

## 1 概述

LPC802 是 LPC800 系列的一个新成员，它满足了人们对更低功耗 MCU 的需求。它是从 8 位机向 32 位机迁移的理想入门产品，而且成本非常低。LPC802 具有高达 15 MHz 的 Cortex-M0+ 内核，具有 16 kB 闪存和 2 kB RAM。适用于 I/O 扩展器和小型可编程逻辑单元。

本应用笔记讨论了将 LPC802 用作类似 EEPROM 的设备。由于 LPC802 的内部闪存可承受至少 200 k R/W 周期，因此将 LPC802 的内部闪存用作 EEPROM 的非易失性存储器。借助专用固件，LPC802 可以转换为具有 UART 或 SPI 等可选接口的智能 EEPROM。

本应用笔记使用 LPC802 的 I<sup>2</sup>C 总线接口和片上闪存来模拟传统的 I<sup>2</sup>C 总线 EEPROM 器件。您可以从中学到如下知识：

- LPC802 概述、内存资源和布局、可用外围设备
- 使用 flash IAP 和示例代码
- 使用 I<sup>2</sup>C 总线模块，尤其是 I<sup>2</sup>C 总线从机的操作

### 目录

|     |                              |    |
|-----|------------------------------|----|
| 1   | 概述.....                      | 1  |
| 2   | 硬件.....                      | 1  |
| 2.1 | MCU 概述.....                  | 1  |
| 2.2 | 硬件连接.....                    | 3  |
| 3   | 软件.....                      | 3  |
| 3.1 | IAP (应用程序编程).....            | 3  |
| 3.2 | I <sup>2</sup> C 总线接口编程..... | 4  |
| 3.3 | 通信协议.....                    | 5  |
| 3.4 | 总结.....                      | 6  |
| 4   | 测试与结果.....                   | 7  |
| 4.1 | 环境设置.....                    | 7  |
| 4.2 | 硬件和连接.....                   | 8  |
| 4.3 | 测试步骤.....                    | 8  |
| 4.4 | 测试结果.....                    | 9  |
| 4.5 | 性能.....                      | 9  |
| 5   | 结论.....                      | 10 |

### 注意

需要了解 I<sup>2</sup>C 总线的知识。有关 I<sup>2</sup>C 总线规范，参阅 <http://www.i2c-bus.org/specification/>。

## 2 硬件

### 2.1 MCU 概述

LPC802 是基于 ARM<sup>®</sup> Cortex<sup>®</sup>-M0+ 的低成本 32 位 MCU 系列，可在高达 15 MHz 的 CPU 频率下运行。LPC802 支持 16 kB 闪存和 2 kB SRAM，提供 TSSOP16，TSSOP20，HVQFN33 和 WLCSP16 封装。

此外，双电源系统还提供电平转换器功能，从而减少了相应的外部组件和总的系统 BOM 成本。

LPC802 的外围设备包括：

- 一个 I<sup>2</sup>C 总线接口
- 最多两个 USART
- 一个 SPI 接口
- 一个多速率计时器，自唤醒计时器，一个通用 32 位计数器/计时器
- 一个 12 位 ADC，一个模拟比较器
- 通过开关矩阵和多达 17 个通用 I/O 引脚的功能可配置的 I/O 端口。三个 I/O 引脚具有高驱动能力，可提供高达 20 mA 的源电流

#### 2.1.1 Flash 功能

LPC802 的片上闪存（总共 16 kB）包含 16 个扇区。每个扇区的大小为 1 kB，包含 16 页。每页的大小为 64 个字节。IAP 命令支持：



- 页擦除
- 页写入
- 扇区擦除

有关闪存分区和配置，请参阅 [表 1](#)。

**表 1. Flash 分区和配置**

| 虚拟扇区编号 | 虚拟扇区大小 (KB) | 页码数       | 地址范围                          | 16 KB 闪存 | 32 KB 闪存 |
|--------|-------------|-----------|-------------------------------|----------|----------|
| 0      | 1           | 0 - 15    | 0 × 0000 0000 - 0 × 0000 03FF | 是        | 是        |
| 1      | 1           | 16 - 31   | 0 × 0000 0400 - 0 × 0000 07FF | 是        | 是        |
| 2      | 1           | 32 - 47   | 0 × 0000 0800 - 0 × 0000 0BFF | 是        | 是        |
| 3      | 1           | 48 - 63   | 0 × 0000 0C00 - 0 × 0000 0FFF | 是        | 是        |
| 4      | 1           | 64 - 79   | 0 × 0000 1000 - 0 × 0000 13FF | 是        | 是        |
| 5      | 1           | 80 - 95   | 0 × 0000 1400 - 0 × 0000 17FF | 是        | 是        |
| 6      | 1           | 96 - 111  | 0 × 0000 1800 - 0 × 0000 1BFF | 是        | 是        |
| 7      | 1           | 112 - 127 | 0 × 0000 1C00 - 0 × 0000 1FFF | 是        | 是        |
| 8      | 1           | 128 - 143 | 0 × 0000 2000 - 0 × 0000 23FF | 是        | 是        |
| 9      | 1           | 144 - 159 | 0 × 0000 2400 - 0 × 0000 27FF | 是        | 是        |
| 10     | 1           | 160 - 175 | 0 × 0000 2800 - 0 × 0000 2BFF | 是        | 是        |
| 11     | 1           | 176 - 191 | 0 × 0000 2C00 - 0 × 0000 2FFF | 是        | 是        |
| 12     | 1           | 192 - 207 | 0 × 0000 3000 - 0 × 0000 33FF | 是        | 是        |
| 13     | 1           | 208 - 223 | 0 × 0000 3400 - 0 × 0000 37FF | 是        | 是        |
| 14     | 1           | 224 - 239 | 0 × 0000 3800 - 0 × 0000 3BFF | 是        | 是        |
| 15     | 1           | 240 - 255 | 0 × 0000 3C00 - 0 × 0000 3FFF | 是        | 是        |

以下是 IAP Flash 命令的执行时间 ( 在 MDK , V5.06 O3 上测试 ) :

- 扇区擦除 : 19.921 ms
- 页面擦除 : 1.466 ms
- 页面写入 : 1.526 ms

## 2.1.2 I2C 总线功能

在本应用笔记中，LPC802 具有一个 I2C 总线接口，该接口用于从属模式。I2C 总线接口具有以下特点：

- 独立的主、从和监控功能
- 支持多主机和具有从机功能的多主机

- 硬件支持多个 I2C 总线从机地址
- 一个从机地址可以有选择地限定位掩码或地址范围

在此应用笔记中，I2C0 被配置为从机模式，并用作 EEPROM 器件的通信接口。

## 2.2 硬件连接

硬件连接很简单，仅使用两个引脚。

- I2C\_SDA: P0\_10
- I2C\_SCL: P0\_16

I2C 总线需要两个 4.7 kΩ 上拉电阻用于 SDA 和 SCL 线路。

## 3 软件

### 3.1 IAP (应用程序编程)

本节提供了如何使用 LPC802 的闪存 IAP 功能的示例。

1. 使用寄存器 r0 中的字指针调用 IAP 例程，该指针指向包含命令代码和参数的内存 (RAM)。
2. 在地址 0x0F001FF1 中定义一个 IAP\_Call 函数指针。
3. 使用 IAP\_Call 设置 IAP 函数并将这些函数调用组织到更高级别的 API 中。

有关 IAP 命令的详细信息，请参阅 LPC802 用户手册中的第 4 章：LPC802 ISP 和 IAP。

#### 注意

在调用 IAP 时，可能会暂时禁用 Flash。最简单的解决方法是在执行 Flash IAP 调用时禁用中断。

声明 IAP 调用：

```
1  /* IAP Call */
2  struct sIAP
3  {    // IAP Structure
4      unsigned long cmd;           // Command
5      unsigned long par[4];       // Parameter
6      unsigned long stat;        // Status
7      unsigned long res[2];      // Result
8  }    IAP;
9  typedef void (*IAP_Entry) (unsigned long *cmd, unsigned long *stat);
10 #define IAP_Call ((IAP_Entry) 0x0F001FF1)
11 #define PAGE_SIZE    (64)
12 #define CCLK         (15000)
```

擦除页面功能：

```
13 uint8_t FLASH_ErasePage(uint32_t addr)
14 {
15     unsigned long n;
16     unsigned long page;
17
18     n = GetSecNum(addr);           // Get Sector Number
19
20     IAP.cmd    = 50;               // Prepare Sector for Erase
21     IAP.par[0] = n;               // Start Sector
22     IAP.par[1] = n;               // End Sector
23     __disable_irq();
```

```

24     IAP_Call (&IAP.cmd, &IAP.stat);    // Call IAP Command
25     __enable_irq();
26     if (IAP.stat) return (1);          // Command Failed
27
28     page = addr/PAGE_SIZE;
29
30     IAP.cmd    = 59;
31     IAP.par[0] = page;
32     IAP.par[1] = page;
33     IAP.par[2] = CCLK;
34     __disable_irq();
35     IAP_Call (&IAP.cmd, &IAP.stat);
36     __enable_irq();
37     if (IAP.stat) return (1);
38
39     return (0);
40 }
41

```

写入页面功能：

```

42     uint8_t FLASH_WritePage(uint32_t addr, const uint8_t *buf)
43     {
44         unsigned long n;
45
46         n = GetSecNum(addr);            // Get Sector Number
47         IAP.cmd = 50;                   // Prepare Sector for Write
48         IAP.par[0] = n;                 // Start Sector
49         IAP.par[1] = n;                 // End Sector
50         __disable_irq();
51         IAP_Call (&IAP.cmd, &IAP.stat); // Call IAP Command
52         __enable_irq();
53         if (IAP.stat) return (1);       // Command Failed
54
55         IAP.cmd = 51;                   // Copy RAM to Flash
56         IAP.par[0] = addr;               // Destination Flash Address
57         IAP.par[1] = (unsigned long)buf; // Source RAM Address
58         IAP.par[2] = PAGE_SIZE;         // Fixed Page Size
59         IAP.par[3] = CCLK;              // CCLK in kHz
60         __disable_irq();
61         IAP_Call (&IAP.cmd, &IAP.stat); // Call IAP Command
62         __enable_irq();
63         if (IAP.stat) return (1);       // Command Failed
64
65         return 0;
66     }

```

## 3.2 I<sup>2</sup>C 总线接口编程

在此应用中，I<sup>2</sup>C 总线被配置为从功能。要使用 I<sup>2</sup>C 总线模块，必须在使用它之前执行正确的初始化步骤。它包括时钟门控，时钟路由，引脚 MUX 等。以下代码段显示了如何初始化 I<sup>2</sup>C 总线并启用 I<sup>2</sup>C 总线中断：

```

67     void app_i2c_slave_init(uint8_t slv_addr)
68     {
69         /* pin mux */
70         ConfigSWM(I2C0_SDA, P0_10);
71         ConfigSWM(I2C0_SCL, P0_16);
72
73         /* using main clock */

```

```

74     LPC_SYSCON->I2C0CLKSEL = 1;
75
76     /* give I2C a reset */
77     LPC_SYSCON->PRESETCTRL[0] &= (I2C0_RST_N);
78     LPC_SYSCON->PRESETCTRL[0] |= ~(I2C0_RST_N);
79
80     LPC_I2C0->DIV = 2;
81     LPC_I2C0->CFG = CFG_MSTENA | CFG_SLVENA;
82
83     LPC_I2C0->SLVADR0 = (slv_addr << 1) | 0;
84
85     // Enable the I2C0 slave pending interrupt
86     LPC_I2C0->INTENSET = STAT_SLVPEND | STAT_SLVDESEL;
87     NVIC_EnableIRQ(I2C0_IRQn); 88     }

```

I<sup>2</sup>C 总线从机操作通过软件中断处理完成。使用两个主要的 I<sup>2</sup>C 总线中断源：

- SLVPENDING：指示从机功能正在等待继续在 I<sup>2</sup>C 总线上进行通信，并且需要软件中断。
- SLVDESEL：发生停止条件或总线上的新地址与当前从机地址不匹配。

在 I<sup>2</sup>C 总线中断中，软件应检查中断状态寄存器。发生 STOP 条件时，将生成 SLVDESEL 中断。软件将消息发布到主线程以处理更多操作，例如将接收到的数据写入 NVM。生成 SLVPENDING 时，软件应检查 I2C0->STAT 寄存器中 SLVSTATE 中的状态码，以确定下一步操作。

- SLVSTATE = 0x00 (已接收并匹配从机地址)

软件应通过读取 SLVDATA 来记录地址，并确定是读操作还是写操作。

- SLVSTATE = 0x01 (从机接收到一个新字节)

软件应读取 SLVDATA 以获取传输的数据并存储在 RAM 中。然后将 SLVCTRL 寄存器中的 CTL\_SLVCONTINUE 位置 1，以使 I<sup>2</sup>C 总线硬件继续处理总线事务。

- SLVSTATE = 0x02 (从机需要向主站发送一个新字节)

软件应将数据写入 SLVDATA，然后将 SLVCTRL 寄存器中的 CTL\_SLVCONTINUE 位置 1，以使 I<sup>2</sup>C 总线硬件继续处理总线转换。

### 3.3 通信协议

R/W 时序与市场其他 EEPROM 器件类似，但只支持顺序读写。

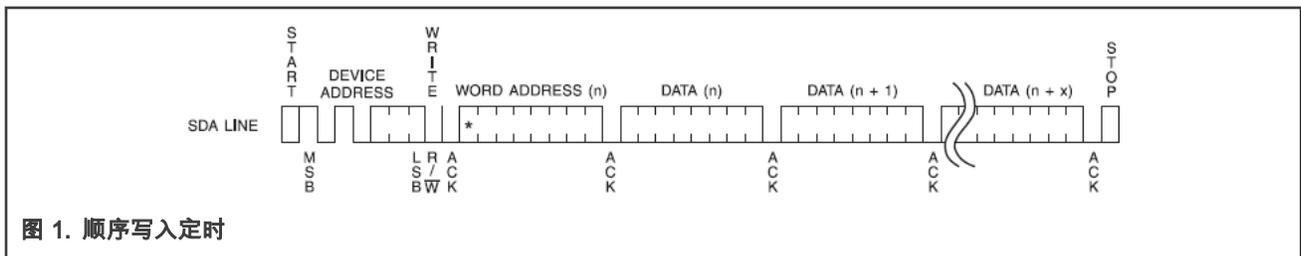
详细的通信协议如下：

- 顺序写入

对于写入操作，从机写入地址发出后需要两个字节的地址。这两个字节从内存中的 65535 个字节中选择一个，先传输低位字节。

格式为：

CHIP\_ADDR (7 位 + W (1)) + DATA\_ADDR\_L + DATA\_ADDR\_H + DATA (0) + ... + DATA(N)



- 顺序读取

顺序读取操作与写入操作类似，但是需要一个 I<sup>2</sup>C 总线重启信号和从机读取地址。主机现在用确认响应，表示它需要额外的数据。设备继续为接收到的每个应答输出数据。

格式为：

CHIP\_ADDR (7 bits + W (0)) + DATA\_ADDR\_L + DATA\_ADDR\_H + CHIP\_ADDR (7 bits + R (1)) + DATA(0) + ... + DATA(N)

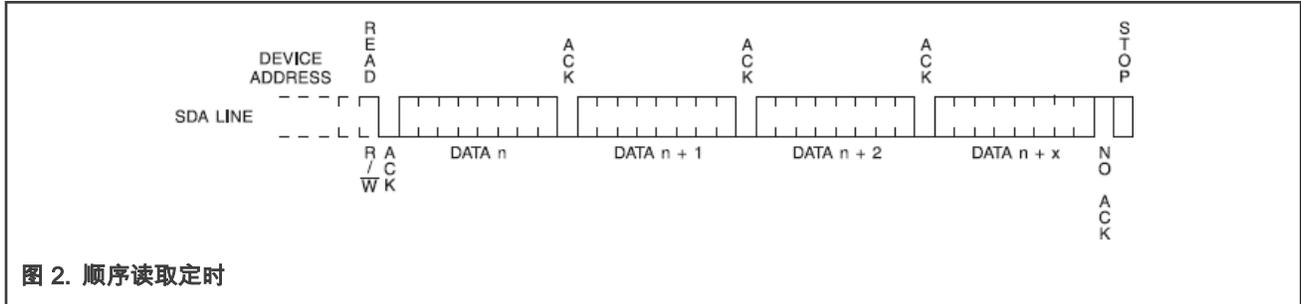


图 2. 顺序读取定时

对于顺序写入，在接收到每个数据字节时，双字节地址在内部递增。当地址到达页面边界时，以下字节将被丢弃。因此，写操作的最大数据长度是页面长度（64 字节）。

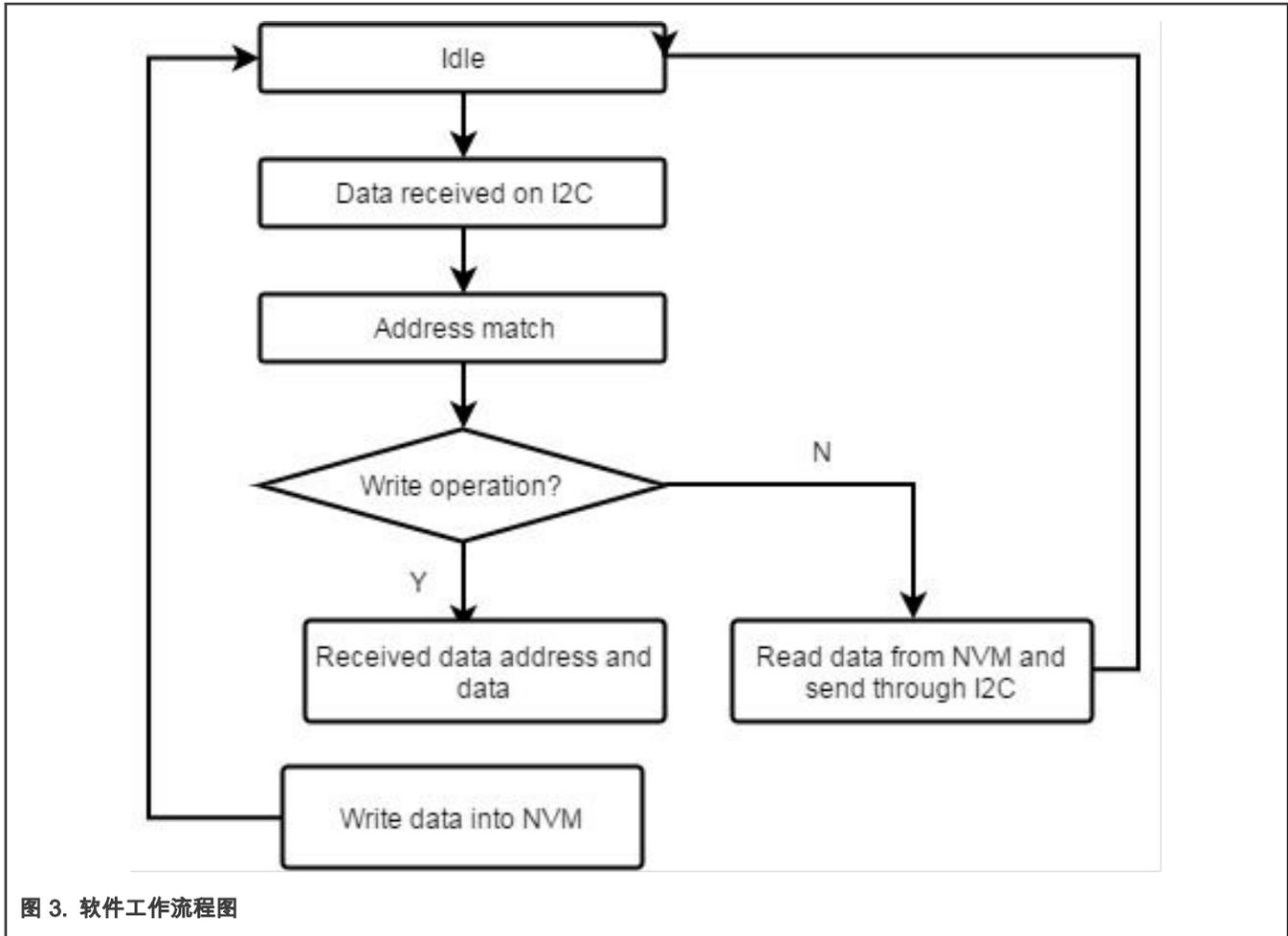
### 3.4 总结

这个 demo 使用 LPC802 上从 0X1000 到 0X4000 的闪存地址作为 EEPROM 存储器。前 4kB 的闪存保留给固件本身。因此，总共有 12 kB 的 flash 可用作 EEPROM 存储器。

对于从机地址，选择 0X50 是为了与市场上当前的 EEPROM 兼容。

对于软件工作流程，使用典型的前台和后台系统。仅允许 I<sup>2</sup>C 总线中断。软件等待 I<sup>2</sup>C 总线转换，一旦 I<sup>2</sup>C 总线中断产生，软件处理上半部分进程，并将消息推送到主线程以执行下半部分任务。主线程处理诸如将数据写入闪存之类的任务。

图 3 显示了软件工作流程：

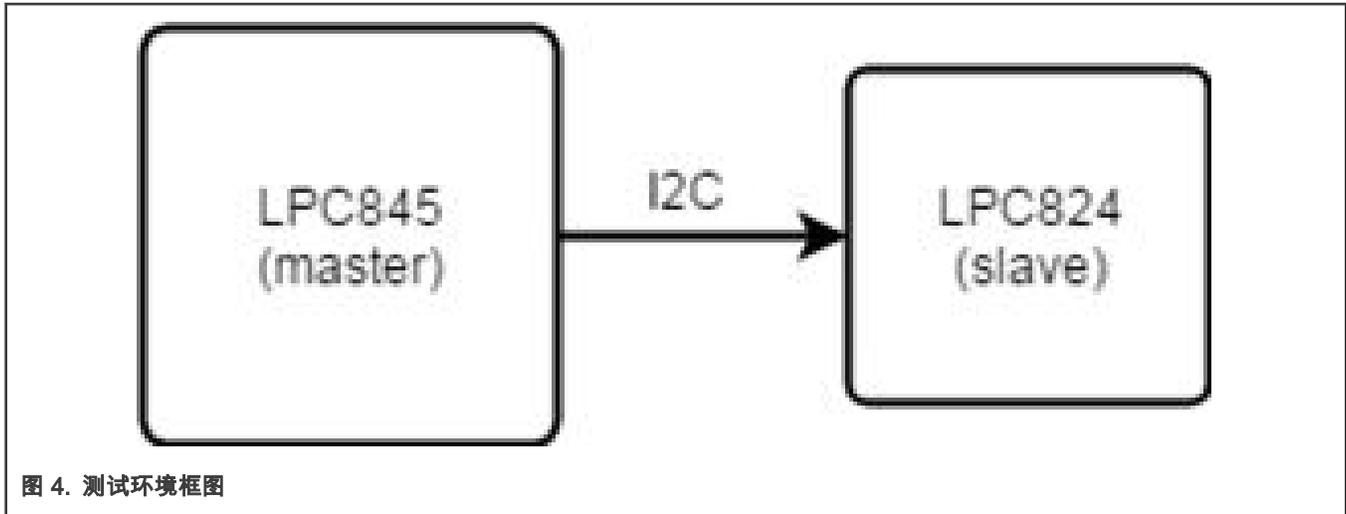


## 4 测试与结果

### 4.1 环境设置

在本节中，将使用 LPC845 作为主机构建测试环境，以通过预定义的 I<sup>2</sup>C 总线接口读取/写入 LPC802。详情请见 [I<sup>2</sup>C 总线接口编程](#)。

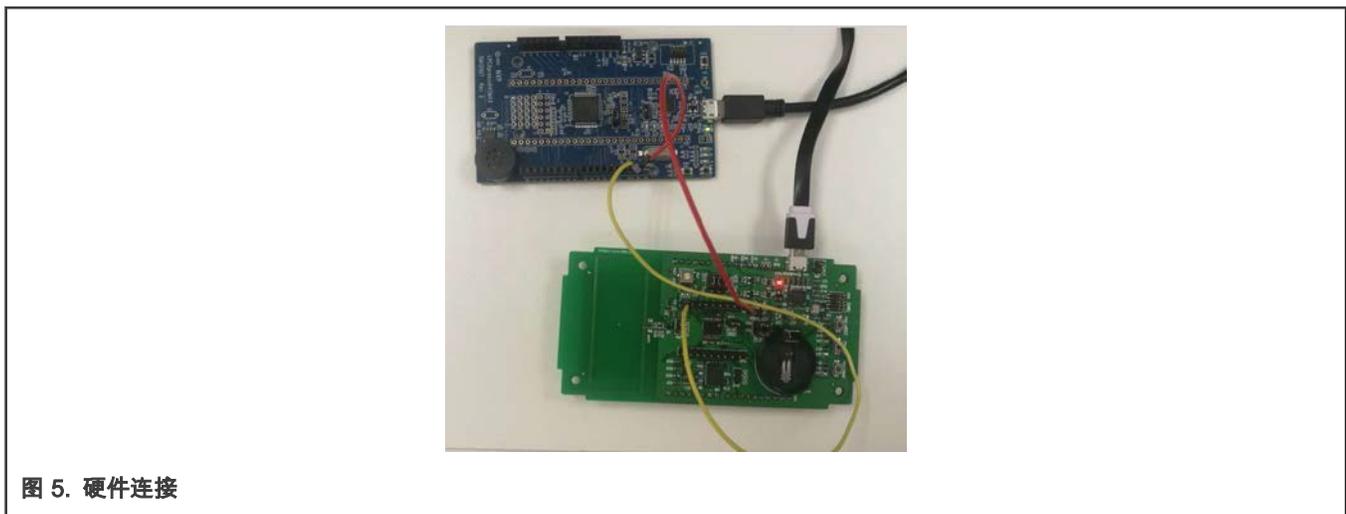
图 4 显示了测试环境框图。



## 4.2 硬件和连接

- 主机：LPC845 XpressoMAX 板
  - I<sup>2</sup>C\_SDA: P0\_11
  - I<sup>2</sup>C\_SCL: P0\_10
- 从机：LPC802 演示板
  - I<sup>2</sup>C\_SDA: P0\_10
  - I<sup>2</sup>C\_SCL: P0\_16

图 5 显示硬件连接。



## 4.3 测试步骤

- 按照[硬件和连接](#)所述，准备并连接硬件。
- 将固件下载到每个板上。
  - 对于主演示项目（LPC845）：请在“lpc845\_EEPROM\_master.zip”下编译项目，并将镜像下载到 LPC845 板上，或将“/binary/lpc845\_EEPROM\_master\_demo.bin”下的预编译镜像下载到 LPC845 中。

— 对于从固件项目 (LPC802) : 请在“lpc845\_EEPROM\_master”下编译项目, 然后将镜像下载到 LPC845 板上, 或将“binary/lpc802\_EEPROM\_firmware.bin”下的预编译镜像下载到 LPC802 中。

- 打开 LPC845 Xpresso 板上的串行终端, 将波特率设置为 115200-N-8-N-1。输出见 图 6。

```
I2C master test for lpc802 eeporm demo CoreClock:30000000Hz
chip test begin...
chip test finish...
write page(64byte) time:6056 us
read page(64byte) time:2499 us
```

图 6. 测试记录

### 4.4 测试结果

使用外部 MCU 进行测试以发送测试顺序读/写命令。

**注意**  
当前固件仅支持顺序读/写命令。见通信协议。

图 7 和 图 8 显示了典型的读写时序 :

- 四个字节的顺序读取操作时序 :



图 7. 顺序读取时序

- 双字节顺序写入操作时序 :

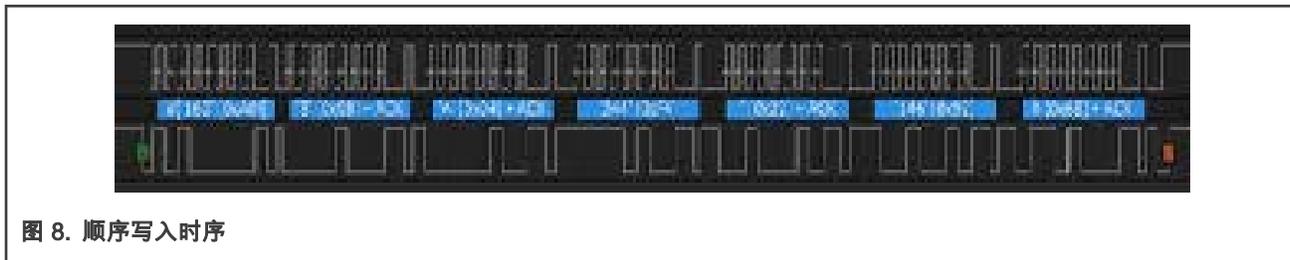


图 8. 顺序写入时序

### 4.5 性能

测试条件为 :

- 测试平台 : LPC845 Xpresso 板
- LPC845 核心时钟 : 30 MHz
- I<sup>2</sup>C 总线时钟 : 373 kHz
- LPC802 核心时钟 : 15 MHz

主机测试项目的固件方案和优化设置对性能影响不大。这是因为, 读/写的大部分时间都花在 I<sup>2</sup>C 总线传输和闪存擦除/写入操作上。

- 写一页 ( 64 字节 )

— 时间 6.074 ms

- 此时间是 I<sup>2</sup>C 总线传输时间加上 LPC802 内部闪存页面擦除时间和页面写入时间之和。有关闪存擦除和写入时间，请参阅 [I<sup>2</sup>C 总线功能](#)。
- 总线传输时间取决于 I<sup>2</sup>C 总线时钟速度和传输大小。在这种情况下，一页的传输需要 64 (数据) +1 (芯片地址) +2 (数据地址) =67 字节。

- 读取一页 (64 字节)

— 时间 : 2.504 ms

- 该值几乎等于 I<sup>2</sup>C 总线传输时间，因为 LPC802 从其内部闪存获取数据并将其写入 I<sup>2</sup>C 总线只需几微秒。

## 5 结论

本应用笔记主要讨论以下主题：

- LPC802 闪存功能：包括闪存分区、IAP 使用和示例代码。
- LPC802 I<sup>2</sup>C 总线特性、从模式、有关如何编写与 I<sup>2</sup>C 总线模块协同工作的软件以及如何处理 I<sup>2</sup>C 总线转换的知识。
- 使用 LPC802 作为 I<sup>2</sup>C 总线 EEPROM 的演示软件，以演示上述两个功能。
- 基于 LPC845 Xpresso 板的测试演示，该板充当 LPC802 的 R/W NVM 的 I<sup>2</sup>C 总线主设备，还提供源代码和项目文件。

使用这个 demo 可以使得 LPC802 成为更智能的类似 EEPROM 的设备，例如添加 UART/SPI 接口，内部处理页面 Flash 边界问题以及添加 I/O 扩展器功能。

**How To Reach Us**

**Home Page:**

[nxp.com](http://nxp.com)

**Web Support:**

[nxp.com/support](http://nxp.com/support)

**Limited warranty and liability** — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

**Right to make changes** - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

**Security** — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at [PSIRT@nxp.com](mailto:PSIRT@nxp.com)) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE, VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

© NXP B.V. 2018-2021.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 2018 年 2 月 20 日

Document identifier: AN12125

