

文档版本	V1.0.0
发布日期	20221026

APT32F110x 基于 CSI 库 I2C 应用指南

HPT



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 I2C 配置	1
3.2 I2C 主机模式	2
3.3 I2C 从机模式	4
4. 程序下载和运行	6

1 概述

本文介绍了在APT32F110x中使用I2C的应用范例。

2. 适用的硬件

该例程使用于 APT32F110x 学习板

3. 应用方案代码说明

3.1 I2C 配置

基于 APT32F110x 完整的 CSI 库文件系统，对 I2C 进行配置。

- 硬件配置：

I2C 主-从设备常见硬件连接电路

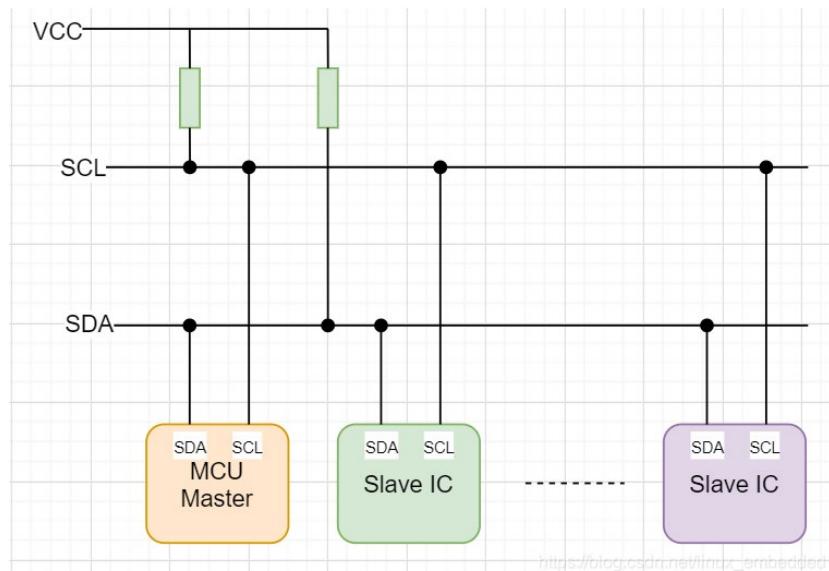


图 3.1.1 I2C 硬件

通讯引脚：

Pin Name	Function	I/O Type	Active Level	Comments
SDA	串行数据线	I/O	高有效	-
SCL	串行时钟线	I/O	高有效	-

图 3.1.2 I2C 引脚

三种传输模式：标准模式传输速率为 100kbit/s，快速模式为 400kbit/s，超快速模式下可达 1000kbit/s。

支持 7 位、10 位设备地址、支持发送和接收缓冲 (FIFO)
从机发送模式下支持大量传输模式

● 软件配置：

编程要点：

1. 通过 csi_pin_output_mode 设置目标引脚为开漏模式
2. 通过 csi_pin_set_mux 设置目标引脚为 I2C 功能
3. 通过 csi_iic_master_init 配置 I2C 主机或者 I2C 从机.

3.2 I2C 主机模式

```
void i2c_master_config()
{
    csi_iic_master_config_t tlicMasterCfg;
    //

    csi_pin_output_mode(PA07,GPIO_OPEN_DRAIN);
    csi_pin_output_mode(PA08,GPIO_OPEN_DRAIN);
    csi_pin_set_mux(PA07,PA07_I2C0_SCL);
    csi_pin_set_mux(PA08,PA08_I2C0_SDA);
    //

    tlicMasterCfg.byAddrMode = IIC_ADDRESS_7BIT;
    tlicMasterCfg.byReStart = ENABLE;
    tlicMasterCfg.bySpeedMode = IIC_BUS_SPEED_FAST;
    tlicMasterCfg.hwiInt = IIC_INTSRC_NONE;
    tlicMasterCfg.wSdaTimeout = 0xFFFF;
    tlicMasterCfg.wSclTimeout = 0xFFFF;
    csi_iic_master_init(I2C0,&tlicMasterCfg);
}

void main()
{
    system_init();
    board_init();
    i2c_master_config();
```

```

while(1)
{
    csi_iic_write_nbyte(I2C0,0xa0,0x55,1,&data[0],2);
    mdelay(100);

    csi_iic_read_nbyte(I2C0,0xa0,0x55,1,&data1[0],2);
    mdelay(100);
}

```

- 代码说明：

csi_pin_output_mode(); --- 用于设置 GPIO 模式
csi_pin_set_mux(); --- 用于 GPIO 配置为 I2C 功能
csi_iic_master_init(); --- 用于配置 I2C 参数
csi_iic_write_nbyte(); --- 用于 I2C 写入数据函数
csi_iic_read_nbyte(); --- 用于 I2C 读取数据函数

- 参数说明：

csi_iic_master_init(I2C0,&tlicMasterCfg); 对结构体 tlicMasterCfg 进行赋值操作，从而配置 I2C 主机模式。

tlicMasterCfg.byAddrMode --- 设置主机地址模式 7/10 bit

tlicMasterCfg.byReStart --- 使能重复起始位

tlicMasterCfg.bySpeedMode --- 设置主机速度模式 100k/ 400k /1Mhz

tlicMasterCfg.hwInt --- 使能需要的中断

tlicMasterCfg.wSdaTimeout --- SDA 超时时间设置 (1/主频 * wSdaTimeout ms)

tlicMasterCfg.wSclTimeout --- SCL 超时时间设置 (1/主频 * wSclTimeout ms)



● 测试 I2C 读写 24C02

5		csi_iic_write_nbyte(I2C0,0xa0,0x55,1,&data[0],2);	
0	b	mdelay(100);	
1	c	csi_iic_read_nbyte(I2C0,0xa0,0x55,1,&data1[0],2);	
2	d	mdelay(100);	
3			
4		Frame Info	
5	Expression	Value	Type
6	g_hwAdcBuf[0]		volatile
7	data1	[10]	volatile
8	0	0x00000001	volatile
9	1	0x00000002	volatile
0	2	0x00000000	volatile
1	3	0x00000000	volatile
2	4	0x00000000	volatile
3	5	0x00000000	volatile
4	6	0x00000000	volatile
5	7	0x00000000	volatile
6	8	0x00000000	volatile
7	9	0x00000000	volatile
8	data	[10]	volatile
9	0	0x00000001	volatile
0	1	0x00000002	volatile
1	2	0x00000003	volatile
2	3	0x00000004	volatile

图 3.2.1 I2C 测试数据

3.3 I2C 从机模式

从机地址不能设置成保留地址，0x00-0x07 或者 0x78-0x7f.

```

volatile uint8_t bySendBuffer[32] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20};

volatile uint8_t_t byWriteBuffer[32];

void i2c_slave_config()

{
    csi_iic_slave_config_t tlicSlaveCfg;
    //

    csi_pin_output_mode(PA07,GPIO_OPEN_DRAIN);
    csi_pin_output_mode(PA08,GPIO_OPEN_DRAIN);
    csi_pin_set_mux(PA07,PA07_I2C0_SCL);
    csi_pin_set_mux(PA08,PA08_I2C0_SDA);

    tlicSlaveCfg.byAddrMode = IIC_ADDRESS_7BIT;
    tlicSlaveCfg.bySpeedMode = IIC_BUS_SPEED_STANDARD;
    tlicSlaveCfg.hwSlaveAddr = 0xa0;
    tlicSlaveCfg.hwiInt = IIC_INTSRC_SCL_SLOW | IIC_INTSRC_STOP_DET |
        IIC_INTSRC_RD_REQ | IIC_INTSRC_RX_FULL | IIC_INTSRC_TX_ABRT;
    csi_iic_set_slave_buffer(byWriteBuffer,32,bySendBuffer,32);
    tlicSlaveCfg.wSdaTimeout = 0xFFFF;
    tlicSlaveCfg.wSclTimeout = 0xFFFF;
    csi_iic_slave_init(I2C0,&tlicSlaveCfg);
}

```

```
void main()
{
    system_init();
    board_init();
    i2c_slave_config();
    while(1);
}

void i2c_int_handler(void)
{
    // ISR content ...
    i2c_irqhandler(I2C0);
}
```

- 代码说明：

csi_iic_slave_init(); --- 用于配置 I2C 从机模式参数

csi_iic_set_slave_buffer(); --- 用于配置 I2C 从机的中断接收。

- 参数说明：

csi_iic_slave_init(I2C0,&tlicSlaveCfg);

根据 tlicSlaveCfg 结构体对 I2C 从机进行配置。

tlicSlaveCfg.byAddrMode --- 从机地址的模式 7 位/10 位

tlicSlaveCfg.bySpeedMode --- 从机地址的速度 100K/400K/1M

tlicSlaveCfg.hwSlaveAddr --- 从机地址

tlicSlaveCfg.hwlnt --- 对应的从机中断

tlicSlaveCfg.wSdaTimeout --- SDA 超时时间设置

tlicSlaveCfg.wSclTimeout --- SCL 超时时间设置

注：110x 系列做从机时，请打开 I2C 的中断处理函数。

- 测试

I2C 从机测试, 中断中接收数据

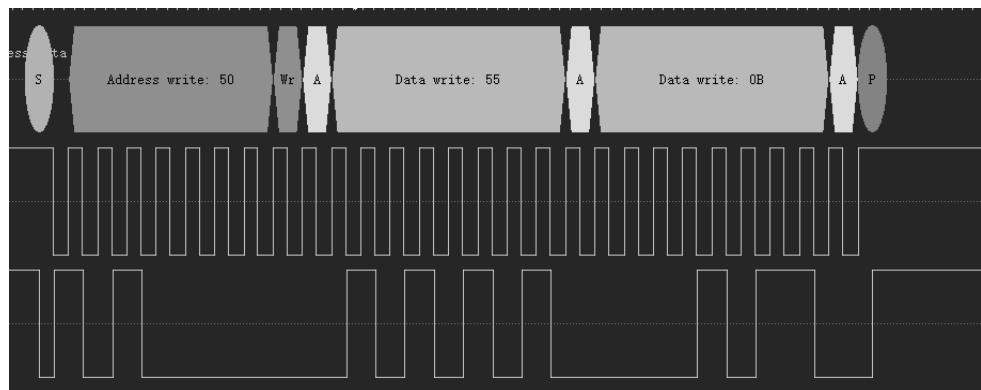


图 3.3.1 I2C 从机接收波形

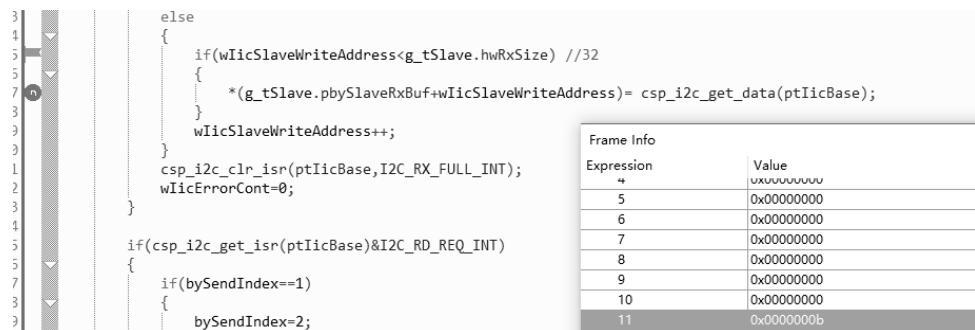


图 3.3.2 I2C 从机接收数据

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. 将主机 I2C 功能脚与对应的从机设备进行连接。
3. 程序编译后仿真运行
4. 写入数据，并读出进行查看验证。