

文档版本	V1.0
发布日期	20191108

# APT32F172 IIC 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 .....	4
4. 程序下载和运行 .....	7
5. 改版历史 .....	8

## 1 概述

本文介绍了在APT32F172中使用IIC的应用范例。

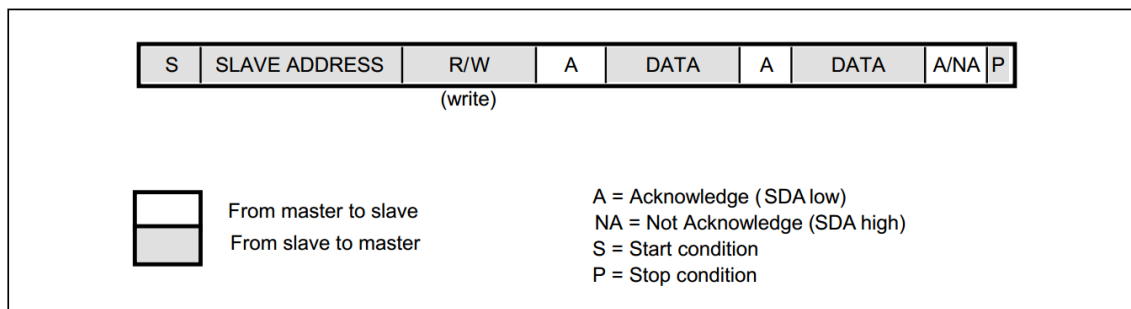
## 2. 适用的硬件

该例程使用于 APT32F172 开发板 APT-DB172

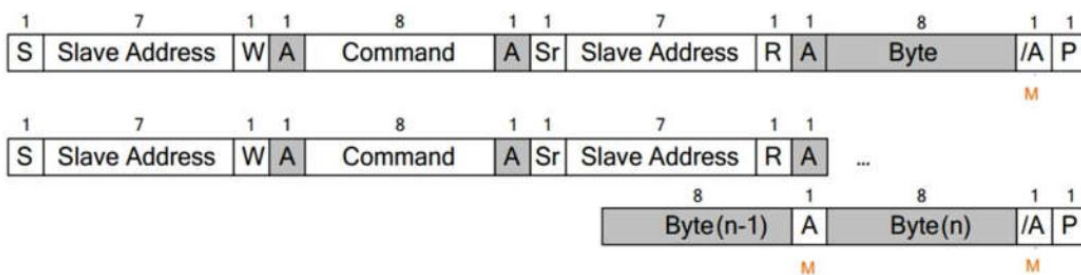
## 3. 应用方案代码说明

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

写数据传输格式如下：



读数据传输格式如下：



### 3.1 主机配置

软件配置：

开启内部主频 20MHz, 并作为系统时钟。

设置 SDA (PA0.9)，SCL (PA0.8) 且使能漏极开路输出，使能上拉 4.7k。

I2C 做主机模式

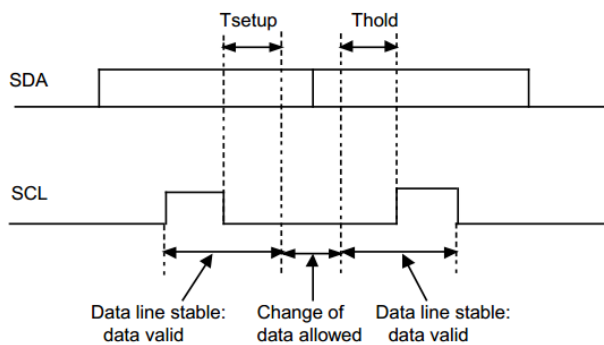
I2C 速度=PCLK/(PRV+4)

e. g:  $20\text{MHz} / (0x040+4) = 294\text{kHz}$

I2C hold/setup time = Htime \* PCLK

e. g:  $0x20 * (1/20\text{MHz}) = 1.6\mu\text{s}$

hold/setup time 如下:



I2C 器件地址设置为 0X56

0X01 地址中写 0XAA

0X01 地址中回读 0XAA

```

/*****/
//GPIO Functions
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
void GPIO_CONFIG(void)
{
    GPIO_OpenDrain_EN(GPIOA0,8); //PA0.8 漏极开路输出
    GPIO_OpenDrain_EN(GPIOA0,9); //PA0.9 漏极开路输出
    GPIO_PullHigh_Init(GPIOA0,8); // PA0.8 上拉使能
    GPIO_PullHigh_Init(GPIOA0,9); // PA0.9 上拉使能
}

/*****/
//I2C MASTER Initial
//EntryParameter:NONE
    
```

```

//ReturnValue:NONE

/*****/

void I2C_MASTER_CONFIG(void)

{
    I2C_DeInit();                //所有寄存器复位赋值
    I2C_Master_Init(I2C_G0,FAST_MODE,0x040,0x20); // 管脚配置 SDA(PA0.9), SCL(PA0.8), 通讯速度配置
}

/*****/

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

void APT32F172_init(void)

{sm
    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
    I2C_MASTER_CONFIG ();         //I2C 主机初始化
}
    
```

```
U32_T R_i2c_read_data;
/*****/
//main
/*****/
int main(void)
{
    APT32F172_init();
    while(1)
    {
        SYSCON_IWD CNT_Reload();           //清狗
        I2C_WriteByte(0x01,0xaa);
        delay_nms(40);                     //延时 1ms
        R_i2c_read_data= I2C_ReadByte(0x01);
    }
}
```

### 3.1 从机配置

开启内部主频 20MHz, 并作为系统时钟。

设置 SDA (PA1.1), SCL (PA1.2) 且使能漏极开路输出, 使能上拉 4.7k。

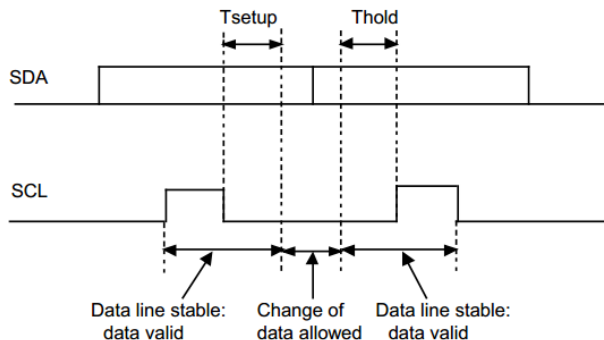
I2C 速度=PCLK/ (PRV+4)

e. g: 20Mhz/ (0x04F+4)=254kHz

I2C hold/setup time=Htime\*PCLK

e. g:0x40\* (1/20Mhz)=3.2us

hold/setup time 如下:



I2C 器件地址设置为 0X56

0X01 地址中写 0XAA

0X01 地址中回读 0XAA

```

/*****/
//GPIO Functions
//EntryParameter:NONE
//ReturnValue:NONE
/*****/

void GPIO_CONFIG(void)
{
    GPIO_OpenDrain_EN(GPIOA1,1);    //PA1.1 漏极开路输出
    GPIO_OpenDrain_EN(GPIOA1,2);    //PA1.2 漏极开路输出
    GPIO_PullHigh_Init(GPIOA1,1);    // PA1.1 上拉使能
    GPIO_PullHigh_Init(GPIOA1,2);    // PA1.2 上拉使能
}

/*****/
//I2C SLAVE Initial
//EntryParameter:NONE
//ReturnValue:NONE
/*****/

void I2C_SLAVE_CONFIG(void)
{

```

```

I2C_DeInit();                //所有寄存器复位赋值

I2C_Slave_Init(I2C_G2,FAST_MODE,0x4F,0x40,0xAC);

//从机地址=0xac(8bit), 快速模式, sysclock=20M, 通讯速度=sysclock/(0x4f+4)=240K

I2C_Int_Enable();           //I2C 中断向量使能
}

/*****/

//I2C Interrupt

//EntryParameter:NONE

//ReturnValue:NONE

/*****/

void I2CIntHandler(void)

{

    I2C_Slave_Receive();

}

/*****/

//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
I2C_SLAVE_CONFIG ();               //I2C 从机初始化
}

/*****/

//main
/*****/

int main(void)
{
    APT32F172_init();

    while(1)
    {
        SYSCON_IWDCNT_Reload();     //清狗
        //I2CWrBuffer[1] 从机写数据
        //I2CRdBuffer[1] 从机读数据
    }
}
```

## 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 IIC 接口
3. 用逻辑分析观察波形是否正确

4. 查看从机程序中 I2CWrBuffer[1], I2CRdBuffer[1]值是否正确

## 5. 改版历史

版本	修改日期	修改概要
V1.0	2019-11-08	初版

