ANALOG DEVICES

AN-891
APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

ADuC703x Series LIN Baud Rate Calculations

by Aude Richard

INTRODUCTION

The purpose of this application note is to familiarize the user with calculating the divisor values, COMDIV0, COMDIV1 and COMDIV2, for use with UART on the ADuC703x series devices from Analog Devices, Inc. This application note assumes that the user is familiar with Local Interconnect Network (LIN) 2.0 specifications.

This document is divided into three sections:

- LIN Frame Header. This section explains the LIN Frame header and the Sync byte.
- LIN Baud Rate Calculations. This section explains one approach to calculating the UART divisor values using the LIN Hardware Synchronization (LHS) functionality.
- LIN Baud Rate Calculations Example C Code. This section provides an example of C code implementation of the calculations explained in the LIN Baud Rate Calculations section.

LIN FRAME HEADER

A standard LIN communication frame is shown in Figure 1. It is broken down into a break symbol, a Sync byte, the protected identifier, data and a checksum.

- The break symbol signifies the start of the LIN packet.
- The Sync byte calibrates the slave's baud rate.
- The protected identifier identifies the slave.
- The checksum is either a classic checksum, calculated over the data transmitted, or the extended checksum, which is calculated over the protected identifier and data.

The Sync byte is shown in more detail in Figure 2. The Sync byte is 0xAA transmitted at the master's desired baud rate. The standard method for determining the master's bit rate is to measure the time from the first falling edge to the fifth falling edge. This value is divided by eight, which gives the desired bit rate. This value is then used to calculate the UART divisor values. The calculations are described in more detail in subsequent sections. This application note assumes that the user has set up the LHS MMR to time the full eight bits of the Sync byte (LHSCON1 = 0x62).

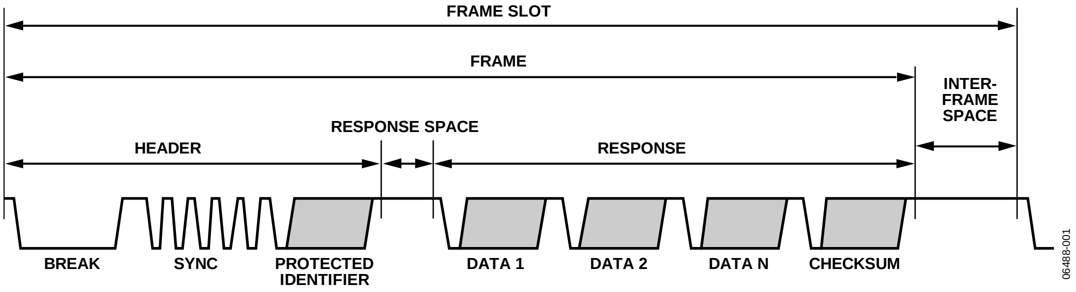

Figure 1. LIN Frame

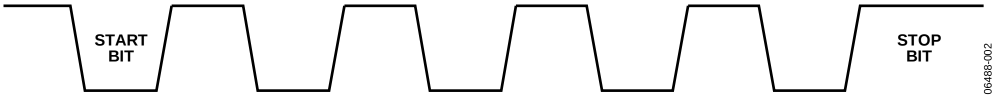

Figure 2. LIN Sync Byte

LIN BAUD RATE CALCULATIONS

Using the LHS system, the user gets a value in LHSVAL0 after the Sync byte is received. The LHSVAL0 contains the equivalent of 8 T_{BIT}. This value is used to generate the values for the UART dividers COMDIV0, COMDIV1, and the fractional divider COMDIV2. For more information on the UART refer to the relevant ADuC703x data sheet.

To calculate the COMDIV0/COMDIV1 values using the standard baud rate generator, the following basic UART equation is used:

$$DL = \frac{20.48\ \text{MHz}}{Baud\ Rate \times 2^{CD} \times 16 \times 2}$$

where:

DL is the value of COMDIV0 and COMDIV1.
CD is the clock divider.

In terms of LHSVAL0, the desired baud rate is as follows:

$$Desired\ Baud\ Rate = \frac{5.12\ \text{MHz} \times 8}{LHSVAL0}$$

(LHSVAL0 is clocked from an internal 5.12 MHz clock and it is assumed that LHSCON1 is configured to measure 8 T_{BIT}.)

If the standard baud rate equation and the desired baud rate equation are combined,

$$DL = \frac{20.48\ \text{MHz} \times LHSVAL0}{5.12\ \text{MHz} \times 2^{CD} \times 16 \times 2 \times 8}$$

$$DL = \frac{LHSVAL0}{2^{CD} \times 16 \times 2 \times 2}$$

$$DL = \frac{LHSVAL0}{2^{CD+6}}$$

Using only the standard baud rate generator equation gives the required values of COMDIV0/COMDIV1.

For increased accuracy, use the ADuC703x fractional divider with the DL value (COMDIV0/COMDIV1) calculated previously for the standard baud rate generator. The equation using the fractional divider is as follows:

$$Baud\ Rate = \frac{20.48\ \text{MHz}}{DL \times 2^{CD} \times 16 \times 2 \times \left(M + \frac{N}{2048}\right)}$$

where *M* and *N* are COMDIV2 values.

$$M + \frac{N}{2048} = \frac{20.48\ \text{MHz}}{Baud\ Rate \times DL \times 2^{CD} \times 16 \times 2}$$

Replacing *Baud Rate*,

$$M + \frac{N}{2048} = \frac{20.48\ \text{MHz} \times LHSVAL0}{5.12\ \text{MHz} \times 8 \times DL \times 2^{CD} \times 16 \times 2}$$

This simplifies to

$$M + \frac{N}{2048} = \frac{LHSVAL0}{DL \times 2^{CD} \times 2 \times 16 \times 2}$$

$$M + \frac{N}{2048} = \frac{LHSVAL0}{DL \times 2^{CD+6}}$$

To reduce the need for complex math used in the calculation of the fractional divider, limit the values of DL (COMDIV0/COMDIV1) to the power of two. For example, if DL = 17, use DL = 16 = 2^4 in the calculation of N. This automatically adjusts the value of N to compensate for the error introduced by modifying DL.

$$DL = 2^{DL\_Power}$$

$$M + \frac{N}{2048} = \frac{LHSVAL0}{2^{DL\_Power} \times 2^{CD+6}}$$

$$M + \frac{N}{2048} = \frac{LHSVAL0}{2^{DL\_Power+CD+6}}$$

If M is set equal to 1,

$$N = \frac{2^{11} \times LHSVAL0}{2^{DL\_Power+CD+6}} - 2048$$

$$N = 2^{5-DL\_Power-CD} \times LHSVAL0 - 2048$$

For example, for a baud rate of 19,200 bps, with CD = 0, DL = 33 and LHSVAL0 = 2133, then N = 21 and the baud rate is 19,197 bps. If DL = 32 and N = 85 is used, the baud rate is then 19,203 bps.

LIN BAUD RATE CALCULATIONS EXAMPLE C CODE

When programming in C, the preceding equations can be simply written by using the << and >> shift commands.

```
// DL = LHSVAL0 >> CD_Bits + 6
iDL = LHSVAL0 >> (( POWCON & 0x7)+6);

// writing DL as 2^iDL_Power
iDL_Power = 0;
iDL_temp = iDL;
while(iDL >> (iDL_Power +1 ))
{
      iDL_Power++;
}

// Configuration of the fractional divider:
// M = 1
// N = LHSVAL0 × 2 ^ (5 – (iDL_Power + CD)) - 2048
iDL_temp = iDL_Power + (POWCON & 0x7);
if (iDL_temp > 5)
{
      iDL_temp = (LHSVAL0 >> (iDL_temp -5)) - 2048;
}
else
{
      iDL_temp= (LHSVAL0 << (5 - iDL_temp)) - 2048;
}

COMDIV2 = 0x8800 + iDL_temp;

COMCON0 = 0x080;   // Setting DLAB
// Setting DIV0 and DIV1 to DL calculated
COMDIV0 = (1<< iDL_Power) & 0xff;
COMDIV1 = (1<< iDL_Power) & 0xff00;
COMCON0 = 0x03;    // Setting DLAB
COMIEN0 = 0x1;    // Enable RX interrupt
```

NOTES

AN06488-0-11/06(0)