新型的高性能双通信模块—USCI

Thomas Kot

德州仪器**MSP430**资深市场工程师

thomas-kot@ti.com

会议议程

- **USCI** 介绍
- **UART/LinBUS** 异步模式
- **SPI** 同步模式
- **I^2C** 同步模式
- 选择正确的总线
- 设备选择
- 模块的同时工作

通用串行通信 I/F

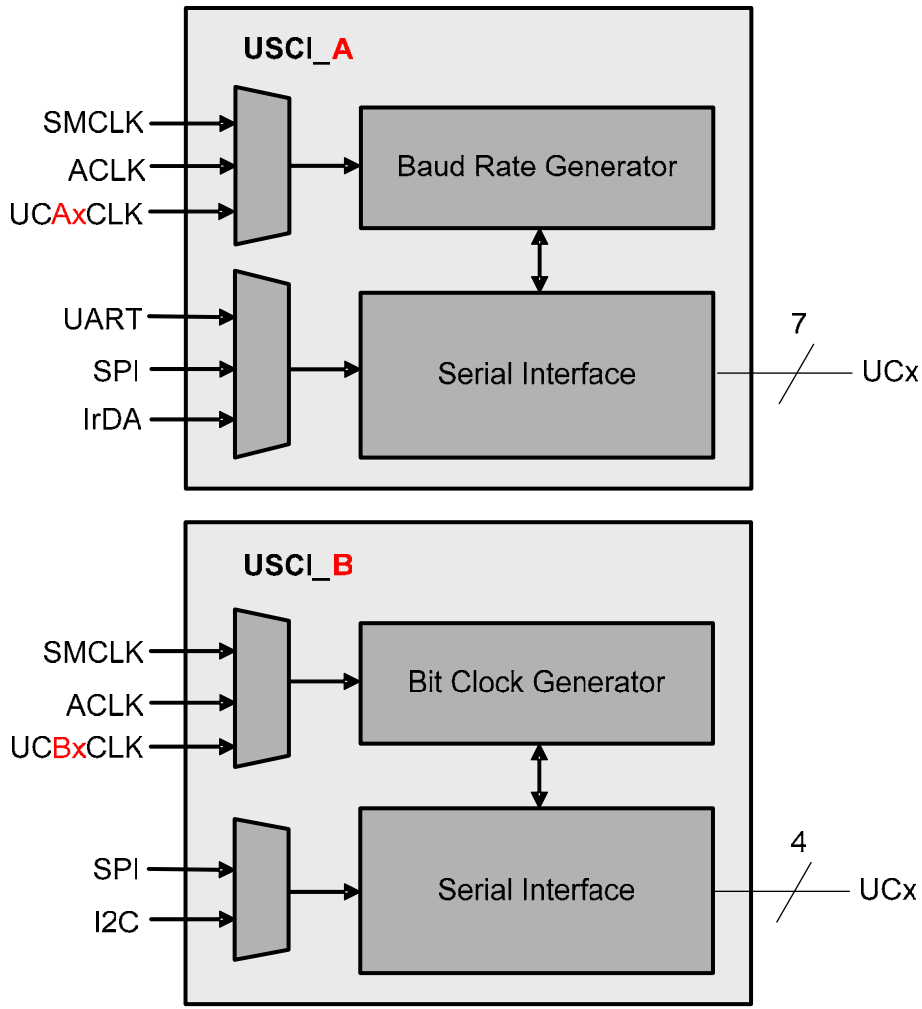
- 超低功耗

  LPMx 工作
- **2** 个独立块
- 双缓存 **TX/RX**
- **RX** 干扰抑制
- 波特率发生器

  灵活的时钟源

  自动检测

  生成
- 启用 **DMA**
- 中断驱动



特性概览

- 新型标准 **MSP430** 串行接口
- 两个相互独立的通信块
- 异步通信模式

UART 标准与多处理器协议

带自动波特率检测的 UART（LIN 支持）

IrDA（SIR——低红外，最大 115k 波特）

LPMx 唤醒

- 同步通信模式

SPI（主从模式，3 或 4 线）

I2C（主从模式）

LPMx 工作

USCI 与 USART 的差异

USCI

- UART

两个调制器可支持 n/16 计时
自动波特率检测功能：LIN
集成式 IrDA 编码器与解码器
同步 USCI_A/USCI_B

- SPI

双 SPI：USCI_A 与 B 各一个

- I^2C

经简化，方便易用

USART

- UART

仅一个调制器

N/A

N/A

N/A

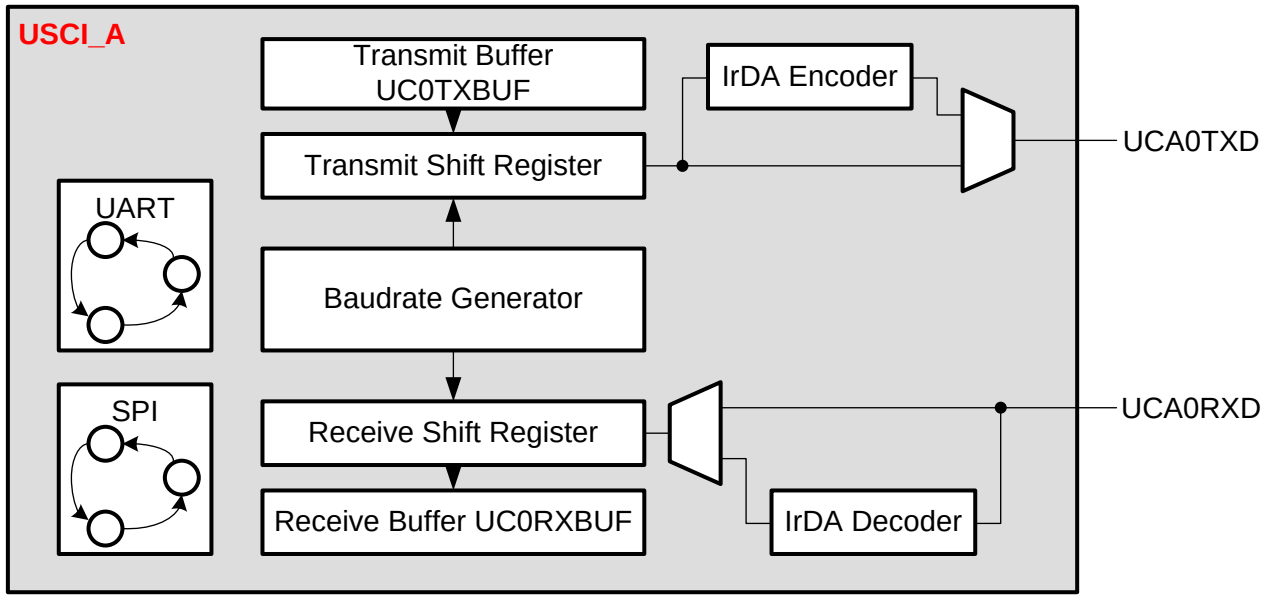
- SPI

仅提供一个 SPI

- I^2C

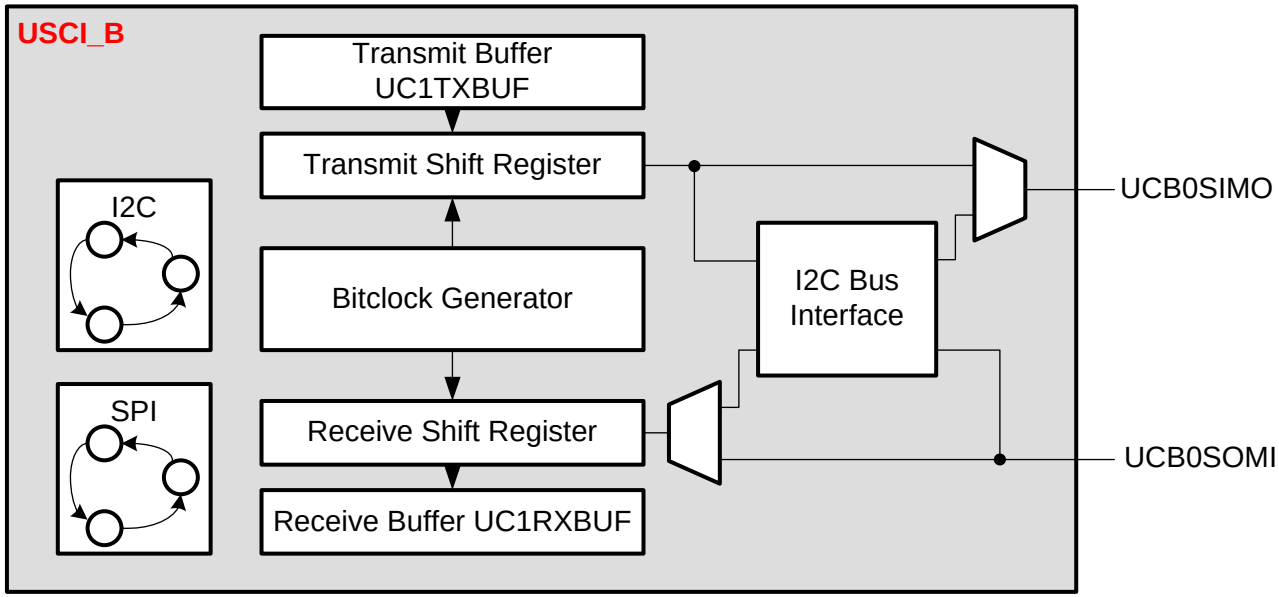
复杂特性

USCI_A



- 支持 **IrDA/LIN** 或 **SPI** 的 **UART**
- 双缓存 **TX/RX**
- 可自动检测波特率的波特率发生器

USCI_B



- **I²C** 主 / 从，最大 **400kHz**，或 **SPI**
- 比特时钟发生器
- 双缓存 **TX/RX**

USCI A UART 模式

- 超低功耗
- 灵活

7 或 8 位数据

奇、偶或非奇偶

LSB 或 MSB 优先

干扰抑制

- 通信方案

IrDA

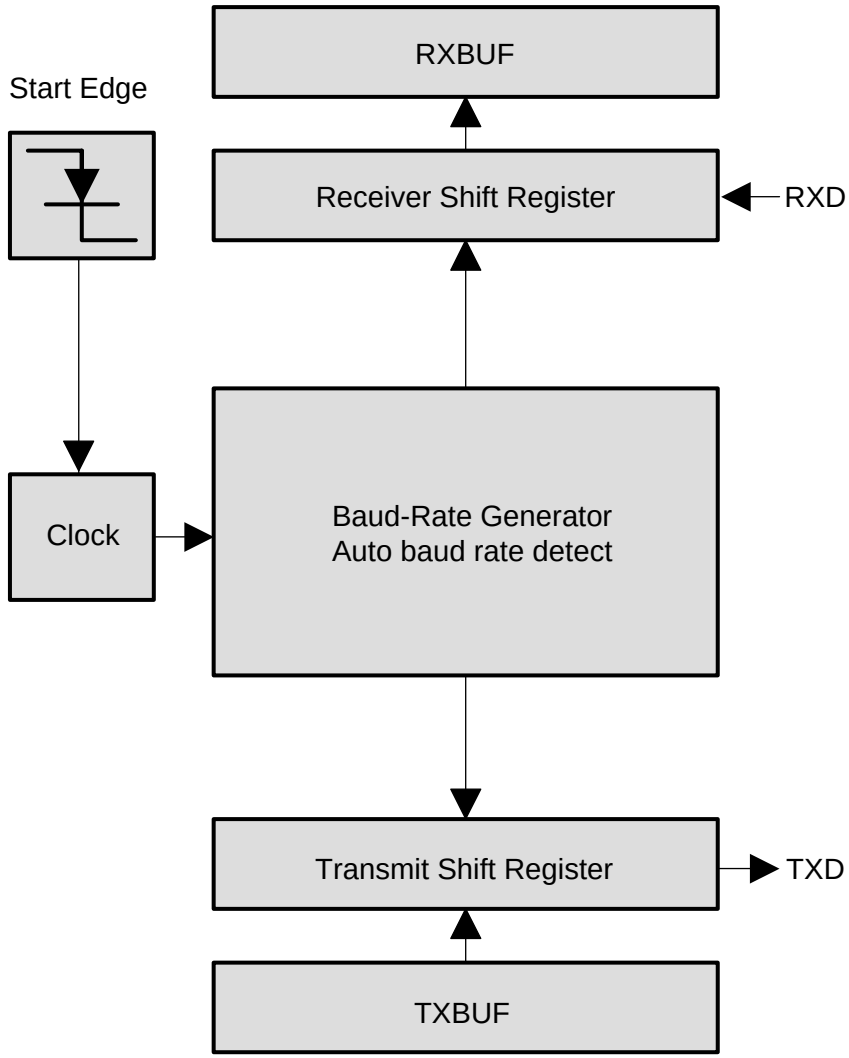
多处理器

自动波特 (S/W LIN)

- 中断驱动

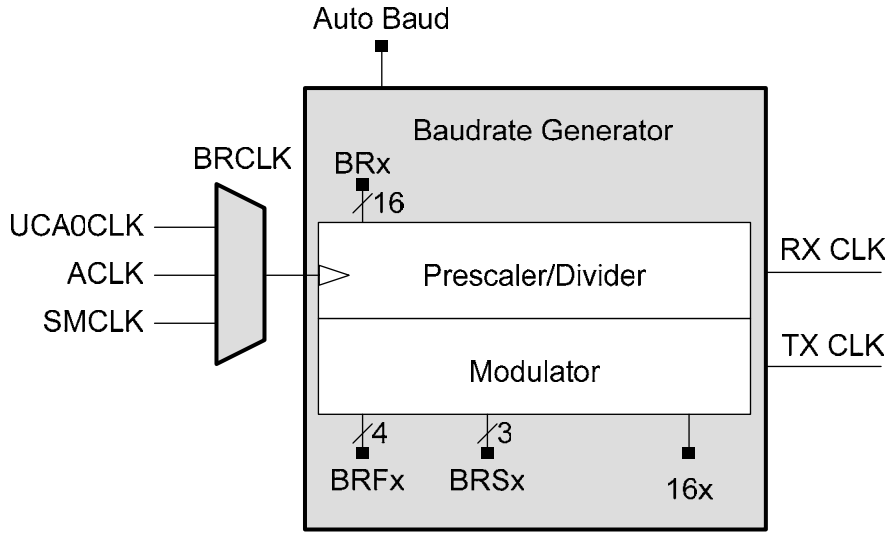
故障检测

TX/RX

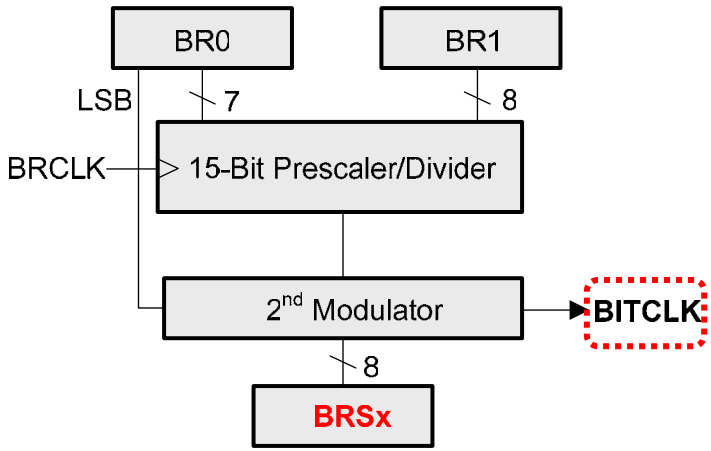


UART 波特率生成器

- 低功耗低频率模式
  - 可使用较慢的时钟
  - n/8 调制器
  - 最大波特率 1/3 BRCLK
  - 大多数情况 3x BITCLK
- **16x 过采样模式**
  - 准标准 UART、LIN、IrDA
  - n/16 调制器
  - 最大波特率 1/16 BRCLK
  - 大多数情况 16x BITCLK
- 灵活的时钟源
  - 外部 UCA0CLK 输入
  - ACLK
  - SMCLK



32KHz XTAL实现 UART 9600 波特率通信

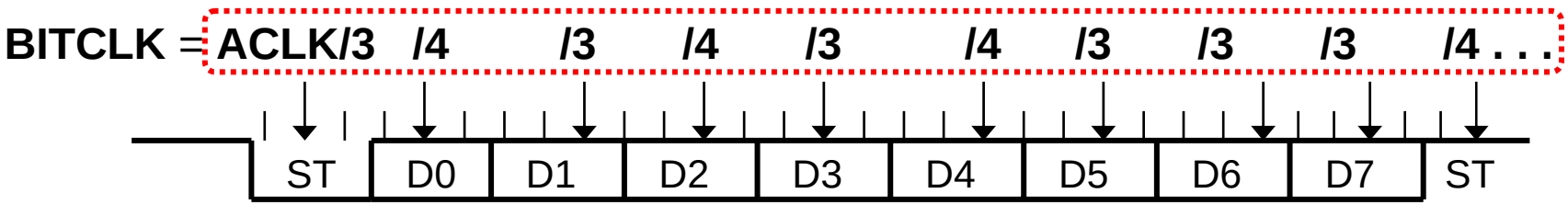


9600 波特率：

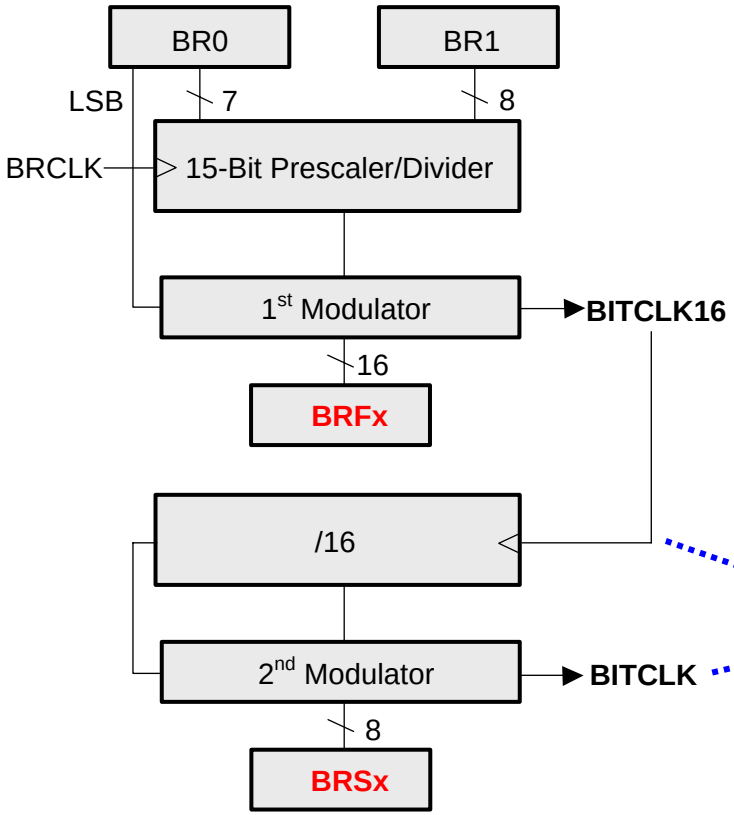
BRCLK = ACLK = 32768 Hz

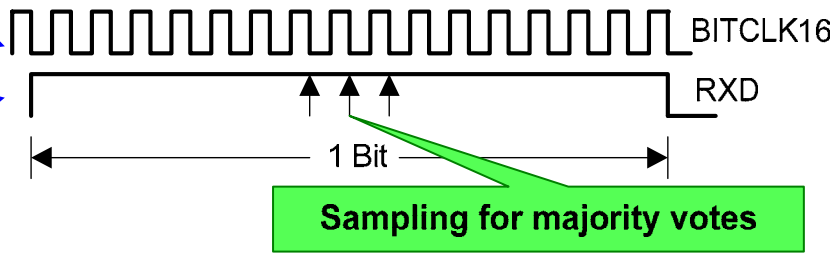
Baud = 32768/9600 = 3.41

BR1 | BR0 | MCTL = 00h 03h 06h: modulation wraps around after 8 bits



过采样模式波特率





- 双调制器
- 用 **BITCLK16** 进行 **RX** 采样
- **BITCLK16** 支持 **IrDA TX** 脉冲发生
- 准标准 **UART、IrDA & LIN**

UART 超低功耗工作

- 从 **LPMx** 唤醒
- 如果 **LPMx** 关闭 **BRCLK** 源，则 **RX** 起始沿或自动写入 **TXBUF** 可打开内部 **BRCLK** 源
- **BRCLK** 源在 **LPMx** 中的 **TX** 或 **RX** 终点后关闭
- 无需软件处理
- 瞬时开启的时钟避免了丢字符的损失
- **LPM3** 模式中，**SMCLK** 可支持高波特率！

LPMx 操作为什么重要？

演示：在起始沿上唤醒 LPM3

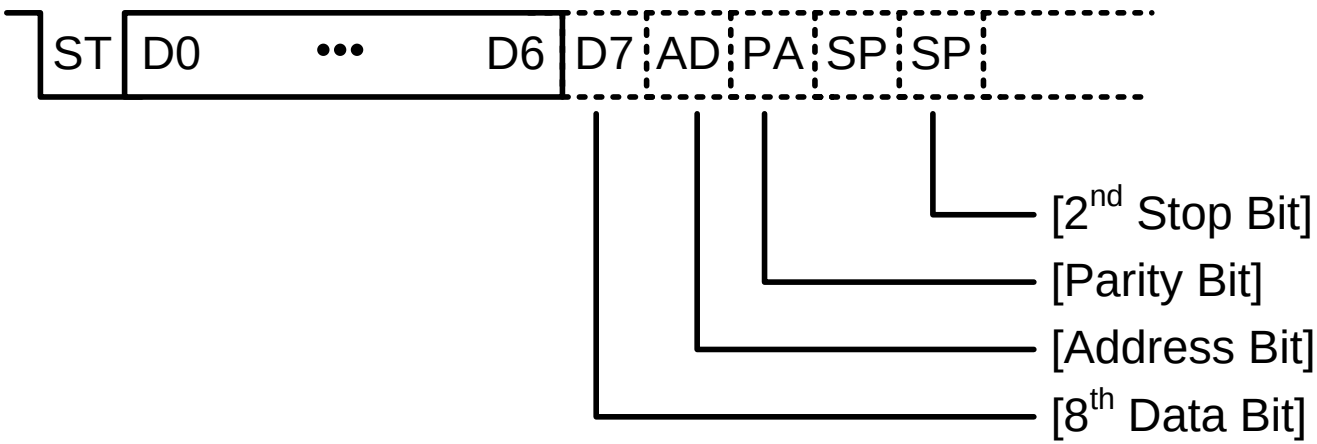
```
SetupUSCIA  mov.b #UCSWRST,&UC0CTL1     ; module reset
            bis.b #UCSSEL1,&UC0CTL1     ; SMCLK
            mov.b #0x09,&UC0BR0         ; 115k at 1048576Hz
            mov.b #0x00,&UC0BR1         ;
            mov.b #0x08,&UC0MCTL        ; modulation values
            bic.b #UCSWRST, &UC0CTL1    ; state m/c start
            bis.b #UC0RXIE, &IE2        ; enable RXinterrupt

Mainloop    bis.b #LPM3+GIE,SR          ; enter LPM3, enable
                                        ; interrupts
USCI01RX_ISR //Function: Echo received character

            bit.b #UC0TXIFG,&IFG2       ; TX buffer ready?
            jz    TX0                   ; jump if not ready
            mov.b &UC0RXBUF,&UC0TXBUF   ; RX -> TX
TX0         reti                        ;
```

- **LPM3 中115.2 kbps 采用 DCO/FLL 作为时钟源**

UART 通信协议



- 支持四种通信协议
  - 标准
  - 闲置线路多处理器
  - 地址位多处理器
  - LIN 的自动波特率检测
- 标准协议
  - 带故障检测的标准 UART 通信

UART 错误检测

- 干扰抑制器滤掉 **RX bit level < de-glitch time**
- 自动错误检测

  帧错误 FE

  如停止位在接收帧中缺失，则设置 FE =1

  奇偶校验错误 PE

  如接收帧中存在奇偶不匹配，则设置 PE=1

  接收溢出错误 OE

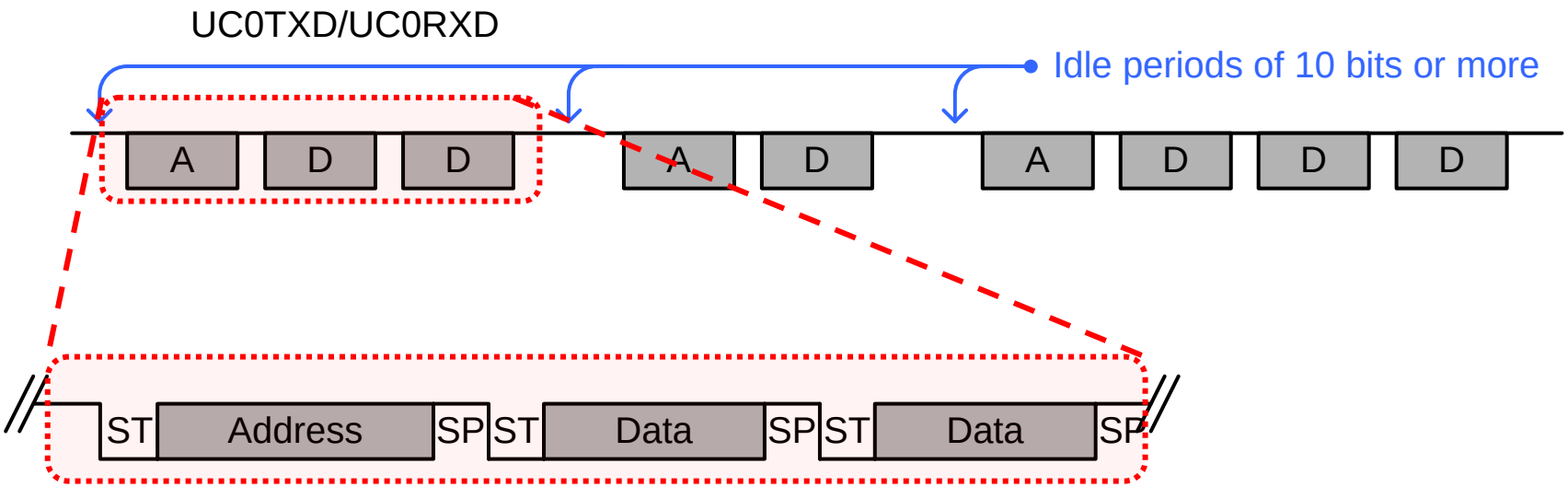
  如 RXBUF 被覆盖，则设置 OE =1

  中断条件 BRK

  如接收帧中所有位为 0，则设置 BRK=1，

  如对 BRKIE 与 RXIFG 进行设置=1 ，则 BRK=1
- **UART** 经编程后，仅向 **RXBUF** 传输无错误的字符

UART 闲置线路多处理器



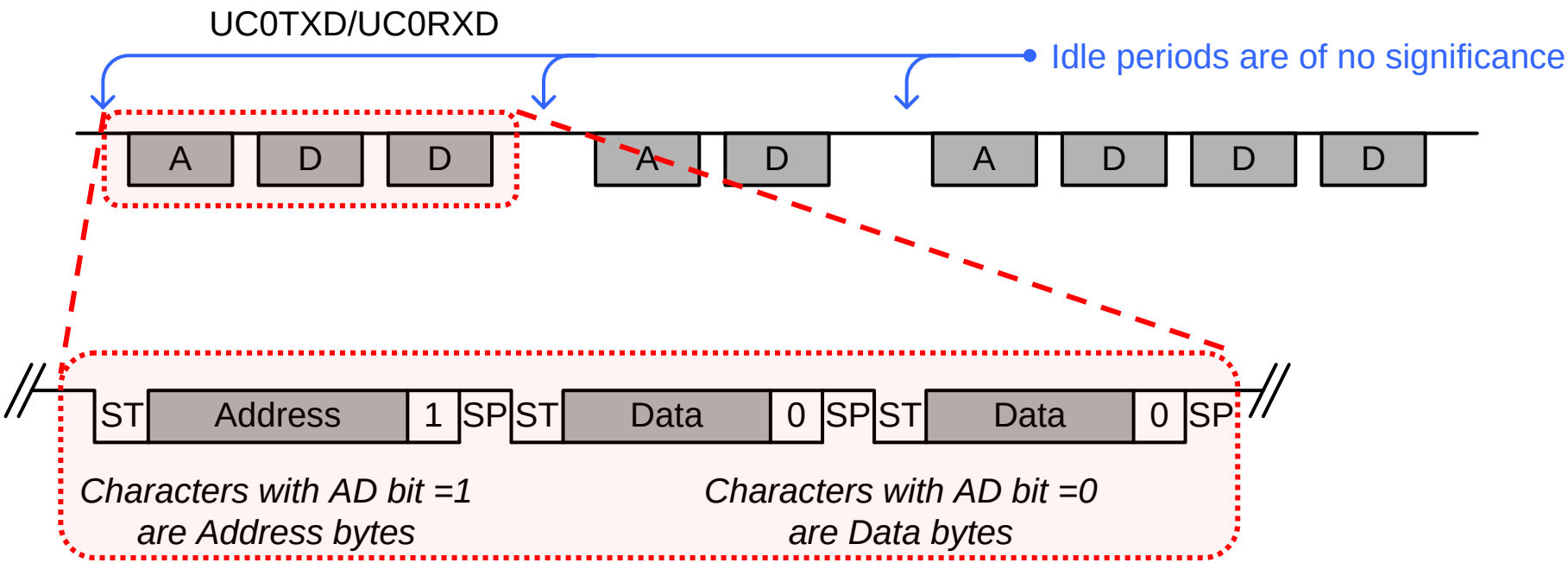
- 闲置线路多处理器协议

  如停止位后出现连续 10 个周期的标志，则检测到闲置。

  闲置后的第一个字符是地址

  可对 UART进行编程，仅接收地址字符

UART 地址位多处理器

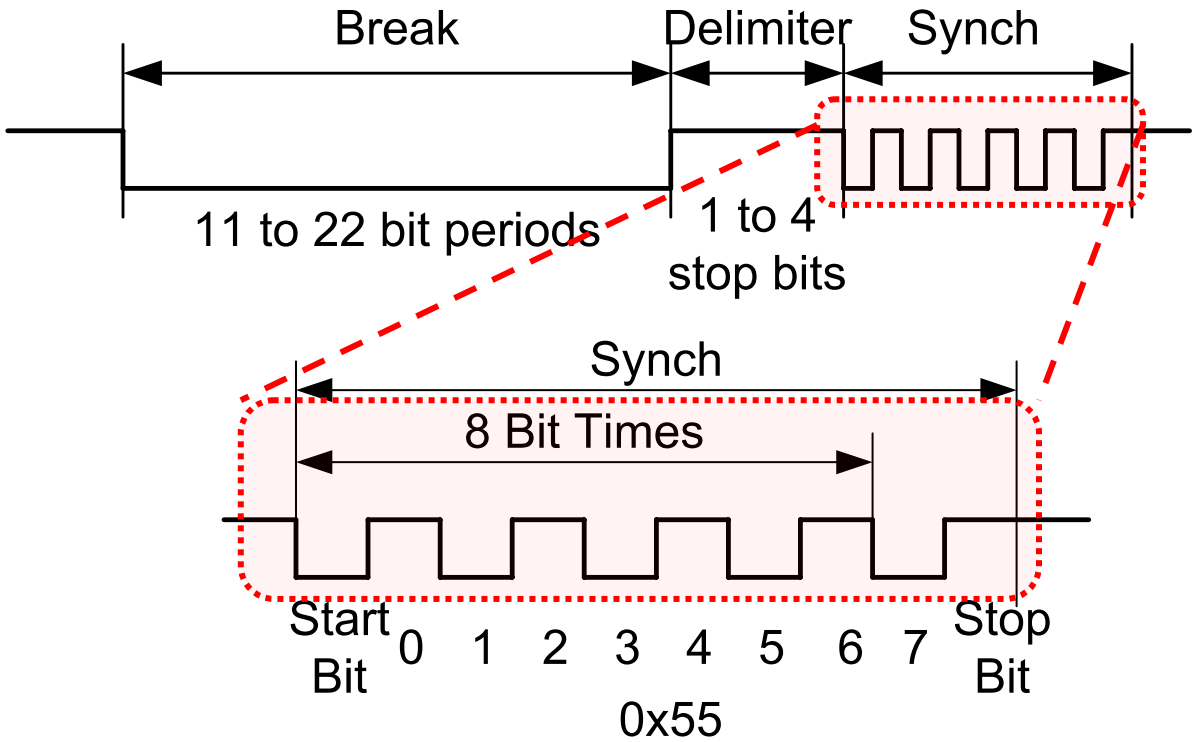


- 地址位多处理器协议

  接收字符中的额外位状态标记为地址字符

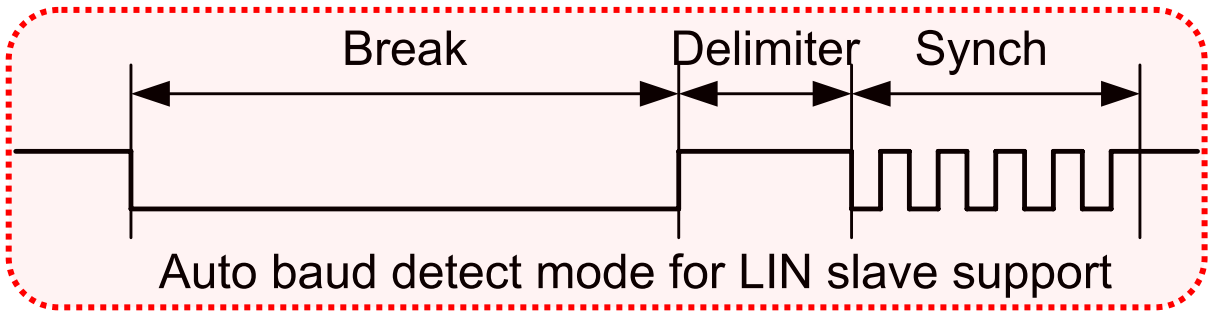
  可对 UART 进行编程，仅接收地址字符

UART 自动波特率检测



- 自动波特率检测协议
  - 波特率是由有效的 SYNC 计算出来的
  - 自动波特率值存储在 BR1、BR0 和调制器位中
  - 硬件BREAK 超时检测
  - 可编程的DELIMITER时间

UART LIN 支持



- 汽车本地互连网络
- 要求 **UART** 自动波特率检测
- **LIN** 模式 **UART = 8 bit**、**LSB** 优先、无奇偶位、**1** 停止位
- **LIN** 设备驱动程序通过软件实施
- 需要外部 **LIN** 总线驱动
- 支持 **LIN** 从模式

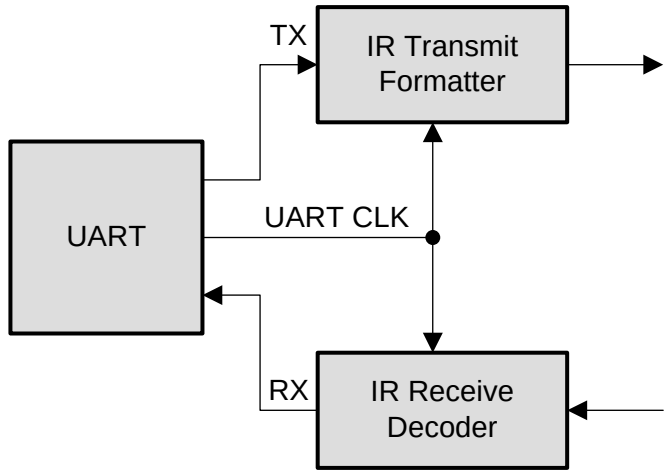
  根据接收到的 LIN break和synch 场‘0x55’ 进行break synch检测和自动波特率测量

支持 LIN 主模式

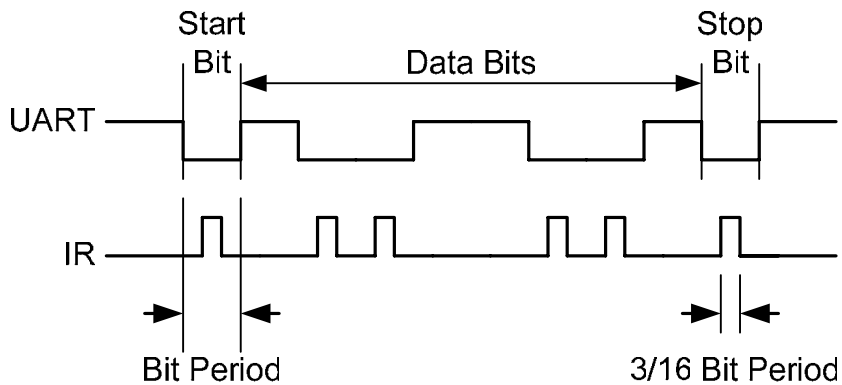
- 支持**LIN** 主模式
  自动break-synch生成
- 软件序列传输一个 **LIN** 主模式要求的中断同步字段
  1. 选择自动波特率检测模式且 UCTXBRK = 1
  2. DELIMx 指定break delimiter的长度，默认为 1 位周期
  3. 检查 TXBUF 是否准备就绪，并向 TXBUF 写入 LIN synch ‘0x55h’
- 传输**13** 位间隔场后，然后是**break delimiter**和**synch**场
  UCTXBRK 在同步载入 TX 移位寄存器后自动重设
  写入 TXBUF 的数据正常传输

UART IrDA 支持

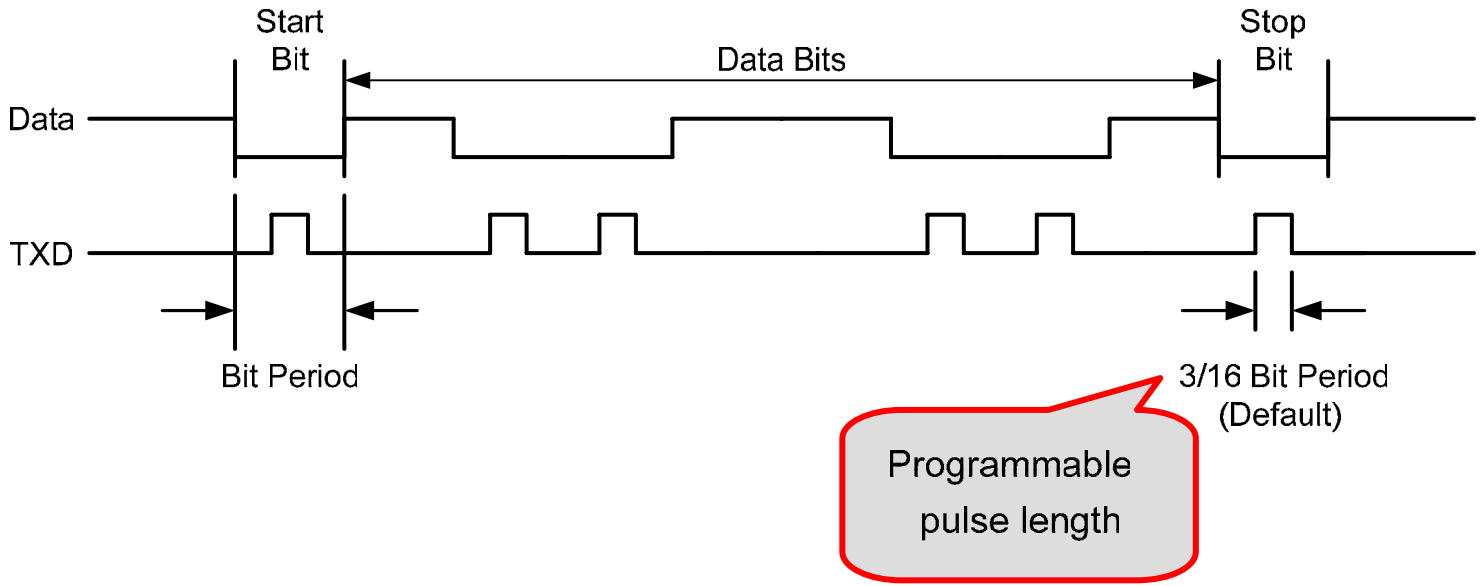
- 集成 **IrDA** 编码器和解码器
- 直接连接至外部 **IR** 模拟电路
- **32KHz** 频率可产生 **9.6kbps**
- 接受脉冲过滤的数字滤波级
- 支持 **IrDA** 标准 **3/16** 位周期脉冲
- **IrDA** 协议栈由软件实施



UART frame vs IR frame

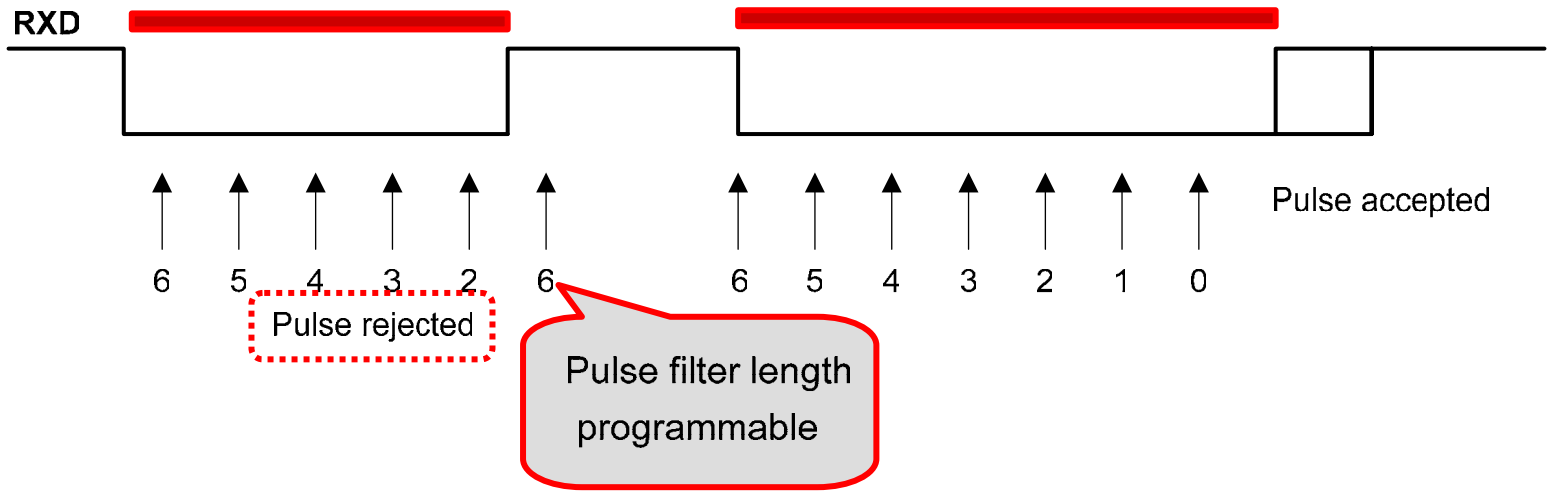


IrDA 编码器



- 可编程的发射机脉冲长度
- 过采样波特率发生器可选择 **IrDA** 标准 **3/16** 位长度

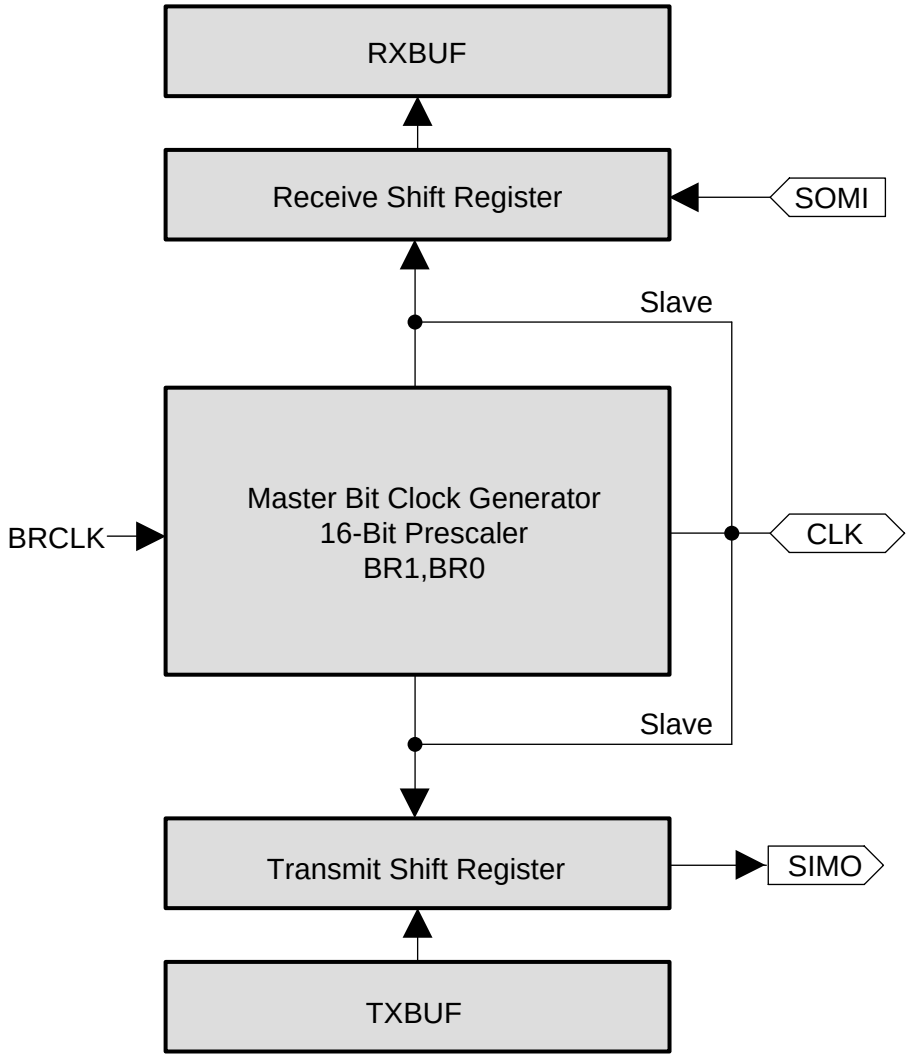
IrDA 解码器



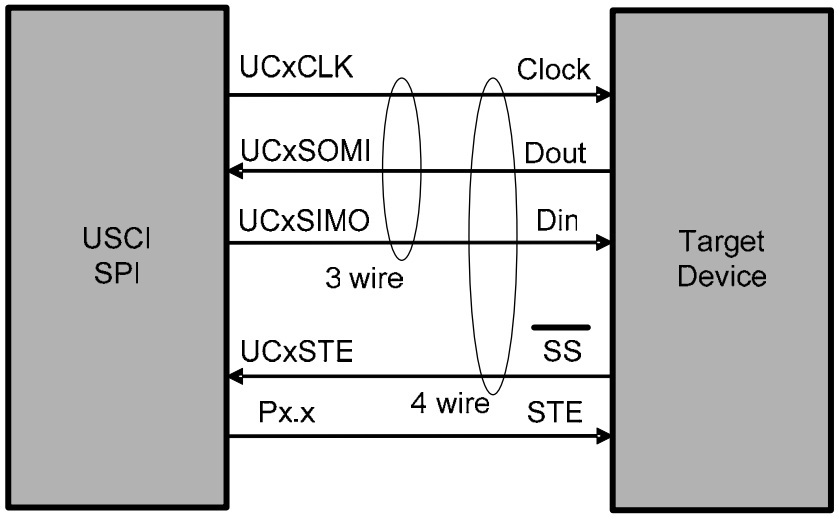
- 可编程的高或低脉冲检测能支持两种类型的 **IrDA** 收发器
- 可编程的接收脉冲长度滤波器，增强了抗干扰的能力
- 上例演示了检测低脉冲的情况

USCI A & USCI B SPI 模式

- 灵活的接口
  - 3 或 4 引脚 SPI
  - 7 或 8 位数据长度
  - 主或从
  - LSB 或 MSB 优先
- 软件可配置的时钟相位和极性
- 可编程的 **SPI** 主时钟
- 双缓存 **TX/RX**
- 中断驱动 **TX/RX**
- 可启用 **DMA**
- **LPMx** 操作

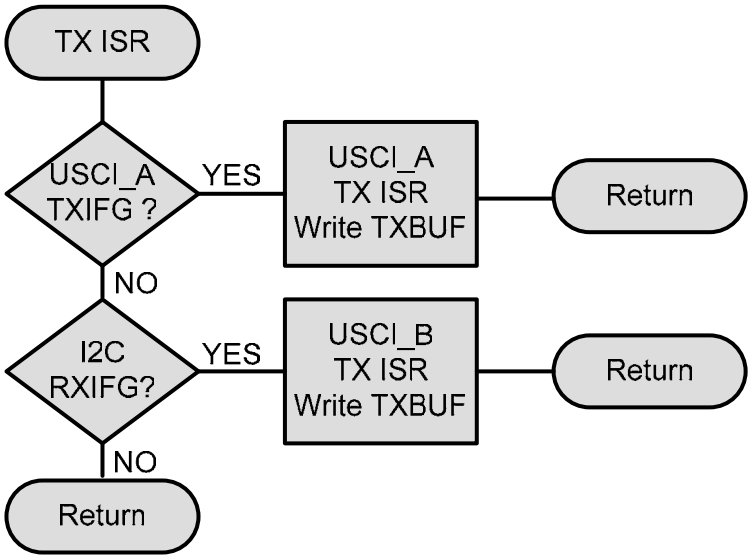


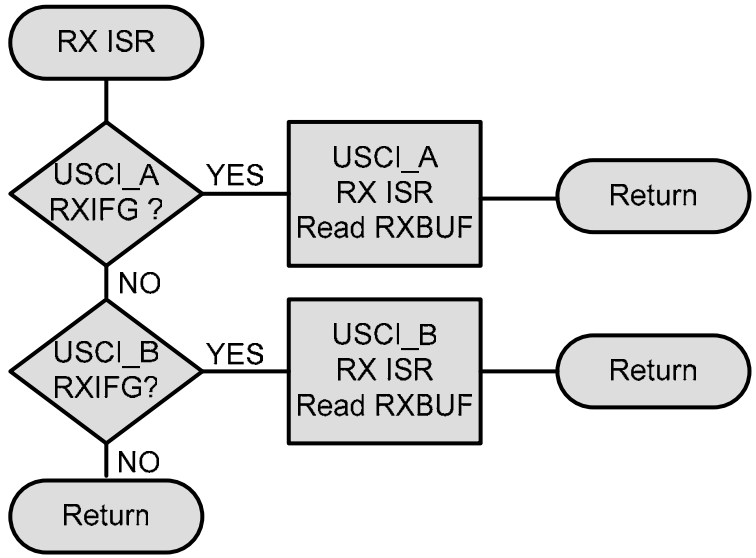
SPI 连接



- **3** 线模式支持主从模式
- **4** 线主模式**STE** 防止与其他主设备发生冲突
- **4** 线从设备中，**TX** 及 **RX** 用 **STE** 进行外部控制

USCI SPI 中断处理

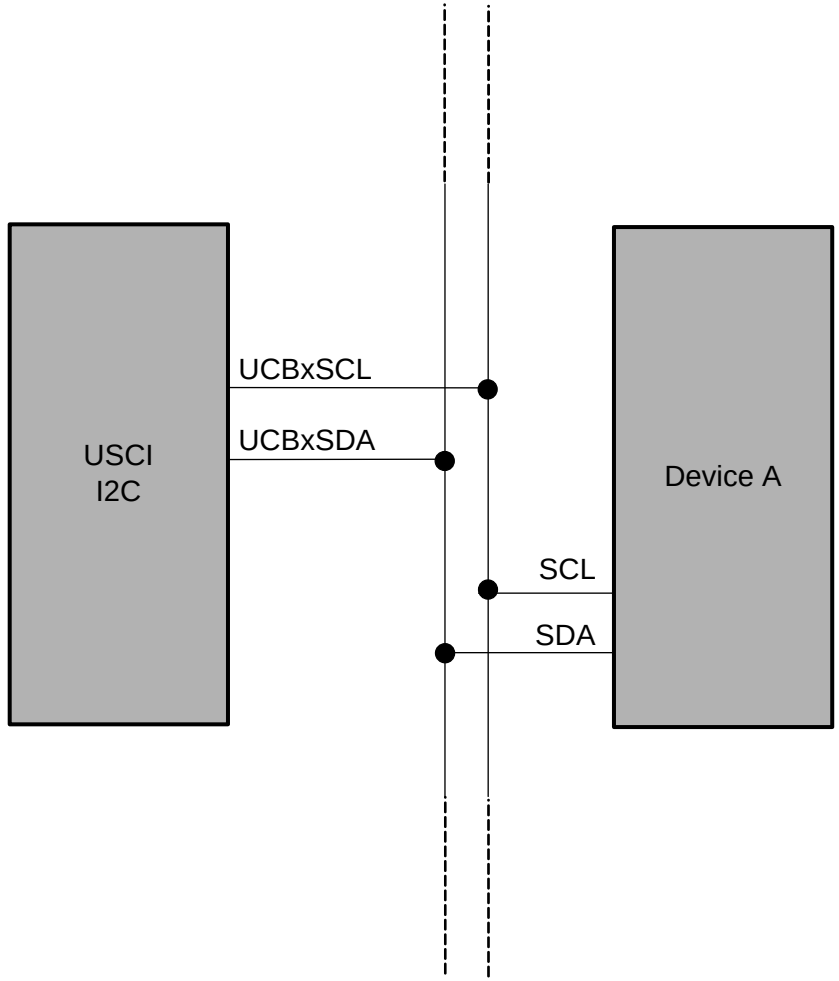




- **TX & RX ISR** 建议流程
- **USCI_A & USCI_B** 共享 **TX** 向量和 **RX** 向量
- 软件检测 **ISR** 是否响应正确的事件

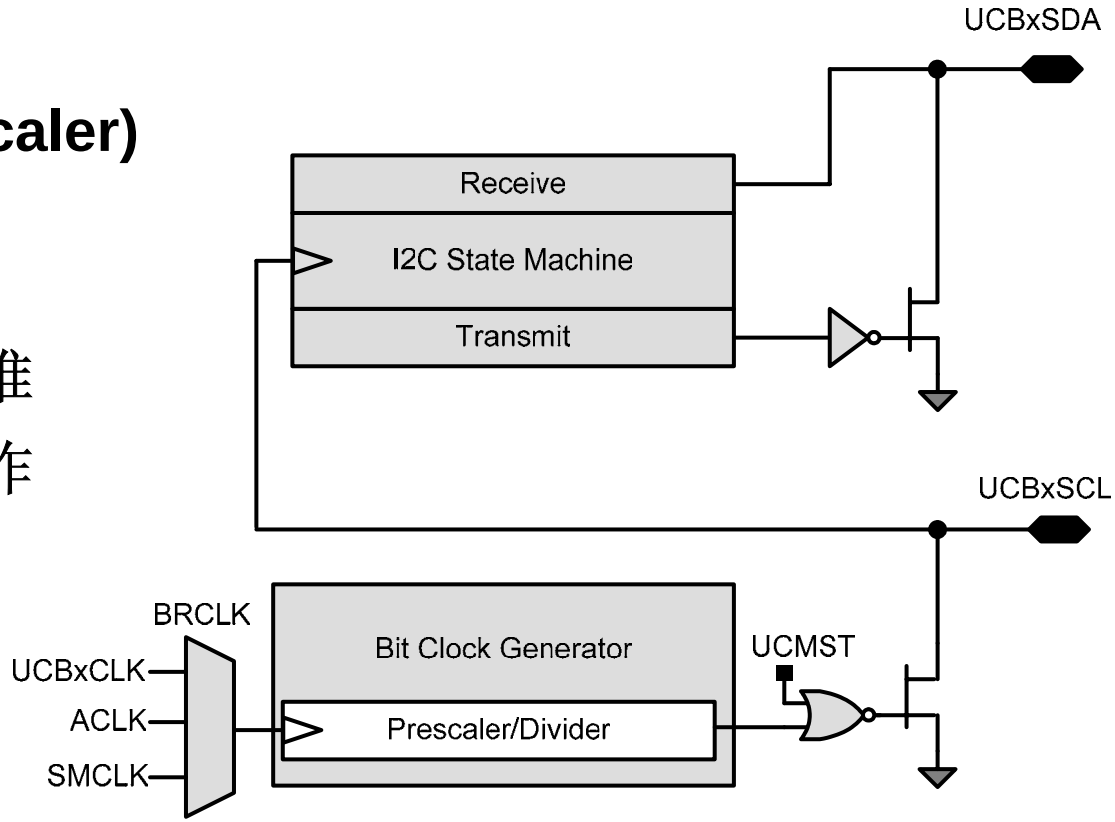
USCI B I²C 模式

- 超低功耗
  - LPMx 从接收器 START 检测
  - LPM4 中的从操作
- **Philips 规范 2.1 版**
  - 7 和 10 位地址
  - 支持多个主设备
  - 从模式
  - 高达 400kbps
- 中断驱动
  - 无应答
  - 判优丢失
  - 启动 / 停止条件
  - TX/RX



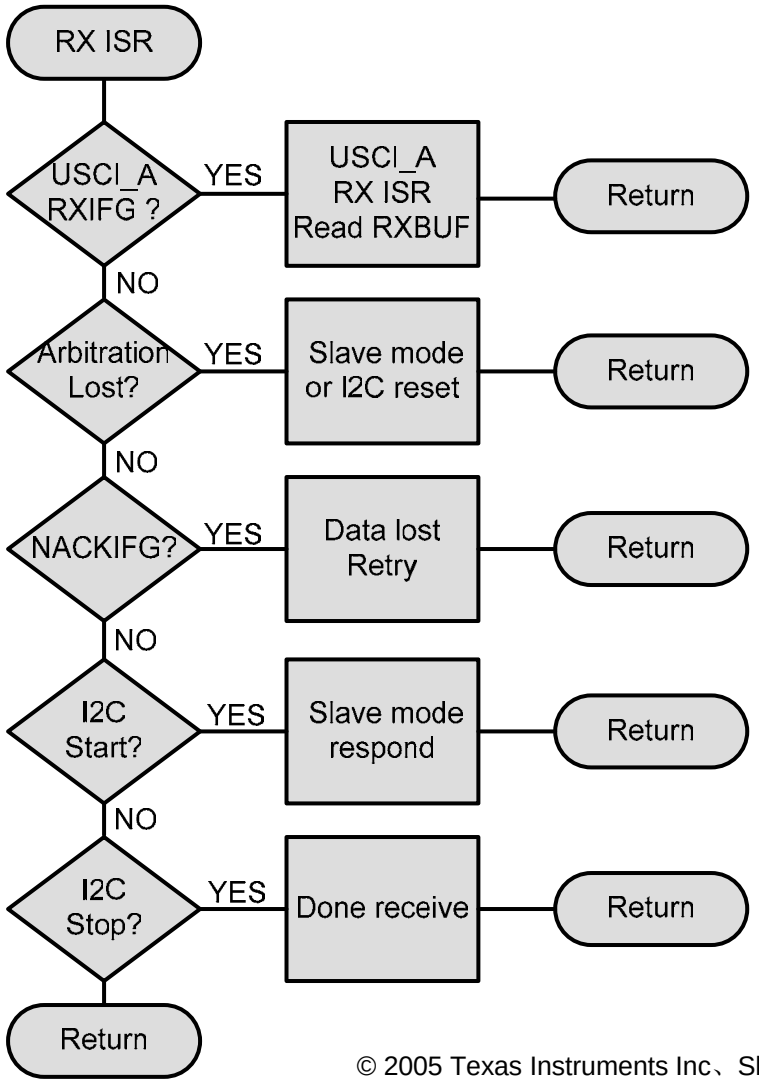
I²C 方块示意图

- 集成前置分频器 **(pre-scaler)** 的比特时钟发生器
- 灵活的时钟选择
- 开漏输出，符合 **I²C** 标准
- 从**LPMx** 返回后自动工作



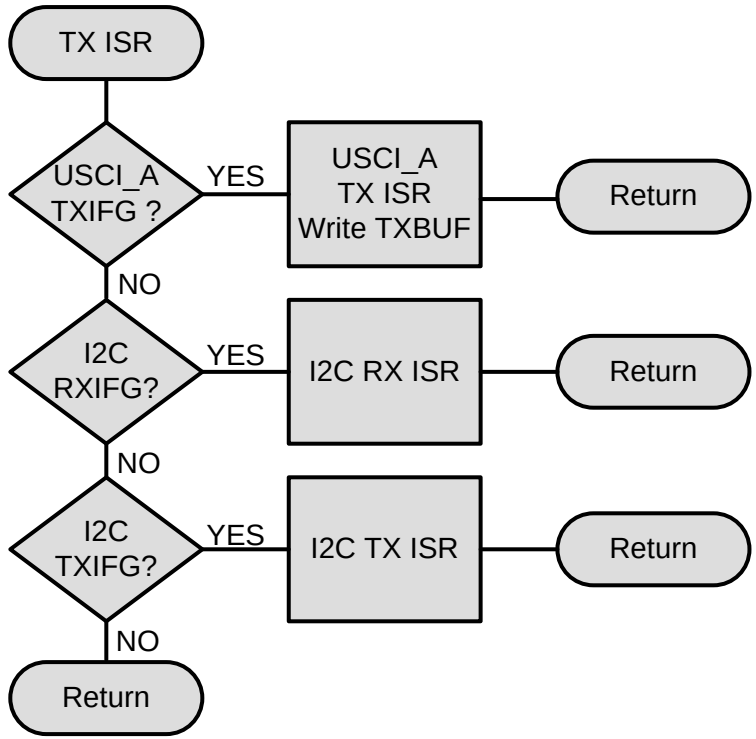
I²C 状态的中断处理

- I²C工作状态的辨别标志
  - ALIFG　仲裁丢失标志
  - NACKIFG　未接收到应答的标志
  - STTIFG　从模式启动条件的标志
  - STPIFG　从模式停止条件的标志



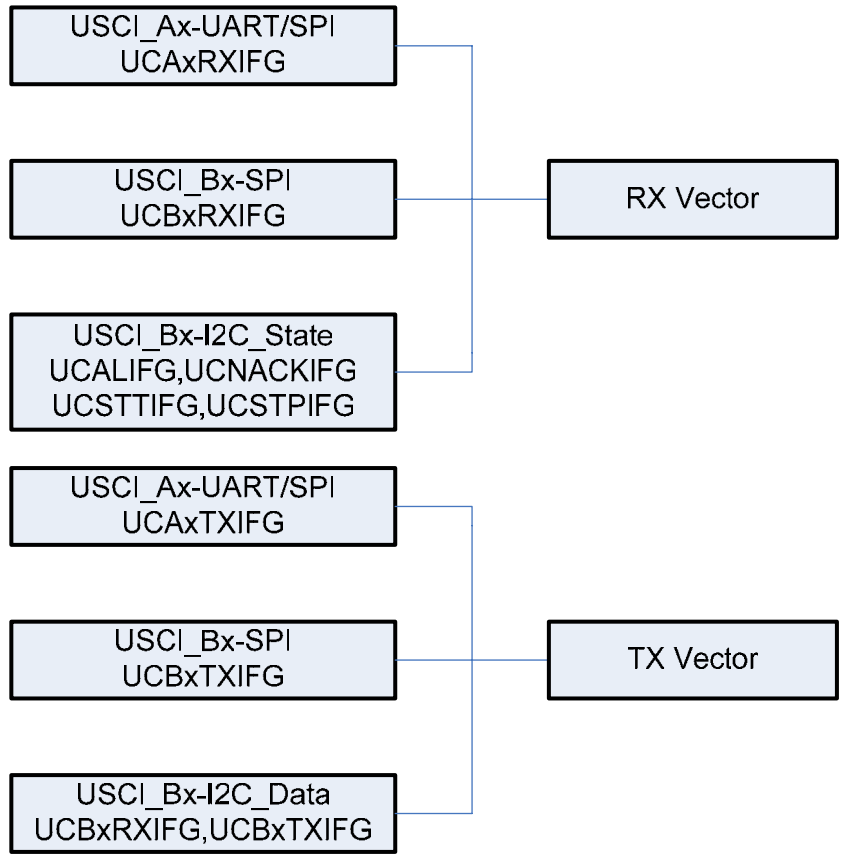
I²C 数据中断处理

- **I2C传输标记**
  UCBxTXBUF 为空时 UCBxTXIFG置位
- **I2C 接收标记**
  UCBxRXBUF 接收到字符UCBxRXIFG置位



USCI 中断

- **USCI_A & USCI_B 共享 TX & RX 中断矢量**
- **USCI_A flags**
  - UART – TXIFG、RXIFG
  - SPI – TXIFG、RXIFG
- **USCI_B 标志**
  - SPI – TXIFG、RXIFG
  - I2C 状态 – ALIFG、NACKIFG、STTIFG、STPIFG
  - I2C 数据 – TXIFG、RXIFG



常见的串行总线选项

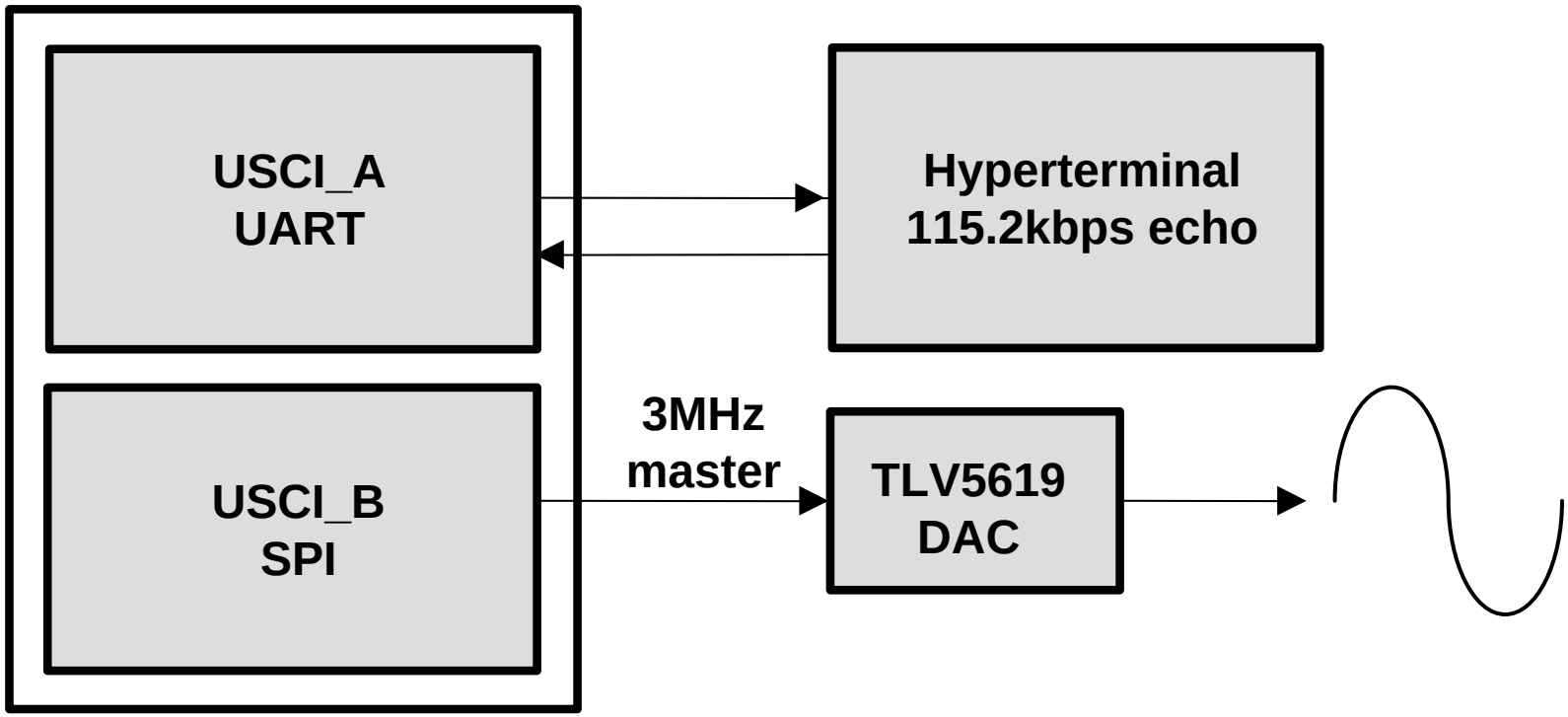
- 异步
  - UART 半双工或全双工
  - 最大波特率达 BRCLK/3
  - 是否要求无线？考虑到 IrDA
  - 网络化异步？考虑到 LIN
- 高速通信
  - SPI 模式
  - 根据所选择的设备系列，可实施 4 或 8MHz 主模式
  - 根据所选择的设备系列，可实施 8 或 16MHz 从模式
  - 自然全双工增加吞吐量
- I2C 环境
  - 符合 Philips v2.1 标准
  - 最大支持 400kHz 的高速
  - 主 / 从操作
  - 支持多主设备环境

未来的 USCI 设备

Device family	Pins	USCI	DMA
MSP430FG461x	100	A, B	Yes
MSP430F22x4	38/40	A, B	No
MSP430F261x	80	C, B	Yes
MSP430F23x/24x	64	C, B	No

- **USCI_A = UART、IrDA、LIN 支持、SPI**
- **USCI_B = SPI、I2C**
- **USCI_C = UART、SPI**

演示：同步工作的 USCI_A/USCI_B



演示：USCI 配置

```
SetupUSCIA
          mov.b #UCSWRST,&UC0CTL1     ; state machine reset
          bis.b #UCSSEL1,&UC0CTL1     ; SMCLK
          mov.b #0x34,&UC0BR0         ; 115k at 6029312Hz
          mov.b #0x00,&UC0BR1         ;
          mov.b #0x20,&UC0MCTL        ; modulation values
          bic.b #UCSWRST,&UC0CTL1     ; state machine start
SetupUSCIB
          mov.b #SWRST,&UC1CTL1       ; state machine reset
          mov.b #UCCKPL+UCCKPH+UCMSB+UCMST+UCSYNC,&UC1CTL0
                                      ; SPI-M,MSB,3-pin,CK-
          mov.b #UCSSEL1,&UC1CTL1     ; SMCLK
          mov.b #0x02,&UC1BR0         ; SMCLK/2
          mov.b #0x00,&UC1BR1         ;
          bic.b #SWRST,&UC1CTL1       ; state machine start
          bis.b #UC0RXIE+UC1TXIE,&IE2; enable RX/TX int.
```

演示：USCI 中断子例程

```
USCIARX_ISR// Echo back RXed character, TXBUF ready?
TX0         bit.b #UC0TXIFG,&IFG2       ; TXBUF ready?
            jz    TX0                   ; Jump if not ready
            mov.b &UC0RXBUF,&UC0TXBUF   ; RX -> TX
            reti                        ;

USCIBTX_ISR// External DAC Sine table update
            bic.b #0x01,&P3OUT          ; FS reset
L1          bit.b #UC1TXIFG,&IFG2       ; USCI Transmit ready?
            jnc   L1                    ;
            mov.b Sin_tab(R6),&UC1TXBUF; load from sine table
            incd.w R6                   ;
            and.w #0x3E,R6              ; only 32 words

L2          bit.b   #UCBUSY,&UC1STAT    ; USCI Transmit done?
            jnz     L2                  ;
            bis.b   #0x01,&P3OUT        ; FS set
            reti                        ;
```

为什么 USCI 不能发挥作用？

```
SetupUSCIA  mov.b #UCSWRST,&UC0CTL1     ; module reset
            bis.b #UCSSEL1,&UC0CTL1     ; SMCLK
            mov.b #0x09,&UC0BR0         ; 115k at 1048576Hz
            mov.b #0x00,&UC0BR1         ;
            mov.b #0x08,&UC0MCTL        ; modulation values
            bic.b #UCSWRST, &UC0CTL1    ; state m/c start
            bis.b #UC0RXIE, &IE2        ; enable RXinterrupt
```

- 正确的 **USCI_A** 配置步骤
  1. 置位 **SWRST**
  2. 配置 **USCI** 寄存器
  3. 清除 **SWRST**
  4. 需要时启用中断

USCI：以一变应千变

- **USCI** 是 **MSP430**新标准的串行接口
- 两个相互独立的块可同时工作
- 所有模式都能从任意 **LPMx** 进行操作
- **USCI** 由中断驱动
- **USCI** 可启用 **DMA**
- **USCI_A** 支持 **UART**、**IrDA**、自动波特 **LinBUS**
- **UART** 集成了波特率发生器和调制器，可支持部分比特率
- **USCI_B** 支持 **SPI**、**I2C**