SPC11x8_SPD11x8_SPC2168 SIO_UART-V2.2.1 使用指南

概述

通用异步收发器（UART）能够灵活地与外部设备进行全双工数据交换，常用于短距离、低速的串行通信中。本文介绍了使用 SIO 模块实现的 UART 的特性。

注：本文档主要以 SPC1168 为例进行介绍。

目录

图片列表

未找到图形项目表。

表格列表

版本历史

版本	日期	作者	状态	变更
A/0	2023 年 10 月 9 日	Haiyang Wang	Outdated	首次发布。
A/1	2023 年 11 月 6 日	Haiyang Wang	Outdated	优化中断使用方法，满中断只触发一次，不需要在软件上清除中断 pending。
A/2	2023 年 11 月 26 日	Haiyang Wang	Released	修复全双工通讯模式下，会导致接收出错的问题

术语或缩写

术语或缩写	描述
MCU	Microcontroller Unit，微控制器单元
UART	Universal Asynchronous Receiver/Transmitter，通用异步收发传输器
SIO	Spintrol IO，可编程 Spintrol 输入输出
PLA	Program Logic Array，可编程逻辑阵列

1 UART 特性

SPC11X8/SPD11X8/SPC2168 使用 SIO 实现一个 UART 单元，具有以下特点：

- 8 位数据，1 停止位，无校验位
- 可配置波特率最高支持 2.5Mbps；
- 发送 FIFO：深度 6，位宽 8；
- 接收 FIFO：深度 6，位宽 8；
- 中断支持：
  - 接收 FIFO 满中断；
  - 接收帧超时中断：RXD 电平保持未变化超过 1.5 个字节；
- 标志支持：
  - RX FIFO 数据丢失标志

在 SIO_UARTV2 的驱动库中，已经有下列驱动函数可供调动，可大大方便用户使用和理解。表 1-1 和表 1-2 中 SIOx 表示 SIO 模块编号，取值为 SIO0。

表 1-1：SIO_UARTV2 宏定义列表

宏名	功能及参数说明
SIO_UARTV2_Enable(SIOx)	使能 SIO_UARTV2
SIO_UARTV2_Disable(SIOx)	禁用 SIO_UARTV2
SIO_UARTV2_IsEnable(SIOx)	SIO_UARTV2 是否使能
SIO_UARTV2_WriteFIFO(SIOx, u8Data)	向 SIO_UARTV2 发送 FIFO 写数据 u8Data：要写的数据
SIO_UARTV2_ReadFIFO(SIOx)	从 SIO_UARTV2 接收 FIFO 读数据
SIO_UARTV2_GetRxFIFOLossDataFlag(SIOx)	获取 SIO_UARTV2 接收 FIFO 是否发生丢失数据
SIO_UARTV2_GetStatus(SIOx, u32Query)	获取 SIO_UARTV2 FIFO 状态 u32Query：要查询的状态，支持以下状态： SIO_UARTV2_STATUS_RX0FIFO_OVERFLOW SIO_UARTV2_STATUS_RX0FIFO_EMPTY SIO_UARTV2_STATUS_RX0FIFO_FULL SIO_UARTV2_STATUS_TX0FIFO_EMPTY SIO_UARTV2_STATUS_TX0FIFO_FULL

表 1-2：SIO_UARTV2 函数定义列表

函数名	功能及参数说明
void SIO_UARTV2_SetBaudRate(SIO_REGS* SIOx, uint32_t u32BaudRate)	设置 SIO_UARTV2 的波特率 SIOx：SIO 模块基址 u32Baudrate: 通讯速率
void SIO_UARTV2_SetRxTimeoutThreshold(SIO_REGS* SIOx, uint32_t u32BaudRate)	根据 SIO_UARTV2 设置的波特率设置超时时间 SIOx：SIO 模块基址 u32Baudrate: 通讯速率
void SIO_UARTV2_Init (SIO_REGS* SIOx, uint32_t u32BaudRate)	初始化 SIO_UARTV2，编程 PLA 和 uCore SIOx：SIO 模块基址 u32Baudrate: 通讯速率
void SIO_UARTV2_WriteBytes (SIO_REGS* SIOx, uint8_t u8Data);	向 SIO_UARTV2 TX FIFO 写数据 SIOx：SIO 模块基址 U8Data: 要发送的数据
void SIO_UARTV2_RxFIFOFullCallback(void)	接收满中断函数
void SIO_UARTV2_RxTimeoutCallback(void)	接收超时中断函数

2 SIO_UARTV2 使用注意事项

使用 SIO_UARTV2 时，以下几点需要特别注意，本节以 SPC1168 为例进行说明。

- SIO_UARTV2 超时寄存器是 16 位，最大值为 65535。当启用此功能时，为了满足超时时间等于 1.5 帧（15 位），最小波特率（以 SIO 时钟频率 100MHz 为例）：

$$f_{baud} = 65535 * 10 / 15 = 43690 \text{ bps}$$

即使用时设置的波特率要大于 43.69Kbps 才可以正常使用超时功能。若设置的波特率小于 43.69Kbps 时，超时时间等于 $65535T_{baud}$；

- CPU 接收到 FIFO 满中断后，必须在一帧 UART 数据传输时间内（$10T_{baud}$）将 FIFO 数据取走，否则会导致下一帧数据丢失。

3 SIO_UARTV2 实例

3.1 SIO_UARTV2 初始化

本示例中，描述了 SIO 如何初始化为 SIO_UARTV2。

Example Code

```
/* Configuration CPU clock */
CLOCK_InitWithRCO(CLOCK_HCLK_200MHZ);

/* Configure SIO clock */
CLOCK_EnableModule(SIO0_MODULE);
CLOCK_SetModuleDiv(SIO0_MODULE, 2);

/* Configure SIO as SIO_UARTV2 */
u32BaudRate = 3000000;
SIO_UARTV2_Init (SIOx, u32BaudRate);

/* Enable SIO_UARTV2 RXFIFO full interrupt */
NVIC_EnableIRQ(SIO0A_IRQn);
/* Enable SIO_UARTV2 RX timeout interrupt */
NVIC_EnableIRQ(SIO0B_IRQn);
```

3.2 SIO_UARTV2 发送数据

下面的示例中，描述了 SIO_UARTV2 如何使用查询 TXFIFO 状态的方式发送数据。

Example Code

```
for (i = 0; i < sizeof(u8Data)/sizeof(u8Data[0]); i++)
{
    SIO_UARTV2_WriteBytes (SIOx, u8Data[i]);
}
```

3.3 SIO_UARTV2 接收数据

下面的示例中，描述了 SIO_UARTV2 如何使用 RXFIFO 满中断的方式接收数据。

Example Code

```
void SIO_UARTV2_RxFIFOFullCallback()
{
    while (SIO_UARTV2_GetStatus(SIOx, SIO_UARTV2_STATUS_RX0FIFO_EMPTY) !=
SIO_UARTV2_STATUS_RX0FIFO_EMPTY)
    {
        u8RxData[u32RxCnt] = SIO_UARTV2_ReadFIFO(SIOx);
        u32RxCnt++;
    }
}
```

3.4 SIO_UARTV2 超时中断处理

下面的示例中，描述了 SIO_UARTV2 如何使用超时中断接收剩余数据。

Example Code

```
void SIO_UARTV2_RxTimeoutCallback()
{
    while (SIO_UARTV2_GetStatus(SIOx, SIO_UARTV2_STATUS_RX0FIFO_EMPTY) !=
SIO_UARTV2_STATUS_RX0FIFO_EMPTY)
    {
        u8RxData[u32RxCnt] = SIO_UARTV2_ReadFIFO(SIOx);
        u32RxCnt++;
    }
}
```