

概述

SIO DUART 是在 SIO 上实现的两个通用异步收发器 (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。UART 通过可编程的波特率产生器提供了多种波特率。

目录

1 基于 SIO 的 DUART 模块概述	7
1.1 SIO DUART 特性.....	7
1.2 SIO DUART 信号描述.....	8
2 SIO DUART 使用方式	9
2.1 配置 SIO 时钟.....	9
2.2 配置 SIO 为 SIO_DUART	9
3 API 函数.....	11
4 代码示例	12
5 寄存器	13
5.1 SIO_DUART 寄存器表.....	13

图片列表

未找到图形项目表。

SPIN TROL

表格列表

表 1-1: 管脚分配	7
表 1-2: SIO DUART 信号方向描述	8
表 3-1: API 函数列表	11
表 5-1: SIO_DUART 模块基地址	13

SPIN TROL

版本历史

版本	日期	作者	状态	变更
C/0	2024-04-10	Haiyangwang	Released	引脚更换： SIO0_TXD0 : SIO0_PIN13 -> GPIO39 SIO0_RXD0 : SIO0_PIN12 -> GPIO38

SPIN
TROL

术语或缩写

术语或缩写	描述
MCU	Microcontroller Unit, 微控制器单元
PLA	ProgrammableLogicArray, 可编程逻辑阵列
SIO	Smart Input Output, SIO

SPIN TROL

1 基于 SIO 的 DUART 模块概述

1.1 SIO DUART 特性

基于 SIO 的 DUART 单元有以下特点：

- 两个 UART 同时工作最高支持 1Mbps；
- 两个 UART 收发各自有宽度为 8，其中 SPC1198/SPC2168 的 FIFO 深度为 8，其它产品 SPC11x8 的 FIFO 深度为 6；
- 数据格式：8 比特数据位，无校验/奇校验/偶校验（需要联系 Spintrol 工程师选择配置模式，目前只支持 3 选 1），1 停止位；
- 接收中断模式：接收一个字节中断；
- 可检测错误：奇偶校验位错误（选择带有校验位时），接收 FIFO 数据丢失错误（RXFIFO 满后若再收到一个数据则触发此中断）；
- 每个比特采样 8 次，去掉第一次采样，采样 5 次有 3 次即认为采样成功；
- 关闭模块后自动恢复到初始启动时状态；
- 两个 UART 只能工作在相同波特率下。

表 1-1: 管脚分配

SIO 模块名	SIO 管脚编号	GPIO 管脚编号	功能
SIO0	-	-	-
SIO0	-	-	-
SIO1	13	39	TXD0
SIO1	12	38	RXD0
SIO2	-	-	-
SIO2	-	-	-

注意事项：

- SIO 时钟频率 $F_{sio} \leq 100\text{MHz}$ ；
- 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 DUART 信号描述

表 1-2: SIO DUART 信号方向描述

信号名	方向	描述
SIO_UART0_TXD	输出	SIO UART0 串行数据输出
SIO_UART0_RXD	输入	SIO UART0 串行数据输入
SIO_UART1_TXD	输出	SIO UART1 串行数据输出
SIO_UART1_RXD	输入	SIO UART1 串行数据输入

2 SIO DUART 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《SPC21X8 Technical Reference Manual》的第 3 章。当 SIO 被配置用作 SIO_DUART 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 配置 SIO 为 SIO_DUART

提供了 SIO_DUART_Init()函数，下载固件到 SIO 模块，将其初始化为 SIO_DUART,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_DUART 初始化代码如下：

示例代码 2-1: SIO_DUART 初始化

```
/* Clock Init */
CLOCK_InitWithRCO(CLOCK_HCLK_200MHZ);

/* Configure SIO Clock, maximum clock is 100MHz */
CLOCK_SetModuleDiv(SIO0_MODULE, 2);
CLOCK_EnableModule(SIO0_MODULE);

/* SIO_DUART Init */
SIO_DUART_Init(SIO0, 256000);

/* Enable SIO_DUART SIOA IRQ */
NVIC_EnableIRQ(SIO0A_IRQn);
NVIC_EnableIRQ(SIO0B_IRQn);
```

示例代码 2-2: SIO_DUART 发送数据

```
/* UART0 Send data */
SIO_DUART_WriteBytes(SIO0, SIO_DUART_UART0, au8UART0TxData, u8Size);

/* UART1 Send data */
SIO_DUART_WriteBytes(SIO0, SIO_DUART_UART1, au8UART1TxData, u8Size);
```

示例代码 2-3: SIO_DUART 数据接收中断

```
/* UART Receive data, use interrupt*/
void SIO_DUART_UART0RxOneByteCallback ()
{
    while (SIO_DUART_GetStatus(SIOx, SIO_DUART_STATUS_RX0FIFO_EMPTY) !=
SIO_DUART_STATUS_RX0FIFO_EMPTY)
    {
        au8UART0RxData[u32UART0RxDataCnt++] = SIO_DUART_ReadUART0FIFO(SIO0);
    }
}

void SIO_DUART_UART1RxOneByteCallback ()
{
    while (SIO_DUART_GetStatus(SIOx, SIO_DUART_STATUS_RX1FIFO_EMPTY) !=
SIO_DUART_STATUS_RX1FIFO_EMPTY)
    {
        au8UART1RxData[u32UART1RxDataCnt++] = SIO_DUART_ReadUART1FIFO(SIO0);
    }
}
```

注意: SIO DUART TXFIFO 为空不代表 TXD 发送完成, 需要等待一个字节的时间才代表发送完成。

3 API 函数

表 3-1: API 函数列表

API 函数名	说明
SIO_DUART_GetStatus(SIOx, u32Query)	根据 mask 查询并获取 FIFO 状态
SIO_DUART_GetUART0RxFIFODataLossFlag(SIOx)	获取 UART0 FIFO 是否丢数据
SIO_DUART_GetUART1RxFIFODataLossFlag(SIOx)	获取 UART1 FIFO 是否丢数据
SIO_DUART_ClearUART0RxFIFODataLossFlag(SIOx)	清除 UART0 接收数据丢失标志
SIO_DUART_ClearUART1RxFIFODataLossFlag(SIOx)	清除 UART1 接收数据丢失标志
SIO_DUART_WriteUART0FIFO(SIOx, u8Data)	向 UART0 FIFO 写数据
SIO_DUART_WriteUART1FIFO(SIOx, u8Data)	向 UART1 FIFO 写数据
SIO_DUART_ReadUART0FIFO(SIOx)	读 UART0 FIFO 的数据
SIO_DUART_ReadUART1FIFO(SIOx)	读 UART1 FIFO 的数据
SIO_DUART_Enable(SIOx)	开启 DUART
SIO_DUART_Disable(SIOx)	关闭 DUART 并复位
void SIO_Init(SIO_REGS* SIOx)	下载 PLD 比特流和微码机器码到存储体
void SIO_PinInit(SIO_REGS* SIOx)	映射 SIO PIN 到芯片的 GPIO
void SIO_DUART_GetIntFlag(SIO_REGS* SIOx, uint32_t u32Query)	获取 SIO DUART 中断标志
void SIO_DUART_Init(SIO_REGS* SIOx)	下载 PLA 实现 DUART 功能
void SIO_Enable(SIO_REGS* SIOx)	SIO 使能
void SIO_DUART_Enable(SIO_REGS* SIOx)	SIO UART 使能
void SIO_DUART_SetBaudRate(SIO_REGS* SIOx)	设置 UART 波特率
void SIO_DUART_WriteBytes(SIO_REGS* SIOx, SIO_DUART_InstanceEnum eUARTx, uint8_t *pu8DataBuf, uint32_t u32Size)	SIO DUART 写数据
void SIO_DUART_ReadBytes(SIO_REGS* SIOx, SIO_DUART_InstanceEnum eUARTx, uint8_t *pu8DataBuf, uint32_t u32Size)	SIO DUART 读数据

4 代码示例

以 SIO0 为例子，参考 demos 目录下例程。

SPIN TROL

5 寄存器

5.1 SIO_DUART 寄存器表

表 5-1: SIO_DUART 模块基地址

外设模块	基地址
SIO_DUART	0x4000B000

SIO_DUART 的寄存器和功能定义请参考响应驱动函数中的描述，在此不在赘述。