NUC240使用I²C驱动DS3231

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

代码简述	本范例代码使用NUC240的I²C驱动DS3231
BSP 版本	NUC230_240 Series BSP CMSIS v3.01.001
开发平台	NuEdu-EVB-NUC240 v1.0

1 功能介绍

1.1 简介

本范例代码使用I²C接口，去读写DS3231缓存器，达成显示时钟功能。本范例提供了2个API，让使用者呼叫：

API	功能
void DS3231_RTC_Write(uint8_t u8Reg, uint8_t u8Value)	写入DS3231缓存器
uint8_t DS3231_RTC_Read(uint8_t u8Reg)	读取DS3231缓存器

1.2 原理

DS3231特点如下：

- 是低成本、高精度 I²C 实时时钟(RTC)。
- 保存秒、分、时、星期、日期、月和年信息。
- I²C 界面。
- 0~40 ℃范围内，精度正负 2 ppm；-40~85 ℃范围内，精度正负 3.5 ppm。

1.3 执行结果

NUC240开发版的PA9 (SCK)连接至DS3231的SCL；PA8 (SDA)连接至DS3231的SDA。编译项目并进入除错模式执行代码，DS3231时间数值会打印在串口窗口 UART #1。

```
19/11/5 15: 58: 02
19/11/5 15: 58: 03
19/11/5 15: 58: 04
19/11/5 15: 58: 05
19/11/5 15: 58: 06
19/11/5 15: 58: 07
19/11/5 15: 58: 08
19/11/5 15: 58: 09
19/11/5 15: 58: 10
19/11/5 15: 58: 11
19/11/5 15: 58: 12
19/11/5 15: 58: 13
19/11/5 15: 58: 14
```

2 代码介绍

2.1 写入 DS3231 缓存器值

DS3231_RTC_Write此函式主要工作为写入DS3231缓存器的值，在初始化时可利用此函式来设定起始的时间。

```
void DS3231_RTC_Write(uint8_t u8Reg, uint8_t u8Value)
{
    /* DS3231 I2C slave address */
    g_u8DeviceAddr = 0x68;

    /* DS3231 address map */
    g_au8TxData[0] = u8Reg;

    /* Write data to DS3231 */
    g_au8TxData[1] = dec_to_bcd(u8Value);
    g_u8DataLen = 0;
    g_u8EndFlag = 0;

    /* I2C function to write data to slave */
    s_I2C0HandlerFn = (I2C_FUNC)I2C_MasterTx;

    /* I2C as master sends START signal */
    I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_STA);

    /* Wait I2C Tx Finish */
    while (g_u8EndFlag == 0);

    g_u8EndFlag = 0;
}
```

2.2 读取 DS3231 缓存器值

DS3231_RTC_Read 此函式主要工作为读取DS3231缓存器的值，本范例程序在主程序中更新最新读取的时间。

```
uint8_t DS3231_RTC_Read(uint8_t u8Reg)
{
```

```
    /* DS3231 I2C slave address */
    g_u8DeviceAddr = 0x68;

    /* DS3231 address map */
    g_au8TxData[0] = u8Reg;

    g_u8DataLen = 0;
    g_u8EndFlag = 0;

    /* I2C function to read data from slave */
    s_I2C0HandlerFn = (I2C_FUNC)I2C_MasterRx;

    /* I2C as master sends START signal */
    I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_STA);

    /* Wait I2C Rx Finish */
    while (g_u8EndFlag == 0);

    g_u8EndFlag = 0;

    return bcd_to_dec(g_u8RxData);
}
```

3 软件与硬件环境

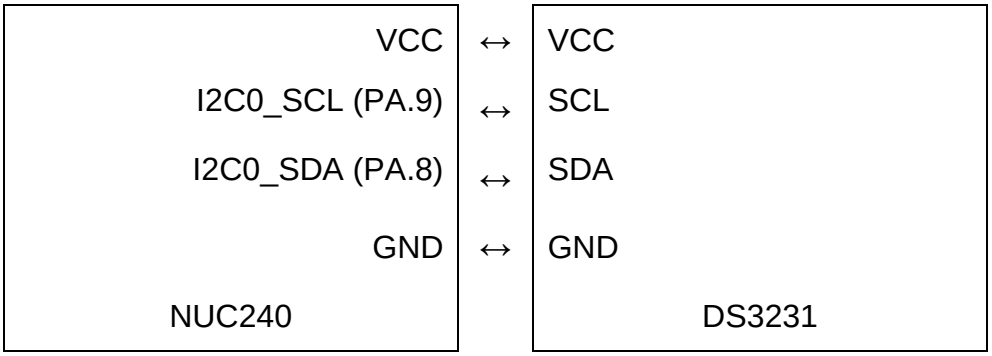
● 软件环境

- ■ BSP 版本
  - ◆ NUC230_240 Series BSP CMSIS v3.01.001
- ■ IDE 版本
  - ◆ Keil uVersion 4.74

● 硬件环境

- ■ 电路组件
  - ◆ NuEdu-EVB-NUC240 V1.0
  - ◆ DS3231
- ■ 示意图

NUC240使用I^{2}C脚位读写DS3231的缓存器值。



4 目录信息

- EC_ NUC240_I2C_DS3231_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - NuEdu — Library for NuEdu-SDK-NUC240 board
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 如何执行范例程序

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 NUC240_I2C_DS3231.uvproj。
2. 进入编译模式接口
   a. 编译
   b. 下载代码至内存
   c. 进入 / 离开除错模式
3. 进入除错模式接口
   a. 执行代码

6 修订纪录

Date	Revision	Description
Nov. 1, 2019	1.00	1. 初始发布

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.