NUC240使用GPIO模擬I²C存取EEPROM

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GPIO模拟I²C存取EEPROM
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

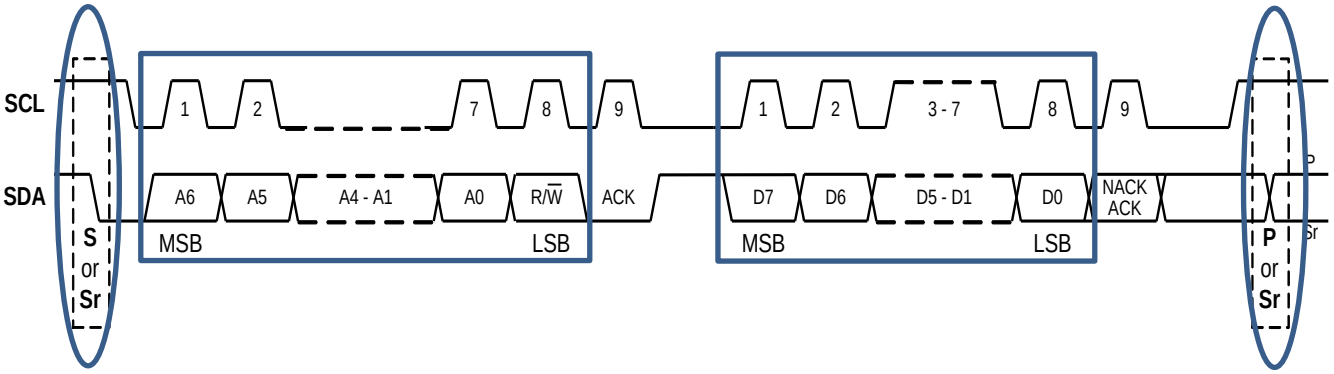
产品开发时，难免会碰到硬件I²C组数不够用，此时便可以用GPIO产生I²C的波形。本范例用软件操作GPIO实现I²C的功能，外部接一个EEPROM，程控去写读EEPROM。本范例提供了2个API，让使用者呼叫：

API	功能
void I2C_GPIO_Write_Byte(uint16_t u16Addr, uint8_t u8Data)	送出字符
void I2C_GPIO_Read_Byte(uint16_t u16Addr, uint8_t* pu8Data)	接收字符

1.2 原理

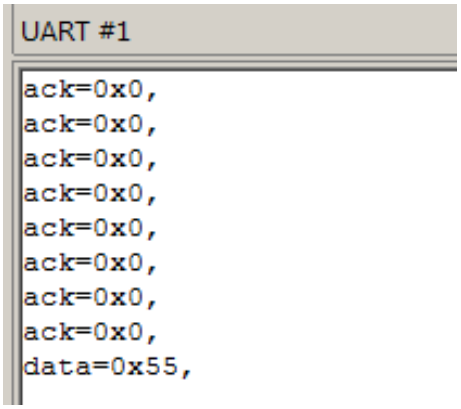
I²C协议分为四个步骤：

1. 产生 START 或 Repeated START 讯号
2. 传输 Slave 地址与读/写 bit
3. 数据传输
4. 产生 STOP 讯号



1.3 执行结果

NUC240开发版的PA11(SCK)连接至NuEdu-Basic01的NU6_10； PA10(SDA)连接至NuEdu-Basic01的NU6_9。编译项目并进入除错模式执行代码， EEPROM透过I²C响应的ACK与读写数值会打印在串口窗口 UART #1。


UART #1
ack=0x0,
ack=0x0,
ack=0x0,
ack=0x0,
ack=0x0,
ack=0x0,
ack=0x0,
ack=0x0,
data=0x55,

2 代码介绍

2.1 软件 I²C 初始化

I2C_Init此函式主要工作为软件I²C的初始化，即将GPIO(PA.10与PA.11)脚位设定成开漏模式，且在空闲时I/O为高电位。

```
void I2C_Init(void)
{
       /* Configure PA10 as open-drain mode */
       GPIO_SetMode(PA, BIT10, GPIO_PMD_OPEN_DRAIN);

       /* Configure PA11 as open-drain mode */
       GPIO_SetMode(PA, BIT11, GPIO_PMD_OPEN_DRAIN);

       /* Control PA11 output status */
       I2C_CLK = 1;

       /* Control PA10 output status */
       I2C_DAT = 1;
}
```

2.2 I²C 字符传送

I2C_GPIO_Write_Byte此函式主要工作为传送I²C字符，需指定EEPROM的地址与欲传输的字符。EEPROM协议请参考EEPROM产品规格手册。

```
void I2C_GPIO_Write_Byte(uint16_t u16Addr, uint8_t u8Data)
{
       uint8_t u8AddrH, u8AddrL;

       u8AddrH = u16Addr >> 8;
       u8AddrL = (uint8_t)u16Addr;

       /* Send Start bit to I2C EEPROM */
       I2C_Start();

       /* Send control byte to I2C EEPROM */
       printf("ack=0x%x,\n\r",I2C_Write(I2C_ADDRESS_W));
```

```
	/* Send I2C EEPROM's High Byte Address */
	printf("ack=0x%x,\n\r",I2C_Write(u8AddrH));

	/* Send I2C EEPROM's Low Byte Address */
	printf("ack=0x%x,\n\r",I2C_Write(u8AddrL));

	/* Send data byte to I2C EEPROM */
	printf("ack=0x%x,\n\r",I2C_Write(u8Data));

	/* Send Stop bit to I2C EEPROM */
	I2C_Stop();
}
```

2.3 I²C 字符接收

I2C_GPIO_Read_Byte此函式主要工作为I²C字符，需指定EEPROM的地址。EEPROM协议请参考EEPROM产品规格手册。

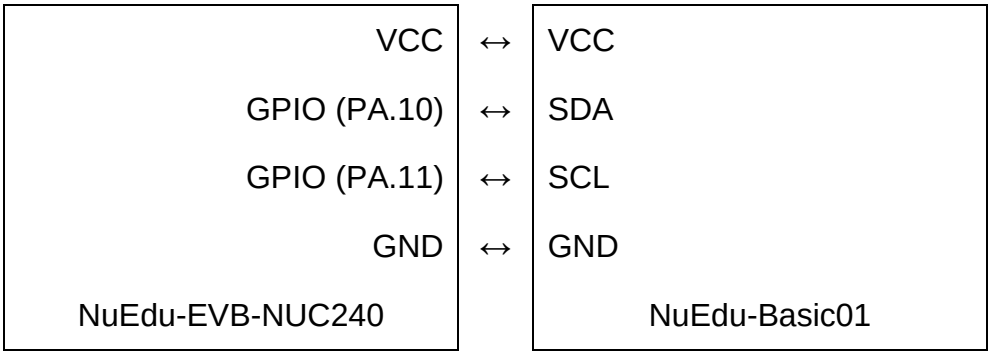
```
void I2C_GPIO_Read_Byte(uint16_t u16Addr, uint8_t* pu8Data)
{
	uint8_t u8AddrH, u8AddrL;

	u8AddrH = u16Addr >> 8;
	u8AddrL = (uint8_t)u16Addr;

	/* Send Start bit to I2C EEPROM */
	I2C_Start();

	/* Send control byte to I2C EEPROM */
	printf("ack=0x%x,\n\r",I2C_Write(I2C_ADDRESS_W));

	/* Send I2C EEPROM's High Byte Address */
	printf("ack=0x%x,\n\r",I2C_Write(u8AddrH));

	/* Send I2C EEPROM's Low Byte Address */
	printf("ack=0x%x,\n\r",I2C_Write(u8AddrL));

	/* Send Start bit to I2C EEPROM */
	I2C_Start();
```

```
    /* Send control byte to I2C EEPROM */
    printf("ack=0x%x,\n\r",I2C_Write(I2C_ADDRESS_R));

    /* Read data byte from EEPROM */
    *pu8Data = I2C_Read(NO_ACK);

    /* Send Stop bit to I2C EEPROM */
    I2C_Stop();
}
```

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
    - NuEdu-Basic01 V2.1
  - 示意图

    NuEdu-EVB-NUC240 使用 PA10 与 PA11 脚位模拟 I^2C 的 SDA 与 SCL 波型来控制在 NuEdu-Basic01 上的EEPROM。



4 目录信息

EC_NUC240_GPIO_I2C_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_GPIO_I2C.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.