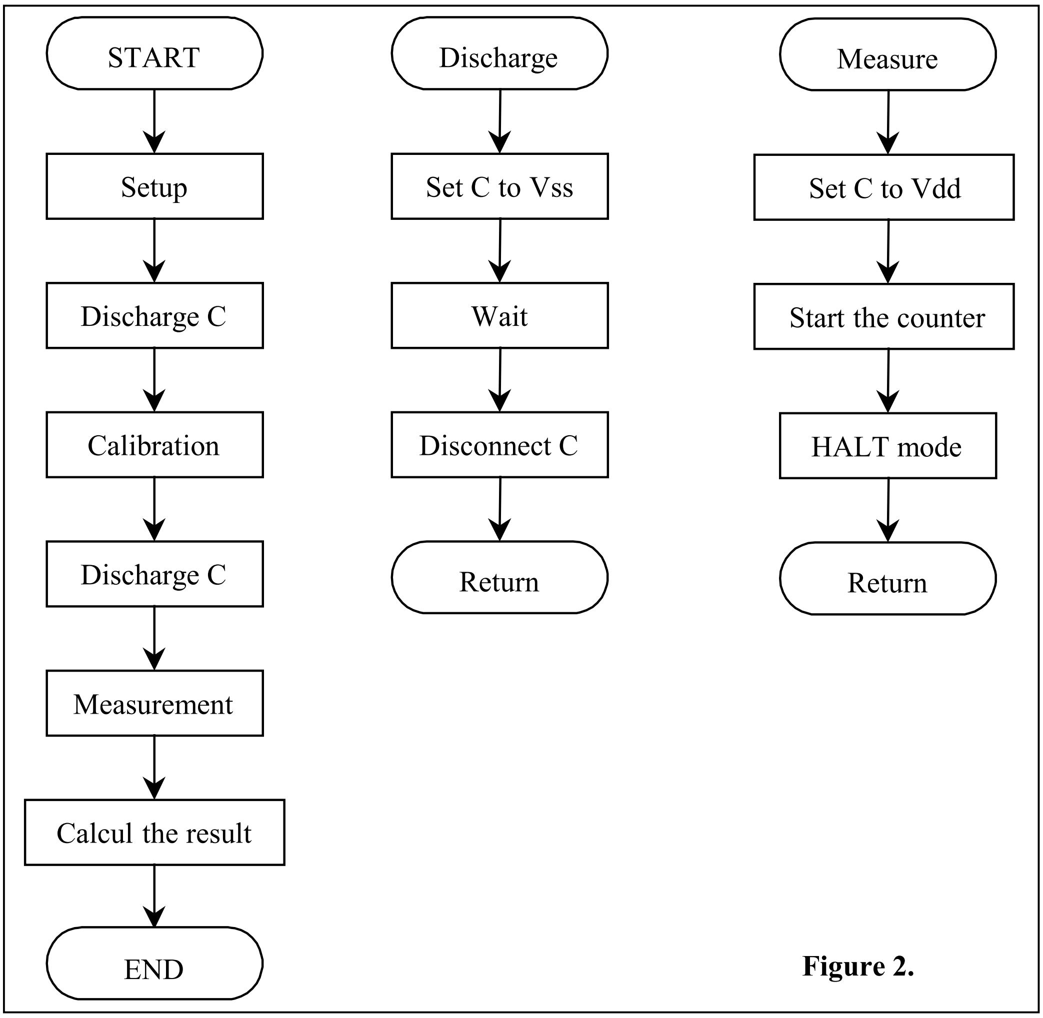

Figure 2.

Appendix A: Ohmmeter program

```
;-----------------------------------------------------------------------------------------------------
;        PROGRAM          :       OM_D01.ASM
;        Date last mod    :       22/12/1998              Ch.Mayer
;        Ohmmeter by the measurement of resistors
;        Uses the V6620REG.ASM Include file
;
;
;        Between PB0 and PA0 it's the measurement's resistor.
;        Between PB1 and PA0 it's the calibration's resistor.
;        Between Vss and PA0 it's the condensator.
;
;-----------------------------------------------------------------------------------------------------
;------------------------------------------------------------------------------------------
;     6520 registers
;------------------------------------------------------------------------------------------

         INCLUDE V6620REG.ASM

;------------------------------------------------------------------------------------------
;     Constants
;------------------------------------------------------------------------------------------
RCal1          EQU    06H                  ; Calibration resistor (LSB)
RCal2          EQU    0CH                  ; Calibration resistor
RCal3          EQU    00H                  ; Calibration resistor
RCal4          EQU    00H                  ; Calibration resistor (MSB)

;------------------------------------------------------------------------------------------
;     Variables
;------------------------------------------------------------------------------------------
Stack0         EQU    00H                  ; Stack0
Errors         EQU    01H                  ; Save if a error is present.
Cnt_Ext1       EQU    02H                  ; Counter extention n°1
Cnt_Ext2       EQU    03H                  ; Counter extention n°2
Sel_Meas       EQU    04H                  ; Select the resistor for the measurement
OP1_1          EQU    05H                  ; First number for the calcul (LSB)
OP1_2          EQU    06H                  ; First number for the calcul
OP1_3          EQU    07H                  ; First number for the calcul
OP1_4          EQU    08H                  ; First number for the calcul (MSB)
OP2_1          EQU    09H                  ; Second number for the calcul (LSB)
OP2_2          EQU    0AH                  ; Second number for the calcul
OP2_3          EQU    0BH                  ; Second number for the calcul
OP2_4          EQU    0CH                  ; Second number for the calcul (MSB)
ComptL         EQU    0DH                  ; Count the position of the division (LSB)
Compt          EQU    0DH                  ; Count the position of the multiplication
Carry          EQU    0EH                  ; Save the carry in a calcul, first memory
Carry2         EQU    0FH                  ; Save the carry in a calcul, second memory
Res1           EQU    10H                  ; Result of the calcul (LSB)
Res2           EQU    11H                  ; Result of the calcul
Res3           EQU    12H                  ; Result of the calcul
Res4           EQU    13H                  ; Result of the calcul
Res5           EQU    14H                  ; Result of the calcul
Rst1           EQU    14H                  ; Rest from the division (LSB)
Res6           EQU    15H                  ; Result of the calcul
Rst2           EQU    15H                  ; Rest from the division
Res7           EQU    16H                  ; Result of the calcul
Rst3           EQU    16H                  ; Rest from the division
Res8           EQU    17H                  ; Result of the calcul (MSB)
Rst4           EQU    17H                  ; Rest from the division (MSB)
```

```
ComptH      EQU   18H          ; Count the position of the division (MSB)
Stack1      EQU   19H          ; Tempory variable
Stack2      EQU   1AH          ; Tempory variable
Stack3      EQU   1BH          ; Tempory variable
Stack4      EQU   1CH          ; Tempory variable
                               ; from 20H to 29H and 2EH-2FH is used for setup the display.
TCal1       EQU   2AH          ; Time's calibration resistor (LSB)
TCal2       EQU   2BH          ; Time's calibration resistor
TCal3       EQU   2CH          ; Time's calibration resistor
TCal4       EQU   2DH          ; Time's calibration resistor (MSB)
                               ; from 30H to 39H and 3EH-3FH is used for setup the display.
Num1_1      EQU   3AH          ; Number to display on the lcd (LSB)
Num1_2      EQU   3BH          ; Number to display on the lcd
Num1_3      EQU   3CH          ; Number to display on the lcd
Num1_4      EQU   3DH          ; Number to display on the lcd (MSB)

;-------------------------------------------------------------------------------------------
;     Code Offset
;-------------------------------------------------------------------------------------------

        ORG 0

Reset:  JMP Main

;-------------------------------------------------------------------------------------------
;         Interrupt Handler
;-------------------------------------------------------------------------------------------
Handler:
        STA     STACK0            ; Save ACCU value

        LDI     01H               ; \
        AND     RegIRQ1           ;  => Test if it's a interrupt PA0
        JPZ     Hand10            ; /
        STI     RegCCnt2, 00H     ; Stop the 10Bit counter

Hand10: LDI     02H               ; \
        AND     RegIRQ3           ;  => Test if it's a interrupt counter
        JPZ     HandEnd           ; /

        LDI     04H
        ADD     Cnt_Ext1          ; Increment the counter extention n°1
        STA     Cnt_Ext1          ; Save the new value
        JPNC    HandEnd

        INC     Cnt_Ext2          ; Increment the counter extention n°2
        STA     Cnt_Ext2          ; Save the new value
        JPNC    HandEnd

        STI     RegCCnt2, 00H     ; Stop the 10 bits counter
        STI     Errors, 01H       ; Set error number

HandEnd:
        LDR     STACK0            ; Reload ACCU
        RTI
```



```
;-------------------------------------------------------------------------------------------
;       Main
;-------------------------------------------------------------------------------------------
Main:
        STI     RegVLDCntl, 03H         ; Disable the watchdog
        STI     INTEN, 08H              ; Enable the general interrupt
        STI     IRQM3, 0010b            ; Enable interrupt counter
        STI     RegLCD1, 0000b          ; Setup the lcd
        STI     RegLCD2, 0010b          ; Turn LCD on 4 MUX
        STI     OPTNoPullPA, 01H        ; Disable the pull on PA0
        STI     OPTDebIntPA, 01H        ; Disable the debouncer on PA0
        STI     OPTNoPdPB, 0011b        ; Disable the pull on port B
        STI     RegPBCntl, 1111b        ; Set port B to output
        STI     RegPBData, 0000b        ; PB0 must be low to discharge resistor
        CALL    DisClear                ; Clear the display
        CALL    DisInit                 ; Init. the display.


Main100:
        CALL    Discharge               ; Discharge the capacitor
        STI     Sel_Meas, 02H           ; Select the measure of the calibration resistor
        CALL    Meas_Res                ; Start the measurement

        CALL    Discharge               ; Discharge the capacitor
        STI     Sel_Meas, 01H           ; Select the measure of the resistor
        CALL    Meas_Res                ; Start the measurement

        LDR     Errors                  ; \
        JPNZ    Mea_Error               ;  => on a error, jump to the error's subroutine

        LDR     Res1                    ; Save the result for display on the lcd
        STA     Num1_1
        LDR     Res2
        STA     Num1_2
        LDR     Res3
        STA     Num1_3

        CALL    Display                 ; Refresh the lcd

        JMP     Main100

;--------------------------------------------------------------------------
;       Errors
;--------------------------------------------------------------------------
Mea_Error:
        STI     Num1_1, 0EH             ; Write "EEE" on the display on a error.
        STI     Num1_2, 0EH
        STI     Num1_3, 0EH

        CALL    Display                 ; Refresh the lcd

        JMP     Main100
;--------------------------------------------------------------------------
;       Measurement resistor
;--------------------------------------------------------------------------
Meas_Res:
        STI     Errors, 00H             ; Clr error
        STI     Cnt_Ext1, 03H           ; Reset the variable
        STI     Cnt_Ext2, 00H           ; Reset the variable
```

```
         STI      RegCCnt2,0000b         ; Stop the 10 bit counter
         STI      RegCCnt1,1001b         ; Count up, System clk/2
         STI      RegCDataL, 00H         ; Start with 00H
         STI      RegCDataM, 00H         ; Start with 00H
         STI      RegCDataH, 00H         ; Start with 00H
         STI      RegCCnt2, 0001b        ; Load counter

         LDR      Sel_Meas               ; \
         STA      RegPBCntl              ;  => Set PB0 to 1 for start the measurement
         STA      PortB                  ; /

         STI      RegCCnt2,1000b         ; Start the 10 Bit counter

         STI      IRQM1, 0001b           ; PA[0] interrupt enable

Meas_Res100:
         HALT                            ; Only PA0 rising edge or 10 Bit overflow will wake up

         LDI      08H                    ; Check if timer is running
         AND      RegCCnt2               ; <> 0 if timer running
         JPNZ     Meas_Res100

         STI      IRQM1, 0000b           ; PA0 interrupt disabled

         LDR      Errors                 ; Test if the counter is in error mode
         JPNZ     Meas_Res400

         SHRR     Sel_Meas               ; What the resistor measured ? ...
         JPC      Meas_Res200            ; ... it's the normal resistor
         SHRA
         JPC      Meas_Res300            ; ... it's the calibration resistor
         JMP      Meas_Res400

Meas_Res200:                             ; Result from the normal resistor
         LDR      RegCDataL              ; Transfer Tmeas in OP1_x for the multiplication
         STA      OP1_1
         LDR      RegCDataM
         STA      OP1_2
         LDI      03H
         AND      RegCDataH
         STA      OP1_3
         LDI      0CH
         AND      Cnt_Ext1
         OR       OP1_3
         STA      OP1_3
         LDR      Cnt_Ext2
         STA      OP1_4

         LDI      RCal1                  ; Transfer Rcal in OP2_x for the multiplication
         STA      OP2_1
         LDI      RCal2
         STA      OP2_2
         LDI      RCal3
         STA      OP2_3
         LDI      RCal4
         STA      OP2_4

         CALL     Multi16                ; Multiplication

         LDR      TCal1                  ; Transfer Tcal in OP2_x for the division
```



```
        STA     OP2_1
        LDR     TCal2
        STA     OP2_2
        LDR     TCal3
        STA     OP2_3
        LDR     TCal4
        STA     OP2_4

        CALL    Divi16              ; Division

        LDR     Res1
        STA     OP1_1
        LDR     Res2
        STA     OP1_2
        LDR     Res3
        STA     OP1_3

        CALL    HEXDEC              ; Convert the result from hexa to decimal

        JMP     Meas_Res400

Meas_Res300:                        ; Result from the calibration resistor
        LDR     RegCDataL
        STA     TCal1
        LDR     RegCDataM
        STA     TCal2
        LDI     03H
        AND     RegCDataH
        STA     TCal3
        LDI     0CH
        AND     Cnt_Ext1
        OR      TCal3
        STA     TCal3
        LDR     Cnt_Ext2
        STA     TCal4

Meas_Res400:
        STI     RegPBCntl, 0FH      ;  => Reset PB0 and PB1 to GND
        STI     PortB, 00H          ; /

        RET



;-----------------------------------------------------------------------
;       Discharge the capacitor
;-----------------------------------------------------------------------
Discharge:
        STI     Cnt_Ext1, 03H       ; Reset the variable
        STI     Cnt_Ext2, 00H       ; Reset the variable

        STI     RegCCnt2, 0000b     ; Stop the 10 bits counter
        STI     RegCCnt1, 1010b     ; Count up, system clk/16
        STI     RegCDataL, 00H      ; Start with 00H
        STI     RegCDataM, 00H      ; Start with 00H
        STI     RegCDataH, 00H      ; Start with 00H
        STI     RegCCnt2, 0001b     ; Load counter

        STI     RegCCnt2, 1000b     ; Start the 10 bits counter
```



```
Disch100:
        HALT                            ; Only 10 bits overflow will wake up

        LDI     01H                     ; \
        SUB     Cnt_Ext2                ;  => Test if "Cnt_Ext2" is < 01H
        JPNC    Disch100                ; /

        STI     RegCCnt2, 0000b         ; Stop the 10 bit counter
        RET

;-------------------------------------------------------------------------
;       Include File
;-------------------------------------------------------------------------

        INCLUDE MULTI16.ASM             ; Include file for multiplication 16 bits.
        INCLUDE DIVIS16.ASM             ; Include file for division 16 bits.
        INCLUDE HEXDEC.ASM              ; Include file for convert hex to dec
        INCLUDE LCDINIT.ASM             ; Include file for drive the lcd.

;-----------------------------------------------------------------------------------------------
        END
;-----------------------------------------------------------------------------------------------
```

You can download the latest version of the source code on our web site (http://www.emmarin.com/).

References

Application Note #13: 16 bit binary division with 4 bit controller.
Application Note #14: 16 bit binary multiplication with 4 bit controller.
Application Note #20: How to convert Hex – Dec.

