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THIS SPEC IS OBSOLETE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: 002-04943

Spec Title: AN204943 - UART0 with an internal timer to generate the baudrate MB90670/675

Replaced by: None

UART0 with an internal timer to generate the baudrate MB90670/675

This application describes how to generate the baudrate MB90670/675 by using UART0 with an internal timer.

OBSOLETE

1 Introduction

In some applications one will have to use the internal timer #0 to generate the baudrate for UART0, because the dedicated baudrate generator cannot be used (e.g. because of an odd frequency of the oscillator). This note shows how to it.

Using the dedicated baudrate generator

UART0 is a serial I/O port for communications with external devices. The easiest way to use the UART is to set a baudrate using the dedicated baudrate generator. The baudrate is calculated using the following equation:

$$b = \frac{\Phi}{8xnR} \quad (1)$$

with: b : baudrate

Φ: machine clock (after PLL) in Hz

x: clock divider (3,4 or 5) bits BCH,BCH0

n: factor (1,2,4 or 8) bits RC3..RC2

R: factor (12 or 13) bit RC0

Where the specified bits are included in the URD(rate and data-) register :

Table 1. The URD-Register bit meaning

BCH	RC3	RC2	RC1	RC0	BCH0	P	D8

Note: The values in the manual on page 91 are NOT correct! The right values for calculating the dedicated baudrate for an asynchronous communication are:

Table 2. Bit settings for calculating decimated baudrates

RC3	RC2	RC1	n		RC0	R		BCH	BCH0	x
0	0	0	*1*		0	*12*		0	1	*4*
0	1	0	*2*		1	*13*		1	0	*3*
0	1	1	*4*					1	1	*5*
1	0	0	*8*							

Example: for a 16 MHz clock (e.g. 4 MHz clock and PLL=4x) the appropriate values for 9600 baud would be x=4, n=4 and R=13 → URD=$3C_{hex}$

Using timer0 to generate the baudrate

If using the internal timer (recource-name : 16-bit reload timer #0), the RC3...RC0-bits in the control register have to be set to “1101”. Then the reload value is calculated from equation 2 :

$$r = \frac{\Phi}{16xb} - 1 \qquad (2)$$

with : r : reload value (Register TMR0)

Φ: machine clock (after PLL) in Hz

x: clock divider(21,23,25)bits CSL0,1 in TMCSR

b: baudrate

For the previous example (16Mhz clock rate, *b*=9600 baud) the appropriate value for *r* would be 51 (*x*=2 assumed). To set up the UART0 for use with the internal timer, the timer0 control register TMCSR0 should be initialized at least with the reload-mode-bit RELD and the enable bit CNTE set to 1. Interrupts can be selected, but are not useful because the speed of transmission would be lower depending on the interrupt service routine. Setting bit 0 of the TMCSR0 triggers the timer.

The clock signal output pin (TOT0) from timer 0 is internally connected to the UARTs, so external connections are not required.

UART1 works in a similar way, but the reload values for UART1 are calculated from the equation :

$$r = \frac{\Phi}{32xb} - 1 \qquad (3)$$

example code :

```
void main(void)
{

                    /* both function calls of initUART would */
                    /* initialize the UART0 with 9600,8,N,2  */
                    /* async. communication. 16Mhz clock assumed */

/*  initUART(0x059,0 ,0x03C);    /* init UART0 for Baudgenerator-mode */

  initUART(0x059,0 ,0x06C);    /* init UART0 for use with timer0 */


  while(1)                      /* main loop */
  {
    putS("Test \13\10");   /* put string from UART0 */
    wait(10000);                /* short delay */
  }
}

.
.
.
void wait(int i)
```

OBSOLETE

```
{
  for (; i ; i--);   /* very simple delay loop */
}

void initUART (BYTE modus, BYTE interr, BYTE baud)
{
    UMC_SOE = 0;            /* stop UART, clear flags */
    URD = baud;              /* set baud rate */
    USR = interr;            /* enable/disable UART interrupts */
    UMC = modus;             /* set various bits (start,stop,parity etc) */

      if (URD && 0x068)           /* If Timer0 should be used for Baudrate : */
    {
    TMCSR0 = 0x012;         /* Set as Reload timer; 0.125us speed(X=2) */
    TMCSR0_INTE = 0;         /* No INTs for Baud Generator Mode */
    TMR0 = 51;                   /* 51 for 9600 Baud (with X=2 and 16MHz clock) */
    TMCSR0_TRG = 1;         /* Trigger for the timer0 */
    }
}

int  Putch (int ch)              /* sends a single character */
{
    while (!USR_TDRE);      /* wait for empty buffer */

    UIDR = (BYTE)ch;        /* fill Input/Output register */
    return ch;
}

void putS(char *buf)        /* puts a string */
{
    while (*buf != '\0') Putch(*buf++);  /* send every char */
}
```

OBSOLETE

2 Document History

Document Title: AN204943 - UART0 with an internal timer to generate the baudrate MB90670/675

Document Number: 002-04943

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	IKHI	01/30/1997	Initial release
			06/28/2000	New format
*A	5097384	IKHI	02/26/2016	Migrated Spansion Application Note MCU-AN-390010-E-V11 to Cypress This AN is for old product, So this should be 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

cypress.com/psoc

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

Cypress Developer Community

Community | Forums |Blogs | Video |Training

Technical Support

cypress.com/support

OBSOLETE

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corp. 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, 1997-2016. 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 under its copyright rights in the Software, a personal, non-exclusive, nontransferable license (without the right to sublicense) (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. Cypress also grants you a personal, non-exclusive, nontransferable, license (without the right to sublicense) 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 to the minimum extent that is necessary for you to exercise your rights under the copyright license granted in the previous sentence. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. 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 Company shall and hereby does release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. Company 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.