

文档版本	V1.0
发布日期	20210610

# APT32F102 SIO 应用指南

**APT**CHIP



## 目录

1 概述 .....	1
2. 适用的硬件.....	1
3. 应用方案代码说明 .....	1
3.1 SIO 配置.....	1
3.2 SIO 发送.....	3
3.3 SIO 接收.....	5
4. 程序下载和运行 .....	7

# 1 概述

本文介绍了在APT32F102中使用SIO的应用范例。

# 2. 适用的硬件

该例程使用于 APT32F102x 系列学习板

# 3. 应用方案代码说明

## 3.1 SIO 配置

基于 APT32F102 完整的库文件系统，可以对 SIO 进行配置。

- **硬件配置：**

SIO 模块是一个串行输入输出的控制器，可以模拟多种串行通信协议，支持双向数据传输。由 D0, D1, DL, DH 四个对象组合编码，可以用于 MCU 外围硬件接口不够，但又需要和其它设备通信或者器件自定协议的情况。

SIO 具有数据发送缓存寄存器和接收缓存器，数据移动方式可配置。

- **注意：**

DH, DL 发送序列长度由 SIO\_TXCR1[10:8]决定,数据从 HSQ[0]到 HSQ[2]顺序依次移位发送

- **模块框图：**

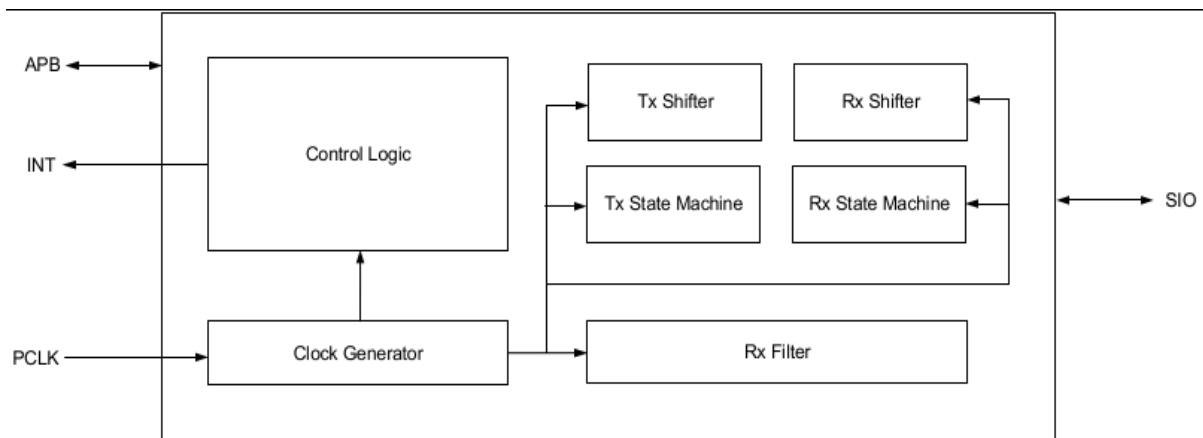


图 3.1.1 功能框图

● 软件配置:

可在 apt32f102\_initial.c 文件中 SIO\_CONFIG 进行初始化的配置

```

/*****/
//SIO Functions
//EntryParameter:NONE
//ReturnValue:NONE
/*****/

void SIO_CONFIG(void)
{
    SIO_DeInit();
    SIO_IO_Init(SIO_PA03);
    SIO_TX_Init(SIOCLK_EN,20);

    SIO_TX_Configure(SIO_IDLE_HIGH,SIO_TX_MSB,15,8,0,0,SIO_OBH_3BIT,SIO_OBL_3BIT,3,1);

    //SIO_TXBUF_Set(TX_DH,TX_DL,TX_DH,TX_DL,TX_DH,TX_DL,TX_DH,TX_DL,TX_D1,TX_D0,TX_D1,TX_D0,TX_D1,TX_D0,TX_D1,TX_D0);

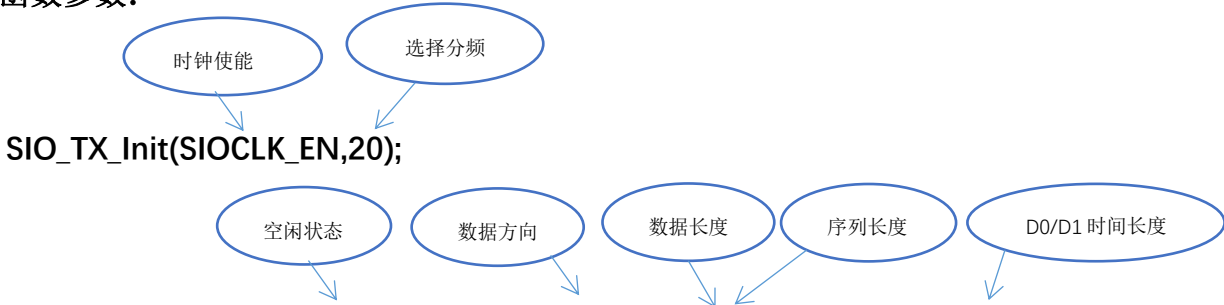
    //SIO_ConfigInterrupt_CMD(ENABLE,SIO_TXDNE);

    //SIO_INT_ENABLE();
}
    
```

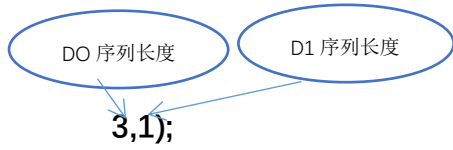
● 代码说明:

- SIO\_DeInit(); ----用于恢复初始值
- SIO\_IO\_Init(); ----用于配置 GPIO 为 SIO 功能
- SIO\_TX\_Init(); ----用于配置时钟和模式。
- SIO\_TX\_Configure();----用于配置 SIO 发送波形
- SIO\_TXBUF\_Set(); ----用于配置 SIO 发送数据缓存
- SIO\_ConfigInterrupt\_CMD();----用于配置中断
- SIO\_INT\_ENABLE(); ----用于使能中断

● 函数参数:



SIO\_TX\_Configure(SIO\_IDLE\_HIGH,SIO\_TX\_MSB,15,8,0,0,SIO\_OBH\_3BIT,SIO\_OBL\_3BIT,



### 3.2 SIO 发送

选择内部时钟 48MHZ 作为系统主频，配置 PA0.3 为 SIO 功能，发送时钟 TXCLK 1us，发送一组序列长度为 9bits 的数据。

● 编程要点：

1. 先配置 SIO 的模式及时钟
2. D0/D1 时间长度，就是固定 bit0 和 bit1 的时间。
3. DL/ DH 的序列长度
4. DL/ DH 序列定义，发送数据的顺序 BIT0-BIT1-BIT3

```

/*****/
//SIO Functions
//EntryParameter:NONE
//Return Value:NONE
/*****/
void SIO_CONFIG(void)
{
    SIO_DeInit();
    SIO_IO_Init(SIO_PA03);
    SIO_TX_Init(SIOCLK_EN,47);
    SIO_TX_Configure(SIO_IDLE_HIGH,SIO_TX_MSB,15,8,0,0,SIO_OBH_3BIT,SIO_OBL_3BIT,3,1);
    SIO_TXBUF_Set(TX_DH,TX_DL,TX_DH,TX_DL,TX_DH,TX_DL,TX_DH,TX_DL,TX_D1,TX_D0,TX_D1,TX_D0,TX_D1,TX_D0,TX_D1,TX_D0);
    //SIO_ConfigInterrupt_CMD(ENABLE,SIO_TXDNE);
    //SIO_INT_ENABLE();
}
void APT32F102_init(void)
{
    //-----/
    //Peripheral clock enable and disable
    //EntryParameter:NONE
    
```

```

//ReturnValue:NONE
//-----/
    SYSCON->PCER0=0xFFFFFFFF;           //PCLK Enable 0x00410071
    SYSCON->PCER1=0xFFFFFFFF;           //PCLK Enable
    while(!(SYSCON->PCSR0&0x1));         //Wait PCLK enabled
//-----/

//ISOSC/IMOSC/EMOSC/SYSCCLK/IWDT/LVD/EM_CMFAIL/EM_CMRCV/CMD_ERR OSC stable interrupt
//EntryParameter:NONE
//ReturnValue:NONE
//-----/
    SYSCON_CONFIG();                   //syscon initial
    CK_CPU_EnAllNormalIrq();           //enable all IRQ
//-----/
//Other IP config
//-----/
    GPIO_Init(GPIOA0,4,0);
    GPIO_Set_Value(GPIOA0,4,1);
    SIO_CONFIG();                       //SIO initial
}
/*****/
//main
/*****/
int main(void)
{
    APT32F102_init();
    while(1)
    {
        //SYSCON_IWDCNT_Reload();

        if(SIO0->RISR&0X01)             //RISR bit1==1 时表示发送完成
        {
            SIO0->ICR=0X01;
            SIO_TXBUF_Set(TX_D1,TX_D0,TX_DL,TX_D1,TX_D1,TX_D0,TX_DH,TX_DH,TX_DH,TX_D0,TX_D1,TX_D0,TX_D1,TX_D0,TX_D1,TX_D0);
            GPIO_Set_Value(GPIOA0,4,0);
        }

        if(SIO0->RISR&0X01)             //RISR bit1==1 时表示发送完成
        {
            SIO0->ICR=0X01;
            SIO_TXBUF_Set(TX_D0,TX_D1,TX_D1,TX_D0,TX_DH,TX_DL,TX_DH,TX_DL,TX_DH,TX_D0,TX_D1,TX_D0,TX_D1,TX_D0,TX_D1,TX_D0);
            GPIO_Set_Value(GPIOA0,4,1);
        }
    }
}
}

```

● 波形输出:

PA0.4 输出低时的波形

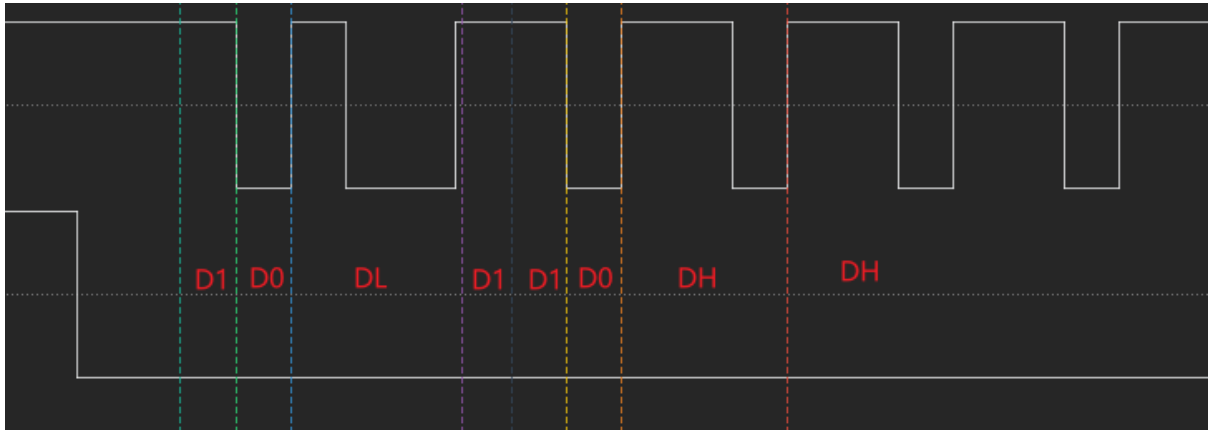


图 3.2.1 SIO 发送波形

由上图得知:

D0: 全 0 序列, 时间 T=1us

D1: 全 1 序列, 时间 T=1us

DL:序列长度设置为 SIO\_OBL\_3BIT, 序列数据设置为 1, 表示将序列数据 0b001, 发送顺序低位开始, 第 1 个 bit 为高,其余两个 bit 为低.

3.3 SIO 接收

```

/*****/
//SIO Functions
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
void SIO_CONFIG(void)
{
    SIO_DeInit();
    SIO_IO_Init(SIO_PA03);
    //RX
    SIO_RX_Init(SIOCLK_EN,SIO_TXDEB_1CYCLE,47);
    SIO_RX_Configure0(SIO_RX_FALL,SIO_RX_DEB,0,1,2,SIO_RX_ALIGNDIS,SIO_RX_MSB,SIO_RMODE0,8,15,47);
    SIO_RX_Configure1(SIO_BREAKDIS,SIO_BREAKLVL_HIGH,0,SIO_TORSTEN,1);
    //
    SIO_ConfigInterrupt_CMD(ENABLE,SIO_RXDNE);
}
    
```

```

    SIO_INT_ENABLE();
}

void APT32F102_init(void)
{
    //-----/
    //Peripheral clock enable and disable
    //EntryParameter:NONE
    //ReturnValue:NONE
    //-----/

    SYSCON->PCER0=0xFFFFFFFF;           //PCLK Enable 0x00410071
    SYSCON->PCER1=0xFFFFFFFF;           //PCLK Enable
    while(!(SYSCON->PCSR0&0x1));         //Wait PCLK enabled
    //-----/

    //ISOSC/IMOSC/EMOSC/SYSCLK/IWDT/LVD/EM_CMFAIL/EM_CMRCV/CMD_ERR OSC stable interrupt
    //EntryParameter:NONE
    //ReturnValue:NONE
    //-----/

    SYSCON_CONFIG();                     //syscon initial
    CK_CPU_EnAllNormalIrq();             //enable all IRQ
    //-----/

    //Other IP config
    //-----/

    GPIO_Init(GPIOA0,4,0);
    GPIO_Set_Value(GPIOA0,4,1);
    SIO_CONFIG();                          //SIO initial
}
/*****/

//main
/*****/

int main(void)
{
    APT32F102_init();
    while(1)
    {
        //SYSCON_IWDCNT_Reload();
    }
}
/*****/

//SIO Interrupt
//EntryParameter:NONE
//ReturnValue:NONE
/*****/

void SIO0IntHandler(void)
{

```



```
if(SIO0->MISR&0X02) //RXDNE
{
    SIO0->ICR=0X02;
    if(R_SIORX_count>=1)
    {
        R_SIORX_buf[R_SIORX_count]=SIO0->RXBUF&0xff000000; //8bit
        nop;
        R_SIORX_count=0;
    }
}
else if(SIO0->MISR&0X08) //RXBUFFULL
{
    SIO0->ICR=0X08;
    if(R_SIORX_count<1)
    {
        R_SIORX_buf[R_SIORX_count]=SIO0->RXBUF; //32bit
        R_SIORX_count++;
    }
}
else if(SIO0->MISR&0X10) //BREAK
{
    SIO0->ICR=0X10;
}
else if(SIO0->MISR&0X20) //TIMEOUT
{
    SIO0->ICR=0X20;
}
}
```

## 4. 程序下载和运行

1. 将目标板与仿真器连接，分别为 VDD SCLK SWIO GND
2. 程序编译后仿真运行
3. 验证波形数据如图 3.2.1 所示