



A76xx Series Open SDK_I2C_应用指导

LTE 模组

芯讯通无线科技(上海)有限公司
上海市长宁区临虹路289号3号楼芯讯通总部大楼
电话: 86-21-31575100
技术支持邮箱: support@simcom.com
官网: www.simcom.com

| | |
|-----|--------------------------------|
| 名称: | A76xx Series Open SDK_I2C_应用指导 |
| 版本: | V1.00 |
| 类别: | 应用文档 |
| 状态: | 已发布 |

版权声明

本手册包含芯讯通无线科技(上海)有限公司(简称:芯讯通)的技术信息。除非经芯讯通书面许可,任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部,并不得以任何形式传播,违反者将被追究法律责任。对技术信息涉及的专利、实用新型或者外观设计等知识产权,芯讯通保留一切权利。芯讯通有权在不通知的情况下随时更新本手册的具体内容。

本手册版权属于芯讯通,任何人未经我公司书面同意进行复制、引用或者修改本手册都将承担法律责任。

芯讯通无线科技(上海)有限公司

上海市长宁区临虹路289号3号楼芯讯通总部大楼

电话: 86-21-31575100

邮箱: simcom@simcom.com

官网: www.simcom.com

了解更多资料, 请点击以下链接:

<http://cn.simcom.com/download/list-230-cn.html>

技术支持, 请点击以下链接:

<http://cn.simcom.com/ask/index-cn.html> 或发送邮件至 support@simcom.com

版权所有 © 芯讯通无线科技(上海)有限公司 2023, 保留一切权利。

Version History

| Version | Date | Owner | What is new |
|---------|------------|-------|-------------|
| V1.00 | 2022-10-31 | | 第一版 |

About this Document

本文档适用于 A1803S open 系列、A1603 open 系列、A1606 open 系列。

SIMCom
Confidential

目录

| | |
|---------------------------|----|
| 版权声明 | 2 |
| Version History | 3 |
| About this Document | 4 |
| 目录 | 4 |
| 1 介绍 | 6 |
| 2 管脚使用及其定义 | 7 |
| 2.1 I2C 管脚使用 | 7 |
| 2.2 I2C 接口管脚定义 | 7 |
| 3 应用层 API | 8 |
| 3.1 实例代码目录 | 8 |
| 3.2 I2C API 介绍 | 8 |
| 4 功能测试验证 | 10 |
| 4.1 I2C 读数据测试 | 10 |
| 4.2 I2C 写数据测试 | 10 |
| 5 驱动调试方法 | 11 |

1 介绍

本模块提供一路 I2C 接口 (CI2C)，且本模块在与 I2C 接口有关的应用中一般只作为主设备。

时钟支持：标注 (100kHz)，fast (400kHz)，默认 Clock 400kHz。

支持 7 位设备寻址，即一个 I2C 总线最多可以挂 127 个从设备，对应数据量比较大的 I2C 设备，建议一条总线上只挂载一个设备。常用的 I2C 设备有 Sensor，Codec 等。

本文的模块硬件相关信息以 A7600 系列为例，仅供参考；具体硬件设计请参考硬件设计文档。

2 管脚使用及其定义

2.1 I2C 管脚使用

I2C_SCL: 模块内部已经 1.8V 上拉, 不用则悬空;

I2C_SDA: 模块内部已经 1.8V 上拉, 不用则悬空;

I2C 接口可复用为 GPIO, 默认功能为 I2C。

2.2 I2C 接口管脚定义

| 引脚名 | 引脚号 | 默认功能 | 复用功能 1 |
|-----|-----|------|--------|
| SCL | 55 | SCL | GPIO49 |
| SDA | 56 | SDA | GPIO50 |

3 应用层 API

3.1 实例代码目录

SIMCOM_SDK\sc_app\sc_demo\demo_i2c.c
SIMCOM_SDK\sc_app\sc_lib\inc\sapi_i2c.h

3.2 I2C API 介绍

1. 初始化接口

| | |
|-------|---|
| 接口: | SC_I2C_ReturnCode sAPI_I2CConfigInit(SC_I2C_DEV *handle) |
| 输入: | Handle i2c 设备信息, 包含 i2c channel 和 i2c clock i2cChannel : 0 (1803S 平台只有 1 路 IIC 出来) i2c clock: 100khz 和 400khz(默认) |
| 输出: | 无 |
| 返回值: | 正确返回 0, 其余返回值参考 sapi_i2c.h 中的 SC_I2C_ReturnCode |
| NOTE: | 无 |

2. 读从设备寄存器的值。

| | |
|-------|---|
| 接口: | SC_I2C_ReturnCode sAPI_I2CWrite(int i2cChannel,UINT8 slaveAddress,UINT8 regAddress,UINT8 *data,UINT16 datasize) |
| 输入: | i2cChannel: i2c 通道, 0 和 1 (1803S 只有 0 通道, 1 通道未引出) slaveAddress:从设备地址, (如: 板载 NAU8810 8bit 地址为: 0x34) regAddress:从设备的寄存器地址 data:存放读取到的数据的指针 datasize:读取长度 |
| 输出: | 无 |
| 返回值: | 正确返回 0, 其余返回值参考 sapi_i2c.h 中的 SC_I2C_ReturnCode |
| NOTE: | 无 |

3. 写从设备寄存器的值。

| | |
|-----|---|
| 接口: | SC_I2C_ReturnCode sAPI_I2CRead(int i2cChannel,UINT8 slaveAddress,UINT8 regAddress,UINT8 *receiveDataBuffer,UINT16 datasize) |
|-----|---|

| | |
|-------|---|
| 输入: | i2cChannel: i2c 通道, 0 和 1 (1803S 只有 0 通道, 1 通道未引出) slaveAddress:从设备地址,(如: 板载 NAU8810 8bit 地址为: 0x35) regAddress:从设备的寄存器地址 receiveDataBuffer:存放读取到的数据的指针 datasize:读取长度 |
| 输出: | 无 |
| 返回值: | 正确返回 0, 其余返回值参考 sapi_i2c.h 中的 SC_I2C_ReturnCode |
| NOTE: | 无 |

除此之外, 还有两个读写函数, 参数跟上述一致。

SC_I2C_ReturnCode sAPI_I2CReadEx(int i2cChannel,UINT8 slaveAddress,UINT8 regAddress,UINT8
*receiveDataBuffer,UINT16 datasize)

SC_I2C_ReturnCode sAPI_I2CWriteEx(int i2cChannel,UINT8 slaveAddress,UINT8 regAddress,UINT8
*data,UINT16 datasize)

两个接口的差异:

sAPI_I2CRead()相比 sAPI_I2CReadEx()来讲, 在 wait i2c busy 总线状态的时候, 会有一点延时。

sAPI_I2CWrite 和 I2CWriteEx 同理。

4 功能测试验证

4.1 I2C 读数据测试

打开 Uldemo 界面

如下图所示：

```
Please select an option to test from the items listed below.

*****
1. NETWORK                2. SDMCARD
3. SMS                    4. UART
5. USB                    6. GPIO
7. PMU                    8. I2C
9. AUDIO                  10. FILE SYSTEM
11. TCPIP                 12. HTTP
13. FTP                   14. MQTT
15. SSL                   16. OTA
17. LBS                   18. NTP
19. HTP                   20. INTERNET SERVICE
21. TTS                   22. CALL
23. WIFI                  24. LCD
25. RTC                   26. TXRX_POWER
27. FLASH                 28. SPI
29. SPINOR                 30. PWM
31. WTD                   32. WEBUI
*****
```

输入 8：

```
*****
1. I2C Write              2. I2C Read
3. I2C ReadEx            4. I2C INIT
99. back
*****
```

然后输入 4，执行初始化操作，再去发送 2 读指令

执行完上面的命令后，用逻辑分析仪，在引脚 55、56 上可以得到下面的 IIC 时序图：

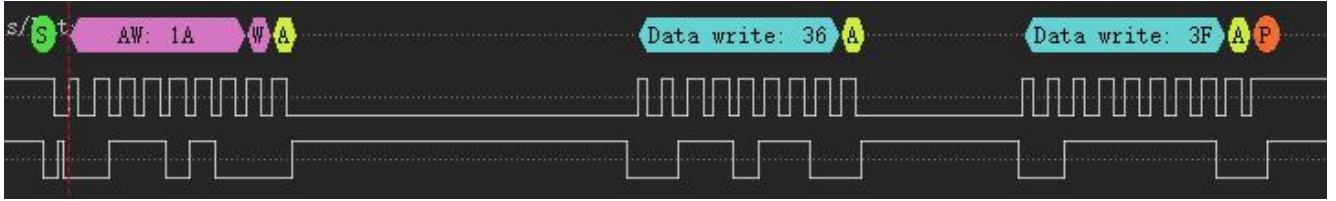


4.2 I2C 写数据测试

如下图所示：

执行 1，写操作

执行完上面的命令后，用逻辑分析仪，在引脚 55、56 上可以得到下面的 IIC 时序图：



5 驱动调试方法

下面使用 NAU8810 举例：

- 1：确认 I2C 的通道选择正确，不同的通道对应绑定的不同的 I/O
- 2：确认 NAU8810 的外围供电是否正常，保证设备正常工作
- 3：以上 2 步确认 ok 后，即可进行读写调试
- 4：如果遇到 **nack**，检查从机地址和寄存器地址是否正确
- 5：遇到写时序或者读时序，数据异常，读写不一致的时候，检查从机的读写时序是否标准，有些个别的，在读写之前需要对数据进行拼接。