



# LPC1100 系列微控制器

---

## 第十章 通用异步收发器(UART)

用户手册 Rev1.00

**广州周立功单片机发展有限公司**

地址：广州市天河北路 689 号光大银行大厦 12 楼 F4

网址：<http://www.zlgmcu.com>

## 销售与服务网络（一）

### 广州周立功单片机发展有限公司

地址：广州市天河北路 689 号光大银行大厦 12 楼 F4

邮编：510630

电话：(020)38730916 38730917 38730972 38730976 38730977

传真：(020)38730925

网址：[www.zlgmcu.com](http://www.zlgmcu.com)



### 广州专卖店

地址：广州市天河区新赛格电子城 203-204 室

电话：(020)87578634 87569917

传真：(020)87578842

### 南京周立功

地址：南京市珠江路 280 号珠江大厦 1501 室

电话：(025) 68123901 68123902

传真：(025) 68123900

### 北京周立功

地址：北京市海淀区知春路 113 号银网中心 A 座  
1207-1208 室（中发电子市场斜对面）

电话：(010)62536178 62536179 82628073

传真：(010)82614433

### 重庆周立功

地址：重庆市石桥铺科园一路二号大西洋国际大厦  
（赛格电子市场）1611 室

电话：(023)68796438 68796439

传真：(023)68796439

### 杭州周立功

地址：杭州市天目山路 217 号江南电子大厦 502 室

电话：(0571)89719480 89719481 89719482

89719483 89719484 89719485

传真：(0571)89719494

### 成都周立功

地址：成都市一环路南二段 1 号数码科技大厦 403 室

电话：(028)85439836 85437446

传真：(028)85437896

### 深圳周立功

地址：深圳市深南中路 2070 号电子科技大厦 C 座 4 楼 D 室

电话：(0755)83781788（5 线）

传真：(0755)83793285

### 武汉周立功

地址：武汉市洪山区广埠屯珞瑜路 158 号 12128 室  
（华中电脑数码市场）

电话：(027)87168497 87168297 87168397

传真：(027)87163755

### 上海周立功

地址：上海市北京东路 668 号科技京城东座 7E 室

电话：(021)53083452 53083453 53083496

传真：(021)53083491

### 西安办事处

地址：西安市长安北路 54 号太平洋大厦 1201 室

电话：(029)87881296 83063000 87881295

传真：(029)87880865

## 销售与服务网络（二）

### 广州致远电子有限公司

地址：广州市天河区车陂路黄洲工业区3栋2楼

邮编：510660

传真：(020)38601859

网址：[www.embedtools.com](http://www.embedtools.com) （嵌入式系统事业部）

[www.embedcontrol.com](http://www.embedcontrol.com) （工控网络事业部）

[www.ecardsys.com](http://www.ecardsys.com) （楼宇自动化事业部）



#### 技术支持：

##### CAN-bus:

电话：(020)22644381 22644382 22644253

邮箱：[can.support@embedcontrol.com](mailto:can.support@embedcontrol.com)

##### MiniARM:

电话：(020)28872684 28267813

邮箱：[miniarm.support@embedtools.com](mailto:miniarm.support@embedtools.com)

##### 无线通讯：

电话：(020) 22644386

邮箱：[wireless@embedcontrol.com](mailto:wireless@embedcontrol.com)

##### 编程器：

电话：(020)22644371

邮箱：[programmer@embedtools.com](mailto:programmer@embedtools.com)

##### ARM 嵌入式系统：

电话：(020) 22644383 22644384

邮箱：[NXPARM@zlgmcu.com](mailto:NXPARM@zlgmcu.com)

#### 销售：

电话：(020)22644249 22644399 22644372 22644261 28872524

28872342 28872349 28872569 28872573 38601786

#### 维修：

电话：(020)22644245

##### iCAN 及数据采集：

电话：(020)28872344 22644373

邮箱：[ican@embedcontrol.com](mailto:ican@embedcontrol.com)

##### 以太网：

电话：(020)22644380 22644385

邮箱：[ethernet.support@embedcontrol.com](mailto:ethernet.support@embedcontrol.com)

##### 串行通讯：

电话：(020)28267800 22644385

邮箱：[serial@embedcontrol.com](mailto:serial@embedcontrol.com)

##### 分析仪器：

电话：(020)22644375

邮箱：[tools@embedtools.com](mailto:tools@embedtools.com)

##### 楼宇自动化：

电话：(020)22644376 22644389 28267806

邮箱：[mjs.support@ecardsys.com](mailto:mjs.support@ecardsys.com)

[mifare.support@zlgmcu.com](mailto:mifare.support@zlgmcu.com)

## 目 录

第 10 章 通用异步接收器/发送器 (UART)	2
10.1 本章导读	2
10.2 特性	2
10.3 管脚描述	2
10.4 计时和功率控制	2
10.5 寄存器描述	3
10.5.1 UART 接收器缓冲寄存器	4
10.5.2 UART 发送保持寄存器	4
10.5.3 UART 除数锁存 LSB 和 MSB 寄存器	5
10.5.4 UART 中断使能寄存器	5
10.5.5 UART 中断标识寄存器	6
10.5.6 UART FIFO 控制寄存器	8
10.5.7 UART Modem 控制寄存器	8
10.5.8 UART 线控制寄存器	11
10.5.9 UART 线状态寄存器	11
10.5.10 UART Modem 状态寄存器	12
10.5.11 UART 高速缓存寄存器	13
10.5.12 UART Auto-baud 控制寄存器	13
10.5.13 自动波特率 (Auto-baud)	14
10.5.14 Auto-baud 模式	15
10.5.15 UART 小数分频器寄存器	16
10.5.16 UART 发送使能寄存器	19
10.5.17 UART RS485 控制寄存器	20
10.5.18 UART RS-485 地址匹配寄存器	20
10.5.19 UART1 RS-485 延时值寄存器	21
10.5.20 RS-485/EIA-485 模式的操作	21
10.5.21 UART FIFO 深度寄存器	22
10.6 结构	22

## 第10章 通用异步接收器/发送器 (UART)

### 10.1 本章导读

对于所有 LPC111x 系列 ARM, UART 模块都相同。 $\overline{\text{DSR}}$ 、 $\overline{\text{DCD}}$  和  $\overline{\text{RI}}$  modem 信号是只用于 LQFP48 和 PLCC44 封装的管脚配置。

### 10.2 特性

- 16 字节收发 FIFO;
- 寄存器的存储单元符合“550 工业标准”;
- 接收器 FIFO 触发点位于 1、4、8 和 14 字节;
- 内置波特率发生器;
- UART 支持软件或硬件流控制执行;
- 支持 RS-458/EIA-485 的 9 位模式和输出使能;
- Modem 控制。

### 10.3 管脚描述

表 10.1 UART 管脚描述

管脚	类型	描述
RXD	输入	串行输入管脚。串行接收数据
TXD	输出	串行输出管脚。串行发送数据
RTS	输出	请求发送。RS-485 方向控制管脚
$\overline{\text{DTR}}$	输出	数据终端就绪
$\overline{\text{DSR}}^{[1]}$	输入	数据设置就绪
$\overline{\text{CTS}}$	输入	清除发送
$\overline{\text{DCD}}^{[1]}$	输入	数据载波检测
$\overline{\text{RI}}^{[1]}$	输入	铃响指示

[1] 仅用于 LQFP48 封装。

$\overline{\text{DSR}}$ 、 $\overline{\text{DCD}}$  和  $\overline{\text{RI}}$  modem 输入在 2 个不同的管脚位置进行多路复用。除了选择 IOCON 寄存器的功能外, 还可以使用 IOCON\_LOC 寄存器 (见第 7 章“IOCON 位置寄存器”小节) 为 LQFP48 管脚封装上的每个功能选择一个物理位置。

$\overline{\text{DTR}}$  输出在 2 个管脚位置都可以使用。 $\overline{\text{DTR}}$  管脚的输出值在两个相同的位置被驱动, 并且可通过为该管脚位置选择 IOCON 寄存器的功能来简单选择任意位置的  $\overline{\text{DTR}}$  功能。

### 10.4 计时和功率控制

UART 模块通过 AHBCLKCTRL 寄存器进行门控。外设 UART 时钟 (由 UART 波特率发生器使用) 通过 UARTCLKDIV 寄存器来控制。

UART\_PCLK 可在 UARTCLKDIV 寄存器 (见第 3 章“UART 时钟分频器寄存器”小节) 中禁能, 并且 UART 模块可通过系统 AHB 时钟控制寄存器的位 12 (见第 3 章“系统 AHB 时钟控制寄存器”小节) 禁能以节省功耗。

注: 在使能 UART 时钟之前, UART 管脚必须在相应的 IOCON 寄存器中配置。

## 10.5 寄存器描述

UART 所包含的寄存器，其结构如表 10.2 所述。除数锁存器访问位（DLAB）包含在 U0LCR[7]中，能够使能除数锁存器的访问。

表 10.2 寄存器概述：UART（基址：0x4000 8000）

名称	访问	地址偏移量	描述	复位值 <sup>[1]</sup>	注
U0RBR	RO	0x000	接收缓冲寄存器。包含下一个要读取的已接收字符	NA	当 DLAB=0
U0THR	WO	0x000	发送保持寄存器。在此写入下一个要发送的字符	NA	当 DLAB=0
U0DLL	R/W	0x000	除数锁存 LSB。波特率除数值的最低有效字节。整个分频器用于产生小数波特率分频器的波特率	0x01	当 DLAB=1
U0DLM	R/W	0x004	除数锁存 MSB。波特率除数值的最高有效字节。整个分频器用于产生小数波特率分频器的波特率	0x00	当 DLAB=1
U0IER	R/W	0x004	中断使能寄存器。包含 7 个潜在的 UART 中断对应的各个中断使能位	0x00	当 DLAB=0
U0IIR	RO	0x008	中断 ID 寄存器。识别等待处理的中断	0x01	-
U0FCR	WO	0x008	FIFO 控制寄存器。控制 UART FIFO 的使用和模式	0x00	-
U0LCR	R/W	0x00C	线控制寄存器。包含帧格式控制和间隔产生控制	0x00	-
U0MCR	R/W	0x010	Modem 控制寄存器	0x00	-
U0LSR	RO	0x014	线状态寄存器。包含发送和接收状态的标志（包括线错误）	0x60	-
U0MSR	RO	0x018	Modem 状态寄存器	0x00	-
U0SCR	R/W	0x01C	高速缓存寄存器。8 位的临时存储空间，供软件使用	0x00	-
U0ACR	R/W	0x020	自动波特率控制寄存器。包含自动波特率特性的控制	0x00	-
-	-	0x024	保留	-	-
U0FDR	R/W	0x028	小数分频器寄存器。为波特率分频器产生时钟输入	0x10	-
-	-	0x02C	保留	-	-
U0TER	R/W	0x030	发送使能寄存器。关闭 UART 发送器，使用软件流控制	0x80	-
-	-	0x034 - 0x048	保留	-	-

续上表

名称	访问	地址偏移量	描述	复位值 <sup>[1]</sup>	注
U0RS485CTRL	R/W	0x04C	RS-485/EIA-485 控制。包含了 RS-485/EIA-485 模式多方面的配置控制	0x00	-
U0ADRMATCH	R/W	0x050	RS-485/EIA-485 地址匹配。包含 RS-485/EIA-485 模式的地址匹配值	0x00	-
U0RS485DLY	R/W	0x054	RS-485/EIA-485 方向控制延迟	0x00	-
U0FIFOLVL	RO	0x058	FIFO 水平寄存器。提供发送和接收 FIFO 的当前填充深度	0x00	-

[1] 复位值仅指使用位中保存的数据，并不包括保留位的内容。

### 10.5.1 UART 接收器缓冲寄存器

U0RBR 是 UART RX FIFO 的最高字节。它包含了最早接收到的字符，并且可通过总线接口进行读取。LSB（位 0）表示最“早”接收的数据位。如果接收到的字符少于 8 位，则未使用的 MSB 用 0 填充。

如果要访问 U0RBR，U0LCR 中的除数锁存访问位（DLAB）必须为 0。U0RBR 为只读寄存器。

由于 PE、FE 和 BI 位（见表 10.14）与 RBR FIFO 顶部的字节（即下次读 RBR 时获取的字节）相关，因此，要正确地成对读出有效的接收字节及其状态位，应先读取 U0LSR 的内容，然后再读取 U0RBR 中的字节。

表 10.3 UART 接收器缓冲寄存器（U0RBR – 0x4000 8000，当 DLAB=0 时，只读）位描述

位	符号	描述	复位值
7:0	RBR	UART 接收器缓冲寄存器包含了 UART RX FIFO 当中最早接收到的字节	未定义
31:8	-	保留	-

### 10.5.2 UART 发送保持寄存器

U0THR 是 UART TX FIFO 的最高字节。它是 TX FIFO 中的最新字符，可通过总线接口进行写入。LSB 代表第一个要发送出去的位。

如果要访问 U0THR，U0LCR 中的除数锁存访问位（DLAB）必须为 0。U0THR 为只写寄存器。

表 10.4 UART 发送保持寄存器（U0THR – 0x4000 8000，当 DLAB=0 时，只写）位描述

位	符号	描述	复位值
7:0	THR	写 UART 发送保持寄存器会使数据保存到 UART 发送 FIFO 中。当字节达到 FIFO 的底部并且发送器可用时，字节就会被发送	NA
31:8	-	保留	-

### 10.5.3 UART 除数锁存 LSB 和 MSB 寄存器

UART 除数锁存器是 UART 波特率发生器的一部分并保存使用的值，它与小数分频器一同使用，来分频 UART\_PCLK 时钟以产生波特率时钟，该波特率时钟必须是所需波特率的 16 倍。U0DLL 和 U0DLM 寄存器一起构成了一个 16 位除数，其中 U0DLL 包含了除数的低 8 位而 U0DLM 包含了除数的高 8 位。值 0x0000 会被作为 0x0001 处理，因为除数不能为 0。如果要访问 UART 除数锁存寄存器，U0LCR 中的除数锁存器访问位（DLAB）必须为 1。关于如何为 U0DLL 和 U0DLM 选择正确的分频值的详细内容，请参见“UART 小数分频器寄存器”小节。

表 10.5 UART 除数锁存器 LSB 寄存器（U0DLL – 0x4000 8000）位描述

位	符号	描述	复位值
7:0	DLLSB	UART 除数锁存 LSB 寄存器与 U0DLM 寄存器一起决定 UART 的波特率	0x01
31:8	-	保留	-

表 10.6 UART 除数锁存器 MSB 寄存器（U0DLM – 0x4000 8004）位描述

位	符号	描述	复位值
7:0	DLMSB	UART 除数锁存 MSB 寄存器与 U0DLL 寄存器一起决定 UART 的波特率	0x00
31:8	-	保留	-

### 10.5.4 UART 中断使能寄存器

U0IER 用于使能 4 个 UART 中断源。

表 10.7 UART 中断使能寄存器（U0IER – 0x4000 8004）位描述

位	符号	值	描述	复位值
0	RBR Interrupt Enable	0 1	使能 UART 的接收数据可用中断。它还控制着字符接收超时中断 禁止 RDA 中断 使能 RDA 中断	0
1	THRE Interrupt Enable	0 1	使能 UART 的 THRE 中断。该中断的状态可从 U0LSR[5]中读出 禁止 THRE 中断 使能 THRE 中断	0
2	RX Line Interrupt Enable	0 1	使能 UART 的 RX 线状态中断。该中断的状态可从 U0LSR[4:1]中读出 禁止 RX 线状态中断 使能 RX 线状态中断	0
3	-	-	保留	-
6:4	-		保留。用户软件不应对其写入 1。从保留位读出的值未定义	NA
7	-	-	保留	0



续上表

位	符号	值	描述	复位值
8	ABEOIntEn	0 1	使能 auto-baud 结束中断 禁止 auto-baud 结束中断 使能 auto-baud 结束中断	0
9	ABTOIntEn	0 1	使能 auto-baud 超时中断 禁止 auto-baud 超时中断 使能 auto-baud 超时中断	0
31:10	-		保留，用户软件不应向保留位写入 1。从保留位读出的值未定义	NA

### 10.5.5 UART 中断标识寄存器

U0IIR 提供状态代码用于指示一个待处理中断的优先级和中断源。在访问 U0IIR 过程中，中断被冻结。如果在访问 U0IIR 过程中产生了中断，那么在下次访问 U0IIR 时该中断会被记录。

表 10.8 UART 中断标识寄存器（U0IIR – 0x4004 8008，只读）位描述

位	符号	值	描述	复位值
0	IntStatus	0 1	中断状态。注意 U0IIR[0]为低电平有效。待处理的中断可通过 U0IIR[3:1]来确定 至少有一个中断正在等待处理 没有等待处理的中断	1
3:1	IntId	011 010 110 001 000	中断标识。U0IER[3:1]指示对应 UART RX FIFO 的中断。下面未列出的 U0IER[3:1]的其它组合都为保留值（100、101、111） 1 – 接收线状态（RLS） 2a – 接收数据可用（RDA） 2b – 字符超时指示器（CTI） 3 – THRE 中断 4 – Modem 中断	0
5:4	-		保留。用户软件不应对其写入 1。从保留位读出的值未定义	NA
7:6	FIFO Enable		这些位等效于 U0FCR[0]	0
8	ABEOInt		Auto-baud 结束中断。若 auto-baud 已成功结束且中断被使能，则为真	0
9	ABTOInt		Auto-baud 超时中断。若 auto-baud 已超时且中断被使能，则为真	0
31:10	-		保留。用户软件不应对其写入 1。从保留位读出的值未定义	NA

位 U0IIR[9:8]由 auto-baud 功能设置，用于发布超时信号或 auto-baud 结束条件信号。而 Auto-baud 中断条件的清除则是通过设置 Auto-baud 控制寄存器中的相应 Clear（清除）位来实现。

如果 IntStatus 位为 1 且没有中断等待处理，此时 IntId 位为 0。如果 IntStatus 位为 0，表示有一个非 auto-baud 中断正等待处理，此时 IntId 位会指示中断的类型，中断的处理见表 10.9。给定了 U0IIR[3:0]的状态，中断处理程序就能确定中断源以及如何清除有效的中断。在退出中断服务程序之前，必须读取 U0IIR 来清除中断。

UART RLS 中断（U0IIR[3:1]=011）是最高优先级中断，只要 UART Rx 输入产生 4 个错误条件（溢出错误 OE、奇偶错误 PE、帧错误 FE 以及间隔中断 BI）中的任意一个，该位就会被置位。设置中断的 UART Rx 错误条件可通过 U0LSR[4:1]来查看。当读取 U0LSR 时，中断就会被清除。

UART RDA 中断（U0IIR[3:1]=010）与 CTI 中断（U0IIR[3:1]=110）共用第二优先级。当 UART Rx FIFO 深度到达 U0FCR[7:6]所定义的触发点时，RDA 就会被激活；当 UART Rx FIFO 深度低于触发点时，RDA 复位。当 RDA 中断激活时，CPU 可读出由触发点所定义的数据块。

CTI 中断（U0IIR[3:1]=110）是一个第二优先级中断，当 UART Rx FIFO 内含有至少一个字符并且在接收到 3.5 到 4.5 字符的时间内没有发生 UART Rx FIFO 操作时，该中断置位。任何 UART Rx FIFO 操作（UART RSR 的读取或写入）将会清除该中断。当接收到的信息不是触发点值的倍数时，CTI 中断将会清空 UART RBR。例如：如果外围设备想要发送一个长度为 105 个字符的信息，而触发值为 10 个字符，那么 CPU 接收 10 个 RDA 中断以使传输前 100 个字符，CPU 接收 1~5 个 CTI 中断（取决于服务程序）将传输剩下的 5 个字符。

表 10.9 UART 中断处理

U0IIR[3:0]值 <sup>[1]</sup>	优先级	中断类型	中断源	中断复位
0001	-	无	无	-
0110	最高	RX 线状态/错误	OE <sup>[2]</sup> 、PE <sup>[2]</sup> 、FE <sup>[2]</sup> 或 BI <sup>[2]</sup>	U0LSR <sup>[2]</sup> 读操作
0100	第二	RX 数据可用	Rx 数据可用或达到 FIFO 的触发点（U0FCR0=1）	U0RBR <sup>[3]</sup> 读操作或 UART FIFO 低于触发值
1100	第二	字符超时指示	RX FIFO 中至少有一个字符，并且在一段时间内没有字符输入或移出，该时间的长短取决于 FIFO 中的字符数以及在 3.5 到 4.5 字符的时间内的触发值 实际时间为：[(字长度)×7-2]×8+[(触发值-字符数)×8+1]RCLK	U0RBR <sup>[3]</sup> 读操作
0010	第三	THRE	THRE <sup>[2]</sup>	U0IIR <sup>[4]</sup> 读操作（如果 U0IIR 是中断源）或 THR 写操作

[1] “0000”、“0011”、“0101”、“0111”、“1000”、“1001”、“1010”、“1011”、“1101”、“1110”、“1111”均为保留值。

[2] 详细信息，请参见“UART 线状态寄存器”小节。

[3] 详细信息，请参见“UART 接收缓冲器寄存器”小节。

[4] 详细信息，请参见“UART 中断标识寄存器”小节以及“UART 发送器保持寄存器”小节。

UART THRE 中断 (U0IIR[3:1]=001) 是第三优先级中断。当 UART THR FIFO 为空且满足特定的初始化条件时, 该中断激活。这些初始化条件是为了让 UART THR FIFO 有机会填入数据, 以免在系统启动时产生许多 THRE 中断。当 THRE=1 时, 且在上次 LSR 寄存器的 THRE=1 事件后, U0THR 中没有出现至少两个字符时, 这些初始化条件就会实现一个字符减去停止位的延时。当没有解码和服务 THRE 中断时, 该延时为 CPU 提供了写数据到 U0THR 的时间。当 UART THR FIFO 中曾经同时出现两个或更多字符, 而当前的 U0THR 为空时, THRE 中断就会立即被设置。当发生 U0THR 写操作或 U0IIR 读操作, 并且 THRE 为最高优先级中断 (U0IIR[3:1]=001) 时, THRE 中断复位。

### 10.5.6 UART FIFO 控制寄存器

U0FCR 控制 UART RX 和 TX FIFO 的操作。

表 10.10 UART FIFO 控制寄存器 (U0FCR – 0x4000 8008, 只写) 位描述

位	符号	值	描述	复位值
0	FIFO Enable	0	UART FIFO 被禁止。禁止在应用中使用	0
		1	高电平有效, 使能对 UART RX FIFO 和 TX FIFO 以及 U0FCR[7:1]的访问。该位必须置位以实现正确的 UART 操作。该位的任何变化都将使 UART FIFO 清空	
1	RX FIFO Reset	0	对两个 UART FIFO 均无影响	0
		1	写 1 到 U0FCR[1]将会清零 UART Rx FIFO 中的所有字节, 并复位指针逻辑, 该位可以自动清零	
2	TX FIFO Reset	0	对两个 UART FIFO 均无影响	0
		1	写 1 到 U0FCR[2]将会清零 UART Tx FIFO 中的所有字节, 并复位指针逻辑。该位会自动清零	
3	-	-	保留	0
5:4	-		保留。用户软件不应对其写入 1。从保留位读出的值未定义	NA
7:6	RX Trigger Level	00	这两个位决定了接收 UART FIFO 在激活中断前必须写入的字符数量	0
		01	触发点 0 (默认 1 字节或 0x01)	
		10	触发点 1 (默认 4 字节或 0x04)	
		11	触发点 2 (默认 8 字节或 0x08)	
31:8	-	-	保留	-

### 10.5.7 UART Modem 控制寄存器

U0MCR 使能 Modem 的回送模式并控制 Modem 的输出信号。

表 10.11 UART0 Modem 控制寄存器 (U0MCR – 0x4000 8010, 只写) 位描述

位	符号	值	描述	复位值
0	DTR Control		选择 Modem 输出管脚 $\overline{\text{DTR}}$ 。当 Modem 回送模式激活时, 该位读出为 0	0

续上表

位	符号	值	描述	复位值
1	RTS Control		选择 Modem 输出管脚 $\overline{\text{RTS}}$ 。当 Modem 回送模式激活时，该位读出为 0	0
3:2	-	NA	保留。用户软件不应对其写入 1。从保留位读出的值未定义	0
4	Loopback Mode Select	0 1	Modem 回送模式提供了执行诊断回送测试的机制。发送器输出的串行数据在内部连接到接收器的串行输入端。输入管脚 $\text{RXD}$ 对回送操作无影响，而输出管脚 $\text{TXD}$ 保持为标记状态（marking state）。4 个 Modem 输入端（ $\overline{\text{CTS}}$ 、 $\overline{\text{DSR}}$ 、 $\overline{\text{RI}}$ 和 $\overline{\text{DCD}}$ ）与外部断开连接。从外部看来，Modem 输出端（ $\overline{\text{RTS}}$ 、 $\overline{\text{DTR}}$ ）无效。而从内部看来，4 个 Modem 输出都连接到 4 个 Modem 输入上。这样连接的结果将导致 $\text{U0MSR}$ 的高 4 位由 $\text{U0MCR}$ 的低 4 位驱动，而不是在正常模式下由 4 个 Modem 输入驱动。这样在回送模式下，写 $\text{U0MCR}$ 的低 4 位可产生 Modem 状态中断 禁能 Modem 回送模式 使能 Modem 回送模式	0
5	-	NA	保留。用户软件不应对其写入 1。从保留位读出的值未定义	0
6	RTSen	0 1	禁止 auto-rts 流控制 使能 auto-rts 流控制	0
7	CTSen	0 1	禁止 auto-cts 流控制 使能 auto-cts 流控制	0

### 1. Auto-flow 控制

如果启动了 auto-RTS 模式，UART 的接收 FIFO 硬件将控制 UART 的  $\overline{\text{RTS}}$  输出。如果启动的是 auto-CTS 模式，UART 的  $\text{U0TSR}$  硬件只会在  $\overline{\text{CTS}}$  输入信号有效时启动发送。

#### (1) Auto-RTS

Auto-RTS 功能是通过设置 RTSen 位来启动的。Auto-RTS 数据流控制在  $\text{U0RBR}$  模块中产生，链接到设定的接收 FIFO 触发点。如果启动了 auto-RTS，数据流的控制如下：

当接收 FIFO 水平达到设定的触发点时， $\overline{\text{RTS}}$  信号被撤销（变为高值）。在达到触发点后，发送 UART 可能还会发送额外的字节（假定发送 UART 有另一个字节要发送），因为 UART 在开始发送额外字节之前可能不知道  $\overline{\text{RTS}}$  信号已撤销。一旦接收 FIFO 达到前面的触发点， $\overline{\text{RTS}}$  就会自动重新生效（变为低值）。 $\overline{\text{RTS}}$  重新生效指示发送 UART 继续发送数据。

如果禁止了 Auto-RTS 模式，RTSen 位控制 UART 的  $\overline{\text{RTS}}$  输出。如果启动了 Auto-RTS 模式，硬件控制  $\overline{\text{RTS}}$  输出，并且  $\overline{\text{RTS}}$  的实际值将会复制到 UART 的 RTS Control 位中。只要启动了 Auto-RTS 模式，软件就只能对 RTS Control 位的值进行读取。

示例：假设 UART 工作在 ‘550 工业标准下，将  $\text{U0FCR}$  中的触发值设置为 0x2，此时如果启动了 Auto-RTS 模式，当接收 FIFO 内出现了 8 个字节时，UART 将使  $\overline{\text{RTS}}$  输出无效。当接收 FIFO 达到先前的触发值（4 个字节）时， $\overline{\text{RTS}}$  输出将会重新生效。

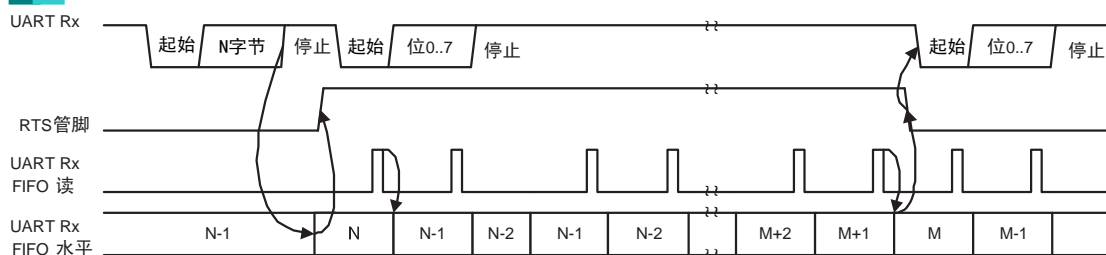


图 10.1 Auto-RTS 功能时序

## (2) Auto-CTS

Auto-CTS 功能是通过设置 CTSen 位来启动的。如果启动了 Auto-CTS 功能，在发送下一个数据字节之前，U0TSR 模块中的发送电路会先对  $\overline{\text{CTS}}$  输入进行检查。当  $\overline{\text{CTS}}$  有效（低电平）时，发送器就会将下一个字节发送出去。如果要发送器停止发送后续的字节，必须在末尾的停止位发送出一半之前释放  $\overline{\text{CTS}}$ 。在 Auto-CTS 模式下，即使 U0MSR 中的 Delta CTS 位被置位， $\overline{\text{CTS}}$  信号的变化也不会触发 Modem 状态中断，除非 CTS Interrupt Enable 位被置位。表 10.12 列出了产生 Modem 状态中断的情况。

表 10.12 Modem 状态中断产生

使能 Modem 状态中断 (U0IER[3])	CTSen (U0MCR[7])	CTS 中断使能 (U0IER[7])	Delta CTS (U0MSR[0])	Delta DCD 或后沿 RI 或 Delta DSR (U0MSR[3]或 U0MSR[2]或 U0MSR[1])	Modem 状态中断
0	x	x	x	x	否
1	0	x	0	0	否
1	0	x	1	x	是
1	0	x	x	1	是
1	1	0	x	0	否
1	1	0	x	1	是
1	1	1	0	0	否
1	1	1	1	x	是
1	1	1	x	1	是

Auto-CTS 功能可减少对主机系统的中断。当流控制被使能时， $\overline{\text{CTS}}$  状态的改变不会触发主机中断，因为器件会自动控制其发送器。不使用 Auto-CTS 时，发送器会将任何存放在发送 FIFO 内的数据都发送出去，从而导致接收器发生溢出错误。图 10.5 给出了 Auto-CTS 的功能时序。



图 10.2 Auto-CTS 功能时序

开始发送第一个字符时， $\overline{\text{CTS}}$  信号有效。一旦处理中的数据传输结束了，发送就会停止。只要  $\overline{\text{CTS}}$  无效（被拉高）时，UART 会不断地发送“1”。一旦  $\overline{\text{CTS}}$  变为无效，传输恢复且发送起始位，然后是下一个字符的数据位。

### 10.5.8 UART 线控制寄存器

U0LCR 决定了要发送和接收的数据字符格式。

表 10.13 UART 线控制寄存器（U0LCR – 0x4000 800C）位描述

位	符号	值	描述	复位值
1:0	Word Length Select	00	5 位字符长度	0
		01	6 位字符长度	
		10	7 位字符长度	
		11	8 位字符长度	
2	Stop Bit Select	0	1 个停止位	0
		1	2 个停止位（若 U0LCR[1:0]=00 时为 1.5 个停止位）	
3	Parity Enable	0	禁止校验的产生和检测	0
		1	使能校验的产生和检测	
5:4	Parity Select	00	奇校验。1s 内的发送字符数和附加校验位为奇数	0
		01	偶校验。1s 内的发送字符数和附加校验位为偶数	
		10	强制“1”奇偶校验（stick parity）	
		11	强制“0”奇偶校验（stick parity）	
6	Break Control	0	禁止间隔传输	0
		1	使能间隔传输。当 U0LCR[6]是高电平有效时，强制使输出管脚 UART TXD 为逻辑 0	
7	Divisor Latch Access Bit (DLAB)	0	禁止对除数锁存器的访问	0
		1	使能对除数锁存器的访问	
31:8	-	-	保留	-

### 10.5.9 UART 线状态寄存器

U0LSR 是一个只读寄存器，提供 UART TX 和 RX 模块的状态信息。

表 10.14 UART 线状态寄存器（U0LSR – 0x4000 8014，只读）位描述

位	符号	值	描述	复位值
0	Receiver Data Ready (RDR)	0	当 U0RBR 包含未读字符时，U0LSR[0]就会被置位；当 UART RBR FIFO 为空时，U0LSR[0]就会被清零	0
		1	U0RBR 为空 U0RBR 包含有效数据	
1	Overrun Error (OE)		一旦发生错误，就设置溢出错误条件。读 U0LSR 会清零 U0LSR[1]。当 UART RSR 已有新的字符就绪，而 UART RBR FIFO 已满时，U0LSR[1]被置位。此时，UART RBR FIFO 将不会被覆盖，UART RSR 内的字符将会丢失	0
		0	溢出错误状态无效	
		1	溢出错误状态有效	

续上表

位	符号	值	描述	复位值
2	Parity Error (PE)	0 1	当接收字符的校验位处于错误状态时,校验错误就会产生。读 U0LSR 会清零 U0LSR[2]。校验错误检测时间取决于 U0FCR[0] 注: 校验错误与 UART RBR FIRO 顶部的字符相关 校验错误状态无效 校验错误状态有效	0
3	Framing Error (FE)	0 1	当接收字符的停止位为逻辑 0 时,就会发生帧错误。读 U0LSR 会清零 U0LSR[3]。帧错误检测时间取决于 U0FCR0。当检测到有帧错误时, RX 会尝试与数据重新同步,并假设错误的停止位实际是一个超前的起始位。但即使没有出现帧错误,它也无法假设下一个接收到的字符是正确的 注: 帧错误与 UART RBR FIRO 顶部的字符相关 帧错误状态无效 帧错误状态有效	0
4	Break Interrupt (BI)	0 1	在发送整个字符(起始位、数据、校验位以及停止位)过程中, RXD1 如果保持在空闲状态(全“0”),则产生间隔中断。一旦检测到间隔条件,接收器立即进入空闲状态,直到 RXD1 进入标记状态(全“1”)。读 U0LSR 会清零该状态位。间隔检测的时间取决于 U0FCR[0] 注: 间隔中断与 UART RBR FIRO 顶部的字符相关 间隔中断状态无效 间隔中断状态有效	0
5	Transmitter Holding Register Empty (THRE)	0 1	当检测到 UART THR 已空时, THRE 就会立即被设置。写 U0THR 会清零 THRE U0THR 包含有效数据 U0THR 为空	1
6	Transmitter Empty (TEMT)	0 1	当 U0THR 和 U0TSR 同时为空时, TEMT 就会被设置;而当 U0TSR 或 U0THR 任意一个包含有效数据时, TEMT 就会被清零 U0THR 和/或 U0TSR 包含有效数据 U0THR 和 U0TSR 为空	1
7	Error in RX FIFO (RXFE)	0 1	当一个带有 RX 错误(如: 帧错误、校验错误或间隔中断)的字符载入到 U0RBR 时, U0LSR[7]就会被置位。当 U0LSR 寄存器被读取并且 UART FIFO 中不再有错误时,该位就会清零 U0RBR 中没有 UART RX 错误或 U0FCR[0]=0 UART RBR 包含至少一个 UART RX 错误	0
31:8	-	-	保留	-

### 10.5.10 UART Modem 状态寄存器



U0MSR 是一个只读寄存器，提供 Modem 输入信号的状态信息。读 U0MSR 会清零 U0MSR[3:0]。需要注意的是 Modem 信号不会对 UART 操作有直接影响，它们有助于 modem 信号操作的软件实现。

表 10.15 UART Modem 状态寄存器（U0MSR – 0x4000 8018，只读）位描述

位	符号	值	描述	复位值
0	Delta CTS	0 1	当输入端 $\overline{\text{CTS}}$ 的状态改变时，该位置位。读 U0MSR 会清零该位 没有检测到 Modem 输入端 $\overline{\text{CTS}}$ 上的状态变化 检测到 Modem 输入端 $\overline{\text{CTS}}$ 上的状态变化	0
1	Delta DSR	0 1	当输入端 $\overline{\text{DSR}}$ 的状态改变时，该位置位。读 U0MSR 会清零该位 没有检测到 Modem 输入端 $\overline{\text{DSR}}$ 上的状态变化 检测到 Modem 输入端 $\overline{\text{DSR}}$ 上的状态变化	0
2	Trailing Edge RI	0 1	当输入端 $\overline{\text{RI}}$ 上低电平到高电平跳变时，该位置位。读 U0MSR 会清零该位 没有检测到 Modem 输入端 $\overline{\text{RI}}$ 上的状态变化 检测到 $\overline{\text{RI}}$ 上的低电平往高电平跳变的变化	0
3	Delta DCD	0 1	当输入端 $\overline{\text{DCD}}$ 的状态改变时，该位置位。读 U0MSR 会清零该位 没有检测到 Modem 输入端 $\overline{\text{DCD}}$ 上的变化 检测到 Modem 输入端 $\overline{\text{DCD}}$ 上的变化	0
4	CTS		清除发送状态。输入信号 $\overline{\text{CTS}}$ 的补码。在 Modem 回送模式下，该位连接到 U0MCR[1]	0
5	DSR		数据设置就绪状态。输入信号 $\overline{\text{DSR}}$ 的补码。在 Modem 回送模式下，该位连接到 U0MCR[0]	0
6	RI		响铃指示状态。输入 $\overline{\text{RI}}$ 的补码。在 Modem 回送模式下，该位连接到 U0MCR[2]	0
7	DCD		数据载波检测状态。输入 $\overline{\text{DCD}}$ 的补码。在 Modem 回送模式下，该位连接到 U0MCR[3]	0

### 10.5.11 UART 高速缓存寄存器

U0SCR 不会对 UART 操作有影响。用户可自由对该寄存器进行读写。不提供中断接口向主机指示 U0SCR 所发生的读或写操作。

表 10.16 UART 高速缓存寄存器（U0SCR – 0x4000 801C）位描述

位	符号	描述	复位值
7:0	Pad	一个可读、可写的字节	0x00
31:8	-	保留	-

### 10.5.12 UART Auto-baud 控制寄存器

在用户测量波特率的输入时钟/数据速率期间，整个测量过程就是由 UART Auto-baud 控制寄存器（U0ACR）进行控制的。用户可自由地读写该寄存器。

表 10.17 Auto-baud 控制寄存器（U0ACR – 0x4000 8020）位描述



位	符号	值	描述	复位值
0	Start	0 1	在 auto-baud 功能结束后，该位会自动清零 Auto-baud 功能停止（auto-baud 功能不运行） Auto-baud 功能启动（auto-baud 功能正在运行）。 Auto-baud 运行位。该位会在 auto-baud 功能结束后自动清零	0
1	Mode	0 1	Auto-baud 模式选择位 模式 0 模式 1	0
2	AutoRestart	0 1	不重新启动 如果超时则重新启动（计数器会在下一个 UART Rx 下降沿重新启动）	0
7:3	-	NA	保留。用户软件不应对其写入 1。从保留位读出的值未定义	0
8	ABEOIntClr	0 1	Auto-baud 中断结束清零位（仅可写访问） 写 0 无影响 写 1 将 U0IIR 中相应的中断清除	0
9	ABTOIntClr	0 1	Auto-baud 超时中断清零位（仅可写访问） 写 0 无影响 写 1 将 U0IIR 中相应的中断清除	0
31:10	-	NA	保留。用户软件不应对其写入 1。从保留位读出的值未定义	0

### 10.5.13 自动波特率（Auto-baud）

UART auto-baud 功能可用于测量基于“AT”协议（Hayes 命令）的输入波特率。如果 auto-baud 特性被使能，那么 auto-baud 特性将测量接收数据流中的位时间，并根据这个结果来设置除数锁存寄存器 U0DLM 和 U0DLL。

Auto-baud 功能是通过置位 U0ACR 起始位来启动的，并通过清零 U0ACR 起始位来停止。Auto-baud 一旦结束，起始位就将自动清零，并且对该起始位进行读取将会返回 auto-baud 的状态（等待处理/完成）。

可通过设置 U0ACR 模式位来使用两种 auto-baud 测量模式。在模式 0 下，波特率是通过对 UART RX 管脚上两个连续的下降沿进行测量（起始位的下降沿和最低有效位的下降沿）来得到的。而在模式 1 下，波特率则是通过测量 UART RX 管脚上的下降沿和后续的上升沿之间的时间（起始位的长度）来得到的。

如果出现超时（速率测量计数器溢出），U0ACR AutoRestart 位可用于自动重启波特率测量。如果该位被置位，速率测量将会在 UART Rx 管脚的下一个下降沿重新启动。

Auto-baud 功能会产生两种中断：

- [1] 若中断使能，则 U0IIR ABTOInt 中断将置位（U0IER ABTOIntEn 置位且自动波特率测量计数器溢出）；
- [2] 若中断使能，则 U0IIR ABEOInt 中断将置位（U0IER ABEOIntEn 置位且 auto-baud 已经成功完成）。

Auto-baud 中断必须通过置位相应的 U0ACR ABTOIntClr 位和 ABEOIntEn 位来清零。

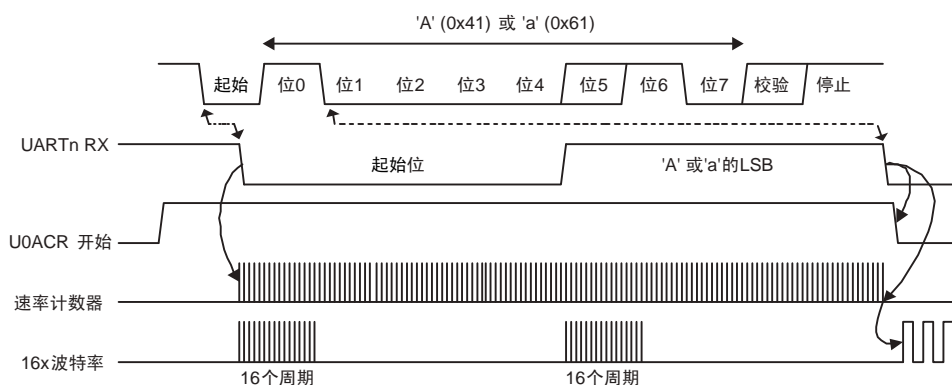
在 auto-baud 期间，小数波特率发生器通常被禁用（即 DIVADDVAL = 0）。同时，当使用 auto-baud 时，任何对 U0DLM 和 U0DLL 寄存器的写操作都必须在写 U0ACR 寄存器之前完成。UART 支持的最小和最大波特率受 UART\_PCLK、数据的位数、停止位以及校验位的影响。

$$\text{ratemin} = \frac{2 \times \text{PCLK}}{16 \times 2} \leq \text{UART}_{\text{baudrate}} \leq \frac{\text{PCLK}}{16 \times (2 + \text{databits} + \text{paritybits} + \text{stopbits})} = \text{ratemax} \quad (6)$$

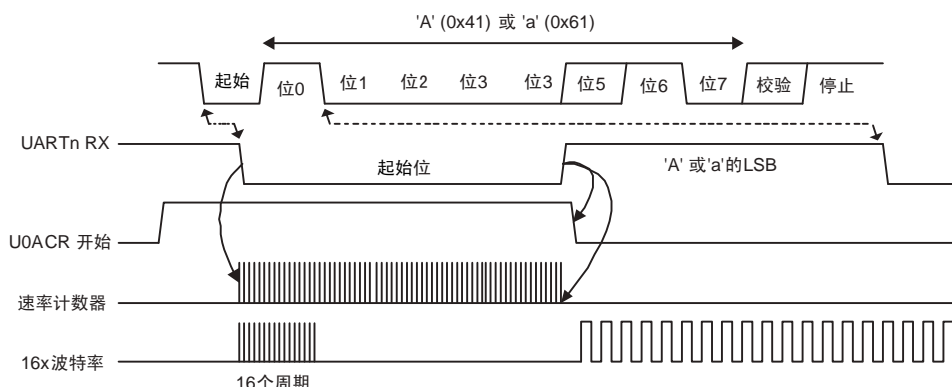
#### 10.5.14 Auto-baud 模式

当软件期望“AT”命令时，它采用期望的字符格式对 UART 进行配置，并置位 U0ACR 起始位。用户不必关心除数锁存器 U0DLM 和 U0DLL 内的初始值。由于字母“A”或“a”ASCII 编码（“A”= 0x41，“a”= 0x61）的关系，UART Rx 管脚所检测的起始位以及期望字符的 LSB 是由两个下降沿来限定的。当 U0ACR 起始位被置位时，auto-baud 协议将执行以下阶段：

- 第 1 步：U0ACR 起始位一置位，波特率测量计数器立即复位，同时 UART U0RSR 复位。U0RSR 波特率切换为最高速率。
- 第 2 步：UART Rx 管脚上的下降沿触发起始位的开始。波特率测量计数器将开始对 UART\_PCLK 周期进行计数。
- 第 3 步：在接收起始位的过程中，RSR 波特率输入端会产生 16 个脉冲，脉冲频率和 UART 输入时钟相同，这样，保证了起始位存放在 U0RSR 中。
- 第 4 步：在接收起始位的过程中（以及模式 0 下的字符 LSB），速率计数器将随着被预分频的 UART 输入时钟（UART\_PCLK）递增。
- 第 5 步：如果是模式 0，那么速率计数器将会在 UART Rx 管脚的下一个下降沿停止。如果是模式 1，那么速率计数器将会在 UART Rx 管脚的下一个上升沿停止。
- 第 6 步：速率计数器的值被载入到 U0DLM/U0DLL 中，并且波特率将会切换到正常操作模式。在设置完 U0DLM/U0DLL 后，如果 auto-baud 结束中断被使能，U0IIR ABEOInt 将会被置位。接着 U0RSR 继续接收“A/a”字符剩下的其它位。



a. 模式 0 (起始位和LSB均用于auto-baud功能)



b. 模式 1 (只有起始位是用于auto-baud功能)

图 10.3 自动波特率 a) 模式 0 和 b) 模式 1 的波形图

### 10.5.15 UART 小数分频器寄存器

UART 小数分频寄存器 (U0FDR) 控制产生波特率的时钟预分频器, 并且用户可自由对该寄存器进行读写操作。该预分频器使用 APB 时钟并根据指定的小数要求产生输出时钟。

要点: 如果小数分频器有效 ( $\text{DIVADDVAL} > 0$ ) 且  $\text{DLM} = 0$ , 则 DLL 寄存器的值必须大于或等于 3。

表 10.18 UART 小数分频寄存器 (U0FDR – 0x4000 8028) 位描述

位	功能	值	描述	复位值
3:0	DIVADDVAL	0	产生波特率的预分频除数值。如果该字段为 0, 小数波特率发生器将不会影响 UART 的波特率	0
7:4	MULVAL	1	波特率预分频乘数值。不管是否使用小数波特率发生器, 为了让 UART 正常运作, 该字段必须大于或等于 1	1
31:8	-	NA	保留。用户软件不应对其写入 1。从保留位读出的值未定义	0

该寄存器控制产生波特率的时钟预分频器。寄存器的复位值会让 UART 的小数功能保持在禁用状态, 从而确保 UART 在软件和硬件方面能够与不具备该特性的 UART 完全兼容。

可如下计算 UART 波特率:

$$\text{UART 波特率} = \frac{\text{PCLK}}{16 \times (256 \times \text{U0DLM} \times \text{U0DLL}) \times \left( 1 + \frac{\text{DivAddVal}}{\text{MulVal}} \right)} \quad (7)$$

其中 UART\_PCLK 为外设时钟, U0DLM 和 U0DLL 为标准 UART 波特率分频器寄存器, 而 DIVADDVAL 和 MULVAL 为 UART 小数波特率发生器的特定参数。

MULVAL 和 DIVADDVAL 的值应符合以下条件:

- 1)  $1 < \text{MULVAL} \leq 15$ ;
- 2)  $0 \leq \text{DIVADDVAL} \leq 14$ ;
- 3)  $\text{DIVADDVAL} < \text{MULVAL}$ 。

U0FDR 的值在发送/接收数据的过程中不应进行更改, 否则可能会导致数据丢失或损坏。

如果 U0FDR 寄存器值不满足上述两个要求, 那么小数分频器输出则为未定义。如果 DIVADDVAL 为 0, 那么小数分频器将被禁能, 并且不会对时钟进行分频。

### 1. 波特率计算

UART 可以与小数分频器一起工作, 也可以不使用小数分频器。在实际应用中, 使用几种不同的小数分频器设置很可能会获得期望的波特率。下面的算法描绘了查找一组 DLM、DLL、MULVAL 以及 DIVADDVAL 值的方法。这组参数产生的波特率与期望值的相对误差小于 1.1%。

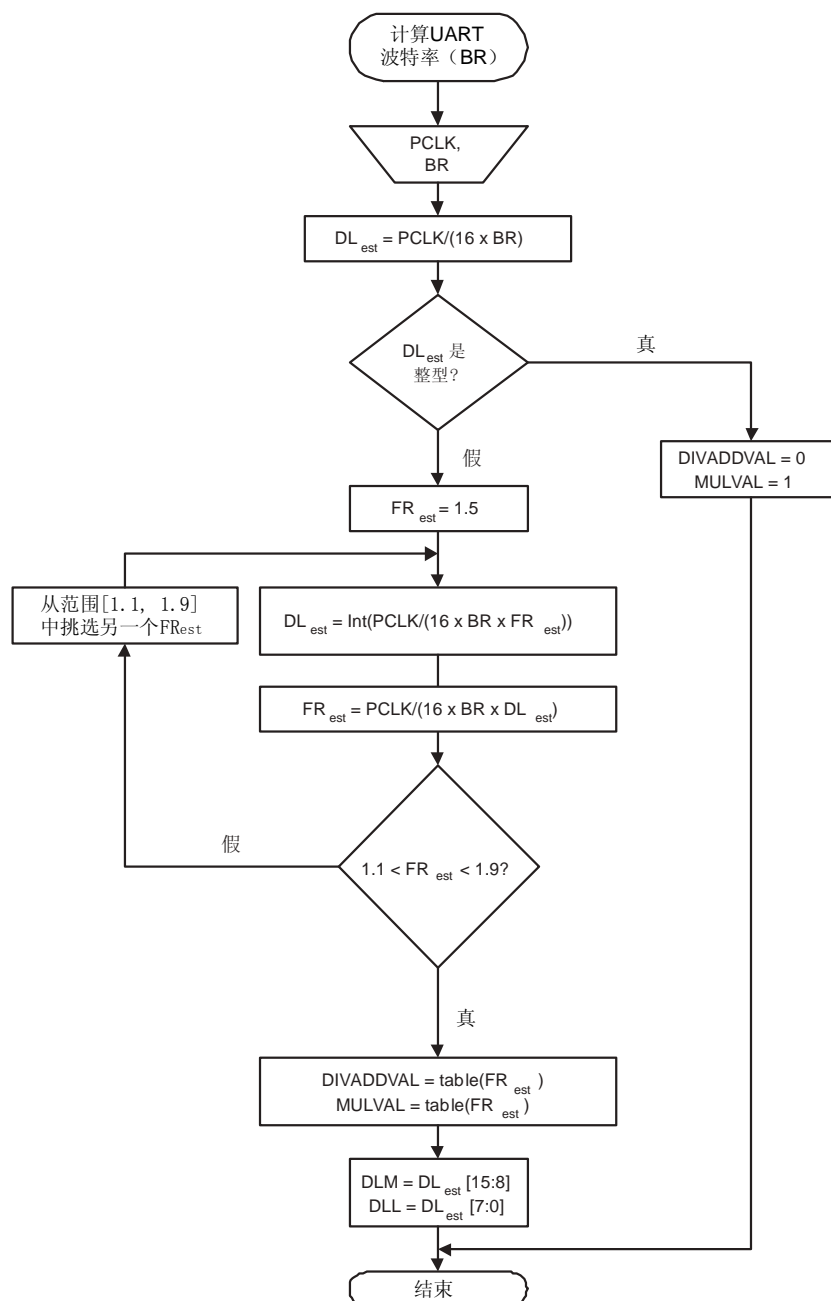


图 10.4 设置 UART 分频器的算法

表 10.19 小数分频器设置查找表

FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal
1.000	0/1	1.250	1/4	1.500	1/2	1.750	3/4
1.067	1/15	1.267	4/15	1.533	8/15	1.769	10/13
1.071	1/14	1.273	3/11	1.538	7/13	1.778	7/9
1.077	1/13	1.286	2/7	1.545	6/11	1.786	11/14
1.083	1/12	1.300	3/10	1.556	5/9	1.800	4/5

续上表

FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal
1.091	1/11	1.308	4/13	1.571	4/7	1.818	9/11
1.100	1/10	1.333	1/3	1.583	7/12	1.833	5/6
1.111	1/9	1.357	5/14	1.600	3/5	1.846	11/13
1.125	1/8	1.364	4/11	1.615	8/13	1.857	6/7
1.133	2/15	1.375	3/8	1.625	5/8	1.867	13/15
1.143	1/7	1.385	5/13	1.636	7/11	1.875	7/8
1.154	2/13	1.400	2/5	1.643	9/14	1.889	8/9
1.167	1/6	1.417	5/12	1.667	2/3	1.900	9/10
1.182	2/11	1.429	3/7	1.692	9/13	1.909	10/11
1.200	1/5	1.444	4/9	1.700	7/10	1.917	11/12
1.214	3/14	1.455	5/11	1.714	5/7	1.923	12/13
1.222	2/9	1.462	6/13	1.727	8/11	1.929	13/14
1.231	3/13	1.467	7/15	1.733	11/15	1.933	14/15

(1) 范例 1: UART\_PCLK = 14.7456 MHz, BR = 9600

根据所提供的算法,  $D_{Lest} = PCLK / (16 \times BR) = 14.7456 \text{ MHz} / (16 \times 9600) = 96$ 。因为这里的  $D_{Lest}$  是一个整数, 所以  $DIVADDVAL = 0$ ,  $MULVAL = 1$ ,  $DLM = 0$  且  $DLL = 96$ 。

(2) 范例 2: UART\_PCLK = 12 MHz, BR = 115200

根据所提供的算法  $D_{Lest} = PCLK / (16 \times BR) = 12 \text{ MHz} / (16 \times 115200) = 6.51$ 。该算式中的  $D_{Lest}$  并不是整数, 因此下一步就要对 FR 参数进行估算。使用  $F_{Rest} = 1.5$  进行首次估算, 得到新的  $D_{Lest} = 4$ , 然后使用  $F_{Rest} = 1.628$  再进行计算。由于  $F_{Rest} = 1.628$  是在 1.1 到 1.9 的指定范围之内, 因此  $DIVADDVAL$  和  $MULVAL$  的值可通过附带的查找表获得。

在表 10.19 中, 最接近  $F_{Rest} = 1.628$  的值为  $FR = 1.625$ 。也就是说  $DIVADDVAL = 5$  而  $MULVAL = 8$ 。

基于这些查找结果, 建议 UART 设置为  $DLM = 0$ ,  $DLL = 4$ ,  $DIVADDVAL = 5$  和  $MULVAL = 8$ 。根据上面的公式(2), UART 的波特率为 115384。该速率与原来指定的 115200 之间存在 0.16% 的相对误差。

### 10.5.16 UART 发送使能寄存器

除了配备完整的硬件流控制 (上述的 auto-CTS 和 auto-RTS 机制) 之外, U0TER 还可以实现软件流控制。当  $TxEn=1$  时, 只要数据可用, UART 发送器就会一直发送数据。一旦  $TxEn$  变为 0, UART 就会停止数据传输。

虽然表 10.20 描述了如何利用  $TxEn$  位来实现软件流控制, 但我们强烈建议用户采用 UART 硬件所实现的自动流控制特性处理软件流控制, 并限制  $TxEn$  位对软件流控制的范围。

表 10.20 描述了如何利用  $TxEn$  位来实现软件流控制。

表 10.20 UART 发送使能寄存器 (U0TER – 0x4000 8030) 位描述

位	符号	描述	复位值
6:0	-	保留。用户软件不应对其写入 1。从保留位读出的值未定义	NA

续上表

位	符号	描述	复位值
7	TxEN	该位为 1 时（复位后），一旦先前的数据都被发送出去后，写入 THR 的数据就会在 TxD 管脚上输出。如果在发送某字符时该位被清零，那么在将该字符发送完毕后就不再发送数据，直到该位被置“1”。也就是说，该位为 0 时会阻止字符从 THR 或 TX FIFO 传输到发送移位寄存器。当检测到硬件握手 TX-permit 信号（ $\overline{\text{CTS}}$ ）变为假时，或者在接收到 XOFF 字符（DC3）时，软件通过执行软件握手可以将该位清零。当检测到 TX-permit 信号变为真时，或者在接收到 XON 字符（DC1）时，软件又能将该位重新置位	1
31:8	-	保留	-

### 10.5.17 UART RS485 控制寄存器

U0RS485CTRL 寄存器控制 UART 在 RS-485/EIA-485 模式下的配置。

表 10.21 UART RS485 控制寄存器（U0RS485CTRL – 0x4000 804C）位描述

位	符号	值	描述	复位值
0	NMMEN	0 1	RS-485/EIA-485 普通多点模式（NMM）禁能 使能 RS-485/EIA-485 普通多点模式（NMM）。在该模式下，当接收字符使 UART 设置校验错误并产生中断时，对地址进行检测	0
1	RXDIS	0 1	使能接收器 禁能接收器	0
2	AADEN	0 1	禁能自动地址检测（AAD） 使能自动地址检测（AAD）	0
3	SEL	0 1	如果使能了方向控制（位 DCTRL =1），管脚 $\overline{\text{RTS}}$ 会被用于方向控制 如果使能了方向控制（位 DCTRL =1），管脚 $\overline{\text{DTR}}$ 会被用于方向控制	0
4	DCTRL	0 1	禁能自动方向控制 使能自动方向控制	0
5	OINV	0 1	该位保留了 $\overline{\text{RTS}}$ （或 $\overline{\text{DTR}}$ ）管脚上方向控制信号的极性 当发送器有数据要发送时，方向控制管脚会被驱动为逻辑“0”。在最后一个数据位被发送出去后，该位就会被驱动为逻辑“1” 当发送器有数据要发送时，方向控制管脚就会被驱动为逻辑“1”。在最后一个数据位被发送出去后，该位就会被驱动为逻辑“0”	0
31:6	-	-	保留。用户软件不应对其写入 1。从保留位读出的值未定义	NA

### 10.5.18 UART RS-485 地址匹配寄存器

U0RS485ADRMATCH 寄存器包含了 RS-485/EIA-485 模式的地址匹配值。

表 10.22 UART RS-485 地址匹配寄存器（U0RS485ADRMATCH – 0x4000 8050）位描述

位	符号	描述	复位值
7:0	ADRMATCH	包含了地址匹配值	0x00
31:8	-	保留	-

### 10.5.19 UART1 RS-485 延时值寄存器

对于最后一个停止位离开 TXFIFO 到撤销  $\overline{\text{RTS}}$ （或  $\overline{\text{DTR}}$ ）信号之间的延时，用户在 8 位的 RS485DLY 寄存器内进行设定。该延迟时间是以波特率时钟周期为单位的。可设定任何从 0 到 255 位时间的延时。

表 10.23 UART RS-485 延时值寄存器（U0RS485DLY – 0x4000 8054）位描述

位	符号	描述	复位值
7:0	DLY	包含了方向控制（RTS 或 DTR）延时值。该寄存器与 8 位计数器一起工作	0x00
31:8	-	保留，用户软件不应向保留位写入 1。从保留位读出的值未定义	NA

### 10.5.20 RS-485/EIA-485 模式的操作

RS-485/EIA-485 特性允许 UART 配置为可寻址的从机。可寻址从机是其中一个由同一主机控制的多从机。

UART 主机发送器通过将校验位（第 9 位）置“1”来标识地址字符。而对于数据字符，校验位会被置“0”。

每一个 UART 从机接收器都会被配给一个唯一地址。从机可设定为手动或自动丢弃那些地址与本机地址不相符的数据。

#### 1. RS-485/EIA-485 正常多点模式（NMM）

该模式是通过置位 RS485CTRL 的位 0 来启动的。在该模式下，当接收到的数据字节使 UART 设置校验错误并产生中断时，就会进行地址检测。

如果接收器被禁能（RS485CTRL 位 1=“1”），任何接收到的数据字节都会被忽略且不会存放到 RXFIFO 中。当检测到地址字节（校验位=“1”）时，接收到的数据就会被存放到 RXFIFO 中，并产生 Rx 数据就绪中断。此时，处理器可读出地址字节，并决定是否使能接收器接收后面的数据。

当接收器被使能（RS485CTRL 位 1=“0”）时，所有接收到的字节（无论是数据还是地址）都会被存放到 RXFIFO 中。当接收到地址字符时，校验错误中断就会产生，同时处理器会决定是否将接收器禁能。

#### 2. RS-485/EIA-485 自动地址检测（AAD）模式

当 RS485CTRL 寄存器的位 0（9 位模式使能）和位 2（AAD 模式使能）同时置位时，UART 就会处于自动地址检测模式。

在该模式下，接收器会将任何接收到的地址字符（校验位=“1”）与 RS485ADRMATCH 寄存器中设定的 8 位地址值进行比较。

如果接收器被禁能（RS485CTRL 位 1=“1”），则任何接收到的字节，无论是数据字节还是地址字节，只要与 RS485ADRMATCH 的值不匹配，都会被丢弃。



当检测到匹配的地址字符时，地址字符和校验位就会被存放到 RXFIFO 中，此外，接收器将会自动使能（RS485CTRL 位 1 将会由硬件清零）。接收器还会产生 Rx 数据就绪中断。

当接收器被使能（RS485CTRL 位 1=“0”）时，所有接收到的字节都会被存放到 RXFIFO 中，直到接收到与 RS485ADRMATCH 的值不匹配的地址字符为止。当出现不匹配时，接收器会通过硬件自动禁能（RS485CTRL 位 1 将会置位）。所接收到的非匹配地址字符将不会被存放到 RXFIFO 中。

### 3. RS-485/EIA-485 自动方向控制

RS485/EIA-485 模式包括选择是否允许发送器自动控制 DIR 管脚的状态作为方向控制输出信号。

该特性是通过将 RS485CTRL 的位 4 置“1”来使能。

如果方向控制使能，当 RS485CTRL 的位 3=0 时，则使用  $\overline{\text{RTS}}$  管脚。当 RS485CTRL 的位 3=1 时，则使用  $\overline{\text{DTR}}$  管脚。

当自动方向控制使能时，被选中的管脚在 CPU 写数据到 TXFIFO 时就会拉低（驱动为低电平）。最后一个数据位一旦发送出去，选中的管脚就会被拉高（驱动为高电平）。请参见 RS485CTRL 寄存器的位 4 和位 5。

除了回送模式，RS485CTRL 位 4 对方向控制管脚的控制优先于所有其它机制。

### 4. RS485/EIA-485 驱动器延时

驱动器延时是最后一个停止位移出 TXFIFO 和撤销  $\overline{\text{RTS}}$  信号之间的延时。该延迟时间是以波特率时钟周期为单位的。可设定任何从 0 到 255 位时间的延时。

### 5. RS485/EIA-485 输出反置

$\overline{\text{RTS}}$ （或  $\overline{\text{DTR}}$ ）管脚上的方向控制信号极性可通过设置 U0RS485CTRL 寄存器的位 5 来改变。当该位置位时，方向控制管脚在发送器有数据要发送时就会驱动为逻辑“1”。在最后一个数据位发送出去后，方向控制管脚就会驱动为逻辑“0”。

## 10.5.21 UART FIFO 深度寄存器

U0FIFOLVL 寄存器是一个只读寄存器，允许软件读取当前的 FIFO 深度状态。发送和接收 FIFO 的深度均存放在该寄存器中。

表 10.24 UART FIFO 深度寄存器（U0FIFOLVL - 0x4000 8058，只读）位描述

位	符号	描述	复位值
3:0	RXFIFILVL	反映 UART 接收 FIFO 的当前水平 0=空，0xF=FIFO 为满	0x00
7:4	-	保留。从保留位读出的值未定义	NA
11:8	TXFIFOLVL	反映 UART 发送 FIFO 的当前水平 0=空，0xF=FIFO 为满	0x00
31:12	-	保留。从保留位读出的值未定义	NA

## 10.6 结构

UART 的结构如下面的结构框图所示。

APB 接口提供了 CPU 或主机与 UART 之间的通信连接。

UART 接收器模块 (U0RX) 监控串行输入线 RXD 的有效输入。UART RX 移位寄存器 (U0RSR) 通过 RXD 接收有效字符。当 U0RSR 接收到一个有效字符时, 它将该字符传送到 UART RX 缓冲寄存器 FIFO 中, 等待 CPU 或主机通过通用主机接口进行访问。

UART 发送器模块 U0TX 接收 CPU 或主机写入的数据, 并且将数据缓存到 UART TX 保持寄存器 FIFO (U0THR) 中。UART TX 移位寄存器 (U0TSR) 读出存放在 U0THR 中的数据, 并对数据进行汇编, 通过串行输出管脚 TXD1 发送出去。

UART 波特率发生器模块 U0BRG 产生 UART TX 模块所使用的时序。U0BRG 时钟输入源为 UART\_PCLK。主时钟由 U0DLL 和 U0DLM 寄存器中所指定的除数分频。分频所得的时钟为过采样时钟 NBAUDOUT 的 16 倍。

中断接口包括 U0IER 和 U0IIR 寄存器。中断接口接收若干个由 U0TX 和 U0RX 模块发出的单时钟宽度的使能信号。

U0TX 和 U0RX 所发送的状态信息会被存放到 U0LSR 中。U0TX 和 U0RX 的控制信息会被存放到 U0LCR 中。

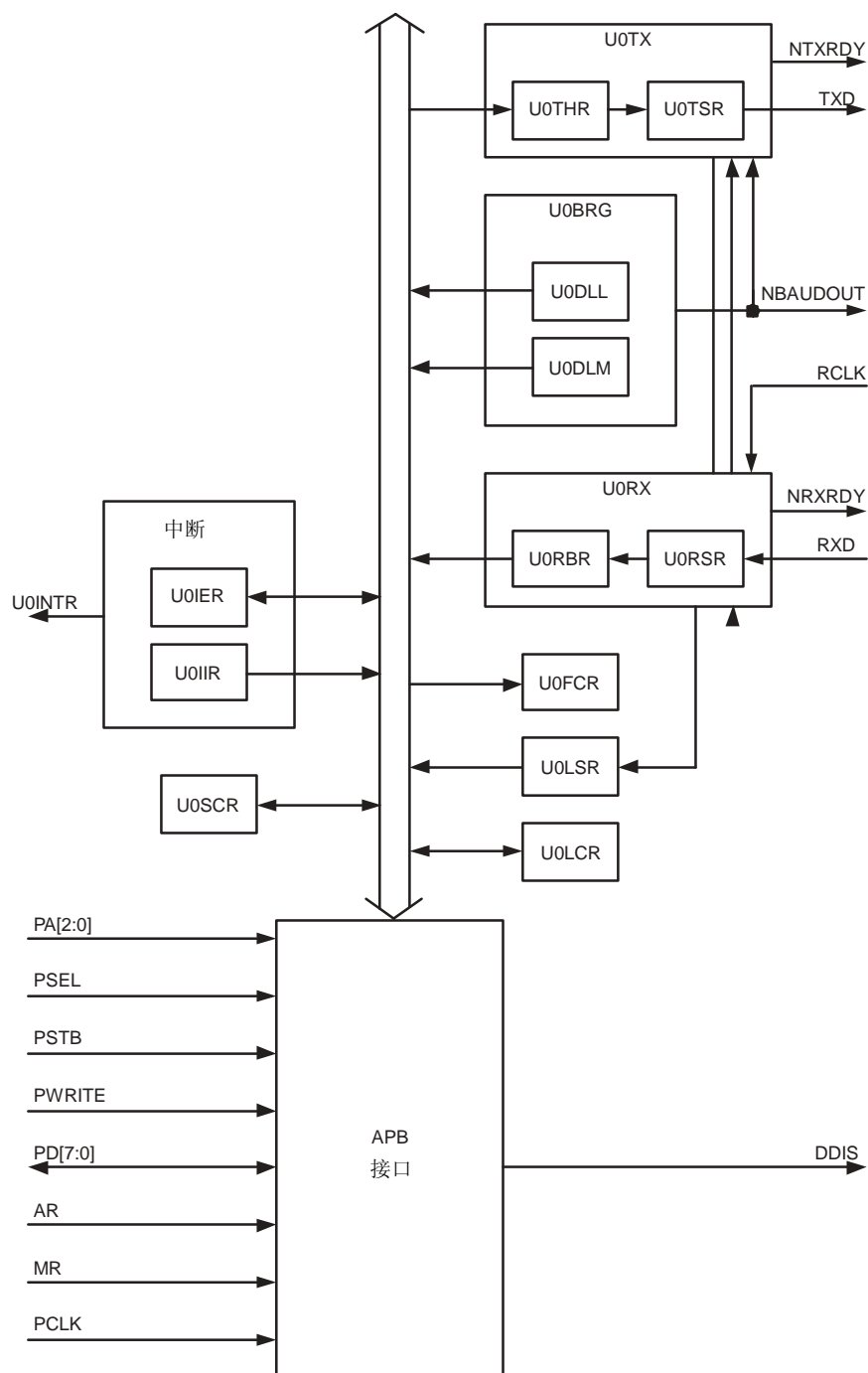


图 10.5 UART 方框图