AN-891
应用笔记

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 系列 LIN 波特率计算

作者：Aude Richard

简介

本应用笔记旨在让用户熟悉用于 ADI 公司 ADuC703x 系列器件 UART 通信的除数值 COMDIV0、COMDIV1 和 COMDIV2 的计算。本应用笔记假定用户熟悉本地互连网络 (LIN) 2.0 规范。

本文分为以下三部分：

- LIN 帧报头：本部分说明 LIN 帧报头和同步字节。
- LIN 波特率计算：本部分说明一种使用 LIN 硬件同步 (LHS) 功能来计算 UART 除数值的方法。
- LIN 波特率计算 C 代码示例：本部分提供“LIN 波特率计算”部分所述计算的 C 代码实现示例。

LIN 帧报头

标准 LIN 通信帧如图 1 所示，它分为如下部分：断开符号、同步字节、受保护标识符、数据和校验和。

- 断开符号表示 LIN 分包的开始。
- 同步字节标定从机的波特率。
- 受保护标识符用于识别从机。
- 校验和既可以是根据发送数据计算的传统校验和，也可以是根据受保护标识符和数据计算的扩展校验和。

图 2 更加详细地显示了同步字节，它是以主机所需的波特率发送的字节 0xAA。确定主机比特率的标准方法是测量从第一个下降沿到第五个下降沿的时间，然后将此值除以 8，便得到所需的比特率。此值可以用于计算 UART 除数值。具体计算详见后续部分。本应用笔记假设用户已设置 LHS MMR 来为同步字节的全部 8 位计时 (LHSCON1 = 0x62)。

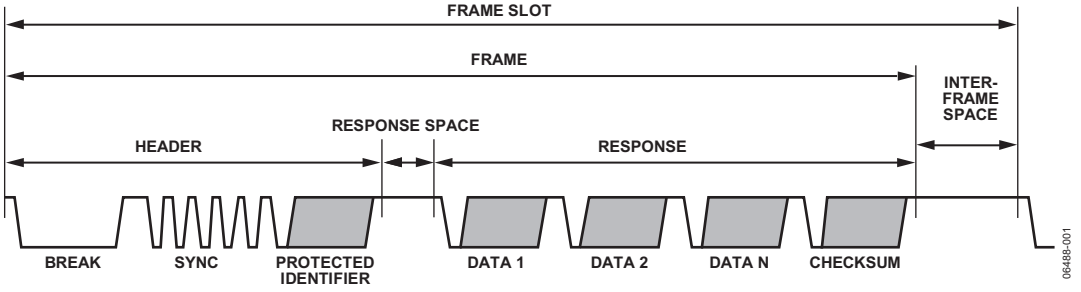

图 1. LIN 帧

图 2. LIN 同步字节

LIN 波特率计算

使用 LHS 系统，用户在收到同步字节后会得到 LHSVAL0 中的值。LHSVAL0 包含 8 T_{BIT} 的等效值，此值用于产生 UART 除数 COMDIV0、COMDIV1 和小数除数 COMDIV2 的值。有关 UART 的更多信息，请参阅相关 ADuC703x 数据手册。

为利用标准波特率发生器计算 COMDIV0/COMDIV1 值，需使用如下基本 UART 方程式：

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

其中：

DL 为 COMDIV0 和 COMDIV1 的值。
CD 为时钟分频比。

就 LHSVAL0 而言，所需的波特率如下：

$$所需波特率 = \frac{5.12\ \text{MHz} \times 8}{LHSVAL0}$$

(LHSVAL0 采用内部 5.12 MHz 时钟，假设将 LHSCON1 配置为测量 8 T_{BIT}。)

将标准波特率方程式与所需波特率方程式合并：

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

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

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

仅使用标准波特率发生器方程式可得出所需的 COMDIV0/COMDIV1 值。

为提高精度，使用 ADuC703x 小数除数和以上针对标准波特率发生器计算的 DL 值 (COMDIV0/COMDIV1)。使用小数除数的方程式如下：

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

其中，*M* 和 *N* 为 COMDIV2 值。

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

替换*波特率*，

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

上式可简化为：

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

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

为减少小数除数计算所用的复杂数学计算，将 DL 的值 (COMDIV0/COMDIV1) 限制为 2 的幂数。例如，如果 DL = 17，则计算 N 时使用 DL = 16 = 2^4，这将自动调整 N 的值以补偿 DL 修改所引入的误差。

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

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

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

如果将 M 设为 1，

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

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

例如，对于 19,200 bps 的波特率，如果 CD = 0、DL = 33、LHSVAL0 = 2133，则 N = 21，波特率为 19,197 bps；如果使用 DL = 32、N = 85，则波特率为 19,203 bps。

LIN 波特率计算 C 代码示例

使用 C 语言编程时，上述方程式可以简单地利用 << 和 >> 移位命令编写。

```
// 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
```

注释