SPC11x8_SPD11x8_SPC2168 SIO_UART-v2.3.0 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
C/0	2024-06-12	Haiyang Wang	Released	首次发布。

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 基于 SIO 的 UARTV2 模块概述

1.1 SIO 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；
- 8 个采样点，从第 3 点开始采样 3 次，取 2 次相同的电平为采样值；
- 支持波特率切换；
- 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

表 1-1：管脚分配

SIO 模块名	SIO 管脚编号	GPIO 管脚编号	功能
SIO0	15	GPIO19	TXD
SIO0	17	GPIO21	RXD

注意事项：

- SIO 时钟频率 Fsio <= 100MHz；
- 不支持修改 SIO 配置文件里的 SIO 管脚数组来实现管脚重新分配。需要不同的配置文件，请联系 Spintrol 工程师。

1.2 SIO UART 信号描述

表 1-2：SIO UART 信号方向描述

信号名	方向	描述
SIO_UART_TXD	输出	SIO UART 串行数据输出
SIO_UART_RXD	输入	SIO UART 串行数据输入

2 SIO UARTV2 使用方式

Spintrol 提供了相应的软件库来简化该系统的使用。

2.1 配置 SIO 时钟

用户可以通过 SIOCLKCTL 寄存器来配置 SIO 时钟，包括时钟的使能和分频比。具体可参见《SPC11X8 Technical Reference Manual》的第 3 章。当 SIO 被配置用作 SIO_UARTV2 时，所允许的 SIO 模块时钟最高频率可达 100MHz（对于部分型号芯片，最高频率会达不到 100MHz，具体参考 TRM）。

2.2 SIO_UARTV2 初始化

提供了 SIO_UARTV2_Init()函数，下载固件到 SIO 模块，将其初始化为 SIO_UARTV2,同时配置 PINMUX，将引脚切换至 SIO 通道。用户在代码中直接调用该函数即可。

SIO_UARTV2 初始化代码如下：

示例代码 2-1：SIO_UARTV2 初始化

```
/* 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);
```

2.3 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]);
}
```

2.4 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++;
    }
}
```

2.5 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++;
    }
}
```

2.6 SIO_UARTV2 切换波特率

下面的示例中，描述了 SIO_UARTV2 如何切换波特率。

Example Code

```
/* Change baud rate */
u32BaudRate = 38400;
SIO_UARTV2_ReConfigBuadrate (SIOx, u32BaudRate);
```

3 API 函数

表 3-1：API 函数列表

API 函数名	说明
SIO_UARTV2_GetStatus(SIOx, u32Query)	根据 mask 查询并获取 FIFO 状态
SIO_UARTV2_ReInitial(SIOx)	初始化 UARTV2 的波特率，需在模块不发送和接收帧时使用
SIO_UARTV2_WriteFIFO(SIOx, u8Data)	向 SIO UARTV2 FIFO 中写数据
SIO_UARTV2_ReadFIFO(SIOx)	从 SIO UARTV2 FIFO 中读数据
SIO_UARTV2_SetRxTimeoutCount(SIOx, u16Data) WRITE_REG((SIOx)->SIOM[2].all, u16Data)	设置接收超时中断时间，需在模块不发送和接收帧时使用
SIO_UARTV2_GetRxTimeoutCount(SIOx) READ_REG((SIOx)->SIOM[2].all)	获取接收超时中断时间，需在模块不发送和接收帧时使用
SIO_UARTV2_SetBaudRateCount(SIOx, u16Data) WRITE_REG((SIOx)->SIOM[3].all, u16Data)	设置波特率定时时间，需在模块不发送和接收帧时使用
SIO_UARTV2_GetBaudRateCount(SIOx) READ_REG((SIOx)->SIOM[3].all)	获取波特率定时时间，需在模块不发送和接收帧时使用
SIO_UARTV2_GetRxFIFOLossDataFlag(SIOx)	获取 SIO UARTV2 是否发生数据丢失标志
SIO_UARTV2_ClearRxFIFOLossDataFlag(SIOx)	清除 SIO UARTV2 数据丢失标志
SIO_UARTV2_GetuCoreInitialFinishFlag(SIOx)	获取 SIO UARTV2 内微核是否初始化完成标志
SIO_UARTV2_ClearuCoreInitialFinishFlag(SIOx)	清除 SIO UARTV2 微核初始化完成标志
SIO_UARTV2_SetBaudRate(SIO_REGS* SIOx, uint32_t u32BaudRate)	设置 SIO UARTV2 波特率
SIO_UARTV2_SetRxTimeoutThreshold(SIO_REGS* SIOx, uint32_t u32BaudRate)	设置 SIO UARTV2 接收超时时间
SIO_UARTV2_Init (SIO_REGS* SIOx, uint32_t u32BaudRate)	SIO UARTV2 初始化
SIO_UARTV2_WriteBytes (SIO_REGS* SIOx, uint8_t u8Data)	向 SIO UARTV2 写入字节并发送
SIO_UARTV2_RxFIFOFullCallback(void)	接收 FIFO 满中断函数
SIO_UARTV2_RxTimeoutCallback(void)	接收 FIFO 超时中断函数
SIO_UARTV2_ReConfigBuadrate (SIO_REGS* SIOx, uint32_t u32BaudRate)	重新配置 SIO UARTV2 波特率

4 代码示例

以 SIO0 为例子，参考 demos 目录下例程。

5 寄存器

5.1 SIO_UART 寄存器表

表 5-1：SIO_UART 模块基地址

外设模块	基地址
SIO0_UART	0x4000B000

表 5-2：SIOUART Register Map

Register	Offset	Description	Reset Value
FIFO	0x0000	UART FIFO Register	0x00000000
TIMEOUT	0x0008	Frame Timeout Register	0x00000000
BAUDCNT	0x000C	Baud Rate Divisor Register	0x00000000
CTL	0x0010	UART Control Register	0x00000000
FLAG	0x0014	UART Flag Register	0x00000000
SIOFLAG	0x0078	SIO Flag Register	0x00000000
STATUS	0x007C	UART FIFO Status Register	0x00000000

表 5-3：UART FIFO Register (FIFO) Layout

FIFO (UART FIFO Register) Access: SIOUART->FIFO.all Offset: 0x0000 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
FIFO							
7	6	5	4	3	2	1	0
FIFO							

表 5-4：UART FIFO Register (FIFO) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	FIFO	RW	0x0	Read data from SIO or write data to SIO

表 5-5：Frame Timeout Register (TIMEOUT) Layout

TIMEOUT (Frame Timeout Register) Access: SIOUART->TIMEOUT.all			Offset: 0x0008		Default: 0x00000000		
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
TOCNT							
7	6	5	4	3	2	1	0
TOCNT							

表 5-6：Frame Timeout Register (TIMEOUT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	TOCNT	RW	0x0	Set frame timeout count as (TIMEOUT+1) SIO clock cycles, it can only set before the SIO initial done

表 5-7：Baud Rate Divisor Register (BAUDCNT) Layout

BAUDCNT (Baud Rate Divisor Register) Access: SIOUART->BAUDCNT.all			Offset: 0x000C		Default: 0x00000000		
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
DIVISOR							
7	6	5	4	3	2	1	0
DIVISOR							

表 5-8：Baud Rate Divisor Register (BAUDCNT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	DIVISOR	RW	0x0	BAUDCNT is (SIO module frequency / baud rate) - 1, it can only set before the SIO initial done

表 5-9： UART Control Register (CTL) Layout

CTL (UART Control Register) Access: SIOUART->CTL.all Offset: 0x0010 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							REINIT

表 5-10： UART Control Register (CTL) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	REINIT	RW	0x0	No effect 0: No effect 1: Reinitial UART

表 5-11： UART Flag Register (FLAG) Layout

FLAG (UART Flag Register) Access: SIOUART->FLAG.all Offset: 0x0014 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							RX_OVF

表 5-12： UART Flag Register (FLAG) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	RX_OVF	RW	0x0	Receive FIFO overflow 0: UART0 RX FIFO has overflow 1: UART0 RX FIFO has not overflow

表 5-13： SIO Flag Register (SIOFLAG) Layout

SIOFLAG (SIO Flag Register) Access: SIOUART->SIOFLAG.all			Offset: 0x0078		Default: 0x00000000		
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							INIT_DONE

表 5-14： SIO Flag Register (SIOFLAG) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	INIT_DONE	RW	0x0	SIO uCore initializing or clear by software 0: uCore initializing or clear by software 1: Notify the CPU that uCore is initial finished

表 5-15： UART FIFO Status Register (STATUS) Layout

STATUS (UART FIFO Status Register) Access: SIOUART->STATUS.all			Offset: 0x007C		Default: 0x00000000		
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							SIOINIT
7	6	5	4	3	2	1	0
TF0FULL	TF0EMPTY	RESERVED		RF0FULL	RF0EMPTY	RF0OVRUN	SIOENABLE

表 5-16：UART FIFO Status Register (STATUS) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	SIOINIT	RW	0x0	SIO initial 0: 1: SIO initial
7	TF0FULL	RO	0x0	In UART0, the number of data written in FIFO less than 6 0: 1: UART0 Transmit FIFO full
6	TF0EMPTY	RO	0x0	In UART0, There is at least one or more data in FIFO 0: 1: UART0 Transmit FIFO Empty
5:4	RESERVED_5_4	RO	0x0	Reserved.
3	RF0FULL	RO	0x0	In UART1, the number of data received in FIFO less than 6 0: 1: UART0 Receive FIFO Full
2	RF0EMPTY	RO	0x0	In UART0,At least one or more data in received FIFO 0: 1: UART0 Receive FIFO Empty
1	RF0OVRUN	RO	0x0	In UART0,No data lost 1: UART0 Receive FIFO overrun
0	SIOENABLE	RW	0x0	SIO enable option 0: 1: Enable SIO