Application Note

Using a PWM Timer as the clock input for the UART of the MB89630 series

Vers. 1.0 by E. Bendels

This Application Note gives a short example on how to calculate timer settings for configuring the PWM-Timer so that the UART clock of the MB89630 can be provided by the timer.

Background:

The UART block of the MB89630 series controller allows to select its required clock signal from 4 different sources.
The 1st source is a dedicated baud-rate generator belonging to the UART block, which is most suitable if the controller operates with a 10 MHz main clock to derive popular baud rates of 9600-, 4800, 2400- etc. baud.
If the main-clock is chosen to be different, its most likely that the desired baud rate can not be generated by this dedicated generator.
In this case, the UART can receive its clock signal from either PWM-Timer 1 or 2, which can be programmed to produce the required clock signal, or finally from an external clock input.

Following, this relationship between main-clock, PWM-timer setting and baud rate will be discussed.

It should be noticed, that if a PWM Timer is used for the UART clock generation, the PWM-Timer must be configured to operate as a simple timer and not to generate a PWM-signal, so the name ´PWM-Timer´ might be confusing.

Formulas:

The output clock frequency of the PWM timer can be calculated according to the following formula:

$$f_{PWM} = \frac{f_{MAIN}}{8 * SC * (COMR + 1)}$$

f_{PWM} : PWM-Timer Output Frequency
f_{MAIN} : Main Clock Frequency
SC : PWM-Timer Input Clock Selector
can be set to {1 | 8 | 16 | 64}
COMR : Timer Value
can be set to 0..256

The UART baud rate (assuming it is driven by the PWM timer) is given by :

$$BR = \frac{f_{PWM}}{CR}$$

CR : UART Clock Division Ratio
can be set to {16 | 64}

If we rearrange the equations to derive COMR we get:

$$COMR = \frac{f_{MAIN}}{BR * 8 * SC * CR} - 1$$

Example 1:

Lets assume:	Main Clock	f_{MAIN} = 10 MHz
	PWM-Timer Selector	SC = 1
	UART Clock Ratio	CR = 16
	and the Baud Rate shall be	BR = 9600

COMR = 7.138

Thus, the PWM timer would have to be set to the integer value 7.
If we recalculate the baud rate with this timer value the real baud rate would be 9765.625 baud, which results in a 1.7 % deviation.

The equation can also be used to calculate main-clock frequencies which allow to realize precise baud-rates. The following table lists some of these frequencies to realize a baud rate of 9600 baud.

Main-Clock Frequency	COMR-Setting
(1,2288 MHz)	1
2,4576 MHz	1
3,6864 MHz	2
4,9152 MHz	3
6,144 MHz	4
7,3728 MHz	5
(8,6016 MHz)	6
9,8304 MHz	7
11,0592 MHz	8

```
$lo cy xr cp
;/*+-----------------------------------------------------------------------+*/
;/*ª                          F U J I T S U                                ª*/
;/*ª                                                                       ª*/
;/*ª              M i k r o e l e k t r o n i k   G m b H                  ª*/
;/*ª                                                                       ª*/
;/*ª                                                                       ª*/
;/*ª  Filename:       UT.asm                                             ª*/
;/*ª  Description:    "MB89630 UART Test Program"                      ª*/
;/*ª  Series:         MB89630                                          ª*/
;/*ª  Version:        V01.00                                               ª*/
;/*ª  Design:         Edmund Bendels 22.08.94                              ª*/
;/*ª  Change:                                                              ª*/
;/*ª                                                                       ª*/
;/*+-----------------------------------------------------------------------+*/

        NAME   "FUART"                ; module name

 &SET DebugFunctions 0                ; Flag: Include Debug Functions
 &INCLUDE "c:\FJ_8L\Include\eBIOS.inc"

;-----------------------------------------
;-----------------------------------------
PDR1    EQU  002h
DDR1    EQU  003h

SYCC    EQU  0007h                    ; System Clock Control

PDR4    EQU  00Fh
DDR4    EQU  010h

CNTR1   EQU  028h
CNTR2   EQU  029h
CNTR3   EQU  02Ah
COMR1   EQU  02Bh
COMR2   EQU  02Ch

SMC     EQU  02Dh
SRC     EQU  02Eh
SSD     EQU  02Fh
SIDR    EQU  030h
SODR    EQU  030h


;-----------------------------------------
;--     Stack area defintion          --
;-----------------------------------------
        SSEG
        RB  20
StackTop:
        ENDS

;-----------------------------------------
;--     Dummy Segment Definitions     --
;-----------------------------------------
DIRVAR  DIRSEG
        RB  1
DIRVAR  ENDS
;-----------------------------------------
;-----------------------------------------
DVAR    DSEG
RxStr   RB 82
DVAR    ENDS


CCONST         CSEG
;-----------------------------------------
;-----------------------------------------
IniMel  DB 13,10," [2J** MB89630 UART Test Program **",13,10,0
TxMel   DB 13,10,"Tx String Operation: 0123456789"
        DB 13,10,"String Input :",0
RecMel  DB 13,10,"   Executing Echo Loop",13,10,0

;-----------------------------------------
;--     Some Symbol Info for Monitor  --
;-----------------------------------------
SymTab: DB "SymTab:",0          ; Symbol Table Header
        DB LabL1-LabS1,0      ; Length of 1st Symbol, dummy byte
LabS1   DB "Reset"              ; Symbol Name
LabL1   DW Reset
        DB LabL2-LabS2,0
LabS2   DB "main"
LabL2   DW main
        DB LabL3-LabS3,0
```

```
LabS3   DB "Stop"
LabL3   DW Stop

        DB LabL4-LabS4,0
LabS4   DB "SMC"
LabL4   DW SMC
        DB LabL5-LabS5,0
LabS5   DB "SRC"
LabL5   DW SRC
        DB LabL6-LabS6,0
LabS6   DB "SSD"
LabL6   DW SSD

        DB LabL7-LabS7,0
LabS7   DB "MLop"
LabL7   DW MLop
        DB LabL8-LabS8,0
LabS8   DB "TxProc"
LabL8   DW TxProc

         DB 00,00               ; End of Symbol Table, No BreakPoints
CCONST         ENDS

;-----------------------------------------
;-----------------------------------------
CSEG    CSEG

;-----------------------------------------
;--     Main Program                  --
;-----------------------------------------
Reset:  MOV  SYCC,#H'07                      ; HiSpeed Main Clock, Stabi=0
        MOVW SP,#StackTop
        MOVW SP,#H'200
        PRSTR IniMel                  ; BIOS Message

main:
        MOV  DDR4,#H'08                      ; P43PT01 Output
;       MOV  DDR4,#H'0A                      ; P43,P41 Output
        CLRB PDR4:3
         NOP
;-----------------------------------------
        MOV  COMR1,#H'07
        MOV  COMR2,#H'A0
        MOV  CNTR1,#B'00000000        ; Fastest Clock
        MOV  CNTR2,#B'10000000        ; Start TimerCounter 1
;       MOV  CNTR3,#B'00100000        ; P43 = PT01 function

BP1:    NOP
;-----------------------------------------
        MOV  SSD,#B'00100100          ; Initialize UART Control Reg.
;       MOV  SRC,#B'00011000          ; Use BaudRateGen. as Clock Souce
        MOV  SRC,#B'00001000          ; Use PWM Timer1 as Clock Souce
         MOV  SMC,#B'01011001         ;

;-----------------------------------------
;--     Tx Operation                  --
;-----------------------------------------
TxProc: MOVW EP,#TxMel
        CALL TxUA

        MOV  A,SSD                    ; dummy read in case
        MOV  A,SIDR                   ; there was something received

MLop:   CALL UaRxChar
         CALL UaTxChar
        JMP  MLop


;-----------------------------------------
        MOVW EP,#RxStr
        CALL RxUA

Stop:   NOP
        JMP  Stop




;-----------------------------------------
;--     Receive String via UART              --
;-----------------------------------------
RxUA:                         ; not programmed yet
        RET
;-----------------------------------------
;-----------------------------------------
UaRxChar: NOP
```

```
        BBC SSD:7,UaRxChar    ; wait until character received
        MOV  A,SIDR           ; read character
        RET

;-----------------------------------------
;--     Transmitt String via UART     --
;-----------------------------------------
TxUA:   MOV  A,@EP            ; load character
        INCW EP
        CMP  A,#0             ; end of string ?
        BEQ  ExTxUa           ; then exit
        CALL UaTxChar
        BNC  TxUA
ExTxUa: RET
;-----------------------------------------
UaTxChar: NOP
        BBC  SSD:5,UaTxChar   ; wait until SODR empty !!
        MOV  SODR,A
        CLRC
        RET

;*****************************************
CSEG    ENDS


;-----------------------------------------
;--     Symbol Table Vector           --
;-----------------------------------------
BIOSVAR DSEG ABS
        ORG 824h
        DW  SymTab                    ; Pointer To Symbol Table
BIOSVAR ENDS

;-----------------------------------------
;--     Reset Vector                  --
;-----------------------------------------
RVector CSEG ABS
        ORG 0FFFDh
        DB  1                         ; external Mode
        DW  Reset
RVector ENDS

        END
```