

文档版本	V1.0
发布日期	20191108

APT32F172 SPI 应用开发指南



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 主机配置	1
3.1 从机配置	3
4. 程序下载和运行	5
5. 改版历史	6

1 概述

本文介绍了在APT32F172中使用SPI的应用范例。

2. 适用的硬件

该例程使用于 APT32F172 开发板 APT-DB172

3. 应用方案代码说明

基于 APT32F172 完整的库文件系统，可以很方便的对 SPI 进行配置。

3.1 主机配置

软件配置：

开启内部主频 20MHz, 并作为系统时钟。

功能管脚配置：

SPI_NSS=PA0.7; SPI_SCK=PA0.8; SPI_MISO=PA0.9; SPI_MOSI=PA0.10

通讯速度：

FSSPCLKOUT=20M/10=1M

发送接收模式：

发送数据大小为 8BIT; SCK 工作时为高电平; SCK 第一个时钟沿捕捉; 串行正常输出

```
/*  
//SPI MASTER Initial  
//EntryParameter:NONE  
//ReturnValue:NONE  
*/  
void SPI_MASTER_CONFIG(void)  
{  
    SPI_DelInit();  
    SPI_NSS_IO_Init(1);  
}
```

```

SPI_Master_Init(SPI_G1,SPI_DATA_SIZE_8BIT,SPI_SPO_0,SPI_SPH_0,SPI_LBM_0,SPI_RXIFLSEL_1_8,0,10);
//选择SPI IO group1; 发送数据大小为8BIT; SCK 工作时为高电平; SCK 第一个时钟沿捕捉; 串行正常
//输出; 接收占用 1/8 FIFO 中断触发断点; FSSPCLKOUT=20M/10=1M
}

/*****/

//APT32F172_init
//EntryParameter:NONE
//ReturnValue:NONE
/*****/

void APT32F172_init(void)
{
    SYSCON_WDT_CMD(DISABLE);           //关闭 WDT

    SYSCON->PCER0=0xFFFFFFFF;          //使能 IP
    SYSCON->PCER1=0xFFFFFFFF;          //使能 IP
    while(!(SYSCON->PCSR0&0x1));        //判断 IP 是否使能

    SYSCON_Int_Enable();               //使能 SYSCON 中断向量
    SYSCON->IECR=ISOSC_ST|IMOSC_ST|EMOSC_ST|SYSCLK_ST;
    //使能 ISOSC 时钟稳定中断,使能 IMOSC 时钟稳定中断,使能 EMOSC 时钟稳定中断

    CK_CPU_EnAllNormalIrq();           //打开全局中断
    SYSCON_CONFIG();                   //syscon 参数 初始化
    GPIO_CONFIG();                      //GPIO 初始化
    SPI_MASTER_CONFIG ();               //SPI 主机初始化
}

U32_TR_j2c_read_data;

/*****/
    
```

```
//main
/*****/

int main(void)
{
    APT32F172_init();

    while(1)
    {
        SYSCON_IWDGNT_Reload();           //清狗
        SPI_WRITE_BYTE (0x01);           //发送 0x01
    }
}
```

3.1 从机配置

开启内部主频 20MHz, 并作为系统时钟。

功能管脚配置:

SPI_NSS=PA0.7; SPI_SCK=PA0.8; SPI_MISO=PA0.9; SPI_MOSI=PA0.10

通讯速度:

FSSPCLKOUT=20M/10=1M

使能接收中断

```
/*****/

//SPI SLAVE Initial

//EntryParameter:NONE

//ReturnValue:NONE

/*****/

void SPI_SLAVE_CONFIG(void)
{
    SPI_DeInit();

    SPI_NSS_IO_Init(1);

    SPI_Slave_Init(SPI_G1,SPI_DATA_SIZE_8BIT,SPI_RXIFLSEL_1_8,0,12);

    //选择SPI IO group1; 发送数据大小为8BIT;接收占用1/8 FIFO 中断触发断点 ;FSSPCLKOUT=20M/2=10M
}
```

```

        SPI_ConfigInterrupt_CMD(SPI_RXIM,ENABLE);//使能 FIFO 接收中断
        SPI_Int_Enable();                //使能 SPI 中断向量
    }

/*****/

//APT32F172_init
//EntryParameter:NONE
//ReturnValue:NONE
/*****/

void APT32F172_init(void)
{
    SYSCON_WDT_CMD(DISABLE);            //关闭 WDT

    SYSCON->PCER0=0xFFFFFFFF;          //使能 IP
    SYSCON->PCER1=0xFFFFFFFF;          //使能 IP
    while(!(SYSCON->PCSR0&0x1));        //判断 IP 是否使能

    SYSCON_Int_Enable();                //使能 SYSCON 中断向量
    SYSCON->IECR=ISOSC_ST|IMOSC_ST|EMOSC_ST|SYSCLK_ST;
    //使能 ISOSC 时钟稳定中断,使能 IMOSC 时钟稳定中断,使能 EMOSC 时钟稳定中断

    CK_CPU_EnAllNormalIrq();           //打开全局中断
    SYSCON_CONFIG();                   //syscon 参数 初始化
    GPIO_CONFIG();                      //GPIO 初始化
    SPI_SLAVE_CONFIG ();                //SPI 主机初始化
}

volatile unsigned int  SPI_DATA[8];

/*****/

//SPI Interrupt
    
```

```
//EntryParameter:NONE

//ReturnValue:NONE

/*****/

void SPIIntHandler(void)

{

    if((SPI0->MISR&SPI_RXIM)==SPI_RXIM)           //接收 FIFO 中断,FIFO 占用 1/8,1/4,1/2 中断

    {

        SPI0->ICR = SPI_RXIM;

        /*SPI_DATA[0]=SPI0->DR;

        SPI_DATA[1]=SPI0->DR;

        SPI_DATA[2]=SPI0->DR;

        SPI_DATA[3]=SPI0->DR;

        SPI_DATA[4]=SPI0->DR;

        SPI_DATA[5]=SPI0->DR;

        SPI_DATA[6]=SPI0->DR;

        SPI_DATA[7]=SPI0->DR;

        nop;*/

    }

}
```

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. 连接两块目标板的 SPI 接口
3. 用逻辑分析观察波形是否正确，查看从机程序 SPI_DATA[0]~ SPI_DATA[7]是否与主机发送相符。

5. 改版历史

版本	修改日期	修改概要
V1.0	2019-11-08	初版