计算掉电時储存512字节数据所需时间和电容值

Application Note for 32-bit NuMicro® Family

Document Information

Abstract	此份文件介绍如何在掉电过程中储存512字节的数据，并使用欠压检测(BOD)功能及外挂电容于V_{DD}管脚上，利用延长放电时间使所需数据能顺利储存在数据内存里，内容包含原理介绍、数据分析以及使用建议。附录提供范例程序以及函式库。
Apply to	NuMicro® Cortex®-M0/M4全系列，本文以NuMicro® Mini51 DE系列为例。

目录

1 简介 3

2 掉电储存分析 4

2.1 机制原理 4

2.2 BOD电压选取 5

2.3 电容值选取 5

2.3.1减少所需电容 5
2.3.2计算所需电容 6
2.3.3实验 6

3 结论 17

4 附录：范例程序代码 18

4.1 库函数 19

4.1.1SYS_Init(); 19
4.1.2BOD_IRQHandler(); 19

4.2 范例程序代码 20

4.2.1掉电储存512字节的数据测试代码 20

5 版本历史 22

1 简介

在许多产品设计上都希望有数据保存的功能，在关机时将系统执行中的部分数据储存下来，例如储存目前模式的设定以及其调整的参数，以便下次开机可用此数据来回复系统为掉电前的状态。

但在产品使用上总会有突然停电或断电的情况，若突然断电可能来不及储存所需的参数，造成系统重新上电后必须重新设定模式或参数。因此我们利用欠压检测 (BOD)功能侦测系统是否掉电与V_{DD}外挂电容来延长掉电的时间，并将系统所需之数据储存于数据内存中，即可于掉电期间完成储存数据的动作，使系统再次上电后依然可以维持为掉电前的状态。

本文章着重于使用BOD进行电压侦测时，对所需电容值进行分析，并提供范例程序，利于计算出测量储存数据所需时间，以及对应的电容值。

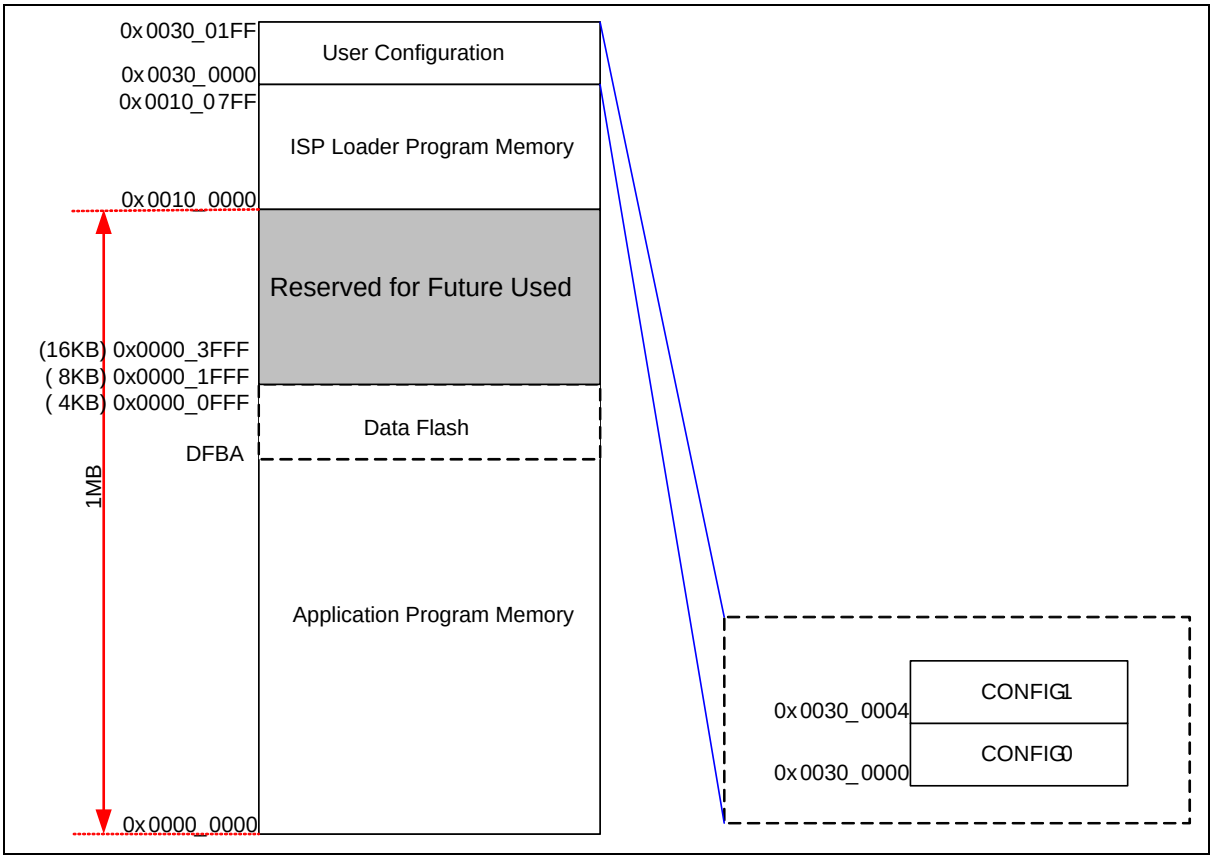


图 1-1 内存结构图

2 掉电储存分析

本节将介绍，如何计算储存512字节数据的机制原理，以及储存耗费的时间，并且依据Mini51的规格书提供之数据，计算掉电后至储存完512字节数据所需外挂电容值。

2.1 机制原理

开启欠压检测 (BOD) 功能，侦测电源是否为掉电状态，进入BOD中断后即开始对数据内存进行数据写入，若储存时间足够，则可以在工作电压之内完成写入。

为使系统掉电后有足够时间进行储存，必须在V_{DD}上外接电容，使放电的时间延长至足以储存完数据。

■ 掉电侦测

NuMicro® Cortex®-M0/M4全系列皆提供欠压检测 (BOD) 的功能可以用来侦测是否为掉电，Mini51的BOD可供选择的电压为 4.4 V / 3.7 V / 2.7 V / 2.2 V，我们可依据系统电压来选择，判定为掉电的电压为多少，当侦测到掉电时，即可开始储存数据至数据内存的工作。

■ 外挂电容延长掉电后可储存数据时间

由于储存数据至数据内存需要时间，Mini51保证的工作电压在2.5 V ~ 5.5 V，当系统电压下降过快低于2.5V时，其放电时间可能无法储存完需要的数据；当低于2V时，则会产生低电压复位(LVR)，我们可以透过外加电容推迟放电的速度，来改善储存时间不足的情况。放电状况比较图如下：

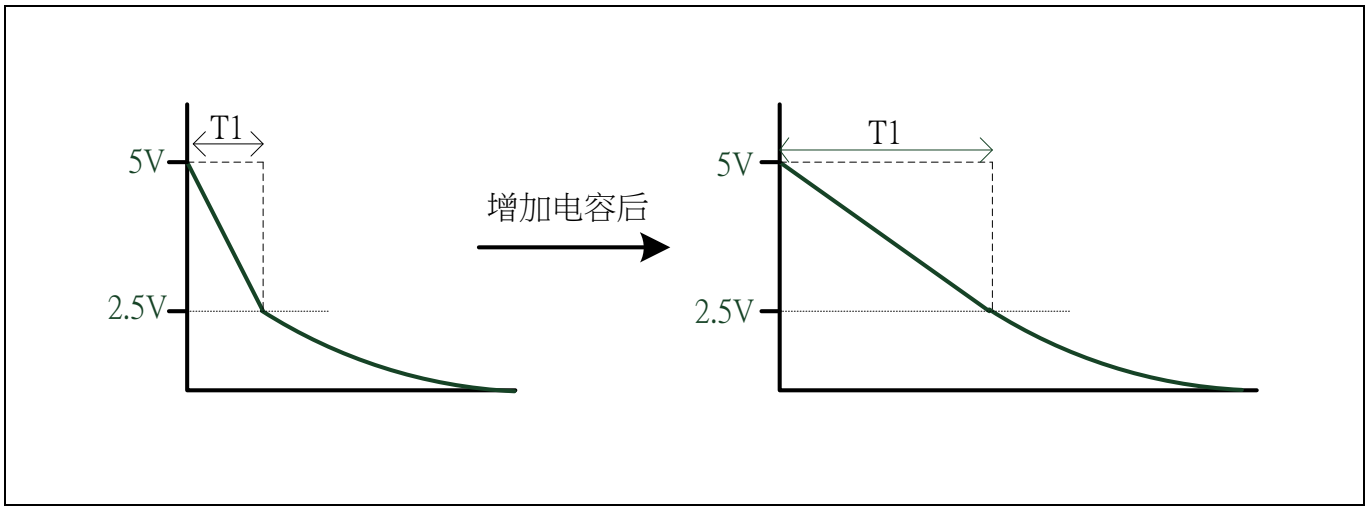


图2-1 利用电容改变可储存数据时间

2.2 BOD 电压选取

由于BOD的侦测电压为4.4 V / 3.7 V / 2.7 V / 2.2V，若系统电压为5V，通常BOD选用4.4V来侦测是否为掉电状态，若BOD选用3.7V来侦测，则储存数据的时间会更少。

2.3 电容值选取

本节介绍如何藉由实验来计算储存完512字节数据所需之外挂电容值

2.3.1 减少所需电容

由Mini51的规格书所提供的表格 2-1可得到，页擦除(Page)耗时20ms，写一个字(Word)耗时60 us，计得写一页512字节需耗时 7.68 ms，所以总计擦除并写一页至少需要约30 ms 的时间，才能保证系统在掉电过程中，将数据储存至数据内存。

由于页擦除的时间较长，若是在系统掉电后才执行页擦除动作，则须要更大的电容。因此当系统上电后先进行页擦除，掉电时就只需要执行写入动作，可大幅减少所需时间与电容量。

符号	参数	最小值	典型值	最大值	单位	条件
V_{FLA}[2]	电源	1.62	1.8	1.98	V	T_A =25℃
N_{ENDUR}	擦写次数	20,000	-	-	cycles[1]	
T_{RET}	数据保存	100	-	-	year	
T_{ERASE}	页擦除时间	-	20	-	ms	
T_{PROG}	编程时间	-	60	-	us	
I_{DD1}	读电流	-	6	-	mA	
I_{DD2}	编程电流	-	8	-	mA	
I_{DD3}	擦除电流	-	12	-	mA	

表格 2-1 Mini51 Flash DC 电气特性

2.3.2 计算所需电容

由于Mini51的保证工作电压为2.5 V ~ 5.5 V，所以为了留有多的余量，我们希望可以在2.7 V以上时，完成储存512字节数据的动作，所需的电容值，与系统电压及耗电流有关，可以透过电容公式 (Q=CV=It) 计算出来，计算过程如下:

Q：电荷量、C：外挂电容量、V：所选BOD电压减2.7V、I：系统电流、t：储存512字节所需时间

假设BOD选取4.4 V，则 V = 4.4 V - 2.7 V = 1.7 V

依表格 2-1可得编程电流为 8 mA，则 I = 8 mA、

依表格 2-1可得一个字(Word)的编程时间为 60 us，则 t = 60 × 128 = 7.68 ms

由 It = CV → 8 mA × 7.68 ms = 1.7 V × C

故 C = 36.14 uF

所以我们可以选择最接近且常用的47 uF作为外挂电容，就可以在掉电时储存完512字节数据，若有其他负载，可以参考 2.3.3.3 或 2.3.3.4 的计算方法

2.3.3 实验

如表格 2-2，我们将进行五个实验，来观察外挂电容值大小，是否能够成功在掉电时储存完512字节数据，实验电路板为图2-2新唐官方所出品的NuTiny-Mini51L与NuTiny-Nano102S，并移除不需要的负载(Power LED)，实验步骤如下， P0.1用于标示流程进度：

1. 电源供电用示波器量测 V_{DD} 电压 (CH1)，及显示状态用的 P0.1 (CH2，预设输为 1)
2. 当电源断开时，MCU 尚在工作，BOD 侦测电压值是否低于设定值，当符合即进入中断
3. 进入中断后，将 P0.1 设为 0，并且开始进行 512 字节的数据写入动作
4. 当 512 字节的数据写入完成后，将 P0.1 设为 1 并离开中断
5. 当 P0.1 于 V_{DD} 下降到 2.7V 前完成 1 → 0 → 1，即代表所选取外挂电容的放电时间可以顺利储存完 512 字节数据

	型号	工作电压	BOD电压	外挂电容	LED负载	进中断关LED	储存成功
实验一	Mini51	5V	4.4V	10 uF	-	-	失败
实验二	Mini51	5V	4.4V	47 uF	-	-	成功
实验三	Mini51	5V	4.4V	100 uF	有	-	成功
实验四	Mini51	5V	4.4V	47 uF	有	有	成功
实验五	Nano102	5V	4.4V	47 uF	有	有	成功

表格 2-2 实验条件

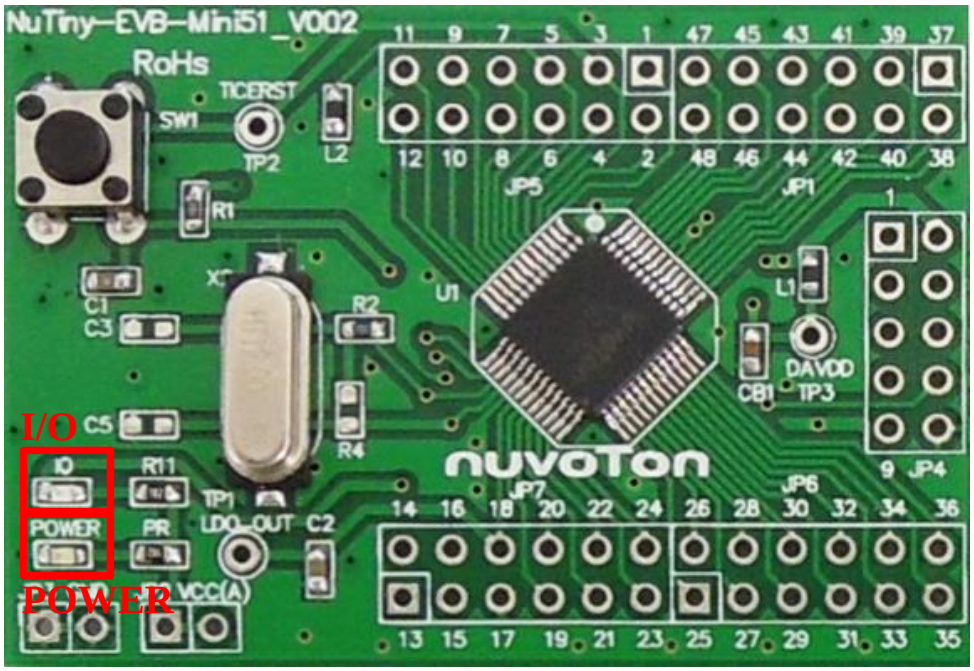
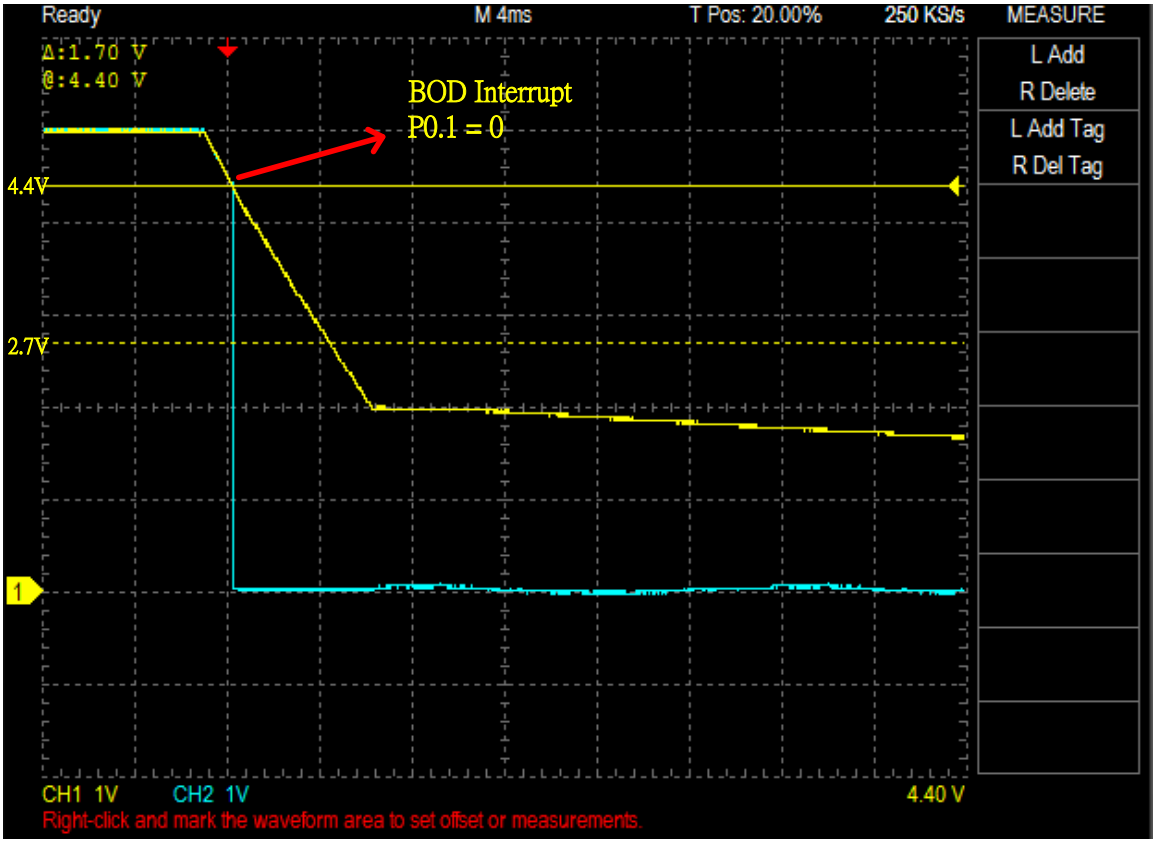


图 2-2 NuTiny-Mini51L

2.3.3.1 实验一：IC：Mini51、工作电压：5V、BOD：4.4V、电容：10 uF

如图2-3，外挂电容10 uF，由于掉电后 P0.1 并没有于V_{DD}下降到 2.7 V 前完成1 → 0 → 1，所以掉电后无法储存完512字节数据。



CH1 : VDD　CH2 : P0.1

图 2-3 V_{DD} 增加 10 uF 电容掉电波形

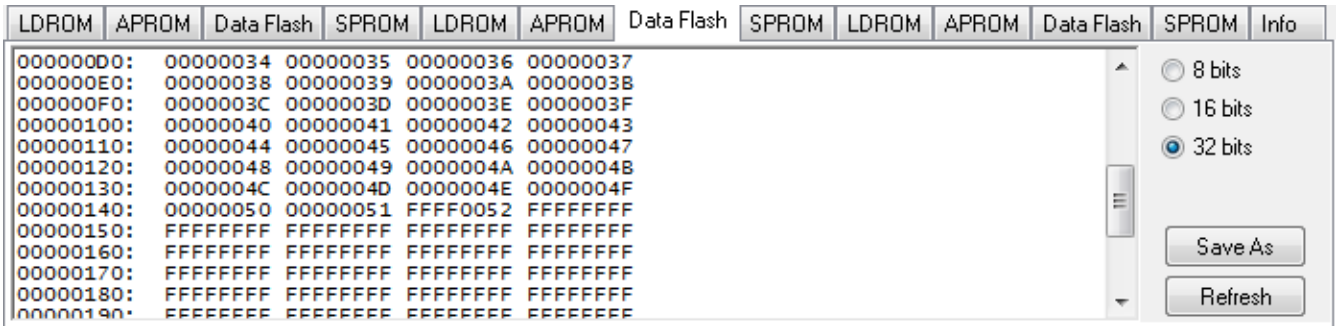
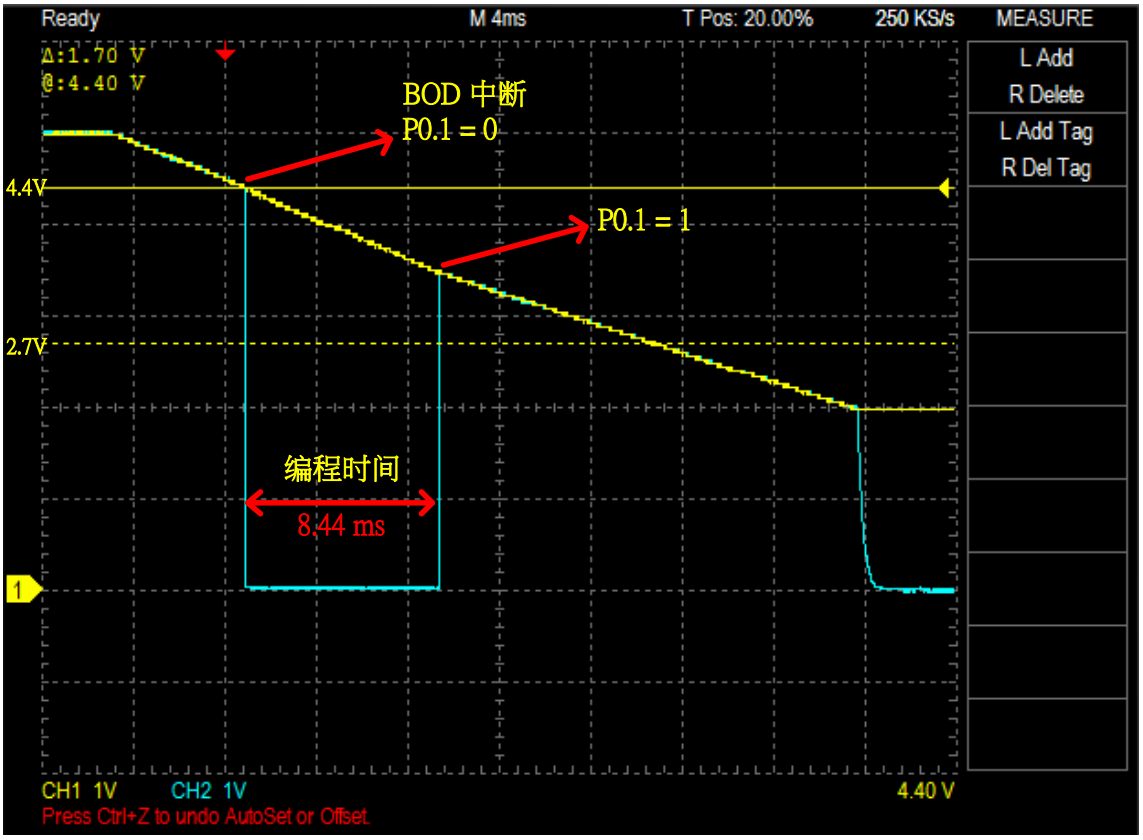


图 2-4 V_{DD} 增加 10 uF 电容储存数据失败

2.3.3.2 实验二：IC：Mini51、工作电压：5V、BOD：4.4V、电容：47 uF

由2.3.2所计算出之电容C = 36.14 uF，我们选择常用的47 uF来做为外挂电容，可以看到波形掉电后P0.1于V_{DD}下降到2.7 V前完成 1 → 0 → 1，所以掉电后可以储存完512字节数据。

另外我们可以看到编程时间是8.44 ms，与表格 2-1所计算出来的7.68 ms有些差距，是因为除了数据内存的的编成时间之外还要再加上程序执行的时间，但由于我们选取的电容通常都大于计算出的电容，因此我们在计算所需电容值的时候，可以忽略此部分。



CH1 : VDD　CH2 : P0.1

图 2-5V_{DD}增加 47 uF 电容掉电波形

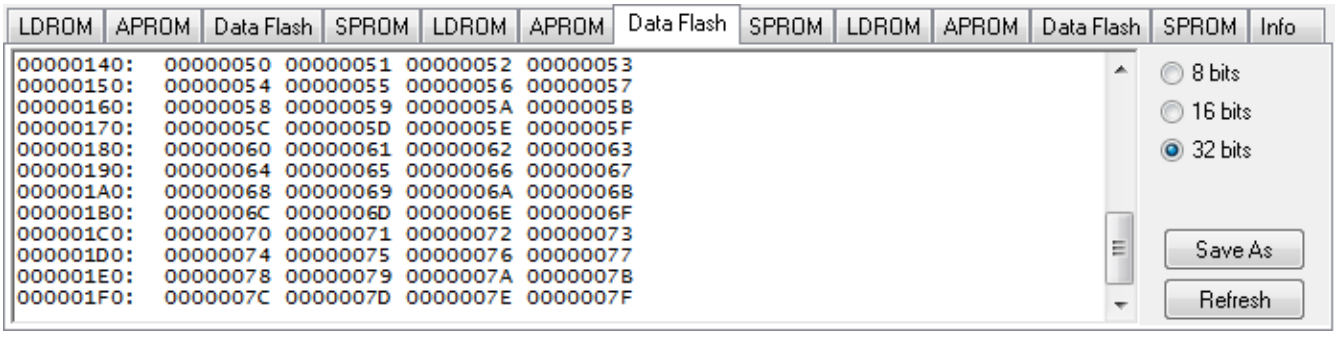


图 2-6 V_{DD}增加 47 uF 电容储存数据成功

2.3.3.3 实验三：IC：Mini51、工作电压：5V、BOD：4.4V、电容：100 uF、并加LED负载

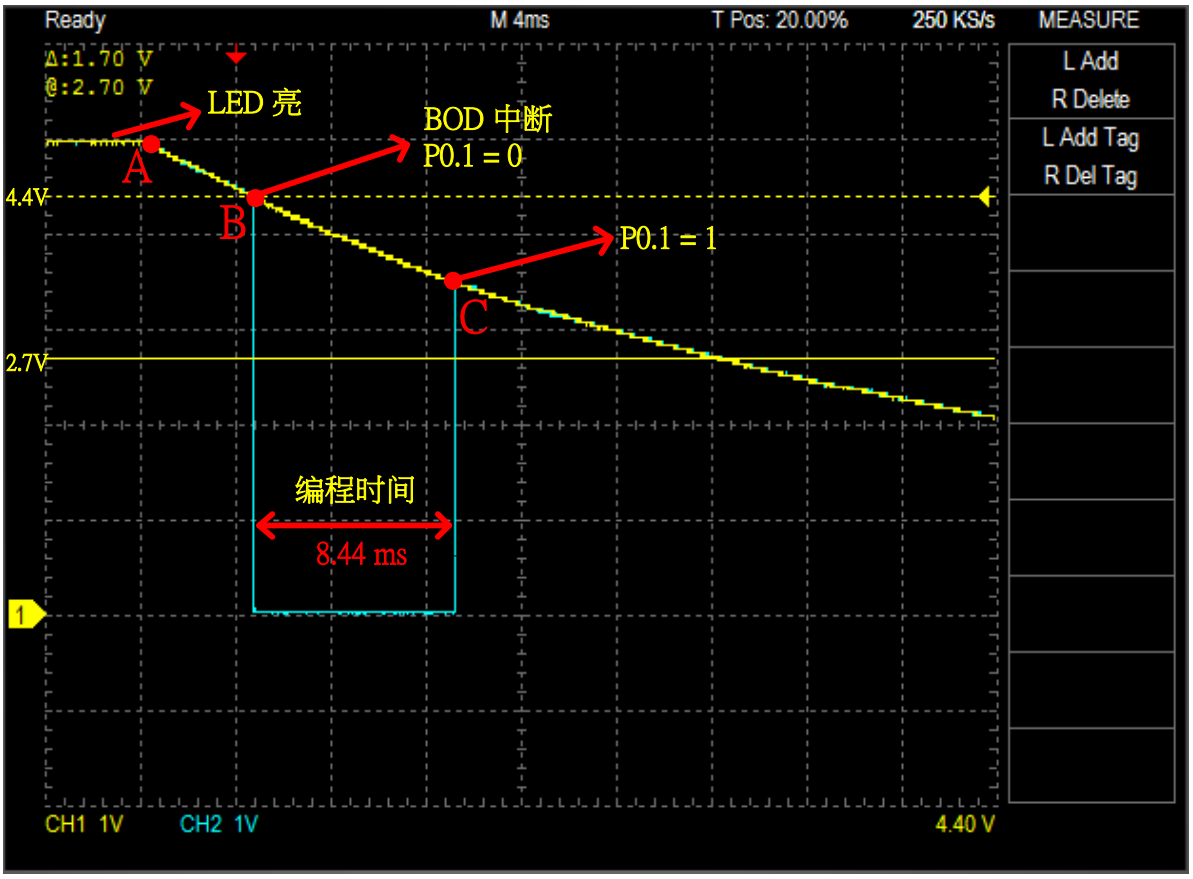
本次实验点亮如图2-2的IO LED作为负载，由于增加LED作为负载，所以电流已经不能由表格2-1的编程电流来计算，必须先量测系统电流，再透过电容公式 (Q = CV = It) 计算出来，计算过程如下:

Q：电荷量、C：外挂电容量、V：所选BOD电压减2.7V、I：系统电流、t：储存512字节所需时间

假设BOD选取4.4V，则 V = 4.4V - 2.7V = 1.7 V、
如图2-8，使用电表量测包含编程与LED负载的系统电流为13 mA，则 I = 13 mA、
依表格 2-1可得写一个字(Word)的编程时间为 60 us，则 t = 60 × 128 = 7.68 ms

由 It = CV → 13 mA × 7.68 ms = 1.7 V × C，得到 C = 58.72 uF

所以我们只能选择最接近且常用的100 uF作为外挂电容，用于完成掉电时储存512字节数据。



CH1 : VDD　CH2 : P0.1

图 2-7 V_{DD} 增加 100 uF 电容与增加 LED 负载波型

图 2-8 加 LED 后的系统总耗电流

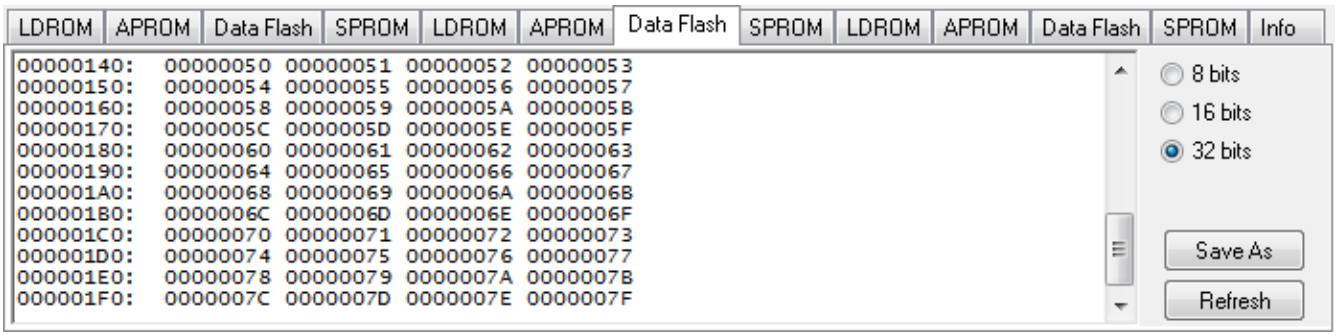


图 2-9 V_{DD} 增加 100uF 电容与增加 LED 负载，储存数据成功

2.3.3.4 实验四：IC：Mini51、工作电压：5V、BOD：4.4V、电容：47uF、加LED负载、BOD中断关LED

由于实验三掉电时持续点亮LED，造成所需的外挂电容增加至100 uF，所以本实验提出方法来减少所需的外挂电容。

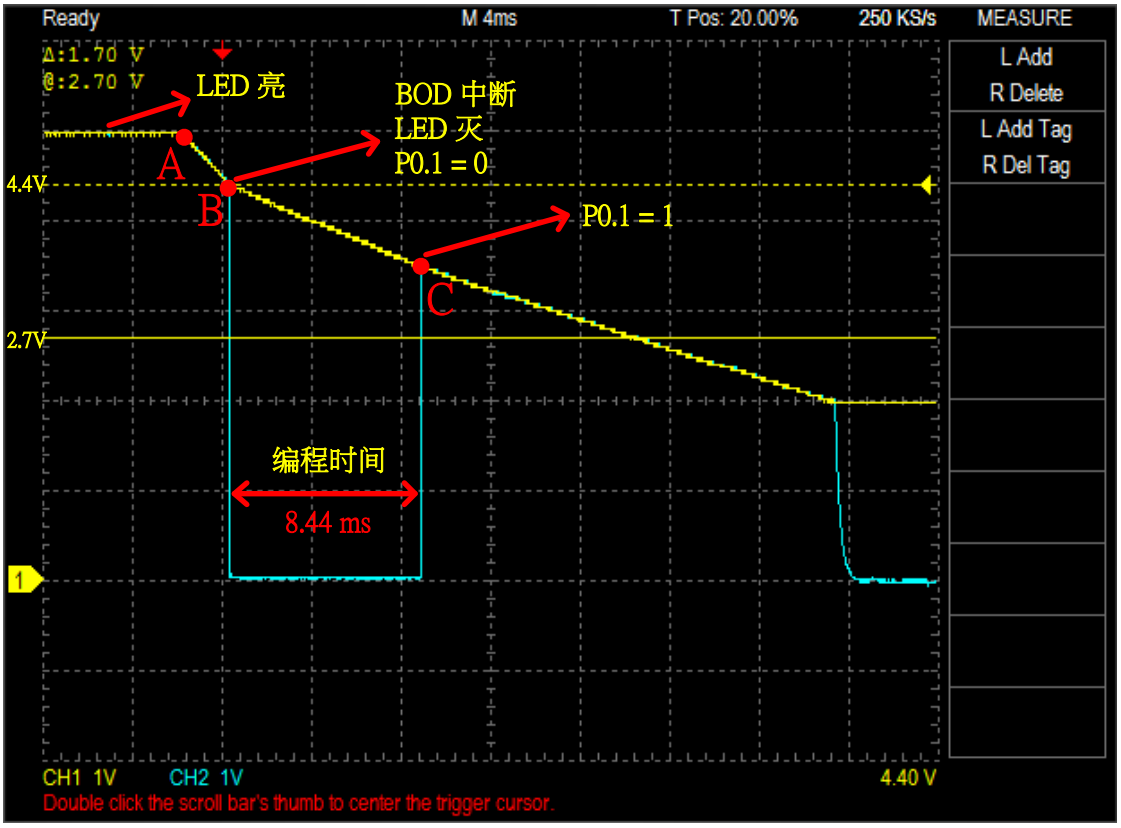
本次实验，同样的LED上电就点亮，量测电流与实验三同样是13 mA，但不同的是当侦测掉电时到进BOD中断，我们可以在中断子程序中关闭LED，再开始做储存数据的动作，所以在储存数据时的总耗电流就不含LED的电流，图2-10中可以观察到AB两点间与BC两点间的斜率不一样是因为关闭LED后电流下降所致，接着可以一样用表格 2-1的编程电流来计算。

Q：电荷量、C：外挂电容量、V：所选BOD电压减2.7V、I：系统电流、t：储存512字节所需时间

假设BOD选取4.4V，则 V = 4.4V - 2.7V = 1.7 V
依表格 2-1可得编程电流为 8 mA，则 I = 8 mA、
依表格 2-1可得写一个字(Word)的编程时间为 60 us，则 t = 60 × 128 = 7.68 ms

由 It = CV → 8 mA × 7.68 ms = 1.7 V × C，故 C = 36.14 uF

所以我们可以选择最接近且常用的47 uF作为外挂电容，用于掉电时储存完512字节数据。



CH1 : VDD　CH2 : P0.1

图 2-10 V_{DD} 增加 47uF 电容，掉电后关闭 LED 负载波型

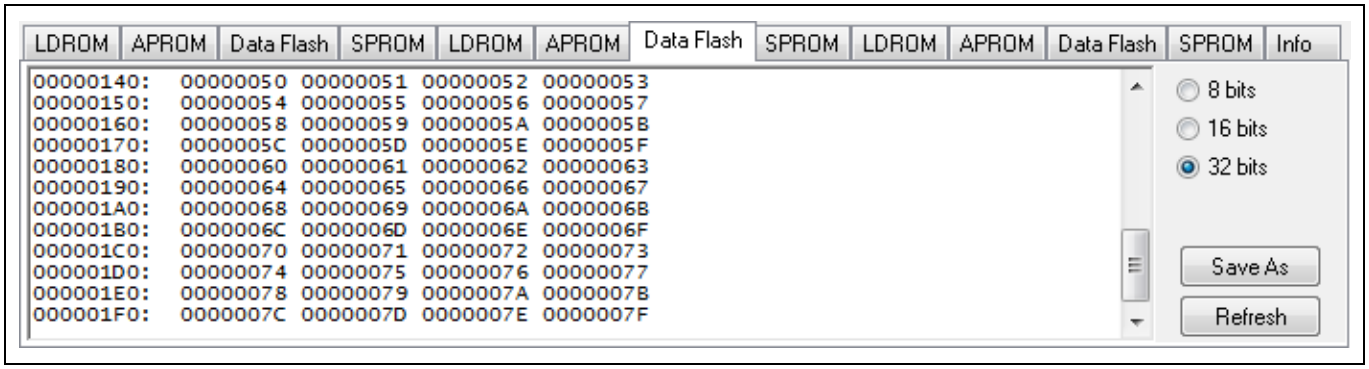


图 2-11 V_{DD} 增加 47uF 电容，掉电后关闭 LED 负载，储存数据成功

2.3.3.5 实验五：IC：Nano102、工作电压：3.3V、BOD：2.5V、电容：100uF、加LED负载、BOD中断关LED

本次实验使用NuTiny-Nano102S來做實驗，并移除不需要的负载(Power LED)，由于Nano102的保证工作电压为1.8 V ~ 3.6 V，所以为了留有多的余量，我们希望可以在2.0 V以上时，完成储存512字节数据的动作。

實驗同样的LED上电就点亮，当侦测掉电时到进BOD中断，即在中断子程序中关闭LED，再开始做储存数据的动作，接着可以一样用表格 2-3的编程电流来计算。

Q：电荷量、C：外挂电容量、V：所选BOD电压减2.0V、I：系统电流、t：储存512字节所需时间

假设BOD选取2.5V，则 V = 2.5V – 2.0V = 0.5 V
依表格 2-1可得编程电流为 7 mA，g则 I = 7 mA、
依表格 2-1可得写一个字(Word)的编程时间为 40 us，则 t = 40 × 128 = 5.12 ms

由 It = CV → 7 mA × 5.12 ms = 0.5 V × C，故 C = 71.68 uF

所以我们可以选择最接近且常用的100 uF作为外挂电容，用于掉电时储存完512字节数据。

符号	参数	最小值	典型值	最大值	单位	条件
V_{FLA}[2]	电源	1.62	1.8	1.98	V	T_A =25℃
N_{ENDUR}	擦写次数	20000	-	-	cycles[1]	
T_{RET}	数据保存	100	-	-	year	
T_{ERASE}	页擦除时间	-	20	-	ms	
T_{PROG}	编程时间	-	40	-	us	
I_{DD1}	读电流	-		0.135	mA	
I_{DD2}	编程电流	-		7	mA	
I_{DD3}	擦除电流	-		10	mA	

表格 2-3 Nano102 Flash DC 电气特性

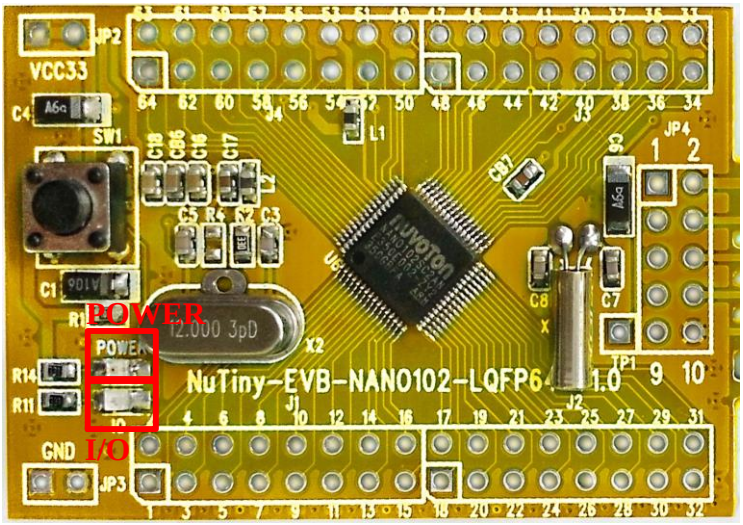
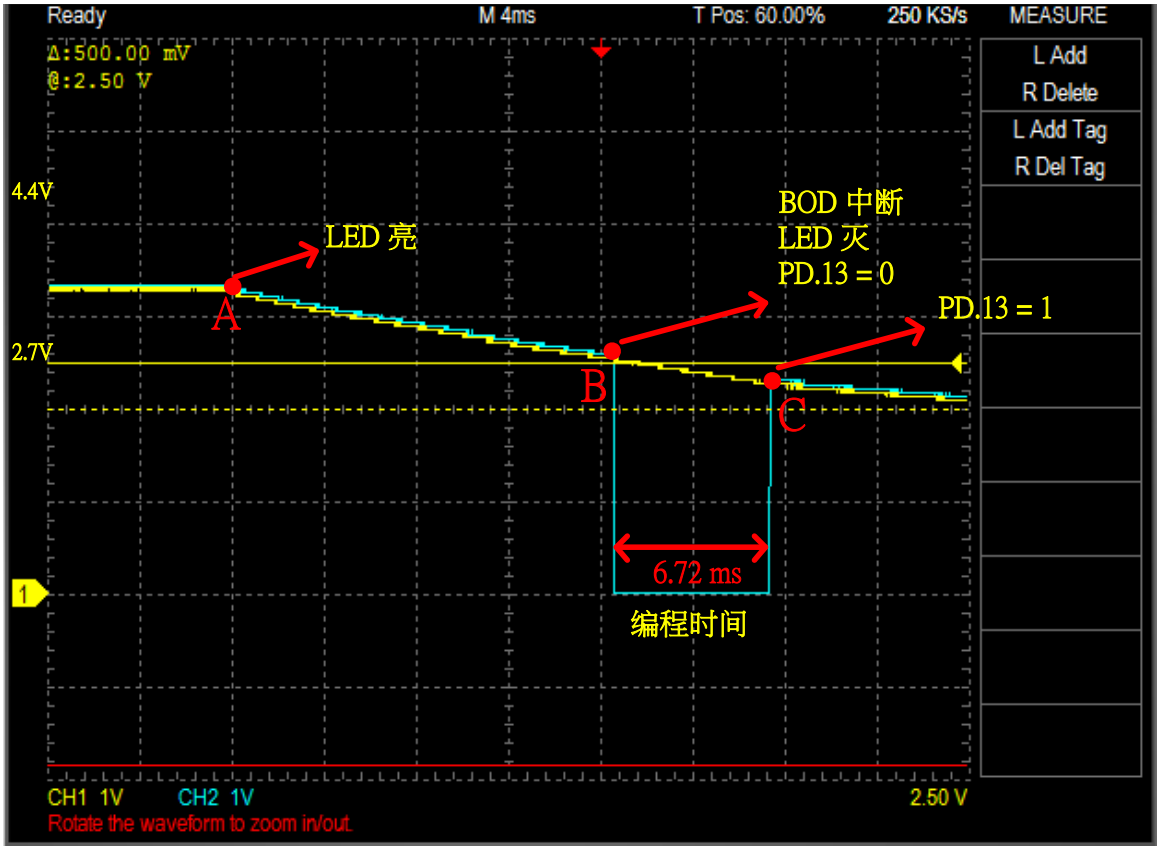


图 2-12 NuTiny-Nano102S



CH1 : VDD　CH2 : PD.13

图 2-13 V_{DD} 增加 100uF 电容，掉电后关闭 LED 负载波型

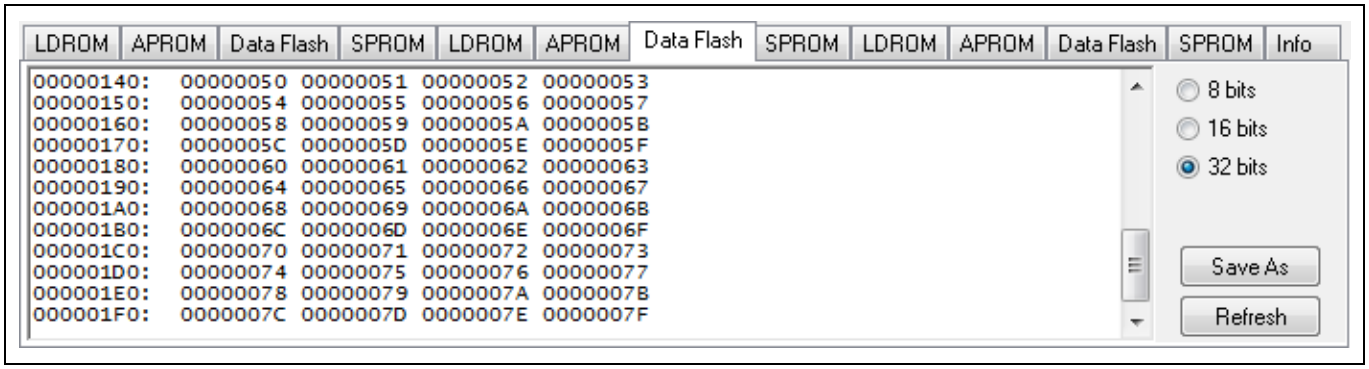


图2-14 V_{DD}增加100uF电容，掉电后关闭LED负载，储存数据成功

3 结论

本篇提出利用开启BOD功能与增加外挂电容，在系统掉电后能够储存512字节内容于数据内存的方法，以避免当系统突然断电，或电池耗尽时对系统造成之影响。

为了能够完整储存所需的数据于数据内存中，如2.3.3.2实验二，我们利用BOD电压设定与Mini51的规格书所提供的数据，来计算出所需之电容。

当系统因有其他负载时必须外挂较大之电容，我们可以如2.3.3.4实验四，于掉电时在BOD中断中关闭其他负载，以减少耗电流，来达成减少外挂电容量的目的，又或者利用如图3-1的二极管来隔绝其他装置消耗外挂电容与搭配GPIO中断来侦测电源是否为掉电。

對於其他系列的NuMicro® Cortex®-M0/M4系列，如2.3.3.5实验五提出使用Nano102也同样可以使用2.3.3.4实验四的計算方法，于掉电时在BOD中断中关闭其他负载後，使用規格書來計算所需電容量。

由于本篇提出储存数据之储存量为512字节数据量，若不需储存如此多资料，则可以依公式计算，以利再减少所需之电容量。

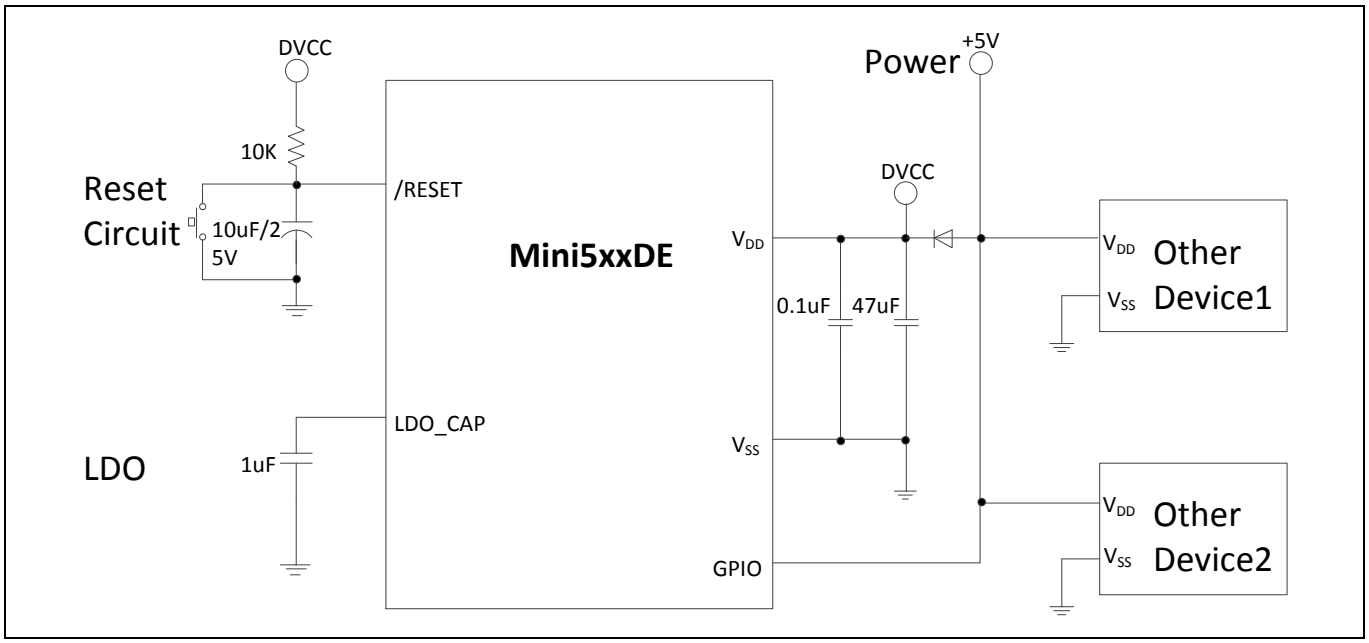


图 3-1 隔离外部装置消耗电容参考电路

4 附录：范例程序代码

如图 4-1，在使用此范例程序前，需要先设定好数据内存；4.1提供函式库的程序代码，提供使用此功能需增加的程序代码；4.2提供范例程序代码，以NuMicro® Mini51 DE系列为例，上电后先将512字节的数据内存擦除，当BOD侦测到之后进入中断并开始将512字节的数据写入数据内存。

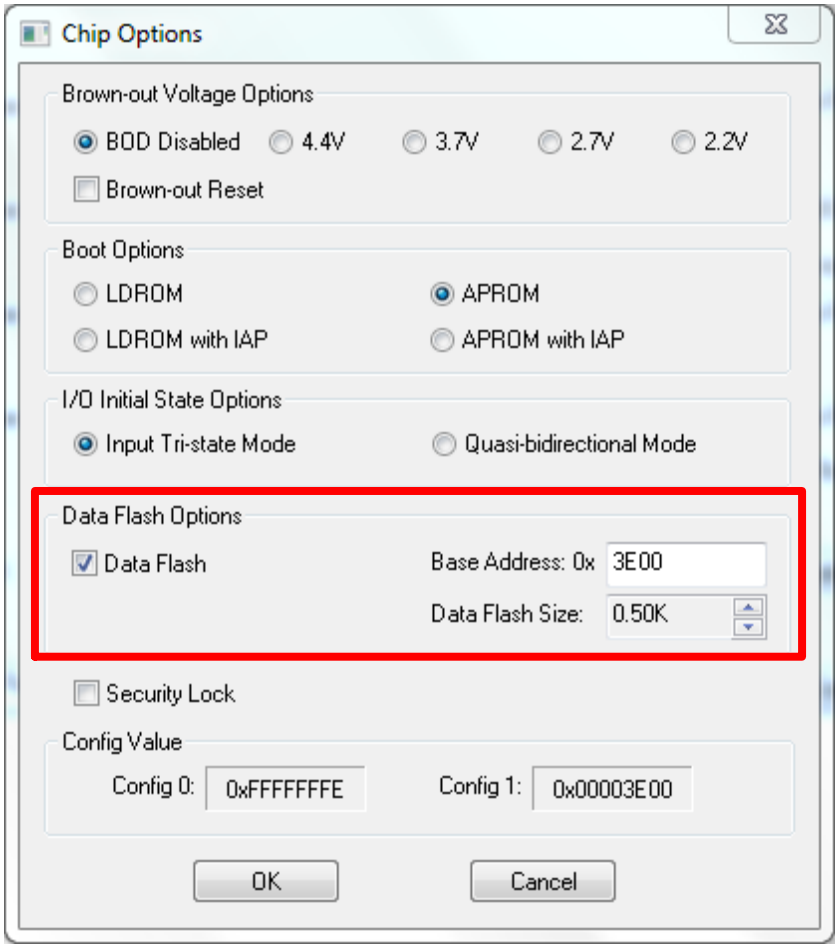


图 4-1 数据内存设定

4.1 库函数

4.1.1 SYS_Init();

```
void SYS_Init(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    P01 = 1;                                  // Set init state to high
    GPIO_SetMode(P0, BIT1, GPIO_PMD_OUTPUT); // Set to output mode
    /* Enable FMC ISP function */
    FMC_Open();
    /* Enable BOD function and setting Brown-out voltage level*/
    SYS_EnableBOD(SYS_BODCR_BOD_INTERRUPT_EN, SYS_BODCR_BOD_VL_4_4V);
    NVIC_EnableIRQ(BOD_IRQn); // Enable BOD interrupt

    /* Lock protected registers */
    SYS_LockReg();
}
```

4.1.2 BOD_IRQHandler();

```
void BOD_IRQHandler(void)
{
    P01 = 0; // Set P0.1 state to low.

    SYS->BODCTL |= SYS_BODCR_BOD_INTF_Msk; // Clear BOD interrupt flag.
    // Disable BOD to avoid into BOD interrupt again.
    SYS_DisableBOD();
    NVIC_DisableIRQ(BOD_IRQn); // Disable BOD interrupt
    // Write test pattern to fill the whole page
    for(i = 0; i < PAGE_SIZE; i++)
    {
        FMC_Write(u32DFBADR + (i*4), u32Data[i]);
    }
    P01 = 1; // Set P0.1 state to high.
}
```

4.2 范例程序代码

4.2.1 掉电储存 512 字节的数据测试代码

```
#include "Mini51Series.h"

#define PAGE_SIZE 128
uint32_t i, u32DFBADR, u32Data[PAGE_SIZE];

void SYS_Init(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    P01 = 1;                                  // Set init state to high
    GPIO_SetMode(P0, BIT1, GPIO_PMD_OUTPUT); // Set to output mode
    /* Enable FMC ISP function */
    FMC_Open();
    /* Enable BOD function and setting Brown-out voltage level*/
    SYS_EnableBOD(SYS_BODCR_BOD_INTERRUPT_EN, SYS_BODCR_BOD_VL_4_4V);
    NVIC_EnableIRQ(BOD_IRQn); // Enable BOD interrupt

    /* Lock protected registers */
    SYS_LockReg();
}

void BOD_IRQHandler(void)
{
    P01 = 0; // Set P0.1 state to low.

    SYS->BODCTL |= SYS_BODCR_BOD_INTF_Msk; // Clear BOD interrupt flag.
    // Disable BOD to avoid into BOD interrupt again.
    SYS_DisableBOD();
    NVIC_DisableIRQ(BOD_IRQn); // Disable BOD interrupt
    // Write test pattern to fill the whole page
    for(i = 0; i < PAGE_SIZE; i++)
    {
        FMC_Write(u32DFBADR + (i*4), u32Data[i]);
    }
    P01 = 1; // Set P0.1 state to high.
}
```

```
int main(void)
{
    /* Before use this sample code, we need to set data flash for
       erasing and writing test pattern */

    /* Init System, set GPIO mode and enable BOD function*/
    SYS_Init();

    /* Fill test pattern */
    for(i = 0; i < PAGE_SIZE; i++)
    {
        u32Data[i] = i;
    }

    // Unlock protected registers for flash erase and program.
    SYS_UnlockReg();

    // Read data flash base address.
    u32DFBADR = FMC_ReadDataFlashBaseAddr();
    /* Erase one page data flash before write test pattern*/
    FMC_Erase(u32DFBADR);

    while (1);
}
```

5 版本历史

日期	版本	描述
2017.11.24	1.00	1. 初次发布

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.