

文档版本	V1.0.0
发布日期	20221026

APT32F110x 基于 CSI 库 CRC 应用指南



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 CRC 配置	1
3.2 CRC 应用	2
4. 程序下载和运行	6

1 概述

本文介绍了在APT32F110x中使用CRC的应用范例。

2. 适用的硬件

该例程使用于 APT32F110x 系列学习板

3. 应用方案代码说明

3.1 CRC 配置

基于 APT32F110x 完整的库文件系统，可以对 CRC 进行配置。

- **概述：**

该模块是用来纠错的 CRC 引擎，支持常用的 CRC。CRC 模块的使用，只需要将数据写入 CRC_SEED 和 CRC_DATAIN 寄存器中，并且在下一个指令读取 CRC_DATAOUT 寄存器即可。

CRC_CR 寄存器用来配置 CRC 引擎。在 CRC 模块工作前，必须使能 CRC_CEDR 寄存器中的时钟使能 CKEN 位。

- **功能框图：**

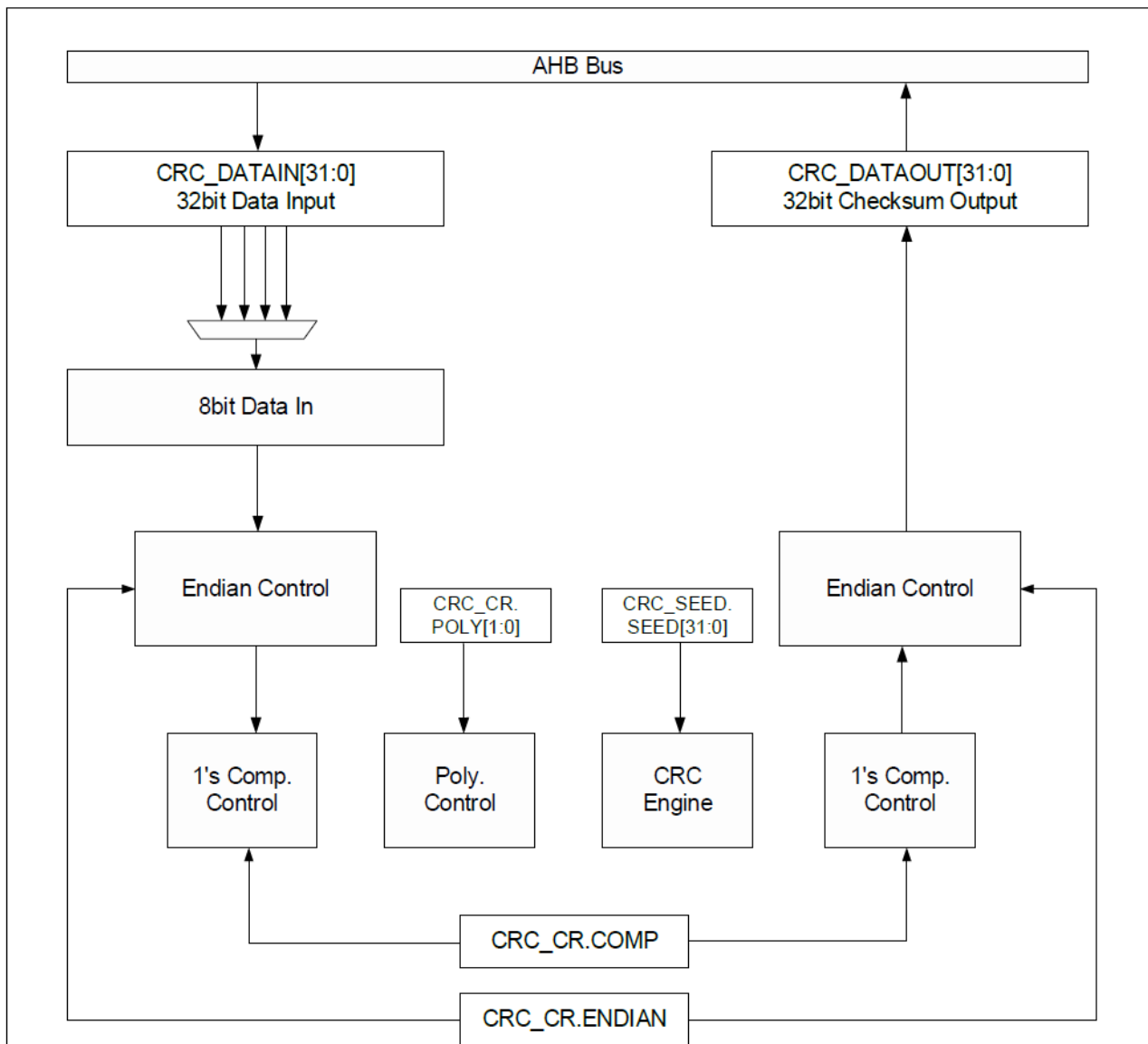


图 3.1.2 CRC 模块框图

3.2 CRC 应用

将数组 byTransData 中的数据进行各种模式的 CRC 计算，并判断计算结果。CRC 的计算结果可参考网址 <http://www.ip33.com/crc.html>。(可在 user_demo.c 文件中的 crc_demo()进行配置)

```

uint8_t byTransData[] = {0x4, 0x3, 0x2, 0x1, 0x8, 0x44, 0x33, 0x22, 0x11, 0x44, 0x33, 0x22, 0x11, 0x44, 0x33, 0x22, 0x11}; //需要 CRC 计算的数据

csi_error_t crc_demo(void)
{
    volatile uint32_t temp1 = 0;

```

```
csi_error_t iRet = CSI_OK;

csi_crc_init(); //CRC 模块初始化

temp1 = csi_crc32_be(0xffffffff, byTransData, 3); //进行 CRC-32 模式计算, 种子值 0xffffffff,byTransData 数组前 3 个数据, 数据长度 3 个字节
if (temp1 != 0xcb6faba1) //判断计算结果并串口打印信息
{
    iRet = CSI_ERROR;
    my_printf("CRC-32 calculation error\n");
}
else
{
    my_printf("CRC-32 calculation is correct.\n");
}

temp1 = csi_crc16_ccitt(0x00, byTransData, 16); //进行 CRC-16/CCITT 模式计算, 种子值 0x00,byTransData 数组前 16 个数据, 数据长度 16 个字节
if (temp1 != 0x1C40) //判断计算结果并串口打印信息
{
    iRet = CSI_ERROR;
    my_printf("CRC-16/CCITT calculation error\n");
}
else
{
    my_printf("CRC-16/CCITT calculation is correct.\n");
}

temp1 = csi_crc16(0x00, byTransData, 5); //进行 CRC-16 模式计算, 种子值 0x00,byTransData 数组前 5 个数据, 数据长度 5 个字节
if (temp1 != 0x1250) //判断计算结果并串口打印信息
{
    iRet = CSI_ERROR;
    my_printf("CRC-16 calculation error\n");
}
else
{
    my_printf("CRC-16 calculation is correct.\n");
}

temp1 = csi_crc16_itu(0x00, byTransData, 3); //进行 CRC-16 XMODEM 模式计算, 种子值 0x00,byTransData 数组前 3 个数据, 数据长度 3 个字节
if (temp1 != 0xa9d1) //判断计算结果并串口打印信息
```

```
{  
    iRet = CSI_ERROR;  
    my_printf("CRC-16 XMODEM calculation error\n");  
}  
else  
{  
    my_printf("CRC-16 XMODEM calculation is correct.\n");  
}  
  
while(1);  
return iRet;  
}
```

- 代码说明:

1. **csi_crc_init():** ----- CRC 模块初始化
2. **csi_crc32_be():** ----- CRC-32 转换函数
3. **csi_crc16_ccitt():** ----- CRC-16/CCITT 转换函数
4. **csi_crc16():** ----- CRC-16 转换函数
5. **csi_crc16_itu():** ----- CRC-16 XMODEM 转换函数

- 函数参数说明:

1. **csi_crc32_be(uint32_t wCrcSeed, uint8_t* pbyData, uint32_t wSize);**

wCrcSeed: CRC 计算种子值

pbyData: 需要计算的数据地址

wSize: 需要计算数据的长度

2. **csi_crc16_ccitt(uint16_t hwCrcSeed, uint8_t *pbyData, uint32_t wSize);**

hwCrcSeed: CRC 计算种子值

pbyData: 需要计算的数据地址

wSize: 需要计算数据的长度

3. **csi_crc16(uint16_t hwCrcSeed, uint8_t* pbyData, uint32_t wSize);**

函数参数参考 `csi_crc16_ccitt()` 函数。

4. `csi_crc16_itu(uint16_t hwCrcSeed, uint8_t* pbyData, uint32_t wSize);`

函数参数参考 `csi_crc16_ccitt()` 函数。

● 验证：

此处以 CRC-32 转换为例子验证。

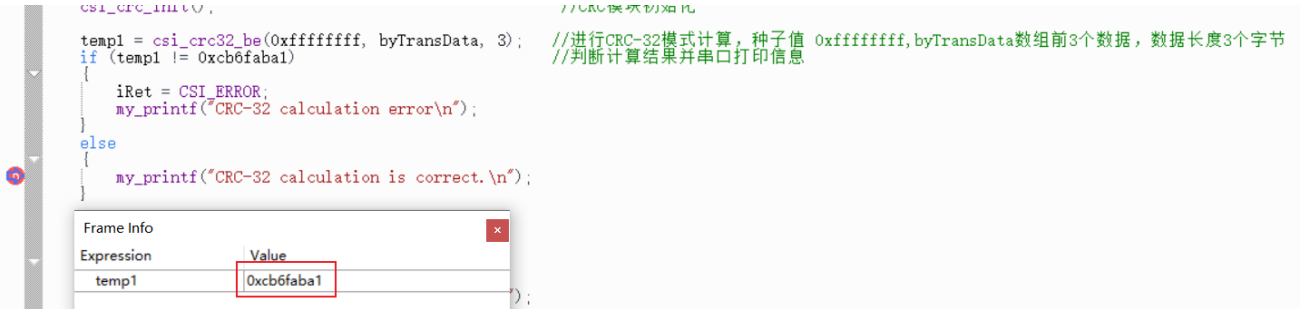


图 3.2.1 CDK 中转换结果



图 3.2.2 在线网页转换结果

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. 运行代码，对照网页转换数据。