

|      |          |
|------|----------|
| 文档版本 | V1.0     |
| 发布日期 | 20191108 |

# APT32F172 USART 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 USART 配置 ..... | 1 |
| 4. 程序下载和运行 .....   | 3 |
| 5. 改版历史 .....      | 4 |

## 1 概述

本文介绍了在APT32F172中使用USART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 USART 进行配置。

### 3.1 USART 配置

软件配置：

开启内部主频 20MHz, 并作为系统时钟。

USART 作为普通模式。

波特率：38400 ， 字节长度 8bit ， 偶校验位 ， 1 个停止位 。

UART 脚位选择：PB0.0->RXD0, PB0.1->TXD0

将 RXD 于 TXD 短接，发送 Uart 数据 0XAA，接收数据同样为 0XAA。

```
/*  
//USART Functions  
//EntryParameter:NONE  
//Return Value:NONE  
*/  
void USART_CONFIG(void)  
{  
    USART_DeInit();           //USART 所有寄存器复位赋值  
    USART_CLK_Enable();      //USART CLK 使能  
    USART_CTRL_Config(RSTRX,ENABLE); //复位接收模块  
    USART_CTRL_Config(RSTTX,ENABLE); //复位发射模块  
    USART_IO_Init(USART_PB0); //USART 使用 PB0.0->RXD0, PB0.1->TXD0  
    USART_MODE_Config(SENDTIME0,PCLK,CHRL8,ASYNC,PAR_EVEN,NBSTOP1,CHMODE_NORMAL);  
    //重发次数为 0, CLKs=PCLK, 字节长度 8bit, 异步模式, 偶校验位, 1 个停止位, 普通模式  
    USART_Baudrate_Cal(38400,20000000,PCLK,ASYNC);  
    //波特率=38400, 主频选择 20M, PCLK 不分频, 异步模式
```

```

USART_CTRL_Config(TXEN,ENABLE);           //USART 发送使能
USART_CTRL_Config(RXEN,ENABLE);           //USART 接收使能
USART_INT_Config(RXRDY_INT,ENABLE);       //USART 接收中断使能
USART_INT_Config(TXRDY_INT,ENABLE);       //USART 发射中断使能
USART_Int_Enable();                        //USART 中断向量使能
}

/*****/
//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
    UART_CONFIG ();                         //UART 初始化
}

volatile U32_T R_Uart_RDBUF;
/*****/
//main
/*****/
int main(void)
{
    APT32F172_init();
    while(1)
    {
        SYSCON_IWDCNT_Reload();             //清狗
        USART_TxByte(0XAA);                 //发送 0xAA
        while(R_Uart_RDBUF!=0XAA);          //判断是否接收到 0XAA
    }
}
    
```

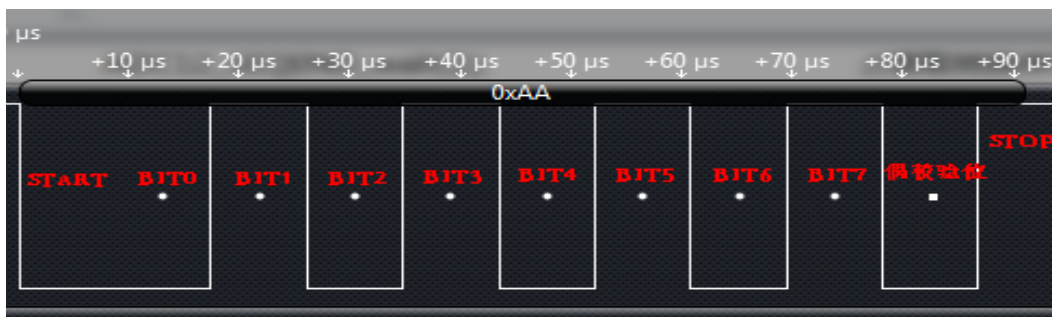
```

}

/*****/
//USART0 Interrupt
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
void USART0IntHandler(void)
{
    unsigned int status;
    status = USART0->MISR & USART0->IMSCR ;
    if( status & USART_RXRDY)
    {
        CSP_USART_SET_ICR(USART0, USART_RXRDY);
        R_Uart_RDBUF = CSP_USART_GET_RHR(USART0);
    }
}
    
```

## 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. 将 RXD 和 TXD 短接
3. 程序编译后仿真运行
4. 查看中断函数中 R\_Uart\_RDBUF 是否等于 0Xaa
5. 用逻辑分析仪出波形如下：



## 5. 改版历史

| 版本   | 修改日期       | 修改概要 |
|------|------------|------|
| V1.0 | 2019-11-08 | 初版   |