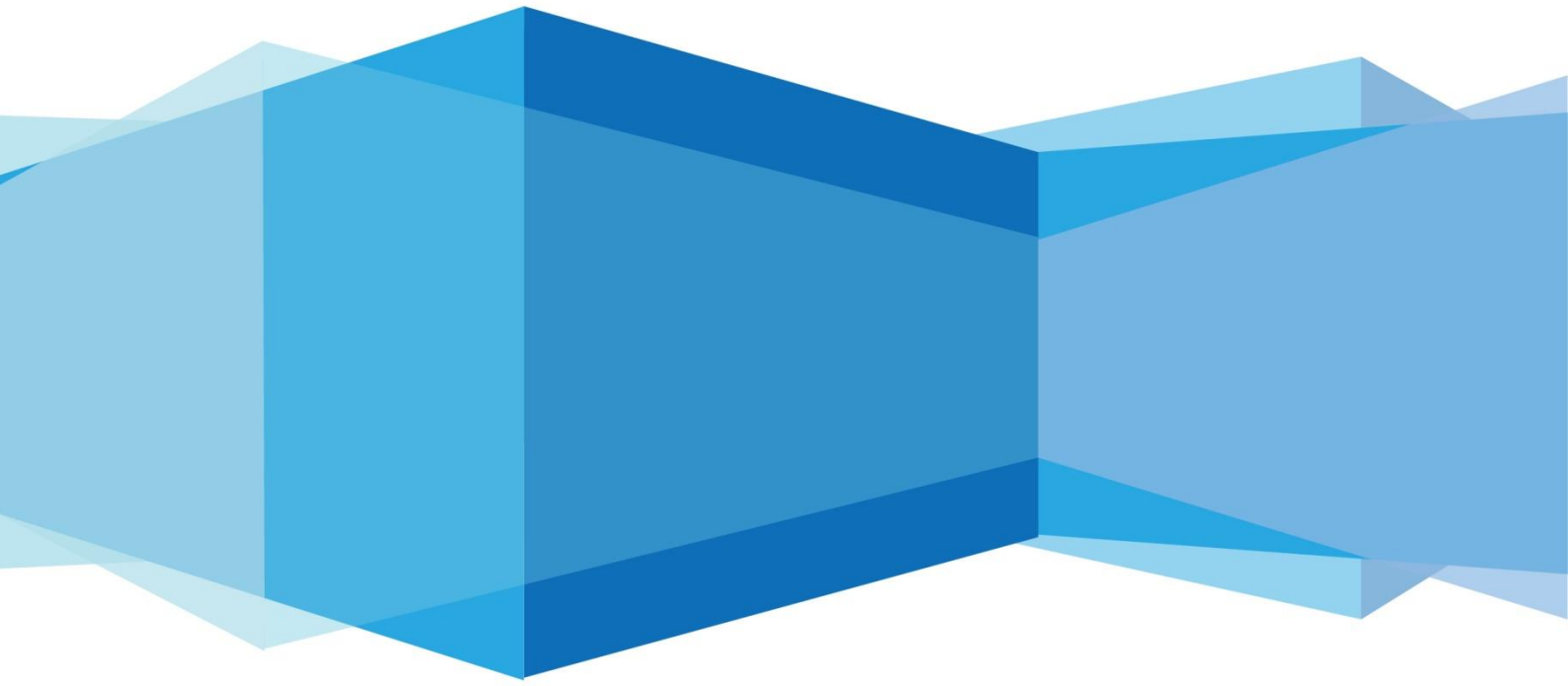
lierda 利尔达科技集团 | 证券代码 832149

Lierda NB86-G

OpenCPU_I2C 应用笔记

版本：Rev1.0

日期：2018-11-30



利尔达科技集团股份有限公司
LIERDA SCIENCE & TECHNOLOGY GROUP CO.,LTD

法律声明

若接收浙江利尔达物联网技术有限公司（以下称为“利尔达”）的此份文档，即表示您已经同意以下条款。若不同意以下条款，请停止使用本文档。

本产品符合有关环境保护和人身安全方面的设计要求，产品的存放、使用和弃置应遵照产品手册、相关合同或者相关法律、法规的要求进行。

本公司保留在不预先通知的情况下，对此手册中描述的产品进行修改和改进的权利；同时保留随时修订或收回本手册的权利。

文件修订历史

版本	修订日期	修订日志
1.0	2018-09-27	新建文档
1.1	2018-10-19	更改文档格式
1.2	2018-10-27	增加A板LED说明
1.3	2018-11-09	增加A板下发命令和A板一键拨测
1.4	2018-11-30	增加I2C应用

适用模块型号

序号	模块型号	模块简介
1	NB86-G	全频段版本，20×16×2.2（mm）
2	NB86-G 宽压型	全频段版本，20×16×2.2（mm）

安全须知

用户有责任遵循其他国家关于无线通信模块及设备的相关规定和具体的使用环境法规。通过遵循以下安全原则，可确保个人安全并有助于保护产品和工作环境免遭潜在损坏。我司不承担因客户未能遵循这些规定导致的相关损失。

	道路行驶安全第一！当您开车时，请勿使用手持移动终端设备，除非其有免提功能。请停车，再打电话！
	登机前请关闭移动终端设备。移动终端的无线功能在飞机上禁止开启用以防止对飞机通讯系统的干扰。忽略该提示项可能会导致飞行安全，甚至触犯法律。
	当在医院或健康看护场所，注意是否有移动终端设备使用限制。RF 干扰会导致医疗设备运行失常，因此可能需要关闭移动终端设备。
	移动终端设备并不保障任何情况下都能进行有效连接，例如在移动终端设备没有花费或 SIM 无效。当您在紧急情况下遇见以上情况，请记住使用紧急呼叫，同时保证您的设备开机并且处于信号强度足够的区域。
	您的移动终端设备在开机时会接收和发射射频信号，当靠近电视，收音机电脑或者其它电子设备时都会产生射频干扰。
	请将移动终端设备远离易燃气体。当您靠近加油站，油库，化工厂或爆炸作业场所，请关闭移动终端设备。在任何有潜在爆炸危险场所操作电子设备都有安全隐患。

目 录

1. 引言

本文旨在帮助基于使用 Lierda NB86-G 模组进行 OpenCPU 开发的用户，让其能快速使用模组本身的各种硬件资源（I2C、GPIO、UART）和 LiteOS 操作系统（创建、删除、挂起、恢复线程。创建、删除、启动、停止软件定时器），文章概述了 openCPU_i2c 的特性、软硬件参考设计、样例说明以及设计要点。

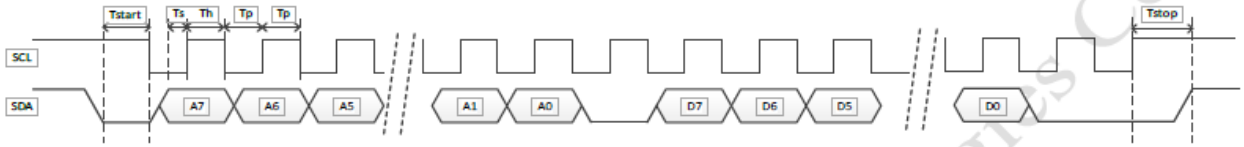
2. 特性概述

本设备有两个 I² C 接口用于与外围设备通信，这些接口可用于活动和待机模式。其中每个都包括以下功能：

- 标准模式双向总线（比特率高达 100 kbit / s）
- 作为主设备或从设备运行，但不作为多主设备系统中的主设备运行
- 支持 7 位或者 10 位寻址
- 支持时钟拉伸检测和生成
- 无需软件干预即可传输/接收最多 16 个字节的数据

I2C 接口需要外部上拉电阻。上拉电阻的允许值范围取决于所使用的 IO 电压和总线电容。建议使用朝向该范围最高的值，以最大限度地降低功耗。

接口速率可编程为 732 Hz 至 3 MHz（必须相应地选择板级上拉电阻值）。



图表 1I2C 时序图

I2C 时序信息	Min	Typ	Max	Unit
Start time，T_{start}	80			ns
Stop time，T_{stop}	80			ns
Steup time，T_s	50			ns
Hold time, T_h	T_p+500			ns
Half-period, T_p	167			ns

表格 1I2C 时序信息

3. 硬件参考设计

建议使用 R2 和 L1 电源域所在的 IO 口，其它电源域所在 IO 不建议使用。

如果模组和终端之间 I2C 电平匹配时，不需要增加电平转换电路，但是必须要增加上拉电阻，阻值建议在 4.7K-10K 之间选择，阻值需要在功耗和通信速率之间做平衡，阻值越大，功耗越低通信速率也越低，反正则相反。

终端 I2C 通信电平不匹配时需要增加电平转换电路，可参考图 3-1 设计。I2C 引脚需要选择同一个电源域的通用 GPIO。Terminal_VCC 上拉电平需要和终端 I2C 通信电平保持一致。建议通信速率控制在 800Kbps 之下，如果需要更高速率，请调整 R1、R2、R3、R4 上拉电阻的阻值到≤4.7K。

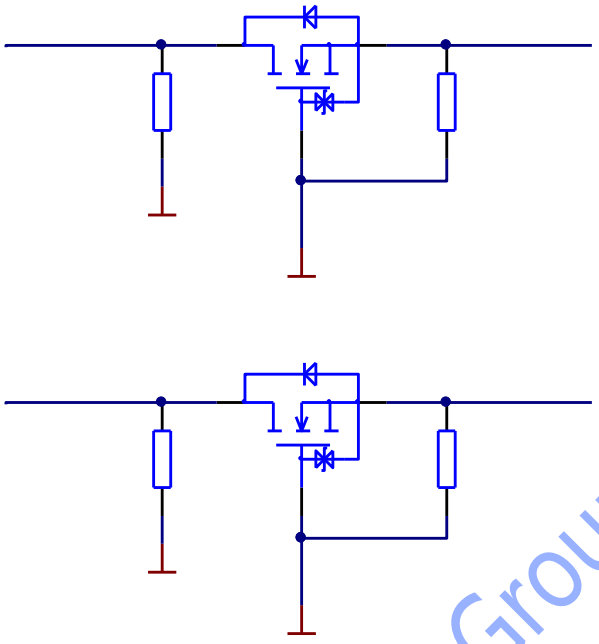
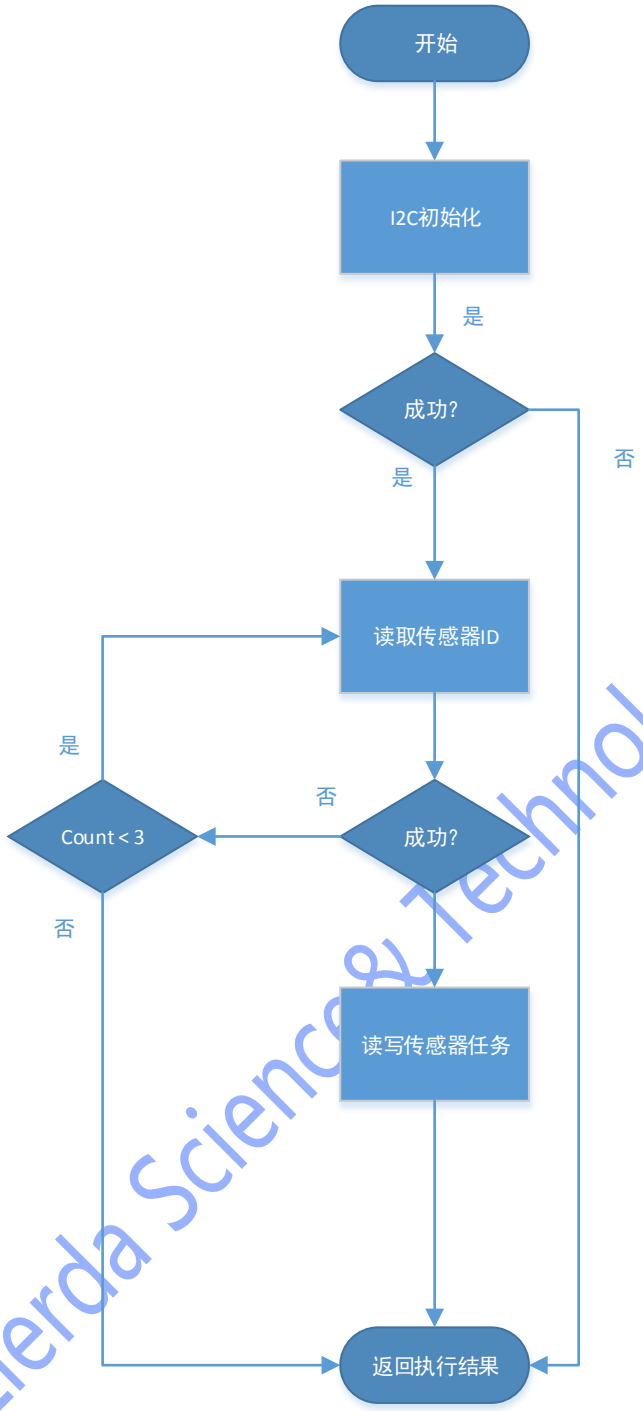


图 3-1 电平转换参考电路

4. 软件参考设计




```
I2C_HandleTypeDef   sensorI2CHandle;
uint8 uGsensorInit(void)
{
    uint8 ucResponse = 0;
    uint8 ucDataTemp;
```

```
#define I2C_SCL             14
#define I2C_SDA             12

 I2C_HandleTypeDef  sensorI2CHandle;

sensorI2CHandle.i2c_bus = I2C_BUS1;
sensorI2CHandle.pin_scl = I2C_SCL;
sensorI2CHandle.pin_sda = I2C_SDA;
sensorI2CHandle.i2c_address_type = HAL_I2C_ADDRESS_TYPE_7_BIT;
sensorI2CHandle.i2c_half_time = 256;
sensorI2CHandle.i2c_mode = HAL_I2C_BUS_MODE_MASTER;

lierdaI2CInit(&sensorI2CHandle);

ucDataTemp = 0x05;

uint8 temp[1] = {0x04};

if(!lierdaI2CWritereg(&sensorI2CHandle,0xc0,0x26,temp,1)){
    lierdaLog("writeReg  succeed");
}//向0xc0寄存器写入0x26


//读取0XC0寄存器的值
if (!lierdaI2CReadreg(&sensorI2CHandle,0xC0,0x26,&ucResponse,1,0))
{
    lierdaLog("get succeed: 0x%X", ucResponse);
} else {
    lierdaLog("get failed: 0x%X", ucResponse);
}
//读取传感器设备ID
while ((ucResponse != 0x13) && ((ucDataTemp--) > 0)) {
    lierdaI2CReadreg(&sensorI2CHandle,0x4c,0x01,&ucResponse,1,0);
    if (ucDataTemp <= 1) {
        lierdaLog("Read who am I failed!");
    }
}
return 1;
 }
```

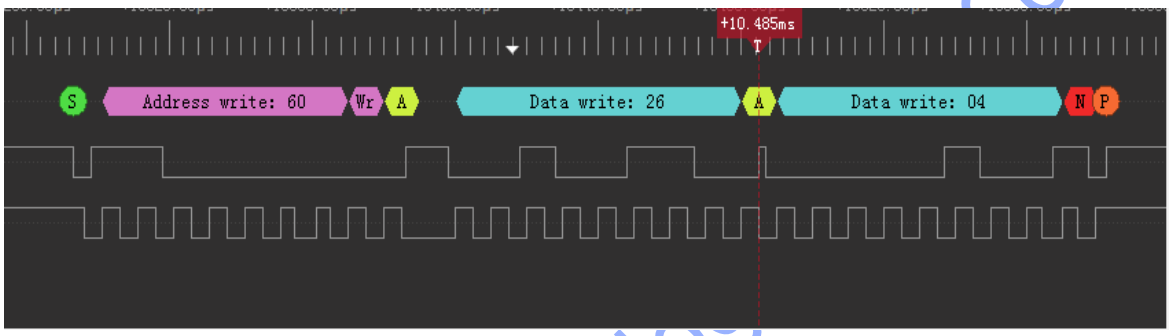
5. 测试样例

5.1. 读写测试

```
uint8 temp[1] = { 0x04 };

    if (!lierdaI2CWritereg(&sensorI2CHandle, 0x60, 0x26, temp, 1)) {
        lierdaLog("writeReg  succeed");
        }
```

如图所示，向从机地址为 0x60 的 I2C 设备，写入 0x04 到寄存器 0x26 中。

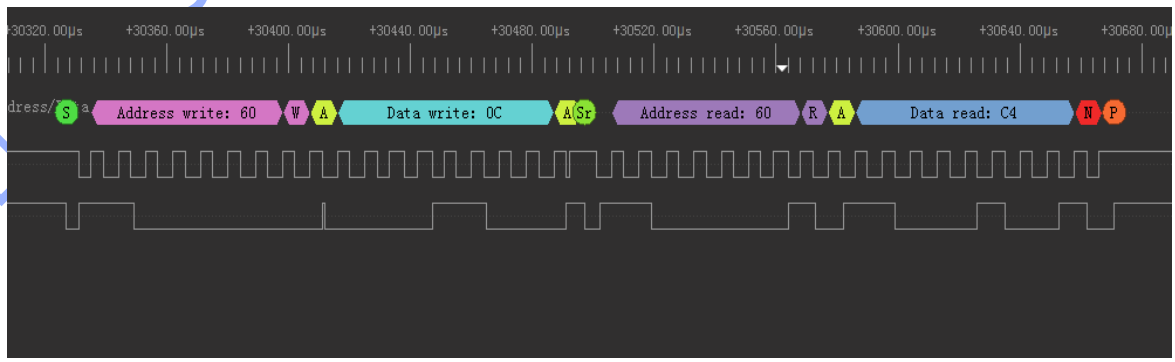


图表 2 写入数据时序图

```
    uint8 ucResponse = 0;

        if (!lierdaI2CReadreg(&sensorI2CHandle, 0x60, 0x0c, &ucResponse,
    1, 0)) {
        lierdaLog("I get succeed: 0x%X", ucResponse);
    } else {
        lierdaLog("I get failed: 0x%X", ucResponse);
    }
```

如图所示，读取从机地址为 0x60 的 I2C 设备中寄存器 0x0c 的数值。



图表 3 读取数据时序图

5.2. 连续读写功耗测试

```
uint8 temp[1] = {0x04};

for(uint8 count = 0;count < 10;count ++)
{
    if(!lierdaI2CWritereg(&sensorI2CHandle,0xc0,0x26,temp,1))
    {
        lierdaLog("writeReg  succeed");
    }

    if (!lierdaI2CReadreg(&sensorI2CHandle,0xC0,0x26,&ucResponse,1,0))
    {
        lierdaLog("I get succeed: 0x%X", ucResponse);
    } else {
        lierdaLog("I get failed: 0x%X", ucResponse);
    }
    }
```

如图所示，在连续对 0x26 寄存器读写了 10 次时，产生的平均电流约为 3mA。

图表 4 连续读写平均电流

6. 设计要点

1. I2C 设备相关属性必须配置正确，譬如“half_time”，（256 对应 100k 的速率）一定要结合传感器芯片手册给出的数据来配置，否则将影响使用。
2. 使用读写函数的时候需要传入正确的参数，尤其是 I2C 从设备的地址，同样需要结合芯片手册使用，严格按照芯片手册的数据对应填写。

7. 相关文档及术语缩写

以下相关文档提供了文档的名称，版本请以最新发布的为准。

表格 2 相关文档

序号	文档名称	注释
[1]	NB86-G硬件应用手册	
[2]	Lierda NB Module V150_AT CommandSet_B300SP2	
[3]	Lierda NB86-EVK测试终端固件烧写教程	
[4]	Lierda NB-IoT模组API使用文档	
[5]	Lierda NB-IoT模组DEMO说明文档	
[6]	Lierda NB-IoT模组V150 OpenCPU开发环境搭建指南	