

---

---

## 第 21 章 UART

---

---

### 目录

本章包括下列主题：

21.1	简介 .....	21-2
21.2	控制寄存器 .....	21-3
21.3	UART 波特率发生器 (BRG) .....	21-9
21.4	UART 配置 .....	21-13
21.5	UART 发送器 .....	21-14
21.6	UART 接收器 .....	21-18
21.7	使用 UART 进行 9 位通信 .....	21-21
21.8	接收间隔字符 .....	21-23
21.9	初始化 .....	21-23
21.10	UART 的其他特性 .....	21-25
21.11	在 CPU 休眠和空闲模式下的 UART 操作 .....	21-27
21.12	UxCTS 和 UxRTS 控制引脚的操作 .....	21-29
21.13	红外线支持 .....	21-31
21.14	与 UART 模块相关的寄存器 .....	21-34
21.15	电气规范 .....	21-35
21.16	设计技巧 .....	21-36
21.17	相关应用笔记 .....	21-37
21.18	版本历史 .....	21-38

## 21.1 简介

通用异步收发器（Universal Asynchronous Receiver Transmitter, UART）模块是 PIC24F 器件系列提供的串行 I/O 模块之一。UART 是可以与外设器件和个人电脑（使用 RS-232、RS-485、LIN 1.2 和 IrDA® 等协议）通信的全双工异步通信通道。模块还通过 UxCTS 和 UxRTS 引脚支持硬件流控制选项，并且还包含有 IrDA 编码器和解码器。

UART 模块的主要特性有：

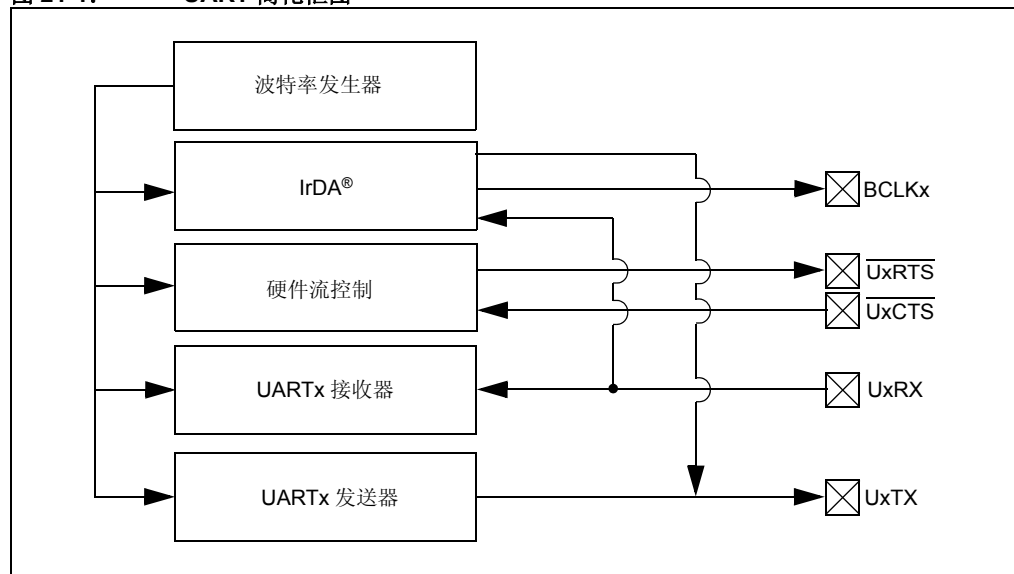
- 通过 UxTX 和 UxRX 引脚进行全双工 8 位或 9 位数据传输
- 偶、奇或无奇偶校验选项（对于 8 位数据）
- 一或两个停止位
- 硬件自动波特率特性
- 通过 UxCTS 和 UxRTS 引脚支持硬件流控制选项
- 完全集成的具有 16 位预分频器的波特率发生器（Baud Rate Generator）
- 16 MIPS 时，波特率范围为 1 Mbps 到 15 bps
- 4 级深度先进先出（First-In-First-Out, FIFO）发送数据缓冲器
- 4 级深度 FIFO 接收数据缓冲器
- 奇偶校验、帧和缓冲器溢出错误检测
- 支持带地址检测的 9 位模式（第 9 位 = 1）
- 发送和接收中断
- 用于诊断支持的环回（Loopback）模式
- IrDA 编码器和解码器逻辑
- LIN 1.2 协议支持
- 用于支持 IrDA 编码器 / 解码器的 16 倍频波特率时钟输出

**注：** 每个不同的 PIC24F 器件可能具有一个或多个 UART 模块。在引脚、控制 / 状态位和寄存器的名称中使用的“x”表示特定的模块。更多详细信息，请参见具体器件数据手册。

图 21-1 所示为 UART 的简化框图。UART 模块由以下主要硬件元件组成：

- 波特率发生器
- 异步发送器
- 异步接收器

**图 21-1: UART 简化框图**



## 21.2 控制寄存器

寄存器 21-1: UxMODE: UARTx 模式寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UARTEN	UFRZ	USIDL	IREN	RTSMD	ALTIO <sup>(1)</sup>	UEN1	UEN0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL
bit 7							bit 0

## 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 值

1 = 置 1

0 = 清零

x = 未知

bit 15

**UARTEN:** UARTx 使能位

1 = UARTx 使能; UEN&lt;1:0&gt; 和 UTXEN 控制位定义了 UARTx 如何控制 UARTx 引脚。

0 = UARTx 禁止; UARTx 引脚由相应的 PORT、LAT 和 TRIS 位控制。

bit 14

**UFRZ:** 调试模式停止位

1 = 仿真器处于调试模式时, 模块停止工作

0 = 仿真器处于调试模式时, 模块继续工作

bit 13

**USIDL:** 空闲模式停止位

1 = 器件进入空闲模式后停止工作

0 = 处于空闲模式时继续工作

bit 12

**IREN:** IrDA 编码器和解码器使能位

1 = IrDA 编码器和解码器使能

0 = IrDA 编码器和解码器禁止

bit 11

**RTSMD:** UxRTS 引脚模式选择位

1 = UxRTS 处于单工 (Simplex) 模式

0 = UxRTS 处于流控制模式

bit 10

**ALTIO:** UARTx 备用 I/O 选择位<sup>(1)</sup>

1 = UART 通过 UxATX 和 UxARX I/O 引脚通信

0 = UARTx 通过 UxTX 和 UxRX I/O 引脚通信

bit 9-8

**UEN<1:0>:** UARTx 使能位

11 = 使能并使用 UxTX、UxRX 和 BCLKx 引脚; UxCTS 引脚由端口锁存器控制

10 = 使能并使用 UxTX、UxRX、UxCTS 和 UxRTS 引脚

01 = 使能并使用 UxTX、UxRX 和 UxRTS 引脚; UxCTS 引脚由端口锁存器控制

00 = 使能并使用 UxTX 和 UxRX 和引脚; UxCTS、UxRTS 和 BCLKx 引脚由端口锁存器控制

bit 7

**WAKE:** 在休眠模式期间检测到启动位唤醒使能位

1 = 使能唤醒

0 = 禁止唤醒

bit 6

**LPBACK:** UARTx 环回模式选择位

1 = 使能环回模式

0 = 禁止环回模式

bit 5

**ABAUD:** 自动波特率使能位

1 = 在下一个字符使能波特率检测。需要收到“同步”字段 (55h); 完成时由硬件清零。

0 = 禁止波特率检测或检测已完成

注 1: 备用 UART I/O 引脚并不是在所有器件上都可用。请参见器件数据手册了解详细信息。

# PIC24F 系列参考手册

---

## 寄存器 21-1: UxMODE: UARTx 模式寄存器 (续)

bit 4	<b>RXINV:</b> 接收奇偶校验翻转位 1 = UxRX 空闲状态为 0 0 = UxRX 空闲状态为 1
bit 3	<b>BRGH:</b> 高波特率选择位 1 = 高速 0 = 低速
bit 2-1	<b>PDSEL&lt;1:0&gt;:</b> 奇偶校验和数据选择位 11 = 9 位数据, 无奇偶校验 10 = 8 位数据, 奇校验 01 = 8 位数据, 偶校验 00 = 8 位数据, 无奇偶校验
bit 0	<b>STSEL:</b> 停止选择位 1 = 2 个停止位 0 = 1 个停止位

**注 1:** 备用 UART I/O 引脚并不是在所有器件上都可用。请参见器件数据手册了解详细信息。

寄存器 21-2: UxSTA: UARTx 状态和控制寄存器

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R-0	R-1
UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/C-0	R-0
URXISEL1	URXISEL0	ADDEN	RIDL	PERR	FERR	OERR	URXDA
bit 7							bit 0

<b>图注:</b>	C = 可清零位		
R = 可读位	W = 可写位	U = 未实现位, 读为 0	
-n = POR 值	1 = 置 1	0 = 清零	x = 未知

- bit 15,13 **UTXISEL<1:0>** 发送中断模式选择位  
 11 = 保留  
 10 = 当一个字符被传输到发送移位寄存器并且发送缓冲器变为空时, 产生中断  
 01 = 当最后一次发送完成 (最后一个字符移出发送移位寄存器) 且所有的发送操作均完成时, 产生中断  
 00 = 当任意字符被传输到发送移位寄存器时产生中断 (这意味着在发送缓冲器中至少有一个单元为空)
- bit 14 **UTXINV**: 发送奇偶校验翻转位  
**IREN = 0:**  
 1 = UxTX 空闲状态为 1  
 0 = UxTX 空闲状态为 0  
**IREN = 1:**  
 1 = IrDA 编码 UxTX 空闲状态为 1  
 0 = IrDA 编码 UxTX 空闲状态为 0
- bit 12 **未实现**: 读为 0
- bit 11 **UTXBRK**: 发送间隔位  
 1 = 无论发送状态如何, 都将 UxTX 引脚驱动为低电平 (同步间隔发送——启动位后跟随 12 个 0, 之后跟随 1 个停止位)  
 0 = 同步间隔发送被禁止或已完成
- bit 10 **UTXEN**: 发送使能位  
 1 = UARTx 发送器使能, UARTx 控制 UxTX 引脚 (如果 UARTEN = 1)  
 0 = UARTx 发送器禁止, 中止所有等待的发送, 缓冲器复位。PORT 控制 UxTX 引脚。
- bit 9 **UTXBF**: 发送缓冲器满状态位 (只读)  
 1 = 发送缓冲器已满  
 0 = 发送缓冲器未满, 可以写入至少一个或多个数据字
- bit 8 **TRMT**: 发送移位寄存器空位 (只读)  
 1 = 发送移位寄存器为空, 同时发送缓冲器为空 (上一个发送已经完成)  
 0 = 发送移位寄存器非空, 发送在进行中或在发送缓冲器中排队
- bit 7-6 **URXISEL<1:0>**: 接收中断模式选择位  
 11 = 接收缓冲器满时 (即, 有 4 个数据字符), 中断标志位置 1  
 10 = 接收缓冲器 3/4 满时 (即, 有 3 个数据字符), 中断标志位置 1  
 0x = 当接收到一个字符时, 中断标志位置 1
- bit 5 **ADDEN**: 地址字符检测位 (接收数据的第 8 位 = 1)  
 1 = 地址检测模式使能。如果没有选择 9 位模式, 该控制位无效。  
 0 = 地址检测模式禁止

# PIC24F 系列参考手册

---

## 寄存器 21-2: UxSTA: UARTx 状态和控制寄存器 (续)

- bit 4      **RIDLE:** 接收器空闲位 (只读)  
1 = 接收器空闲  
0 = 正在接收数据
- bit 3      **PERR:** 奇偶校验错误状态位 (只读)  
1 = 检测到当前字符的奇偶校验错误  
0 = 未检测到奇偶校验错误
- bit 2      **FERR:** 帧错误状态位 (只读)  
1 = 检测到当前字符的帧错误  
0 = 未检测到帧错误
- bit 1      **OERR:** 接收缓冲器溢出错误状态位 (清零 / 只读)  
1 = 接收缓冲器溢出  
0 = 接收缓冲器未溢出 (清零前置置 1 的 OERR 位会将接收缓冲器和 RSR 复位为空状态)
- bit 0      **URXDA:** 接收缓冲器是否有数据位 (只读)  
1 = 接收缓冲器中有数据, 有至少一个或多个字符可读  
0 = 接收缓冲器为空

**寄存器 21-3: UxRXREG: UARTx 接收寄存器**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0
—	—	—	—	—	—	—	URX8
bit 15							bit 8
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
URX<7:0>							
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = POR 值	1 = 置 1	0 = 清零
		x = 未知

- bit 15-9      **未实现:** 读为 0
- bit 8        **URX8:** 接收到字符的第 8 个数据位 (在 9 位模式下)
- bit 7-0      **URX<7:0>:** 接收到字符的第 7-0 个数据位

**寄存器 21-4: UxTXREG: UARTx 发送寄存器 (只写)**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	W-x
—	—	—	—	—	—	—	UTX8
bit 15							bit 8
W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
UTX<7:0>							
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = POR 值	1 = 置 1	0 = 清零
		x = 未知

- bit 15-9      **未实现:** 读为 0
- bit 8        **UTX8:** 已发送字符的第 8 个数据位 (在 9 位模式下)
- bit 7-0      **URX<7:0>:** 已发送字符的第 7-0 个数据位

# PIC24F 系列参考手册

寄存器 21-5: **UxBRG: UARTx 波特率寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	W-x
BRG<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG<7:0>							
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = POR 值	1 = 置 1	0 = 清零
		x = 未知

bit 15-0      **BRG<15:0>**: 波特率除数位



### 21.3 UART 波特率发生器 (BRG)

UART 模块包含一个专用的 16 位波特率发生器。UxBRG 寄存器控制一个自由运行的 16 位定时器的周期。公式 21-1 给出了 BRGH = 0 时计算波特率的公式。

**公式 21-1: BRGH = 0 时的 UART 波特率**

$$\text{波特率} = \frac{F_{CY}}{16 \cdot (UxBRG + 1)}$$

$$UxBRG = \frac{F_{CY}}{16 \cdot \text{波特率}} - 1$$

**注:** F<sub>CY</sub> 表示指令周期时钟频率 (F<sub>OSC</sub>/2)。

例 21-1 给出了如下条件下的波特率误差计算:

- F<sub>CY</sub> = 4 MHz
- 目标波特率 = 9600

**例 21-1: 波特率误差计算 (BRGH = 0)**

$$\text{目标波特率} = F_{CY} / (16 (UxBRG + 1))$$

求解 UxBRG 的值:

$$UxBRG = ((F_{CY} / \text{目标波特率}) / 16) - 1$$

$$UxBRG = ((4000000 / 9600) / 16) - 1$$

$$UxBRG = 25$$

$$\begin{aligned} \text{计算波特率} &= 4000000 / (16 (25 + 1)) \\ &= 9615 \end{aligned}$$

$$\begin{aligned} \text{误差} &= \frac{(\text{计算波特率} - \text{目标波特率})}{\text{目标波特率}} \\ &= (9615 - 9600) / 9600 \\ &= 0.16\% \end{aligned}$$

能得到的最大波特率 (BRGH = 0) 是 F<sub>CY</sub>/16 (UxBRG = 0 时), 最小波特率是 F<sub>CY</sub>/(16 \* 65536)。

公式 21-2 给出了 BRGH = 1 时计算波特率的公式。

**公式 21-2: BRGH = 1 时的 UART 波特率**

$$\text{波特率} = \frac{F_{CY}}{4 \cdot (UxBRG + 1)}$$

$$UxBRG = \frac{F_{CY}}{4 \cdot \text{波特率}} - 1$$

**注:** F<sub>CY</sub> 表示指令周期时钟频率。

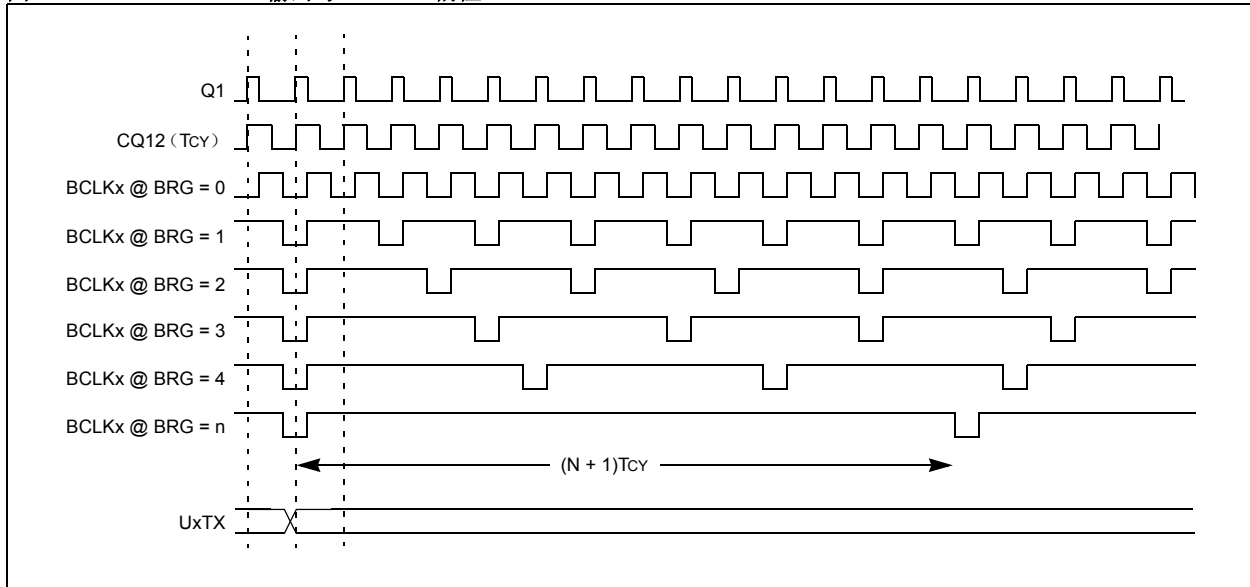
最大波特率 (BRGH = 1) 是 F<sub>CY</sub>/4 (UxBRG = 0 时), 最小波特率是 F<sub>CY</sub>/(4 \* 65536)。

向 UxBRG 寄存器写入新值会使 BRG 定时器复位 (清零)。这可以确保 BRG 无需等待定时器溢出就可以产生新的波特率。

## 21.3.1 BCLKx 输出

如果使能 UART 和 BCLKx 输出 ( $UEN<1:0> = 11$ )，则 BCLKx 引脚将输出 16 倍频波特率时钟。该功能用于支持外部 IrDA 编 / 解码器（见图 21-2）。在休眠模式下，BCLKx 输出保持低电平。只要 UART 保持在该模式 ( $UEN<1:0> = 11$ )，则无论 PORTx 和 TRISx 锁存位如何，BCLKx 都强制作为输出。

图 21-2: BCLKx 输出与 UxBRG 编程



## 21.3.2 波特率表

表 21-1 和表 21-2 给出了普通器件指令周期频率（Fcy）的 UART 波特率。同时也给出了每个频率下最小和最大波特率。

表 21-1: UART 波特率 (BRGH = 0)

波特率	Fcy = 32 MHz			Fcy = 16 MHz			Fcy = 12 MHz		
	实际波特率	% 误差	BRG 值 (十进制)	实际波特率	% 误差	BRG 值 (十进制)	实际波特率	% 误差	BRG 值 (十进制)
110	110.0	0.00	18181	110.0	0.00	9090	110.0	0.00	6817
300	300.0	0.00	6666	300.0	0.01	3332	300.0	0.00	2499
1200	1200.0	0.00	1666	1200.5	0.04	832	1200.0	0.00	624
2400	2400.9	0.03	832	2398.1	-0.08	416	2403.8	0.16	311
9600	9615.3	0.16	207	9615.4	0.16	103	9615.3	0.16	77
19.2K	19230.7	0.15	103	19230.8	0.16	51	19230.7	0.15	38
38.4K	38461.5	0.16	51	38461.5	0.16	25	37500.0	-2.34	19
56K	55555.5	-0.79	35	55555.6	-0.79	17	57692.3	-3.02	12
115K	117647.0	2.30	16	111111.1	-3.38	8			
250K	250000.0	0.00	7	250000.0	0.00	3			
300K									
500K				500000.0	0.00	1			
最小值	31.0	0.00	65535	15.0	0.00	65535	11.0	0.00	65535
最大值	2000000.0	0.00	0	1000000.0	0.00	0	480000.0	0.00	0

波特率	Fcy = 8 MHz			Fcy = 4 MHz			Fcy = 1 MHz		
	实际波特率	% 误差	BRG 值 (十进制)	实际波特率	% 误差	BRG 值 (十进制)	实际波特率	% 误差	BRG 值 (十进制)
110	917.4	0.00	4544	110.0	0.00	2272	110.0	0.00	567
300	299.9	0.00	1666	300.1	0.00	832	300.4	0.10	207
1200	1199.0	0.00	416	1201.9	0.16	207	1201.9	0.16	51
2400	2403.8	0.16	207	2403.8	0.15	103	2403.8	0.15	25
9600	9615.4	0.16	51	9615.4	0.20	25			
19.2K	19230.8	0.16	25	19230.8	0.20	12			
38.4K	38461.5	0.16	12						
56K	55555.6	-0.79	8						
115K									
250K									
300K									
500K									
最小值	8.0	0.00	65535	4.0	0.00	65535	0.95	0.00	65535
最大值	500000.0	0.00	0	250000.0	0.00	0	62500.0	0.00	0

# PIC24F 系列参考手册

表 21-2: UART 波特率 (BRGH = 1)

波特率	Fcy = 32 MHz			Fcy = 16 MHz			Fcy = 12 MHz		
	实际波特率	% 误差	BRG 值 (十进制)	实际波特率	% 误差	BRG 值 (十进制)	实际波特率	% 误差	BRG 值 (十进制)
110	110.0	0.00	72726	110.0	0.00	36363	110.0	0.00	27272
300	300.0	0.00	26665	300.0	0.01	13332	300.0	0.00	9999
1200	1200.1	0.01	6665	1200.1	0.01	3332	1200.0	0.00	2499
2400	2400.2	0.01	3332	2399.5	-0.01	1666	2403.8	0.00	1249
9600	9603.8	0.04	832	9592.3	-0.07	416	9584.6	-0.15	312
19.2K	19184.7	-0.07	416	19230.7	0.16	207	19230.7	0.15	155
38.4K	38461.5	0.16	207	38461.5	0.16	103	38461.5	0.16	77
56K	55944.1	-0.09	142	56338.0	0.60	70	55555.5	-0.79	53
115K	114285.0	-0.62	69	114285.7	-0.62	34	115384.0	0.33	25
250K	250000.0	0.00	31	250000.0	0.00	15			
300K	296296.0	-1.23	26	307692.3	2.50	12			
500K				500000.0	0.00	7			
最小值	122.0	0.00	65535	61.0	0.00	65535	46.0	0.00	65535
最大值	8000000.0	0.00	0	4000000.0	0.00	0	3000000.0	0.00	0

波特率	Fcy = 8 MHz			Fcy = 4MHz			Fcy = 1 MHz		
	实际波特率	% 误差	BRG 值 (十进制)	实际波特率	% 误差	BRG 值 (十进制)	实际波特率	% 误差	BRG 值 (十进制)
110	110.0	0.00	18181	110.0	0.00	9090	110.0	0.00	2272
300	300.0	0.00	6666	300.0	0.00	3332	300.1	0.10	832
1200	1199.7	-0.01	1666	1200.5	0.00	832	1201.9	0.15	207
2400	2400.9	0.04	832	2398.1	-0.07	416	2403.8	0.15	103
9600	9615.4	0.16	207	9615.3	0.16	103	9615.3	0.16	25
19.2K	19230.8	0.16	103	19230.7	0.16	51	19230.7	0.16	12
38.4K	38461.5	0.16	51	38461.5	0.16	25			
56K	55555.6	-0.79	35	55555.5	-0.79	17			
115K	117647.0	2.30	16						
250K									
300K									
500K									
最小值	31.0	0.00	65535	16.0	0.00	65535	3.81	0.00	65535
最大值	200000.0	0.00	0	1000000.0	0.00	0	250000.0	0.00	0

## 21.4 UART 配置

UART 使用标准的不归零 (Non-Return-to-Zero, NRZ) 格式 (1 个启动位、8 或 9 个数据位和 1 或 2 个停止位)。硬件提供奇偶校验, 用户可以配置为偶校验、奇校验或无奇偶校验。最普通的数据格式是 8 位, 没有奇偶校验位, 有 1 个停止位 (用 8、N 或 1 表示), 这是默认的 (POR) 设置。数据位和停止位的数目以及奇偶校验均在 PDSEL<1:0> (UxMODE<2:1>) 和 STSEL (UxMODE<0>) 位中指定。片上专用的 16 位波特率发生器可用于根据振荡器产生标准的波特率频率。UART 首先发送和接收 LSb。UART 模块的发送器和接收器在功能上是独立的, 但使用相同的数据格式和波特率。

### 21.4.1 使能 UART

UART 模块通过将 UARTEN (UxMODE<15>) 位和 UTXEN (UxSTA<10>) 位置 1 来使能。一旦使能, UxTX 和 UxRX 引脚就分别配置为输出和输入, 改写了相应的 I/O 端口引脚的 TRIS 和 PORT 寄存器位设置。UxTX 引脚在没有传输发生时, 处于逻辑 1。

**注:** 在 UARTEN 位置 1 之前, 不应该将 UTXEN 位置 1; 否则将无法使能 UART 发送。

### 21.4.2 禁止 UART

通过清零 UARTEN (UxMODE<15>) 位来禁止 UART 模块。这是任何复位后的默认状态。如果禁止了 UART, 所有 UART 引脚在相应的 PORT 和 TRIS 位控制下用作端口引脚。

禁止 UART 模块将缓冲器复位为空状态。所有缓冲器中的数字字符丢失, 同时波特率计数器也复位。

当 UART 模块禁止时, 所有与之相关的错误和状态标志都复位。URXDA、OERR、FERR、PERR、UTXEN、UTXBRK 和 UTXBF 位被清零而 RIDLE 和 TRMT 被置 1。其他控制位, 包括 ADDEN、URXISEL<1:0> 和 UTXISEL<1:0>, 还有 UxMODE 和 UxBRG 寄存器不受影响。

当 UART 处于活动状态时, 清零 UARTEN 位将中止所有等待的发送和接收, 同时如以上定义那样将该模块复位。再次使能 UART 将使用同样的配置重新启动 UART。

### 21.4.3 备用 UART I/O 引脚

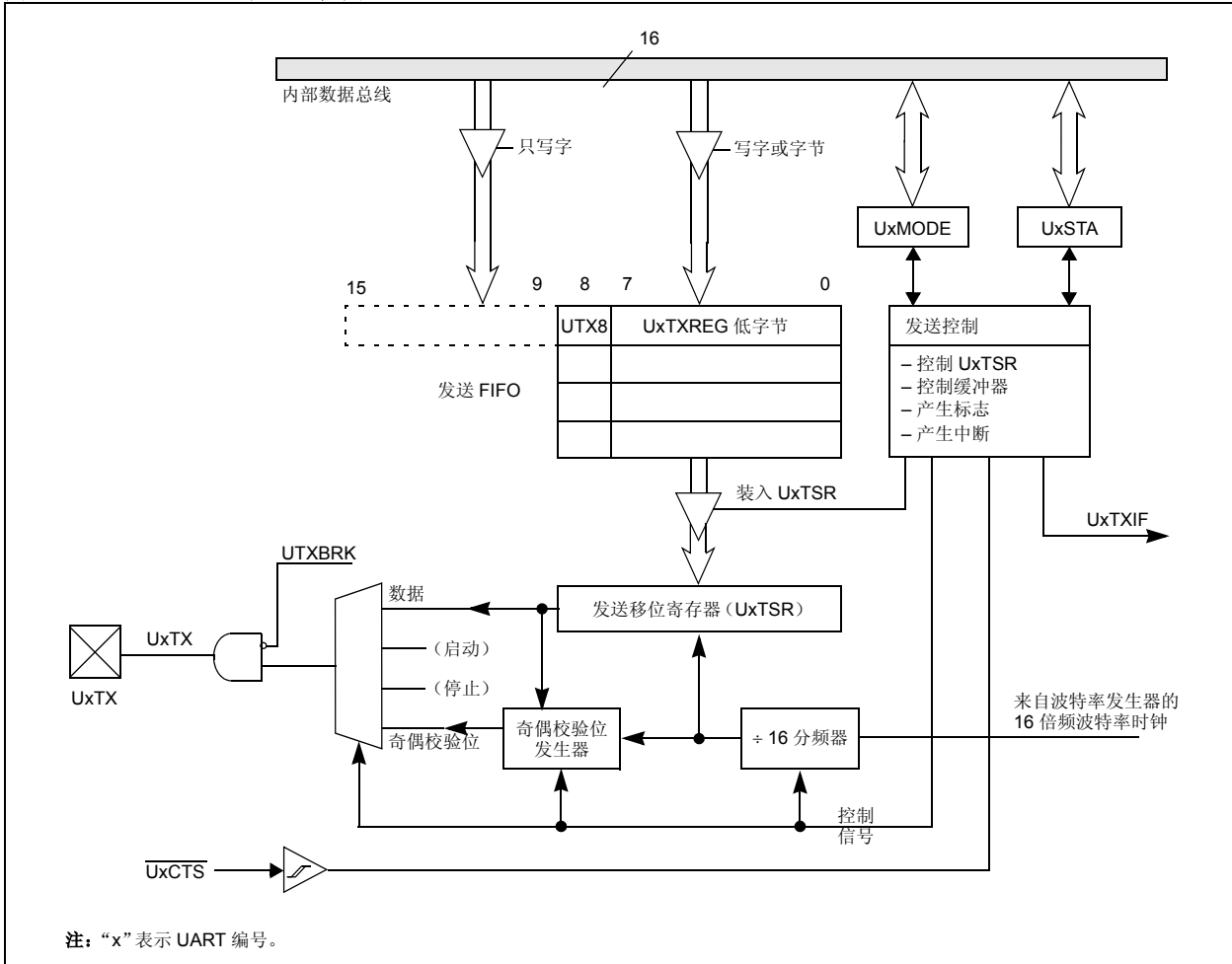
一些 PIC24F 器件有可用来通信的备用 UART 发送和接收引脚。当主 UART 引脚与其他外设共用时, 可以使用备用 UART 引脚。备用 I/O 引脚通过将 ALTIO 位 (UxMODE<10>) 置 1 来使能。如果 ALTIO = 1, UART 模块使用 UxATX 和 UxARX 引脚 (分别为备用发送和备用接收引脚) 代替 UxTX 和 UxRX 引脚。如果 ALTIO = 0, UART 模块使用 UxTX 和 UxRX 引脚。

## 21.5 UART 发送器

图 21-3 所示为 UART 发送器框图。发送器的核心是发送移位寄存器 (UxTSR)。移位寄存器从发送 FIFO 缓冲器 UxTXREG 获取数据。UxTXREG 寄存器中的数据由软件写入。在前一次装入数据的停止位发送前，不会向 UxTSR 寄存器装入新数据。一旦停止位发送完毕，UxTXREG 寄存器中的新数据 (如果有) 就会被装入 UxTSR。

**注：** UxTSR 寄存器并未映射到数据存储中，因此用户不能直接访问它。

图 21-3: UART 发送器框图



通过将 UTXEN 使能位 (UxSTA<10>) 置 1 来使能发送。实际的发送要到 UxTXREG 寄存器装入了数据并且波特率发生器 (UxBRG) 产生了移位时钟 (图 21-3) 后才发生。还可以先装入 UxTXREG 寄存器，然后将 UTXEN 使能位置 1 来启动发送。通常，第一次开始发送的时候，由于 UxTSR 寄存器为空，这样传输数据到 UxTXREG 寄存器会导致该数据立即传输到 UxTSR。发送过程中清零 UTXEN 位将使发送中止并复位发送器。因此，UxTX 引脚将回复到一个高阻抗状态。

为了选择 9 位发送，PDSEL<1:0> 位 (UxMODE<2:1>) 应设置为 11 并且第 9 位应该写入 UTX8 位 (UxTXREG<8>)。应该向 UxTXREG 执行一个字写操作，这样可以同时写入所有的 9 位。

**注：** 在 9 位数据发送的情况下，不采取奇偶校验。

### 21.5.1 发送缓冲器 (UxTXREG)

发送缓冲器有 9 位宽和 4 级深。算上发送移位寄存器 (UxTSR)，用户实际有一个 5 级深的缓冲器。它按先进先出 (FIFO) 的方式组织。一旦 UxTXREG 的内容被传送到 UxTSR 寄存器，当前缓冲单元就可以写入新的数据，下一个缓冲单元将成为 UxTSR 寄存器的数据源。无论何时，只要缓冲器满了，UTXBF (UxSTA<9>) 状态位就会置 1。如果用户试图向已经满的缓冲器执行写操作，则新数据将不会被 FIFO 接收。

FIFO 在任何器件复位时复位，但当器件进入省电模式或从省电模式唤醒时，FIFO 不受影响。

### 21.5.2 发送中断

发送中断标志 (UxTXIF) 位于相应的中断标志状态 (IFS) 寄存器中。UTXISEL<1:0> 控制位 (UxSTA<15,13>) 决定 UART 将何时产生一个发送中断。

1. UTXISEL<1:0> = 00，当一个字符从发送缓冲器传输到发送移位寄存器 (UxTSR) 时，UxTXIF 被置 1。这暗示发送缓冲器中至少有一个单元为空。
2. UTXISEL<1:0> = 01，当最后一个字符移出发送移位寄存器 (UxTSR) 时，UxTXIF 被置 1。这表示所有发送操作都已完成。
3. UTXISEL<1:0> = 10，当一个字符传输到发送移位寄存器 (UxTSR) 且发送缓冲器为空时，UxTXIF 被置 1。

当模块第一次使能时，UxTXIF 位将被置 1。用户应该在 ISR 中清零 UxTXIF 位。

运行时可以在两种中断模式间切换。

**注：** 如果 UTXISEL<1:0> = 00，当 UTXEN 位置 1 时，UxTXIF 标志位也置 1，因为发送缓冲器尚未满（可以向 UxTXREG 寄存器移入待发送数据）。

UxTXIF 标志位指示 UxTXREG 寄存器的状态，而 TRMT 位 (UxSTA<8>) 表明 UxTSR 的状态。TRMT 状态位是只读位，当 UxTSR 寄存器为空时被置 1。因为没有与此位关联的中断逻辑，所以用户必须查询此位以判断 UxTSR 是否为空。

## 21.5.3 设置 UART 发送

设置发送时应该遵循的步骤:

1. 对 UxBRG 寄存器进行初始化, 设置合适的波特率 (见第 21.3 节“UART 波特率发生器 (BRG)”)。
2. 通过写 PDSEL<1:0> (UxMODE<2:1>) 和 STSEL (UxMODE<0>) 位来设置数据位数、停止位数和奇偶校验选择。
3. 如果需要发送中断, 就要将相应的中断允许控制寄存器 (IEC) 中的 UxTXIE 控制位置 1。使用相应的中断优先级控制寄存器 (IPC) 中的 UxTXIP<2:0> 控制位来指定发送中断的中断优先级。同时, 通过写 UTXISEL<1:0> (UxSTA<15,13>) 位来选择发送中断模式。
4. 通过清零 UARTEN (UxMODE<15>) 位来使能 UART 模块。
5. 通过将 UTXEN (UxSTA<10>) 置 1 来使能发送, 与此同时将 UxTXIF 位置 1。在 UART 发送中断服务程序中, UxTXIF 位应该清零。UxTXIF 位的操作由 UTXISEL<1:0> 控制位控制。
6. 将数据装入 UxTXREG 寄存器 (开始发送)。如果选择了 9 位发送, 则装入一个字。如果使用了 8 位发送, 则装入一个字节。数据可以装入缓冲器, 直到 UTXBF 状态位 (UxSTA<9>) 被置 1 为止。

**注:** 在 UARTEN 位置 1 之前, 不应该将 UTXEN 位置 1; 否则将无法使能 UART 发送。

图 21-4: 发送 (8 位或 9 位数据)

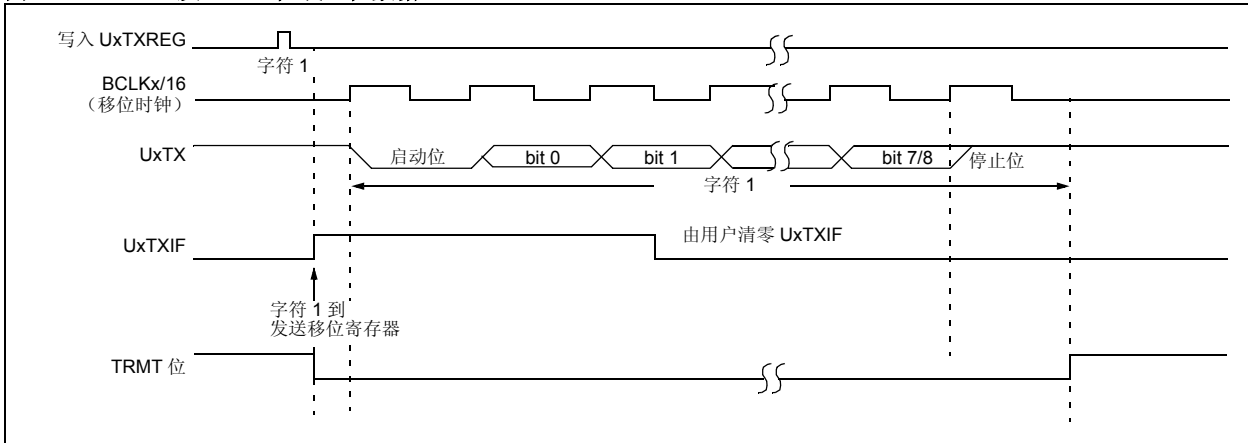
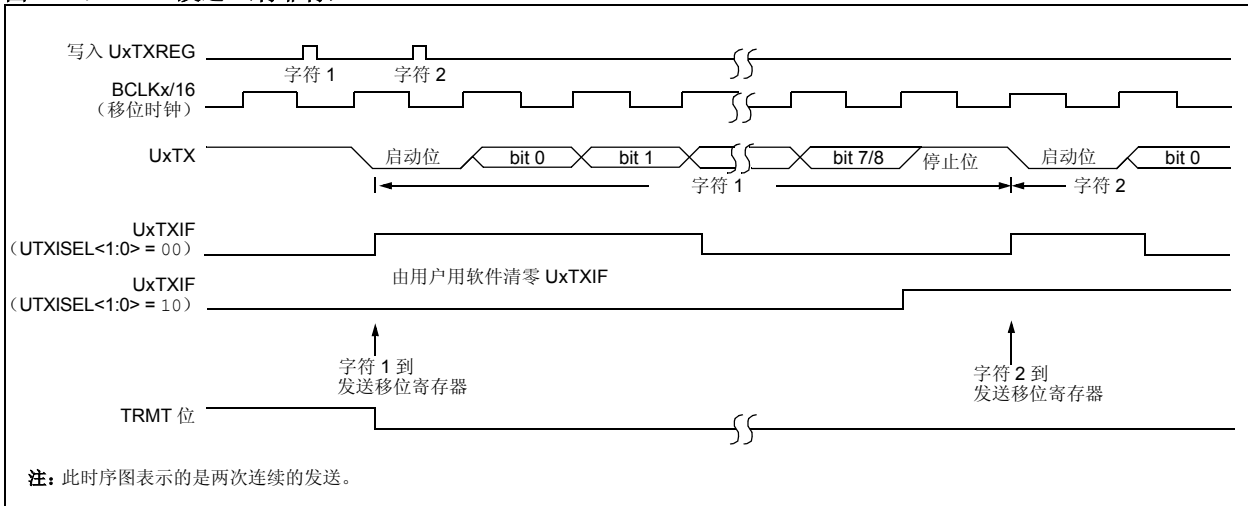


图 21-5: 发送 (背靠背)





### 21.5.4 间隔字符的发送

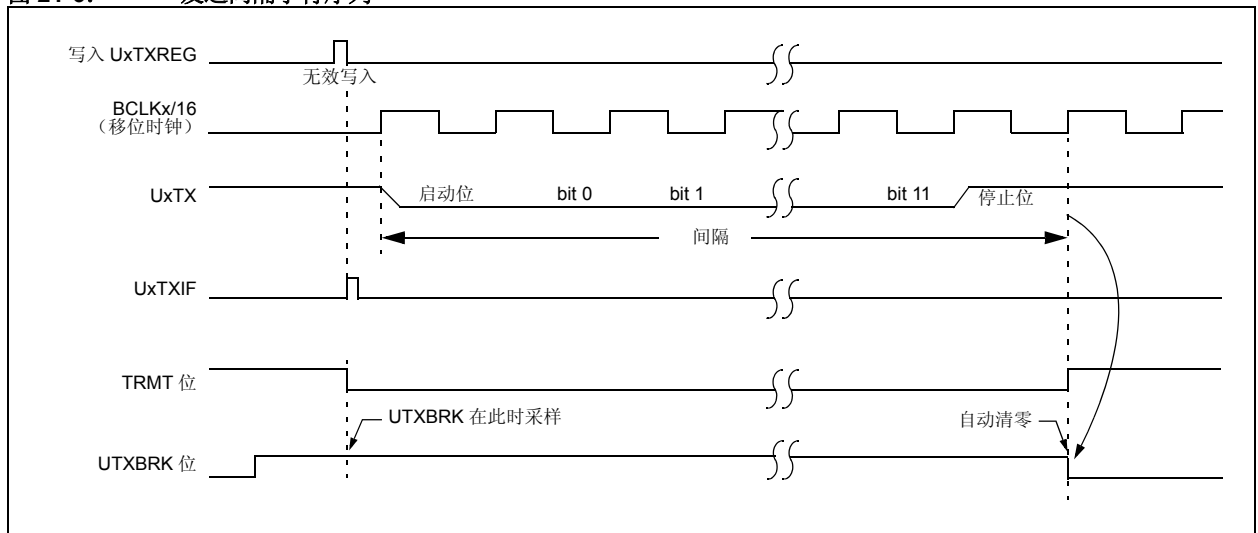
发送的间隔字符包括 1 个启动位，后面跟有 12 个 0 位和 1 个停止位。在发送移位寄存器中装有数据时，只要 **UTXBRK** 和 **UTXEN** 位置 1，就会发送帧间隔字符。必须对 **UxTXREG** 寄存器进行假写操作，才能启动间隔字符发送。请注意在发送间隔字符时写入 **UxTXREG** 的数据值会被忽略。进行写操作只是为了启动适当的序列——发送的数据将为全 0。

在发送了相应的停止位后，硬件会自动将 **UTXBRK** 位复位。这样用户可以在间隔字符（在 LIN 规范中通常是同步字符）后预先将下一个要发送字节装入发送 FIFO 队列。

**注：** 在将 **UTXBRK** 置 1 之前，用户应先等待发送器变为空闲（**TRMT** = 1）。**UTXBRK** 会覆盖所有其他发送器活动。如果用户在序列完成之前清零 **TXBRK** 位，则可能导致意外的模块行为。发送间隔字符不会产生发送中断。

正如其在正常发送操作中一样，**TRMT** 位表明发送移位寄存器是空还是满。关于发送间隔字符的时序，请参见图 21-6。

图 21-6: 发送间隔字符序列



#### 21.5.4.1 间隔和同步发送序列

下述序列会发送一个报文帧头，包括一个间隔字符和其后的一个自动波特率同步字节。此序列适用于典型的 LIN 总线主控器件。

1. 将 **UART** 配置为所需的模式。
2. 将 **UTXEN** 和 **UTXBRK** 置 1——设置间隔字符。
3. 将无效字符装入 **UxTXREG**，启动发送（该值会被忽略）。
4. 向 **UxTXREG** 写 “55h”——将同步字符装入发送 FIFO。

间隔字符发送后，硬件会将 **UTXBRK** 位复位。现在发送同步字符。

## 21.6 UART 接收器

图 21-7 所示为接收器框图。接收器的核心是接收（串行）移位寄存器（UxRSR）。在 UxRX 引脚上接收数据，并发送到数据恢复区中。数据恢复区以 16 倍波特率运行，而主接收串行移位器以波特率运行。在采集到 UxRX 引脚上的停止位后，UxRSR 中接收到的数据传输到接收 FIFO（如果为空）。

**注：** UxRSR 寄存器并未映射到数据存储区中，因此用户不能直接访问它。

择多检测电路（majority detect circuit）对 UxRX 引脚上的数据采样三次，以判定 UxRX 引脚上出现的是高电平还是低电平。

### 21.6.1 接收缓冲器（UxRXREG）

UART 接收器有一个 4 级深、9 位宽的 FIFO 接收数据缓冲器。UxRXREG 是一个存储器映射的寄存器，可提供对 FIFO 输出的访问。在缓冲器溢出发生以前，可以有 4 个字的数据被接收并传输到 FIFO，从第 5 个字开始将数据移位到 UxRSR 寄存器中。

### 21.6.2 接收器错误处理

如果 FIFO 已满（4 个字符），而第 5 个字符已经完全接收到了 UxRSR 寄存器，溢出错误位 OERR（UxSTA<1>）就将置 1。UxRSR 中的字将被保留，但是只要 OERR 位置 1，就被禁止继续向接收 FIFO 传输。用户必须用软件清零 OERR 位，以允许继续接收数据。

如果需要保存溢出前接收到的数据，用户应该先读所有 5 个字符，然后清零 OERR 位。如果这 5 个字符可以丢弃，则用户只要清零 OERR 位即可。这可有效地复位接收 FIFO，同时先前接收到的所有数据都将丢失。

**注：** 接收 FIFO 中的数据应该在清零 OERR 位之前读出。当 OERR 清零时，FIFO 复位，这将导致缓冲器中的所有数据丢失。

如果在逻辑低电平检测到停止位，则帧错误位 FERR（UxSTA<2>）就将置 1。

如果检测到缓冲器顶部的数据字（即，当前的字）有奇偶校验错误，则奇偶校验错误位 PERR（UxSTA<3>）就将置 1。例如，如果奇偶校验设置为偶数，但检测出数据中 1 的总数为奇数，就会产生奇偶校验错误。PERR 位在 9 位模式下是无关系的。FERR 和 PERR 位与相应的字一起被缓冲，并且应该在读取数据字之前读出。

如果发生了这些错误（OERR、FERR 和 PERR）中的任一错误，则会产生中断。所产生的中断仅在一个周期内有效。用户必须允许相应的中断允许位（IEC4<UxERIE>）才能转到相应的中断向量单元。

### 21.6.3 接收中断

UART 接收中断标志（UxRXIF）位于相应的中断标志状态（IFS）寄存器中。URXISEL<1:0>（UxSTA<7:6>）控制位决定 UART 接收器何时将产生一个中断。

- 如果 URXISEL<1:0> = 00 或 01，每当一个数据字从接收移位寄存器（UxRSR）传输到接收缓冲器后，就会产生中断。接收缓冲器中可以有一个或多个字符。
- 如果 URXISEL<1:0> = 10，当一个字从接收移位寄存器（UxRSR）传输到接收缓冲器时就会产生中断，其结果是接收缓冲器中有 3 或 4 个字符。
- 如果 URXISEL<1:0> = 11，当一个字从接收移位寄存器（UxRSR）传输到接收缓冲器时就会产生中断，其结果是接收缓冲器中有 4 个字符（即，缓冲器满）。

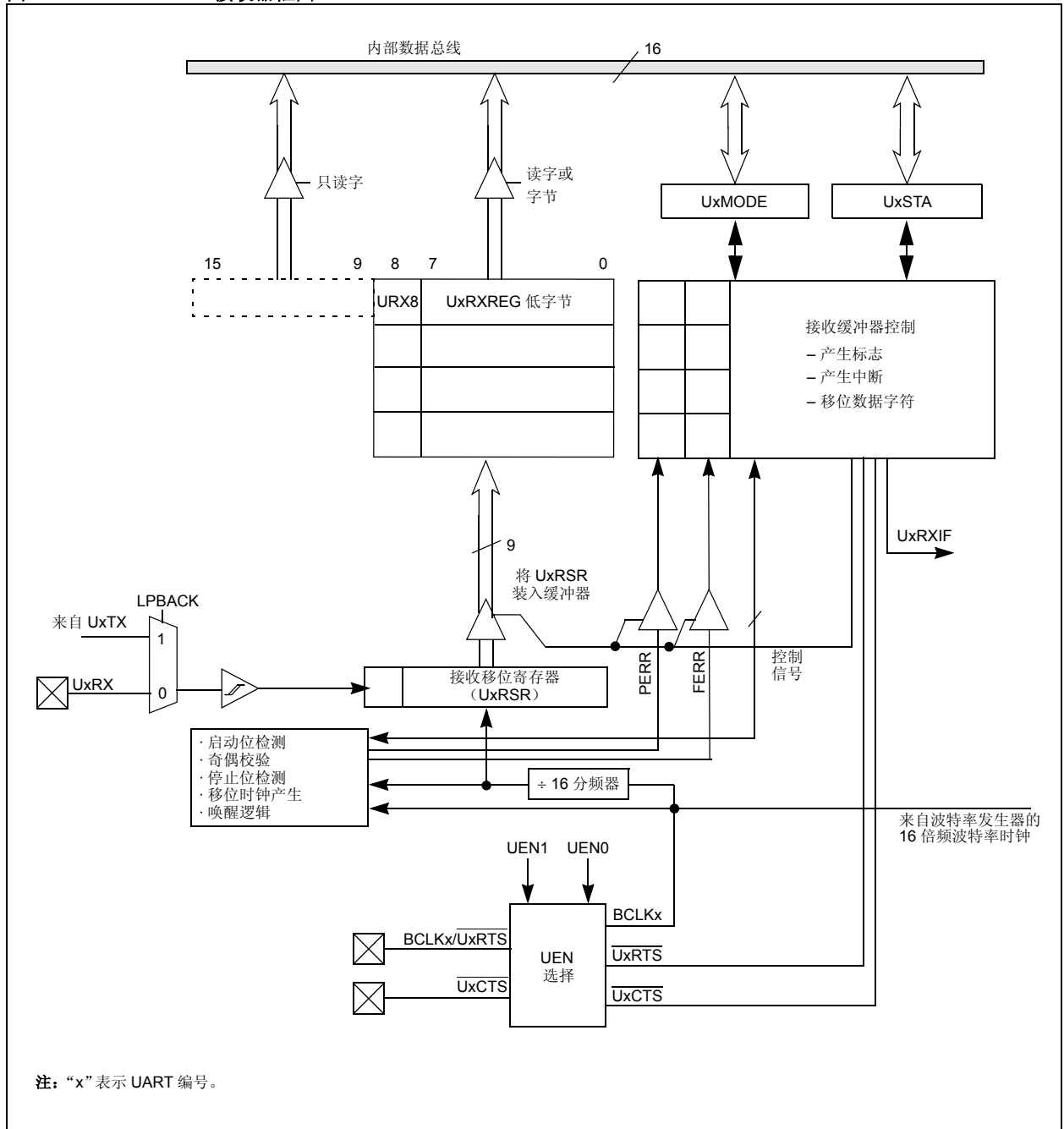
运行时可以在三种中断模式间切换。

URXDA 和 UxRXIF 标志位指示 UxRXREG 寄存器的状态，而 RIDLE 位 (UxSTA<4>) 表明 UxRSR 寄存器的状态。RIDLE 状态位是只读位，在接收器空闲时 (即，UxRSR 寄存器为空) 时被置 1。因为没有与此位关联的中断逻辑，所以用户必须查询此位以判断 UxRSR 是否空闲。

URXDA 位 (UxSTA<0>) 指示接收缓冲器有数据还是为空。只要接收缓冲器中至少有一个可以读出的字符，该位就将置 1。URXDA 是只读位。

图 21-7 所示为 UART 接收器的框图。

图 21-7: UART 接收器框图



## 21.6.4 设置 UART 接收

设置接收时应该遵循的步骤：

1. 将 UxBRG 寄存器初始化，以获得合适的波特率（见第 21.3 节“UART 波特率发生器 (BRG)”）。
2. 通过写 PDSEL<1:0> (UxMODE<2:1>) 和 STSEL (UxMODE<0>) 位来设置数据位数、停止位数和奇偶校验选择。
3. 如果需要中断，就要将相应的中断允许控制 (IEC) 寄存器中的 UxRXIE 位置 1。使用相应的中断优先级控制寄存器 (IPC) 中的 UxRXIP<2:0> 控制位来指定该中断的优先级。同时，通过写 URXISEL<1:0> (UxSTA<7:6>) 位来选择接收中断模式。
4. 通过清零 UARTEN (UxMODE<15>) 位来使能 UART 模块。
5. 接收中断取决于 URXISEL<1:0> 控制位的设置。如果没有允许接收中断，用户可以查询 URXDA 位。在 UART 接收中断服务程序中，UxRXIF 位应该清零。
6. 从接收缓冲器中读取数据。如果选择了 9 位发送，则读一个字；否则，读一个字节。无论何时，只要缓冲器中有数据可读，URXDA 状态位 (UxSTA<0>) 就将被置 1。

图 21-8: UART 接收

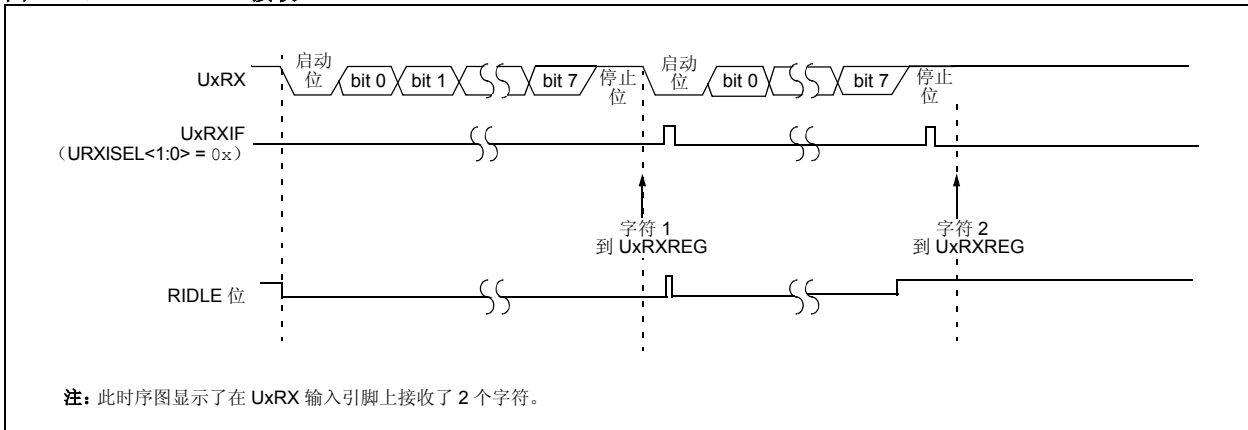
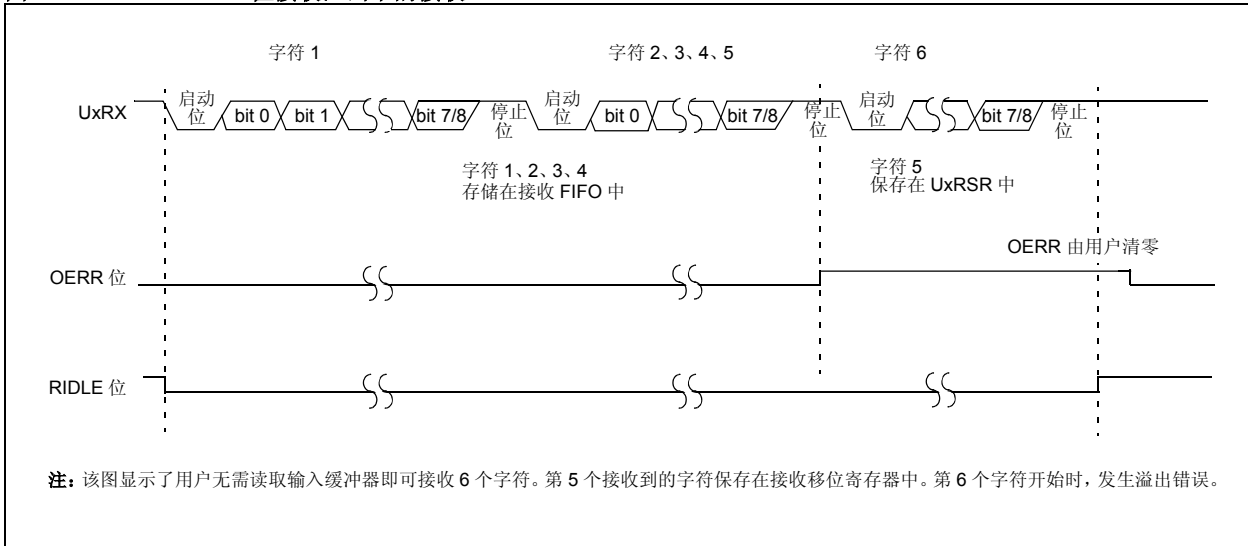


图 21-9: UART 在接收溢出下的接收



## 21.7 使用 UART 进行 9 位通信

在 9 位数据模式下，UART 接收器可用于进行多处理器通信。在 9 位数据模式下，当 ADDEN 位置 1 时，接收器可以在数据的第 9 位为 0 时忽略数据。该功能可以在多处理器环境下使用。

### 21.7.1 多处理器通信

典型的多处理器通信协议会区别数据字节和地址 / 控制字节。一般的方法是使用第 9 个数据位来识别数据字节是地址还是数据信息。如果第 9 位置 1，数据就作为地址或控制信息处理。如果第 9 位清零，接收到的数据字就作为和前面的地址 / 控制字节相关的数据处理。

协议操作如下：

- 主器件发送一个第 9 位置 1 的数据。数据字包含从器件的地址。
- 通信链中的所有从器件接收地址字并检查从地址值。
- 被寻址的从器件将接收和处理主器件发送的后续数据字节。所有其他从器件将丢弃后续的数据字节，直到接收到新的地址字（第 9 位置 1）。

### 21.7.2 ADDEN 控制位

UART 接收器有一个地址检测模式，该模式允许接收器忽略第 9 位清零的数据字。这降低了中断开销，因为第 9 位清零的数据字不被缓冲。这个功能通过将 ADDEN 位 (UxSTA<5>) 置 1 来使能。

使用地址检测模式，UART 必须配置为 9 位数据模式。当接收器配置为 8 位数据模式时，ADDEN 位无效。

### 21.7.3 设置 9 位发送

除了 PDSEL<1:0> 位 (UxMODE<2:1>) 应该设置为 11 外，设置 9 位发送的过程与设置 8 位发送模式类似（见第 21.5.3 节“设置 UART 发送”）。

应该对 UxTXREG 寄存器执行字写操作（开始发送）。

## 21.7.4 设置使用地址检测模式的 9 位接收

除了 PDSEL<1:0> 位 (UxMODE<2:1>) 应该设置为 11 外, 设置 9 位接收的过程和设置 8 位接收模式类似 (见第 21.6.4 节 “设置 UART 接收”)。

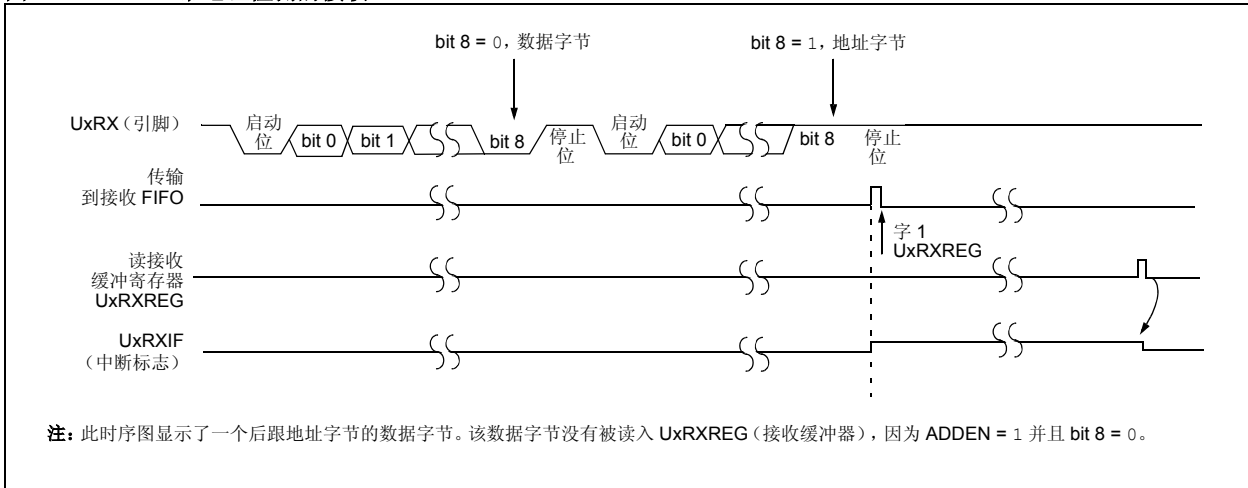
应该通过写 URXISEL<1:0> (UxSTA<7:6>) 位来配置接收中断模式。

**注:** 如果地址检测模式使能 (ADDEN = 1) 了, URXISEL<1:0> 控制位应该配置为接收到每个字后就产生中断。每个接收到的数据字在接收后必须立即用软件检查是否地址匹配。

使用地址检测模式的过程如下:

1. 将 ADDEN (UxSTA<5>) 位置 1 来使能地址检测。确保 URXISEL 控制位配置为每接收一个字就产生一个中断。
2. 读 UxRXREG 寄存器, 检查每个 8 位地址, 确定器件是否被寻址。
3. 如果该器件没有被寻址, 就丢弃接收到的字。
4. 如果器件被寻址, 应将 ADDEN 位清零以允许后续数据字节被读进接收缓冲器并中断 CPU。如果预计收到的是一个长数据包, 则需要改变接收中断模式, 以使中断之间可以缓冲多于一个的数据字节。
5. 当接收到最后的数据字节时, 应将 ADDEN 位置 1 以便以允许地址字节被接收。同样, 确保 URXISEL 控制位配置为每接收一个字就产生一个中断。

图 21-10: 带地址检测的接收 (ADDEN = 1)



## 21.8 接收间隔字符

唤醒功能通过设置 WAKE 位 (UxMODE <7>) = 1 而使能。在该模式下, 模块会接收启动位、数据和无效停止位 (这会将 FERR 置 1), 但接收器会在检测下一个启动位之前先等待有效的停止位。它不会将线上的间隔条件当作下一个启动位。间隔字符将被当作包含的数据位为全 0 的字符, 而 FERR 位置 1。间隔字符会被装入缓冲器中。只有在接收到停止位之后, 才会继续进行接收。在 13 位的间隔字符后又接收到停止位时, WAKE 位将自动清零。请注意, 接收到停止位后, RIDLE 变为高电平。

接收器将根据 PDSEL<1:0> (UxMODE<2:1>) 和 STSEL (UxMODE<0>) 位中设定的值, 计数并等待特定数量的位时间。

如果间隔大于 13 个位时间, 则在经过 PDSEL 和 STSEL 位所指定数量的位时间之后, 就认为接收已完成。此时, URXDA 位置 1, FERR 置 1, 接收 FIFO 中装入 0, 并产生中断。

如果未设置唤醒功能, 即 WAKE (UxMODE <7>) = 0, 则间隔接收并无任何不同。间隔字符将被计为一个字符装入缓冲器 (所有位全为 0), 而 FERR 置 1。

## 21.9 初始化

例 21-2 是在 8 位模式下发送器 / 接收器的初始化程序。例 21-3 所示为 8 位地址检测模式下可寻址 UART 的初始化。在两个示例中, 要装入 UxBRG 寄存器的值取决于所需的波特率和器件频率。

**注:** 在 UARTEN 位置 1 之前, 不应该将 UTXEN 位置 1; 否则将无法使能 UART 发送。

### 例 21-2: 8 位发送 / 接收 (UART1)

```

U1BRG=#BaudRate;                //Set Baudrate

IPC3bits.U1TXIP2 = 1;            //Set Uart TX Interrupt Priority
IPC3bits.U1TXIP1 = 0;
IPC3bits.U1TXIP0 = 0;
IPC2bits.U1RXIP2 = 1;            //Set Uart RX Interrupt Priority
IPC2bits.U1RXIP1 = 0;
IPC2bits.U1RXIP0 = 0;

U1STA = 0;
U1MODE = 0x8000;                 //Enable Uart for 8-bit data
                                   //no parity, 1 STOP bit

U1STAbits.UTXEN = 1;             //Enable Transmit
IEC0bits.U1TXIE = 1;             //Enable Transmit Interrupt
IEC0bits.U1RXIE = 1;             //Enable Receive Interrupt

```

## 例 21-3: 8 位发送 / 接收 (UART1), 地址检测使能

```
U1BRG=#BaudRate; //Set Baudrate

IPC3bits.U1TXIP2 = 1; //Set Uart TX Interrupt Priority
IPC3bits.U1TXIP1 = 0;
IPC3bits.U1TXIP0 = 0;
IPC2bits.U1RXIP2 = 1; //Set Uart RX Interrupt Priority
IPC2bits.U1RXIP1 = 0;
IPC2bits.U1RXIP0 = 0;

U1STA = 0;
U1STAbits.ADDEN = 1; //Address detect enabled
U1MODE = 0x8080; //Enable Uart for 8-bit data
//no parity,1 STOP bit,wake enabled

U1STAbits.UTXEN = 1; //Enable Transmit
IEC0bits.U1TXIE = 1; //Enable Transmit Interrupt
IEC0bits.U1RXIE = 1; //Enable Receive Interrupt
```



## 21.10 UART 的其他特性

### 21.10.1 环回模式下的 UART

将 LPBACK 位置 1 将使能这种特殊模式，在该模式下，UxTX 输出在内部连接到 UxRX 输入。当配置为环回模式时，UxRX 引脚从内部 UART 接收逻辑断开。不过，UxTX 引脚仍正常工作。

要选择该模式，请执行以下步骤：

1. 将 UART 配置为所需的工作模式。
2. 按照第 21.5 节“UART 发送器”中的描述使能发送。
3. 设置 LPBACK = 1 (UxMODE<6>) 来使能环回模式。

环回模式取决于 UEN<1:0> 位，如表 21-3 中所示。

表 21-3: 环回模式引脚功能

UEN<1:0>	引脚功能, LPBACK = 1 <sup>(1)</sup>
00	UxRX 输入连接到 UxTX；UxTX 引脚工作；UxRX 引脚忽略；UxCTS/UxRTS 未使用
01	UxRX 输入连接到 UxTX；UxTX 引脚工作；UxRX 引脚忽略；UxRTS 引脚工作，UxCTS 未使用
10	UxRX 输入连接到 UxTX；UxTX 引脚工作；UxRX 引脚忽略；UxRTS 引脚工作，UxCTS 输入连接到 UxRTS；UxCTS 引脚忽略
11	UxRX 输入连接到 UxTX；UxTX 引脚工作；UxRX 引脚忽略；BCLKx 引脚工作；UxCTS/UxRTS 未使用

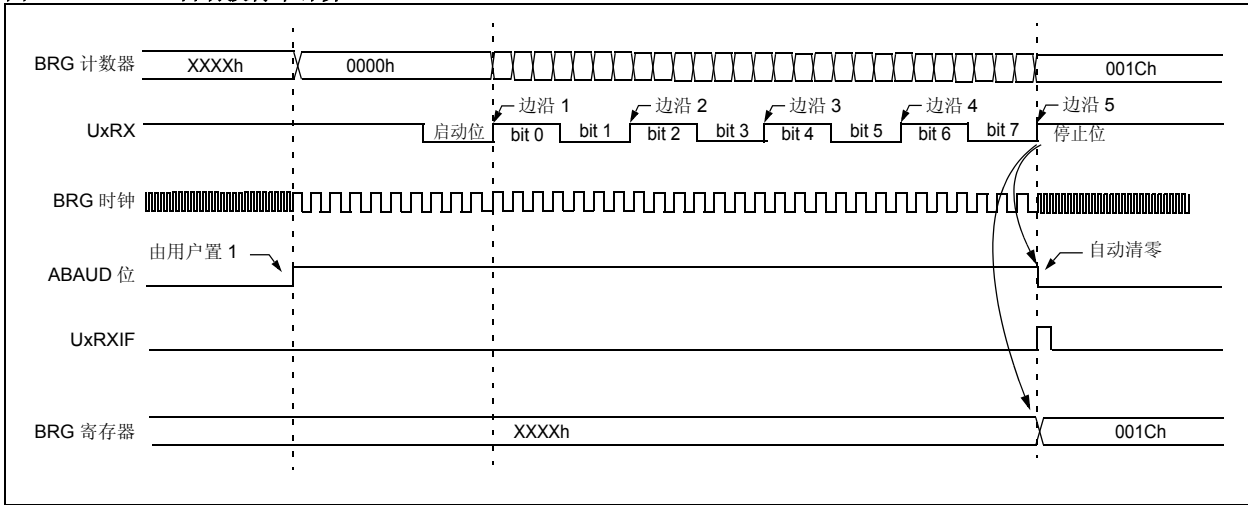
注 1: 应在使能与 UART 模块相关的其他位之后，才设置 LPBACK = 1。

### 21.10.2 自动波特率支持

要允许系统确定所接收字符的波特率，可以使能 ABAUD 位。如果使能自动波特率检测 (ABAUD = 1)，则在接收到启动位时，UART 会开始自动波特率测量序列。波特率计算采用自平均的方式。该功能只有在禁止自动唤醒 (WAKE = 0) 时才有效。此外，对于自动波特率操作，LPBACK 必须等于 0。ABAUD 置 1 后，BRG 计数器值将被清零，并开始检测一个启动位；在此时，启动位规定为高电平到低电平变换后跟随一个低电平到高电平变换。

在启动位之后，自动波特率功能需要接收一个 ASCII “U” (55h)，以计算相应的位速率。为了尽量减少输入信号不对称造成的影响，测量时段内要包含一个高位和一个低位时间。在启动位 (上升沿) 结束时，BRG 计数器开始使用 Tcy/8 时钟计数。在第 5 个 UxRX 引脚上升沿，统计相应 BRG 总周期数的累计 BRG 计数器值被传送到 UxBRG 寄存器。ABAUD 位被自动清零。如果用户在序列完成之前清零 ABAUD 位，则可能导致意外的模块行为。关于 ABD 序列的信息，请参见图 21-11。

图 21-11: 自动波特率计算



在进行自动波特率序列时，UART 状态机保持在空闲状态。无论 URXISEL<1:0> 设置如何，UxRXIF 中断均设为在第 5 个 UxRX 上升沿产生。接收器 FIFO 不会被更新。

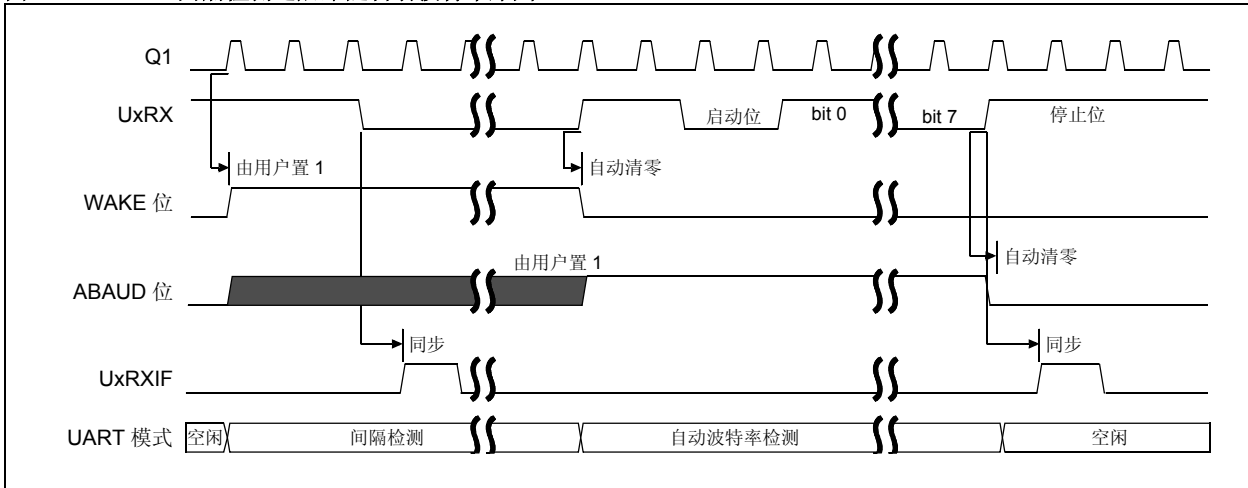
### 21.10.2.1 间隔检测序列

用户可以将模块配置为在间隔检测之后立即自动检测波特率。这可通过将 ABAUD 位置 1 和 WAKE 位置 1 来实现。图 21-12 显示了在间隔检测之后产生波特率序列。WAKE 位的优先级高于 ABAUD 位设置的优先级。

**注：** 如果 WAKE 位与 ABAUD 位同时置 1，自动波特率检测会发生在间隔字符之后的字节。用户必须考虑给定时钟可能提供的波特率，确保进入的字符波特率处于选定的 UxBRG 时钟源的范围之内。

在自动波特率序列期间，不能使用 UART 发送器。此外，用户应确保不要在正在进行发送序列时，将 ABAUD 位置 1。否则，UART 会产生不可预测的行为。

图 21-12: 间隔检测之后跟随自动波特率序列



## 21.11 在 CPU 休眠和空闲模式下的 UART 操作

UART 在休眠模式下不工作。如果在进行发送时进入休眠模式，则发送被终止，UxTX 引脚驱动为逻辑 1。类似地，如果在进行接收时进入休眠模式，则接收被中止。

在休眠模式期间，UART 会将其自身复位。

如果处于掉电模式，则  $\overline{\text{UxRTS}}$  引脚驱动为 0；否则，它驱动为第 21.12 节“UxCTS 和 UxRTS 控制引脚的操作”中规定的值。

BCLCx 引脚（如果使能）驱动为 0。进入或退出休眠模式不会影响以下寄存器。

- UxMODE 和 UxSTA 寄存器
- 发送和接收寄存器和缓冲器
- UxBRG 寄存器

在发送或接收处于等待状态时，没有任何自动方式可用以阻止进入休眠模式。用户可以在进入休眠模式之前检查 RIDLE 位，以避免接收中止。发送器由用户控制，所以用户软件必须将休眠模式与 UART 操作进行同步，以确保发送不会被中止。

对于 UART，USIDL 位用于选择在空闲模式下模块是停止还是继续工作。如果 USIDL = 0，则在空闲模式下模块将继续工作。如果 USIDL = 1，则在空闲模式下模块将停止。在空闲模式下停止时（USIDL = 1），UART 将执行与休眠模式相同的过程。

### 21.11.1 同步间隔字符自动唤醒

自动唤醒功能通过 WAKE 位（UxMODE<7>）来使能。WAKE 生效后，将禁止 UxRX 上的典型接收序列。发生唤醒事件后，模块会产生一个 UxRXIF 中断。

LPBACK 位（UxMODE<6>）必须等于 0 唤醒功能才有效。

唤醒事件是指 UxRX 线上发生高电平到低电平的跳变。这刚好与同步间隔字符或 LIN 协议唤醒信号字符的启动条件一致。WAKE 有效时，无论 CPU 模式如何，都会对 UxRX 线进行监视。在正常的用户模式下，UxRXIF 中断将与 Q 时钟同步产生；在模块因休眠或空闲模式而停止工作时，中断则异步产生。为了确保不会丢失任何实际数据，应在进入休眠模式之前和当 UART 模块处于空闲模式时将 WAKE 位置 1。

发生唤醒事件后，当 UxRX 线上出现由低向高的电平转换时，WAKE 位自动清零。此时，UART 模块将从空闲状态返回正常工作模式，由此用户可知“同步间隔”事件已经结束。如果用户在序列完成之前清零 WAKE 位，则可能导致意外的模块行为。

唤醒事件会通过将 UxRXIF 位置 1 产生一个接收中断。对于该功能，接收中断选择模式位（URXISEL<1:0>）被忽略。如果允许 UxRXIF 中断，则这会唤醒器件。

**注：** 同步间隔（或唤醒信号）字符必须足够长，以便使选定的振荡器有充足的时间起振并保证 UART 正确初始化。为确保部件及时唤醒，用户应读取 WAKE 位的值。如果该位清零，则说明 UART 可能未能及时准备就绪以接收下一个字符，可能需要将模块与总线重新同步。

图 21-13: 正常工作模式下的自动唤醒位 (WAKE) 时序

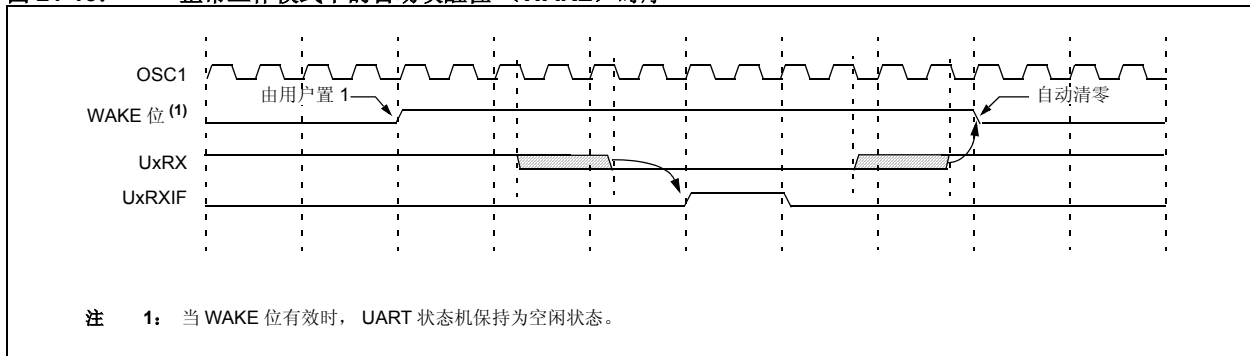
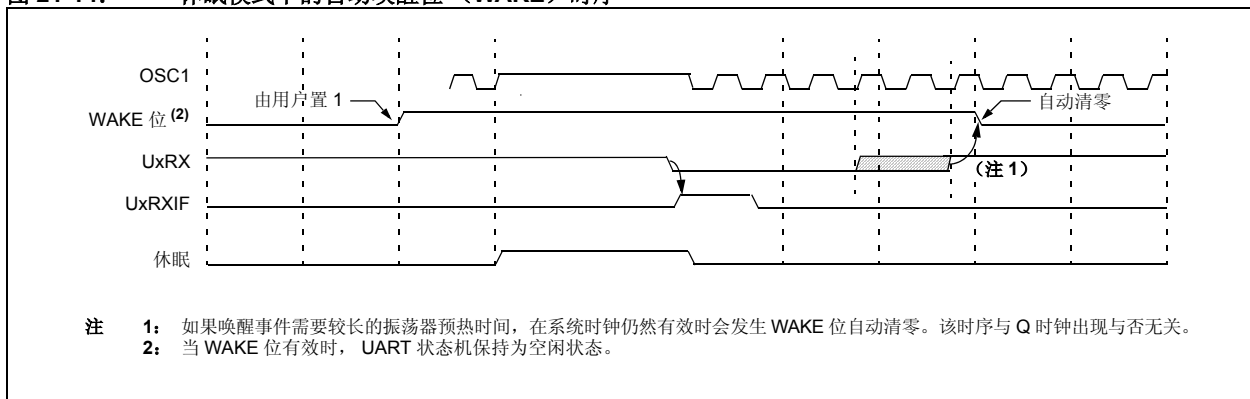


图 21-14: 休眠模式下的自动唤醒位 (WAKE) 时序



## 21.12 $\overline{\text{UxCTS}}$ 和 $\overline{\text{UxRTS}}$ 控制引脚的操作

$\overline{\text{UxCTS}}$  (清零发送) 和  $\overline{\text{UxRTS}}$  (请求发送) 是与 UART 模块相关的两个硬件控制引脚。这两个引脚使 UART 可以工作于单工和流控制模式，第 21.12.2 节“流控制模式下的  $\overline{\text{UxRTS}}$  功能”和第 21.12.3 节“单工模式下的  $\overline{\text{UxRTS}}$  功能”分别详细说明了两种模式。它们用于控制 DTE (数据终端设备) 之间的发送和接收。

### 21.12.1 $\overline{\text{UxCTS}}$ 功能

在 UART 操作中， $\overline{\text{UxCTS}}$  用作可以控制发送的输入引脚。该引脚由另一个器件 (通常为个人计算机) 控制。 $\overline{\text{UxCTS}}$  引脚使用  $\text{UEN}\langle 1:0 \rangle$  进行配置。 $\text{UEN}\langle 1:0 \rangle = 10$  时， $\overline{\text{UxCTS}}$  配置为输入。如果  $\text{UxCTS} = 1$ ，则发送器会装入发送移位寄存器中的数据，但不会启动发送。这使 DTE 可以根据其需求通过控制器相应地控制和接收数据。

$\overline{\text{UxCTS}}$  引脚在发送数据改变的同时 (即在 16 倍频波特率时钟开始时) 进行采样。只有采样到  $\overline{\text{UxCTS}}$  为低电平时才会开始发送。 $\overline{\text{UxCTS}}$  在内部利用 Q 时钟进行采样，这意味着  $\overline{\text{UxCTS}}$  上的时钟脉冲宽度应至少为 1 个  $T_{cy}$ 。不过，这不能作为规范，因为  $T_{cy}$  会随所使用的时钟而变化。

用户也可以通过读相关的端口引脚来读取  $\overline{\text{UxCTS}}$  的状态。

### 21.12.2 流控制模式下的 $\overline{\text{UxRTS}}$ 功能

在流控制模式下，DTE 的  $\overline{\text{UxRTS}}$  连接到 PIC24F 的  $\overline{\text{UxCTS}}$ ，DTE 的  $\overline{\text{UxCTS}}$  连接到 PIC24F 的  $\overline{\text{UxRTS}}$ ，如图 21-15 所示。 $\overline{\text{UxRTS}}$  信号指示器件准备好接收数据。每当  $\text{UEN}\langle 1:0 \rangle = 01$  或  $10$  时， $\overline{\text{UxRTS}}$  引脚就驱动为输出。每当接收器准备好接收数据时， $\overline{\text{UxRTS}}$  引脚就驱动为低电平。当  $\text{RTSMD}$  位 = 0 (器件处于流控制模式) 时， $\overline{\text{UxRTS}}$  引脚在接收器缓冲器未满载或  $\text{OERR}$  位未置 1 时驱动为低电平。当  $\text{RTSMD}$  位 = 0 时， $\overline{\text{UxRTS}}$  引脚在器件未准备好接收时 (即，接收器缓冲器已满载或正在进行移位时)，驱动为高电平。

因为 DTE 的  $\overline{\text{UxRTS}}$  连接到 PIC24F 的  $\overline{\text{UxCTS}}$ ，所以每当它准备好接收数据时， $\overline{\text{UxRTS}}$  就会将  $\overline{\text{UxCTS}}$  驱动为低电平。当  $\overline{\text{UxCTS}}$  变为低电平时，数据发送开始，如第 21.12.1 节“ $\overline{\text{UxCTS}}$  功能”中所述。

### 21.12.3 单工模式下的 $\overline{\text{UxRTS}}$ 功能

在单工模式下，DCE 的  $\overline{\text{UxRTS}}$  连接到 PIC24F 的  $\overline{\text{UxRTS}}$ ，DCE 的  $\overline{\text{UxCTS}}$  连接到 PIC24F 的  $\overline{\text{UxCTS}}$ ，如图 21-16 所示。在单工模式下， $\overline{\text{UxRTS}}$  信号指示 DTE 已准备好发送。每当 DCE 准备好接收发送数据时，DCE 将通过有效的  $\overline{\text{UxCTS}}$  对  $\overline{\text{UxRTS}}$  信号作出答复。当 DTE 接收到有效的  $\overline{\text{UxCTS}}$  时，它将开始发送。

如图 21-17 所示，IEEE-485 系统中也使用单工模式来使能发送器。当  $\overline{\text{UxRTS}}$  指示 DTE 准备好发送时， $\overline{\text{UxRTS}}$  信号将使能驱动器。

每当  $\text{UEN}\langle 1:0 \rangle = 01$  或  $10$  时， $\overline{\text{UxRTS}}$  引脚就驱动为输出。当  $\text{RTSMD} = 1$  时，每当有数据可供发送时 ( $\text{TRMT} = 0$ )， $\overline{\text{UxRTS}}$  就驱动为低电平。当  $\text{RTSMD} = 1$ ，在发送器为空 ( $\text{TRMT} = 1$ ) 时， $\overline{\text{UxRTS}}$  驱动为高电平。

图 21-15: DTE-DTE 的  $\overline{UxRTS}/\overline{UxCTS}$  流控制 (RTSMD = 0, 流控制模式)

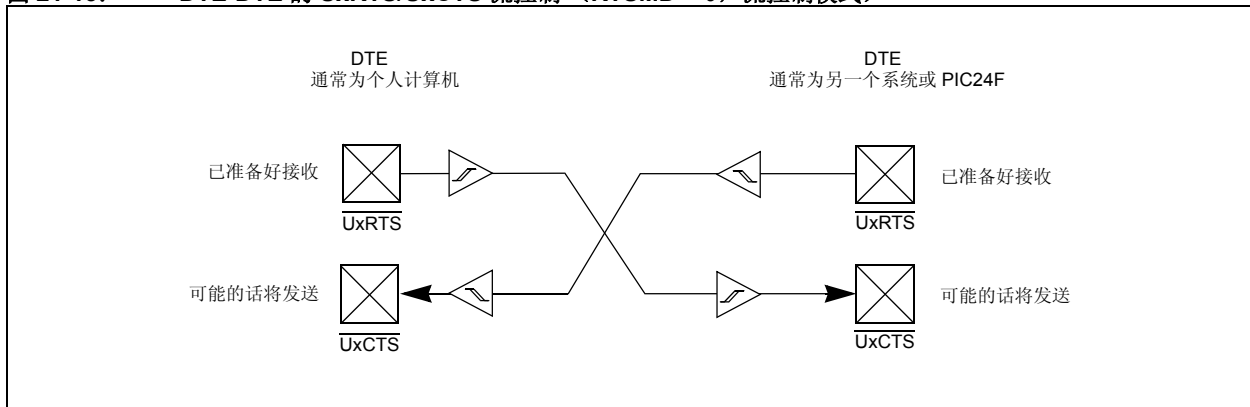


图 21-16: DTE-DCE 的  $\overline{UxRTS}/\overline{UxCTS}$  握手 (RTSMD = 1, 单工模式)

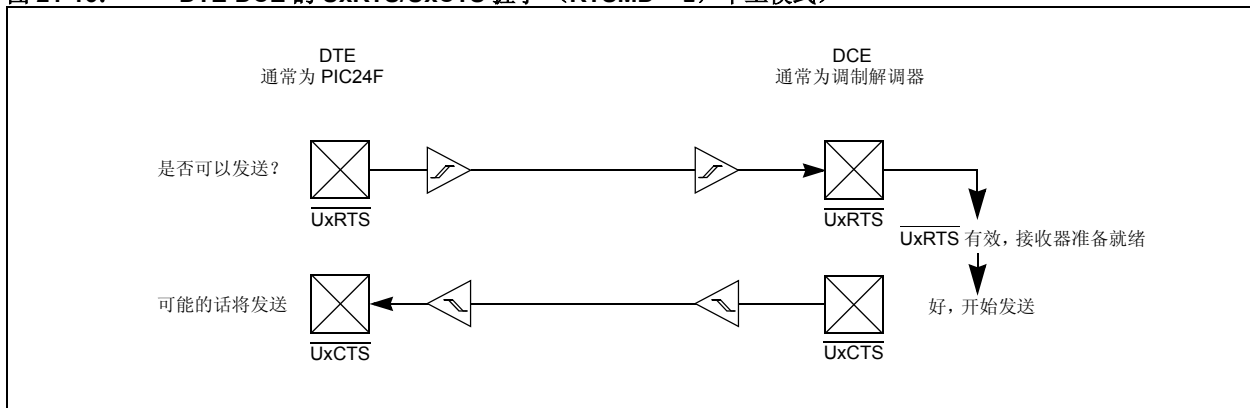
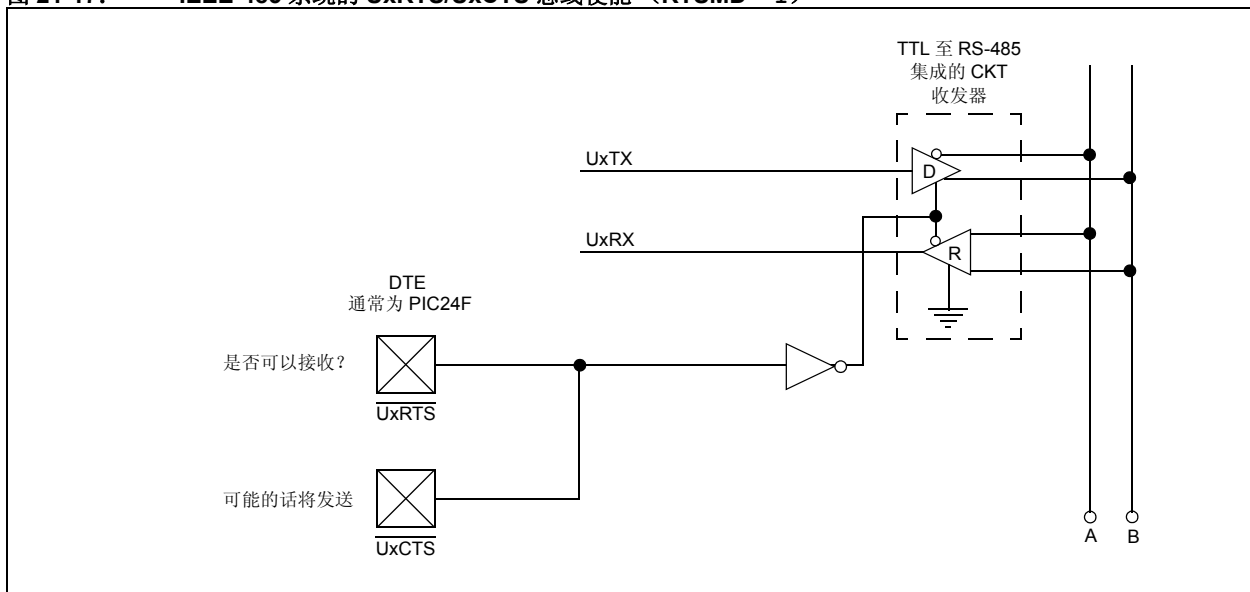


图 21-17: IEEE-485 系统的  $\overline{UxRTS}/\overline{UxCTS}$  总线使能 (RTSMD = 1)



## 21.13 红外线支持

UART 模块提供两种类型的红外线 UART 支持：一种是 IrDA 时钟输出，用以支持外部 IrDA 编码器和解码器（传统模块支持）；另一种是完全实现的 IrDA 编码器和解码器。

### 21.13.1 外部 IrDA 支持——IrDA 时钟输出

要支持外部 IrDA 编码器和解码器，可以将 BCLKx 引脚配置为产生 16 倍频波特率时钟。当 UEN<1:0> = 11 时，如果使能 UART 模块，BCLKx 引脚将输出 16 倍频波特率时钟；它可以用于支持 IrDA 编解码芯片。

### 21.13.2 内置 IrDA 编码器和解码器

UART 具有完全实现的 IrDA 编码器和解码器，作为 UART 模块的组成部分。内置 IrDA 编码器和解码器的功能通过使用 IREN 位 (UxMODE<12>) 来使能。在使能时 (IREN = 1)，接收引脚 (UxRX) 用作来自红外线接收器的输入。发送引脚 (UxTX) 用作到红外线发送器的输出。

#### 21.13.2.1 IrDA 编码器功能

编码器的工作方式为：从 UART 获取串行数据，并使用下面描述的方法替换它。

对于 16 倍频波特率时钟的全部 16 个周期，值为 1 的发送位数据编码为 0。对于 16 倍频波特率时钟的前 7 个周期，值为 0 的发送位数据编码为 0，对于接下来的 3 个周期，编码为 1，对于余下的 6 个周期，则编码为 0。详情请参见图 21-18 和图 21-20。

#### 21.13.2.2 IrDA 发送极性

IrDA 发送极性使用 UTXINV 位 (UxSTA<14>) 进行选择。该位仅在使能 IrDA 编码器和解码器 (IREN = 1) 时影响模块。对于正常的发送和接收，该位既不影响接收器，也不影响模块操作。当 UTXINV = 0 时，UxTX 线的空闲状态为 0 (见图 21-18)。当 UTXINV = 1 时，UxTX 线的空闲状态为 1 (见图 21-19)。

图 21-18: IrDA® 编码方式

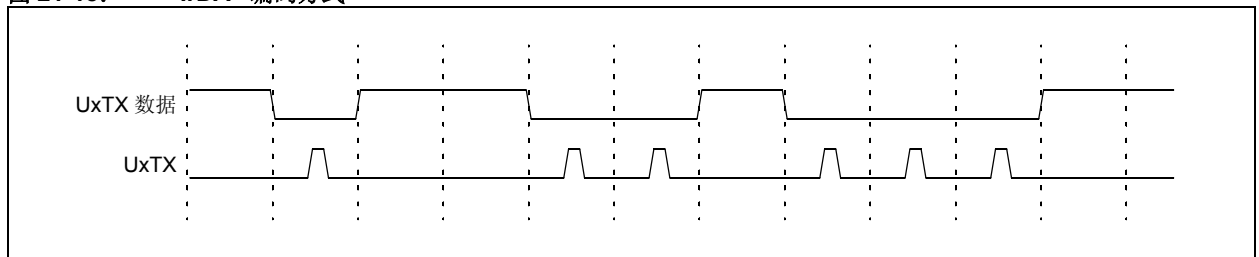


图 21-19: 位数据为 0 时的 IrDA® 编码方式

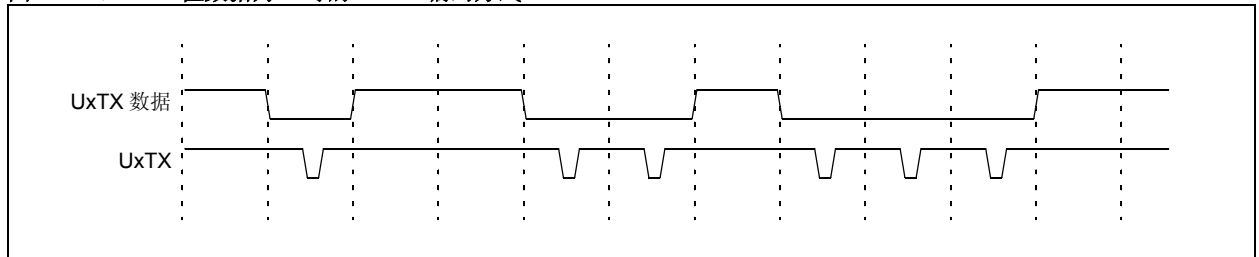
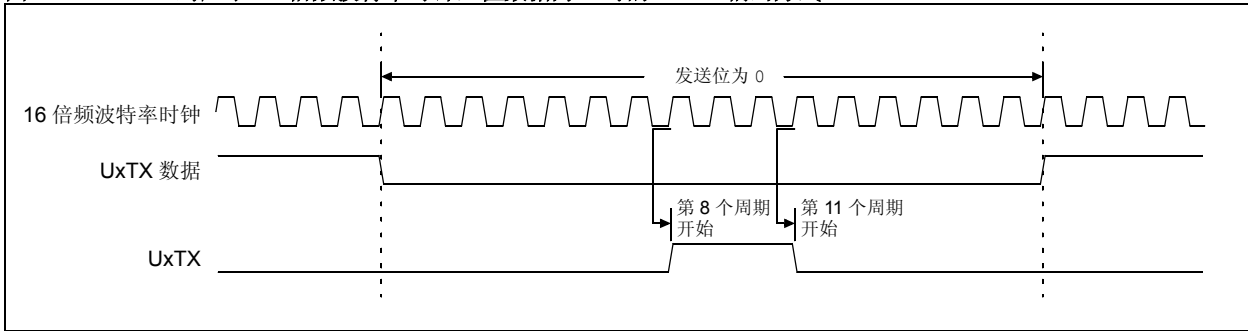


图 21-20: 对应于 16 倍频波特率时钟，位数据为 0 时的 IrDA® 编码方式



### 21.13.2.3 IrDA 解码器功能

解码器的工作方式为：从 UxRX 引脚获取串行数据，并使用解码数据流替换它。数据流根据在 UxRX 输入检测到的下降沿进行编码。

UxRX 的每个下降沿都会使解码数据被驱动为低电平并保持 16 倍频波特率时钟的 16 个周期。如果在 16 个周期计时期满之前，检测到另一个下降沿，则在接下来的 16 个周期，解码数据继续保持为低电平。如果未检测到下降沿，则编码数据驱动为高电平。

请注意，进入器件的数据流与实际报文源之间有 16 倍频波特率时钟 7 至 8 个周期的移位。存在一个时钟的不确定性是由于时钟边沿分辨精度的原因（详情请参见图 21-21）。

### 21.13.2.4 IrDA 接收极性

IrDA 信号的输入可以具有反转的极性。同一逻辑可以解码信号串，但此时，解码数据流与原始报文源之间有 16 倍频波特率时钟的 10 至 11 个周期的移位。同样，存在一个时钟的不确定性是由于时钟边沿分辨精度的原因（详情请参见图 21-22）。

### 21.13.2.5 时钟抖动

由于时钟抖动或器件之间微小的频率差，可能会导致错过某个 16x 周期的下一个下降位边沿。在这种情况下，在解码数据流中会出现一个时钟宽的脉冲。因为 UART 主要在数据位中点附近执行检测，所以这不会导致错误数据（详情请参见图 21-23）。

图 21-21: IrDA® 解码方式的宏观视图

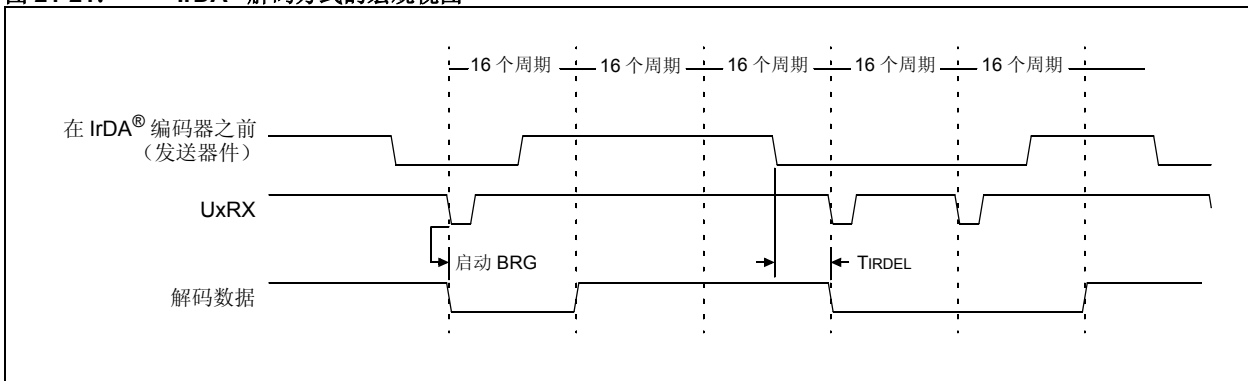




图 21-22: 反转极性解码结果

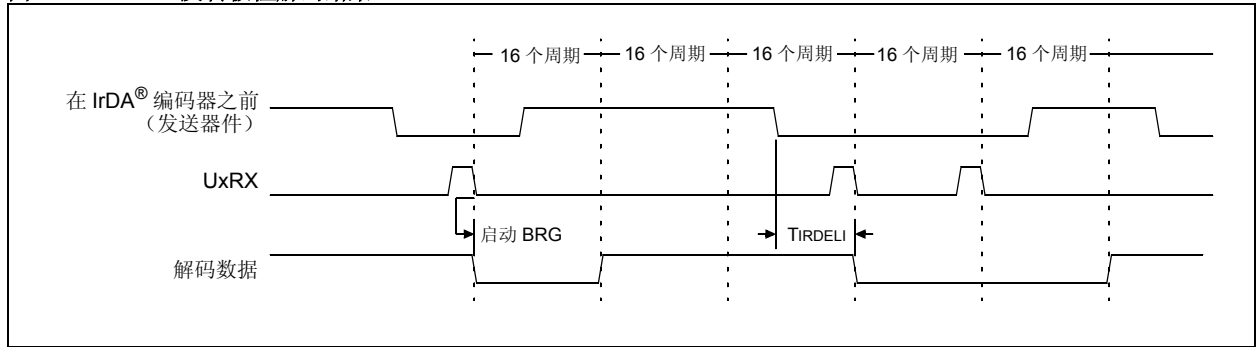
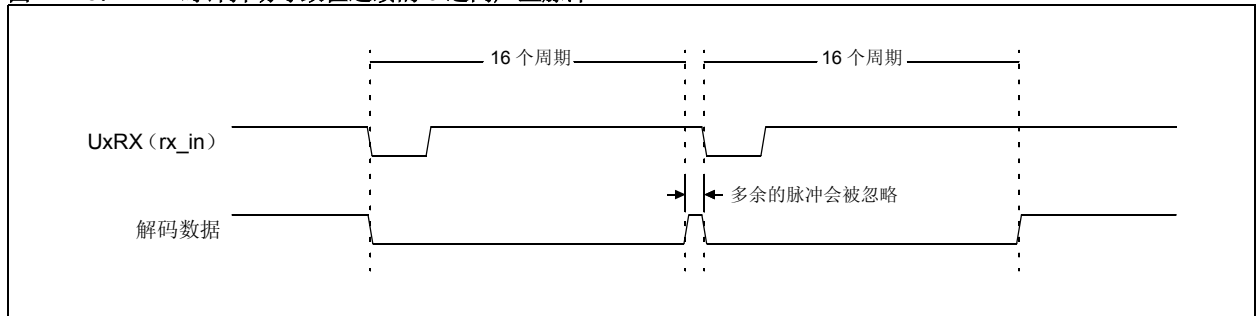


图 21-23: 时钟抖动导致在连续的 0 之间产生脉冲



## 21.14 与 UART 模块相关的寄存器

表 21-4 中提供了与 PIC24F UARTx 相关的寄存器汇总。

表 21-4: 与 UARTx 相关的寄存器

SFR 名称	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时
UxMODE	UARTEN	UFRZ	USIDL	IREN	RTSMO	ALTIO	UEN1	UEN0	WAKE	LPBACK	ABAUD	—	BRGH	PDSEL1	PDSEL0	STSEL	0000
UxSTA	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
UxTXREG	—	—	—	—	—	—	—	UTX8	发送寄存器								xxxx
UxRXREG	—	—	—	—	—	—	—	URX8	接收寄存器								0000
UxBRG	波特率发生器预分频器																0000
IFS0	—	—	AD1IF	UxTXIF	UxRXIF	SPI1IF	SPF1IF	T3IF	T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS4	—	—	—	—	—	—	—	—	—	—	—	—	CRCIF	U2ERIF	UxERIF	—	0000
IEC0	—	—	AD1IE	UxTXIE	UxRXIE	SPI1IE	SPF1IE	T3IE	T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC4	—	—	—	—	—	—	—	—	—	—	—	—	CRCIE	U2ERIE	UxERIE	—	0000
IPC2	—	UxRXIP2	UxRXIP1	UxRXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0	—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0	4444
IPC3	—	—	—	—	—	—	—	—	—	AD1IP2	AD1IP1	AD1IP0	—	UxTXIP2	UxTXIP1	UxTXIP0	0044
IPC16	—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0	—	UxERIP2	UxERIP1	UxERIP0	—	—	—	—	4440

注：与 UARTx 相关的寄存器如上表所示供用户参考。与其他 UART 模块相关的寄存器，请参见器件数据手册。

21.15 电气规范

21.15.1 时序图

图 21-24: 波特率发生器输出时序

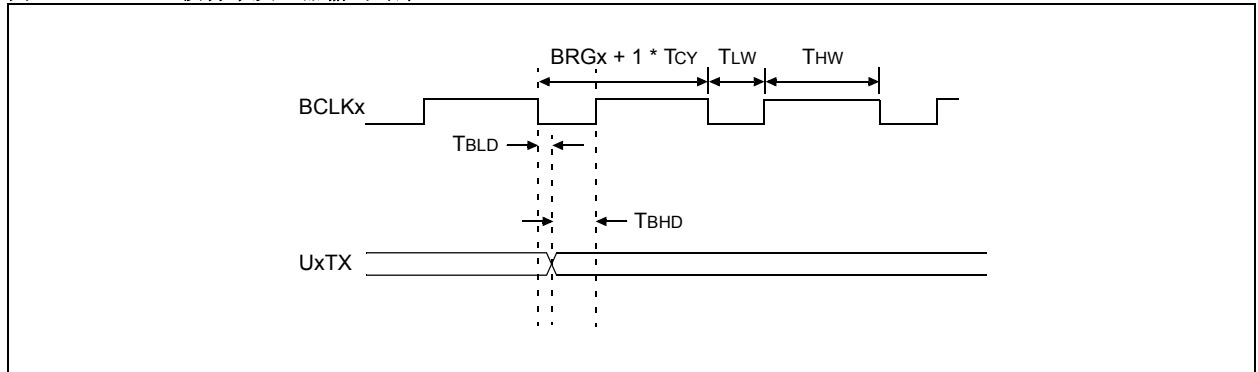


图 21-25: 启动位边沿检测

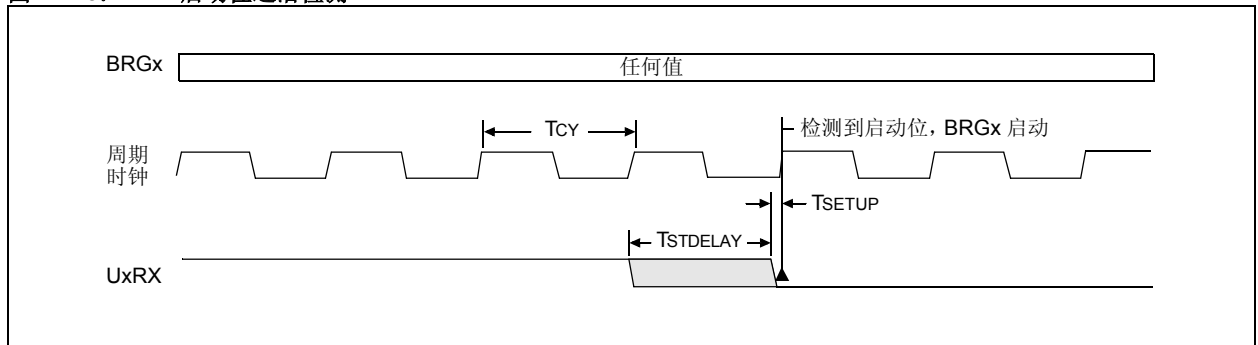


表 21-5: 交流规范

符号	特性	最小值	典型值	最大值	单位
TLW	BCLKx 高电平时间	20	$T_{cy}/2$	—	ns
THW	BCLKx 低电平时间	20	$(T_{cy} * BRGx) + T_{cy}/2$	—	ns
TBLD	来自 UxTX 的 BCLKx 下降沿延时	-50	—	50	ns
TBHD	来自 UxTX 的 BCLKx 上升沿延时	$T_{cy}/2 - 50$	—	$T_{cy}/2 + 50$	ns
TWAK	UxRX 线导致唤醒事件的最小低电平时间	—	1	—	$\mu s$
TCTS	UxCTS 线启动发送的最小低电平时间	$T_{cy}$	—	—	ns
TSETUP	启动位下降沿到系统时钟上升沿的建立时间	3	—	—	ns
TSTDELAY	检测启动位下降沿时的最大延时	—	—	$T_{cy} + T_{SETUP}$	ns

## 21.16 设计技巧

**问 1:** *我用 UART 发送的数据不能正确接收。这是什么原因?*

**答:** 接收错误的最常见的原因是为 UART 波特率发生器计算了一个错误的值。确保写入 UxBRG 寄存器的值是正确的。

**问 2:** *尽管 UART 接收引脚上的信号看上去是正确的，但我还是得到了帧错误。可能是什么原因?*

**答:** 确保以下控制位已正确设置:

- UxBRG: UART 波特率寄存器
- PDSEL<1:0>: 奇偶校验和数据大小选择位
- STSEL: 停止位选择

## 21.17 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC24F 器件系列而编写的，但其概念是相关的，通过适当修改即可使用，但在使用中可能会受到一定限制。当前与 UART 模块相关的应用笔记有：

标题

应用笔记编号

目前没有相关的应用笔记。

**注：** 如需获取更多 PIC24F 系列器件的应用笔记和代码示例，请访问 Microchip 网站 ([www.microchip.com](http://www.microchip.com))。

## 21.18 版本历史

### 版本 A（2006 年 4 月）

这是本文档的初始发行版。