

实验 15 - NV DATA 读写实验

1. 实验目的

掌握 NRF24LE1 的 NV DATA 的读写。

2. 实验内容

按键 S1 按下时，程序向 0xFA00~0xFA04 地址依次写入 0x01、0x02、0x03、0x04、0x05 5 个字节数据，写完后，再读出数据并通过串口打印出数据。

3. 实验原理

NRF24LE1 内部集成 1.5KB 的 NV DATA。

- 1KB NV 非易失存储器；
- 512 字节的耐久扩展存储器；
- 普通 Flash 块提供最少 1000 次擦/写循环；
- 扩展 Flash 块提供最少 20000 次擦/写循环。

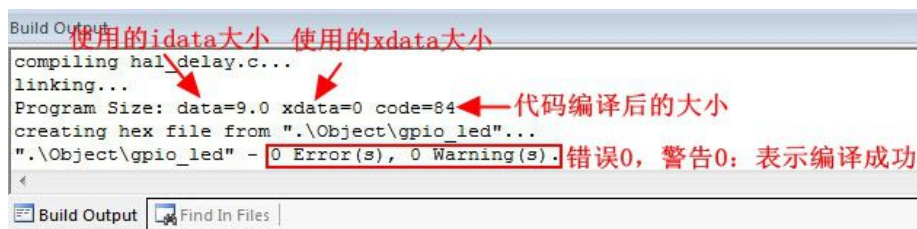
1.5KB 的 NV 非易失数据存储区包括两页各 256 字节的扩展页和两页各 512 字节的普通页。表 1 列出了 MCU 和 SPI 可以访问的这 4 页的映像和用来擦除的页数。

表 1：MCU 和 SPI 可以访问的映像及可擦除的页数

数据存储区	MCU 地址	SPI 地址	页数
长擦写循环次数数据区	0xFA00~0xFAFF	NA	32
	0xFB00~0xFBFF	NA	33
普通擦写循环次数数据区	0xFC00~0xFCFF	0x4400~0x45FF	34
	0xFE00~0xFEFF	0x4600~0x47FF	35

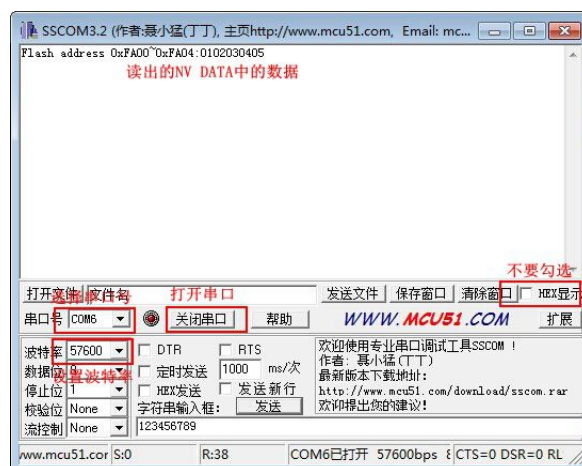
4. 实验步骤

- 在 Keil uVision4 中打开工程“nvdat.uvproj”工程；
- 编译工程，注意查看编译输出栏，观察编译的结果，如果有错误，修改程序，直到编译成功为止；



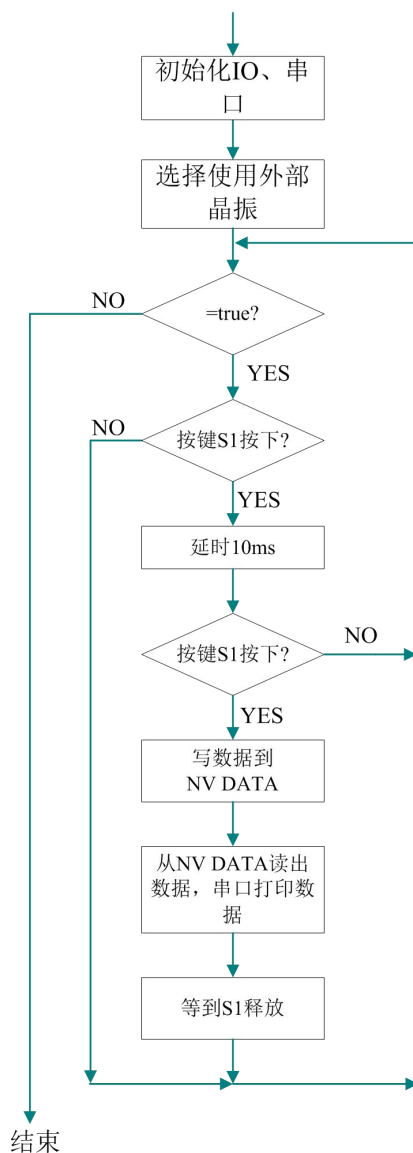
- 将编译生成的 HEX 文件“nvdat.hex”（该文件位于工程目录下的“Object”文件夹中）通过编程器下载到开发板中运行。
- 打开串口调试助手，选择串口号，设置波特率为 57600，打开串口，注意不要勾选

“HEX 显示”。按下 S1 按键再松开，观察串口输出的数据。



5. 实验程序

5.1. 程序流程



5.2. 程序清单

```
#define D1    P00 //开发板上的指示灯 D1
```

```

/*****
*描 述：配置 IO P0.0 为输出，驱动 LED。P03 输出: UART TXD, P04:输入 UART RXD
*入 参：无
*返回值：无
*****/

```

```
void IO_Init(void)
```

```

{
    P0DIR &= ~0x01;    //配置 P0.0 为输出
    P0DIR &= ~0x08;    //P03:输出 UART TXD
    P0DIR |= 0x10;     //P04:输入 UART RXD
    D4 = 1;           //设置 D4 初始状态为熄灭
}

/*****

*描 述：串口打印字符串
*入 参：无
*返回值：无

*****/

void PutString(char *s)
{
    while(*s != 0)
        hal_uart_putchar(*s++);
}

/*****

*描 述：主函数
*入 参：无
*返回值：无

*****/

void main(void)
{
    uint8_t i;
    uint32_t LoopCount = 0;

    hal_clk_set_16m_source(HAL_CLK_XOSC16M); //使用外部 16MHz 晶振

    IO_Init(); //初始化 IO
    hal_uart_init(UART_BAUD_57K6); //初始化 UART，波特率 57600
    while(hal_clk_get_16m_source() != HAL_CLK_XOSC16M) //等待时钟稳定
        ;
    EA = 1;           //开启全局中断

    while(1)
    {
        LoopCount++;
        if(LoopCount == 10000)
        {
            D1 = ~D1; //D1 指示灯闪烁，指示设备工作正常
            LoopCount = 0;
        }
    }
}

```

```
if(S1 == 0)// 按键 S1 按下?
{
    delay_ms(10);
    if(S1 == 0)    //确认按键 S1 按下
    {
        hal_flash_page_erase(32); //写之前先擦除
        delay_ms(30);
        for(i=0;i<5;i++) WriteBuf[i] = i+1;
        hal_flash_bytes_write(0xFA00,WriteBuf,5); //写入数据
        hal_flash_bytes_read(0xFA00,temp_data,5); //读出数据
        PutString("Flash address 0xFA00~0xFA04:"); //串口打印数据
        for(i=0;i<5;i++)PutHexString(temp_data[i]);
        while(S1==0); //等待按键释放
    }
}
}
```