Semiconductors

AN10386

Baud rate calculation for Philips 16C UARTs

Rev. 01 — 3 August 2005

Application note

Document information

Info	Content
Keywords	baud rate, tolerance, frequency error
Abstract	This application note details how baud rates and baud rate tolerance are calculated and determined.

Revision history

Rev	Date	Description
AN10386_1	20050803	Application note. Initial version.

Contact information

For additional information, please visit: **http://www.semiconductors.philips.com**

For sales office addresses, please send an email to: **sales.addresses@www.semiconductors.philips.com**

1. Introduction

The purpose of this application note is to describe how baud rates and a crystal frequency can be determined for any Philips 16C UART. The equation for calculating the baud rates is listed in the data sheet section 'Programmable baud rate generator', but will be given here for reference. This equation is based on the fact that the UART samples the start bit at 16 times the baud rate. This 16× oversampling helps to determine the center of the start bit, and hence the center of all the bits that follow the start bit.

2. Baud rate calculation

The equation to calculate the baud rate is listed in Equation 1:

$$\text{baud} = \frac{\text{X1 freq}}{16 \times DIV} \quad (1)$$

where DIV is the number (in hex) to program the DLL and DLM. 'X1 freq' is the crystal's frequency or the frequency at XTAL1 pin of the UART.

From Equation 1, it can be calculated that:

$$DIV = \frac{\text{X1 freq}}{16 \times \text{baud}} \quad (2)$$

Also from Equation 1 it can be also be shown that:

$$\text{X1 freq} = \text{baud} \times 16 \times \text{DIV} \quad (3)$$

The error between the 'baud' and 'baud (selected)' is given as:

$$\%\ \text{error} = \frac{|\text{baud} - \text{baud (selected)}|}{\text{baud}} \times 100 \quad (4)$$

See the example in Section 2.1.2, for 'baud (selected)'.

2.1 Examples using a 19.2 MHz clock

2.1.1 Example 1

Suppose that we want a 9600 baud rate, and we want to know the values to be written to DLL and DLM. From Equation 2:

$$DIV = \frac{19.2M}{(16 \times 9600)} = 125$$

This number must be converted to hex before writing to DLL and DLM. A simple conversion shows the hex equivalent is 0x7D. Therefore, DLL must be written with 0x7D and DLM must be set to 0x00.

2.1.2 Example 2

Suppose that we want a 57600 baud rate, and we want to know the values to be written to DLL and DLM. Again from Equation 2:

$$DIV = \frac{19.2M}{(16 \times 57600)} = 20.83$$

Because 20.83 is not an even number, we must round it up to the closest even number of 21. Using the formula from Equation 1, we can calculate the new baud rate based on this number:

$$\text{baud (selected)} = \frac{19.2M}{(16 \times 21)} = 57142.86$$

This is not exactly what we want; we want 57600. Therefore the error is:

$$\%\ \text{Error} = \frac{57600 - 57142.86}{57600} \times 100 = 0.79\ \%$$

There is no exact limit on how much frequency error can be tolerated by the UART, since this depends on the baud rates, the precise frequencies used by the two devices, the character length, the number of stop bits, and whether a parity bit is used or not. However, most UARTs will work with a frequency error less than 5 %. This is the theoretical limit based on 16× sampling of the start bit, and 10-bit data format (one start bit, 8 data bits, and one stop bit).

2.1.3 Example 3

Suppose we select the UART to operate at 921.6 kbit/s, and we need to know what frequency is required. This frequency also depends on the setting of the DIV (DLL and DLM), assume that DLM is set to 0x00, and DLL is set to 0x01 for this example.

From Equation 3:

$$\begin{aligned} \text{X1 freq.} &= \text{baud} \times 16 \times \text{DIV} \\ &= 921.6\ \text{k} \times 16 \times 1 \\ &= 14.7456\ \text{MHz} \end{aligned}$$

3. Abbreviations

Table 1: Abbreviations

Acronym	Description
DLL	Divisor Latch L
DLM	Divisor Latch M
UART	Universal Asynchronous Receiver/Transmitter

4. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

5. Trademarks

Notice — All referenced brands, product names, service names and trademarks are the property of their respective owners.

6. Contents

Date of release: 3 August 2005
Document number: AN10386_1

Published in The Netherlands