Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



THIS SPEC IS OBSOLETE

Spec No: 002-04633

Spec Title: AN204633 - MB89630 Series Using a PWM Timer as the Clock Input for the UART

Replaced by: None

MB89630 Series Using a PWM Timer as the Clock Input for the UART

Associated Part Family: MB89630 Series

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.

OBSOLETE

1 Introduction

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.

2 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, it's most likely that the desired baud rate cannot 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.

3 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.

OBSOLETE

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                          C y p r e s s                                  a*/
;/*a                                                                         a*/
;/*a                                                                         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
```

OBSOLETE

```
;---------------------------------------
;--     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
```

OBSOLETE

```
;----------------------------------------
;--     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
```

OBSOLETE

```
;----------------------------------------
;--     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
```

OBSOLETE

Document History

Document Title: AN204633 – MB89630 Series Using a PWM Timer as the Clock Input for the UART

Document Number: 002-04633

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	WOFR	02/20/2014	Original version
*A	5242262	WOFR	05/06/2016	Migrated Spansion Application Note MCU-AN-389015-E-V10 to Cypress format. Document obsoleted.

OBSOLETE

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Lighting & Power Control	cypress.com/powerpsoc
Memory	cypress.com/memory
PSoC	cypress.com/psoc
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless/RF	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

OBSOLETE

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corporation. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2013-2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.
TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.
Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.