

Receive Mode	[1:0]	确定哪个模式可以从 UART 接收缓冲寄存器读数据。 00 = 禁止 01 = 中断请求或轮询模式 10 = DMA0 请求 (仅用于 UART0), DMA3 请求 (请求信号 0)	00
--------------	-------	---	----

注:

(1) $DIV_VAL = UBRDIVn + (\text{在 } UDIVSLOTn \text{ 上 } 1 \text{ 的数量})/16$. 涉及 UART 波特率配置寄存器。

(2) 是 S3C6410 用的一个水平触发中断控制器。因此每次发送时这个位必须设置为 1。

(3) 当 UART 没有达到 FIFO 触发水平或在 FIFO 下 DMA 接收模式中的三个字的时间内没有接收到数据, Rx 中断会发生 (接收超时), 并且用户应该检测 FIFO 状态读取中断。

(4) EXT_UCLK0 是外部时钟。(XpwmECLK PAD 输入)。

EXT_UCLK1 时钟产生是由 syscon。SYSCON 产生 EXT_UCLK1 为分频 EPLL 或 MPLL 输出。

31.4.3. UART 的 FIFO 控制寄存器

UART 模块中含有四个 UART FIFO 控制寄存器。

寄存器	地址	读/写	描述	复位值
UFCON0	0x7F005008	读/写	UART0 通道 FIFO 控制寄存器。	0x0
UFCON1	0x7F005408	读/写	UART1 通道 FIFO 控制寄存器。	0x0
UFCON2	0x7F005808	读/写	UART2 通道 FIFO 控制寄存器。	0x0
UFCON3	0x7F005C08	读/写	UART3 通道 FIFO 控制寄存器。	0x0

位名称	位	描述	初始状态
Tx FIFO Trigger Level	[7:6]	确定发送 FIFO 的触发条件。 00 = 空 01 = 4 字节 10 = 8 字节 11 = 12 字节	00
Rx FIFO Trigger Level	[5:4]	确定接收 FIFO 的触发条件。 00 = 4 字节 01 = 8 字节 10 = 12 字节 11 = 16 字节	00
Reserved	[3]	保留。	0

Tx FIFO Reset	[2]	Tx 复位，该位在 FIFO 复位后自动清除。 0 = 正常 1= Tx FIFO 复位	0
Rx FIFO Reset	[1]	Rx 复位，该位在 FIFO 复位后自动清除。 0 = 正常 1= Rx FIFO 复位	0
FIFO Enable	[0]	0 =FIFO 禁止 1 = FIFO 模式	0

注意：在 FIFO DMA 接收模式下，当 UART 没有达到 FIFO 触发水平或者在三个字的时间内没有接收到数据时，接收中断会产生（接收超时），并且用户应该检测 FIFO 状态读取中断。

31.4.4. UART Modem 控制寄存器

UART 模块中有两个 UART MODEM 控制寄存器 UMC0N0 和 UMC0N1。

寄存器	地址	读/写	描述	复位值
UMC0N0	0x7F00500C	读/写	UART0 通道 Modem 控制寄存器。	0x0
UMC0N1	0x7F00540C	读/写	UART1 通道 Modem 控制寄存器。	0x0
Reserved	0x7F00580C	-	保留。	未定义
Reserved	0x7F005C0C	-	保留。	未定义

位名称	位	描述	初始状态
Reserved	[7:5]	当 AFC 被激活，这个位决定什么时候阻止信号。 000 =接收 FIFO 控制 63 字节 001 =接收 FIFO 控制 56 字节 010 =接收 FIFO 控制 48 字节 011 =接收 FIFO 控制 40 字节 100 =接收 FIFO 控制 32 字节 101 = 接收 FIFO 控制 24 字节	000
Auto Flow Control (AFC)	[4]	AFC 是否允许： 0 = 禁止 1 =激活	0
Reserved	[3:1]	这 3 位必须均为 0。	00

Request to Send	[0]	如果 AFC 位允许，则该位忽略，这时，S3C6410 将自动控制 nRTS。 如果 AFC 位禁止，则 nRTS 必须被软件控制。 0 = 'H' 电平 (nRTS 无效) 1 = 'L' 电平 (nRTS 有效)	0
-----------------	-----	---	---

注：UART2 不支持 AFC 功能，因为 S3C6410 没有 nRTS2 和 nCTS2。

UART3 不支持 AFC 功能，因为 S3C6410 没有 nRTS2 和 nCTS2。

31.4.5. UART 接收 (Rx) / (Tx) 发送状态寄存器

UART 模块有四个 UART 接收/发送状态寄存器：UTRSTAT0、UTRSTAT1、UTRSTAT2 和 UTRSTAT3。

寄存器	地址	读/写	描述	复位值
UTRSTAT0	0x7F005010	读	UART 0 通道 Tx/Rx 状态寄存器。	0x6
UTRSTAT1	0x7F005410	读	UART 1 通道 Tx/Rx 状态寄存器。	0x6
UTRSTAT2	0x7F005810	读	UART 2 通道 Tx/Rx 状态寄存器。	0x6
UTRSTAT3	0x7F005C10	读	UART 3 通道 Tx/Rx 状态寄存器。	0x6

位名称	位	描述	初始状态
Transmitter empty	[2]	在发送缓冲寄存器没有有效数据或发送移位寄存器为空时，该位自动置 1。 0 = 不空 1 = 发送器（发送缓冲寄存器和移位寄存器）空	1
Transmit buffer empty	[1]	当发送缓冲寄存器为空时，该位自动置 1。 0 = 发送缓冲寄存器不空 1 = 空 (在非 FIFO 模式，中断或 DMA 被申请。在 FIFO 模式，当 Tx FIFO 触发水平被设置为 00 (空) 时，中断或 DMA 被申请。) 如果 UART 使用 FIFO，则用户应该检查 UFSTAT 寄存器的 Tx FIFO 计数位和 Tx FIFO 满标志位，以代替检查该位。	1

Receive buffer data ready	[0]	<p>无论何时接收缓冲寄存器包含在 RXDn 接口接收的有效数据，该位自动置 1。</p> <p>0 = 空</p> <p>1 = 接收缓冲寄存器有接收数据（在非 FIFO 模式，中断或 DMA 被申请）。</p> <p>如果 UART 使用 FIFO，则用户应该检查 USTAT 寄存器中的 Rx FIFO 计数位和 Rx FIFO 满标志位以代替检查该位。</p>	0
---------------------------	-----	--	---

31.4.6. UART 错误状态寄存器

UART 模块有四个 UART 错误状态寄存器：UERSTAT0、UERSTAT1、UERSTAT2 和 UERSTAT3。

寄存器	地址	读/写	描述	复位值
UERSTAT0	0x7F005014	读	UART 0 通道错误状态寄存器。	0x0
UERSTAT1	0x7F005414	读	UART 1 通道错误状态寄存器。	0x0
UERSTAT2	0x7F005814	读	UART 2 通道错误状态寄存器。	0x0
UERSTAT3	0x7F005C14	读	UART 3 通道错误状态寄存器。	0x0

位名称	位	描述	初始状态
Break Detect	[3]	<p>自动设置为 1 说明中断信号接收中断信号。</p> <p>0 = 没有中断信号</p> <p>1 = 中断接收 (请求中断)</p>	0
Frame Error	[2]	<p>在接收过程中无论何时发生帧错误，该位自动置 1。</p> <p>0 = 没发生帧错误</p> <p>1 = 发生帧错误 (请求中断)</p>	0
Parity Error	[1]	<p>在接收过程中无论何时发生奇偶错误，该位自动置 1。</p> <p>0 = 没发生奇偶错误</p> <p>1 = 发生奇偶错误 (请求中断)</p>	0
Overrun Error	[0]	<p>在接收过程中无论何时发生溢出错误时，该位自动置 1。</p> <p>0 = 没发生溢出错误</p> <p>1 = 发生溢出错误 (请求中断)</p>	0

注意：当 UART 错误状态寄存器被读时，这些位会自动清 0。

31.4.7. UART 的 FIFO 状态寄存器

UART 模块有四个 FIFO 状态寄存器：UFSTAT0、UFSTAT1、UFSTAT2 和 UFSTAT3。

寄存器	地址	读/写	描述	复位值
UFSTAT0	0x7F005018	读	UART 0 通道 FIFO 状态寄存器。	0x00
UFSTAT1	0x7F005418	读	UART 1 通道 FIFO 状态寄存器。	0x00
UFSTAT2	0x7F005818	读	UART 2 通道 FIFO 状态寄存器。	0x00
UFSTAT3	0x7F005C18	读	UART 3 通道 FIFO 状态寄存器。	0x00

位名称	位	描述	初始状态
保留	[15]	保留。	0
Tx FIFO Full	[14]	无论何时发送 FIFO 满时，该位自动置 1。 0 = 0 字节 ≤ Tx FIFO 中的数据 ≤ 63 字节 1 = Tx FIFO 中的数据满	0
Tx FIFO Count	[13: 8]	Tx FIFO 数据中的数量。	0
Reserved	[7]	保留。	
Rx FIFO Full	[6]	无论何时接收 FIFO 满时，该位自动置 1。 0 = 0 字节 ≤ Rx FIFO 数据 ≤ 63 字节 1 = Rx FIFO 中的数据满	0
Rx FIFO Count	[5: 0]	Rx FIFO 数据中的数量。	0

31.4.8. UART Modem 状态寄存器

UART 模块有两个 UART Modem 状态寄存器：UMSTAT0 和 UMSTAT1。

寄存器	地址	读/写	描述	复位值
UMSTAT0	0x7F00501C	读	UART0 通道 Modem 状态寄存器。	0x0
UMSTAT1	0x7F00541C	读	UART1 通道 Modem 状态寄存器。	0x0
Reserved	0x7F00581C	-	保留。	未定义
Reserved	0x7F005C1C	-	保留。	未定义

位名称	位	描述	初始状态
Reserved	[7: 5]	保留。	000
Delta CTS	[4]	该位指示输入到 S3C6410 的 nCTS 信号自从上次读后是否已经改变状态。（如图 6-56 所示） 0 = 没有改变	0
Reserved	[3:1]	保留。	00
Clear to Send	[0]	0 = CTS 信号没有改变(nCTS 引脚为高电平)。 1 = CTS 信号改变(nCTS 引脚为低电平)。	0

nCTS 和 Delta CTS 时序表的显示，如图 31-6 所示。

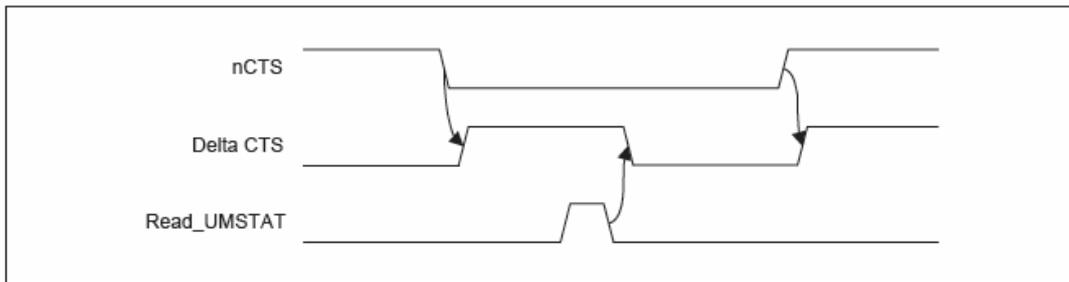


图 31-6 nCTS 和 Delta CTS 时序表

31.4.9. UART 发送缓冲寄存器（保存寄存器和 FIFO 寄存器）

UART 模块有四个发送缓冲寄存器寄存器：UTXH0、UTXH1、UTXH2 和 UTXH3。UTXHn 有一个 8 位数据作为发送数据。

寄存器	地址	R/写	描述	复位值
UTXH0	0x7F005020	写	UART 0 通道发送缓冲寄存器。	-
UTXH1	0x7F005420	写	UART 1 通道发送缓冲寄存器。	-
UTXH2	0x7F005820	写	UART 2 通道发送缓冲寄存器。	-
UTXH3	0x7F005C20	写	UART 3 通道发送缓冲寄存器。	-

位名称	位	描述	初始值
TXDATAn	[7:0]	UARTn 的发送数据。	-

31.4.10 .UART 接收缓冲寄存器（保存寄存器和 FIFO 寄存器）

UART 模块有四个接收缓冲寄存器寄存器：URXH0、URXH1、URXH2 和 URXH3。URXHn 有一个 8 位数据作为发送数据。

寄存器	地址	读/写	描述	复位值
URXH0	0x7F005024	读	UART 0 通道接收缓冲寄存器。	-
URXH1	0x7F005424	读	UART 1 通道接收缓冲寄存器。	-
URXH2	0x7F005824	读	UART 2 通道接收缓冲寄存器。	-
URXH3	0x7F005C24	读	UART 3 通道接收缓冲寄存器。	-

URXHn	位	描述	初始状态
RXDATAn	[7:0]	UARTn 的接收数据。	-

注意：当溢出错误产生时，URXHn 必须被读。否则，即使该 UERSTATn 溢出错误位清 0，下一个接收数据也会产生溢出错误。

31.4.11. UART 波特率分频寄存器

UART 模块有四个波特率分频寄存器：UBRDIV0、UBRDIV1、UBRDIV2 和 UBRDIV3，如表 6-90 所示。UBRDIVn 中的值决定串行 Tx/Rx 时钟波特率，如下：

$$\text{DIV_VAL} = \text{UBRDIVn} + (\text{UDIVSLOTn 中 1 的量}) / 16$$

$$\text{DIV_VAL} = (\text{PCLK} / (\text{b/s} \times 16)) - 1$$

$$\text{DIV_VAL} = (\text{EXT_UCLK0} / (\text{b/s} \times 16)) - 1$$

或者

$$\text{DIV_VAL} = (\text{EXT_UCLK1} / (\text{b/s} \times 16)) - 1$$

除数的范围为 1 到 (216 - 1)，并且 UEXTCLK 应该比 PCLK 小。

利用 UDIVSLOT, 能够得到更准确的波特率。例如，如果波特率是 115200 b/s PCLK、EXT_UCLK0 或

EXT_UCLK1 是 40 MHz ， UBRDIV_n 和 UDIVSLOT_n 是：

$$\begin{aligned} \text{DIV_VAL} &= (40000000 / (115200 \times 16)) - 1 \\ &= 21.7 - 1 \\ &= 20.7 \end{aligned}$$

UBRDIV_n = 20 (DIV_VAL 的整数部分)

(UDIVSLOT_n 中 1 的数量)/16 = 0.7

这时， (UDIVSLOT_n 中 1 的数量) = 11

因此， UDIVSLOT_n 为 16' b1110_1110_1110_1010 或者 16' b0111_0111_0111_0101 等。

UDIVSLOT_n 选择如下表 31-5 所示：

表 31-5 UDIVSLOT_n

Num of 1' s	UDIVSLOT _n	Num of 1' s	UDIVSLOT _n
0	0x0000(0000_0000_0000_0000b)	8	0x5555(0101_0101_0101_0101b)
1	0x0080(0000_0000_0000_1000b)	9	0xD555(1101_0101_0101_0101b)
2	0x0808(0000_1000_0000_1000b)	10	0xD5D5(1101_0101_1101_0101b)
3	0x0888(0000_1000_1000_1000b)	11	0xDDD5(1101_1101_1101_0101b)
4	0x2222(0010_0010_0010_0010b)	12	0xDDDD(1101_1101_1101_1101b)
5	0x4924(0100_1001_0010_0100b)	13	0xDFDD(1101_1111_1101_1101b)
6	0x4A52(0100_1010_0101_0010b)	14	0xDFDF(1101_1111_1101_1111b)
7	0x54AA(0101_0100_1010_1010b)	15	0xFFDF(1111_1111_1101_1111b)

31.4.12. 波特率错误容限

UART 帧错误率应当限制在 1.87%(3/160) 以内。

$t_{UPCLK} = (UBRDIV_n + 1) \times 16 \times 1\text{帧} / PCLK$ t_{UPCLK} : 实际 UART 时钟。

$t_{EXTUARTCLK} = 1\text{帧} / \text{波特率}$ $t_{EXTUARTCLK}$: 理想 UART 时钟。

UART 错误 = $(t_{UPCLK} - t_{EXTUARTCLK}) / t_{EXTUARTCLK} \times 100\%$ 。

注：1FRAME = START 位+DATA 位+ PARITY 位+STOP 位。

寄存器	地址	读/写	描述	复位值
UBRDIV0	0x7F005028	读/写	波特率分频寄存器 0。	0x0000
UBRDIV1	0x7F005428	读/写	波特率分频寄存器 1。	0x0000
UBRDIV2	0x7F005828	读/写	波特率分频寄存器 2。	0x0000
UBRDIV3	0x7F005C28	读/写	波特率分频寄存器 3。	0x0000

UBRDIV _n	位	描述	初始状态
UBRDIV	[15:0]	波特率分频值。UBRDIV _n > 0	-

寄存器	地址	读/写	描述	复位值
UDIVSLOT0	0x7F00502C	读/写	波特率分频寄存器 0。	0x0000
UDIVSLOT1	0x7F00542C	读/写	波特率分频寄存器 1。	0x0000
UDIVSLOT2	0x7F00582C	读/写	波特率分频寄存器 2。	0x0000
UDIVSLOT3	0x7F005C2C	读/写	波特率分频寄存器 3。	0x0000

UDIVSLOT n	位	描述	初始状态
UDIVSLOT	[15:0]	选择由 (UBRDIV + 2) 产生时钟产生分频时钟源的插槽。	-

31.4.13. UART 中断处理寄存器

中断处理寄存器包括产生中断的信息。

寄存器	地址	读/写	描述	复位值
UINTP0	0x7F005030	读/写	UART 0 通道中断处理寄存器。	0x0
UINTP1	0x7F005430	读/写	UART 1 通道中断处理寄存器。	0x0
UINTP2	0x7F005830	读/写	UART 2 通道中断处理寄存器。	0x0
UINTP3	0x7F005C30	读/写	UART 3 通道中断处理寄存器。	0x0

UINTPn	位	描述	初始状态
MODEM	[3]	产生 Modem 中断。	0x0
TXD	[2]	产生发送中断。	0x0
ERROR	[1]	产生错误中断。	0x0
RXD	[0]	产生接收中断	0x0

当 4 位有一位置位逻辑 ‘1’ 时，UART 每个通道都产生中断。在中断服务程序这个寄存器被清理。可以通过置 ‘1’ 在指定的位来清理 UINTP 特殊的位。

31.4.14. UART 中断源处理寄存器

中断源处理寄存器包含产生中断的信息（不管中断屏蔽为何值）。

寄存器	地址	读/写	描述	复位值
UINSP0	0x7F005034	读/写	中断源处理寄存器 0。	0x0
UINSP1	0x7F005434	读/写	中断源处理寄存器 1。	0x0
UINSP2	0x7F005834	读/写	中断源处理寄存器 2。	0x0
UINSP3	0x7F005C34	读/写	中断源处理寄存器 3。	0x0

UINTSPn	位	描述	初始状态
MODEM	[3]	产生 Modem 中断。	00
TXD	[2]	产生发送中断。	0
ERROR	[1]	产生错误中断。	0
RXD	[0]	产生接收中断。	0

31.4.15. UART 中断屏蔽寄存器

中断屏蔽寄存器包含屏蔽中断信息。如果一个特殊位被置为‘1’，尽管相应的中断产生，但不产生到中断控制器的中断请求信号（在这中情况下，UINTSPn 寄存器相应位置为‘1’）。如果屏蔽为是‘0’，中断请求能从相应的中断源得到响应（在这中情况下，UINTSPn 寄存器相应位置为‘1’）。

寄存器	地址	读/写	描述	复位值
UINTM0	0x7F005038	读/写	UART 0 通道中断屏蔽寄存器。	0x0
UINTM1	0x7F005438	读/写	UART 1 通道中断屏蔽寄存器。	0x0
UINTM2	0x7F005838	读/写	UART 2 通道中断屏蔽寄存器。	0x0
UINTM3	0x7F005C38	读/写	UART 3 通道中断屏蔽寄存器。	0x0

UINTMn	位	描述	初始状态
MODEM	[3]	屏蔽 Modem 中断。	0
TXD	[2]	屏蔽发送中断。	0
ERROR	[1]	屏蔽错误中断。	0
RXD	[0]	屏蔽接收中断。	0

31.5 UART 接口应用举例

UART 接口的应用十分广泛，是学习嵌入式开发所必不可少的内容，下面是 UART 接口在 ARM11 处理器中的实例应用。针对以上对 UART 接口特性及操作的理解，再参照各个寄存器功能的描述，具体的程序代码分析如下：

UART 接口的配置：UART_Config 函数主要功能是由用户选择建立 UART 接口。

输入：NONE

输出：NONE

```
u8 UART_Config(void)
```

```
{
```

```
    u8 cCh;
```