

免责声明



Firmware Disclaimer Information

1. The customer hereby acknowledges and agrees that the program technical documentation, including the code, which is supplied by BEST HEALTH ELECTRONIC Inc., (hereinafter referred to as BestHealth) is the proprietary and confidential intellectual property of BestHealth, and is protected by copyright law and other intellectual property laws.
2. The customer hereby acknowledges and agrees that the program technical documentation, including the code, is confidential information belonging to BestHealth, and must not be disclosed to any third parties other than BestHealth and the customer.
3. The program technical documentation, including the code, is provided and for customer reference only. After delivery by BestHealth, the customer shall use the program technical documentation, including the code, at their own risk. BestHealth disclaims any expressed, implied or statutory warranties, including the warranties of merchantability, satisfactory quality and fitness for a particular purpose.

Copyright (C) BEST HEALTH ELECTRONIC Inc.
All rights reserved

EEPROM 说明

EEPROM 数据存储器。由于其非易失的存储结构，即使在电源掉电的情况下存储器内的数据仍然保存完好。这种存储区扩展了存储器空间，对设计者来说增加了许多新的应用机会。EEPROM 可以用来存储产品编号、校准值、用户特定数据、系统配置参数或其它产品信息等。EEPROM 的数据读取和写入过程也会变的更简单。

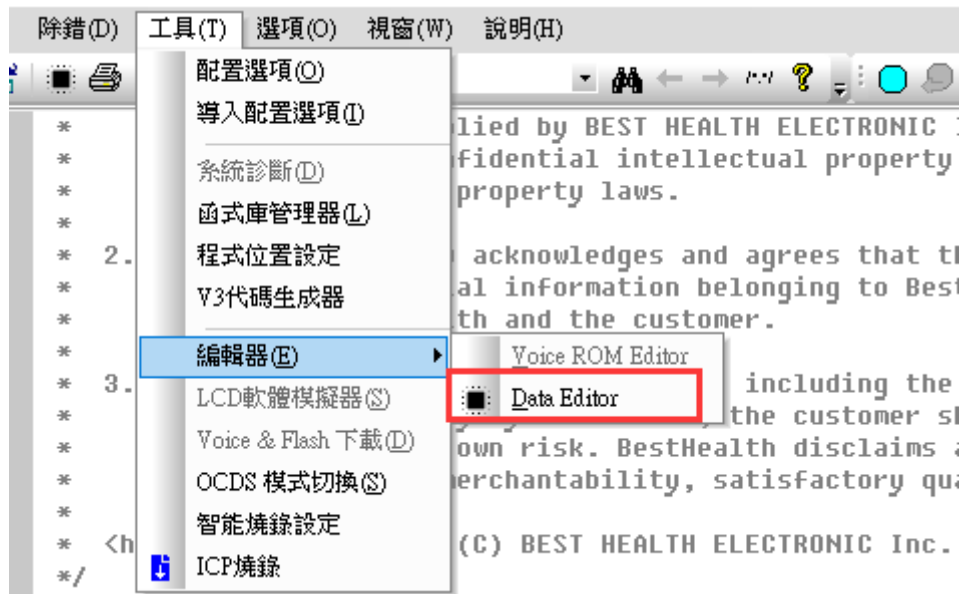
example 说明

此范例演示了 **EEPROM的读写操作** 的使用

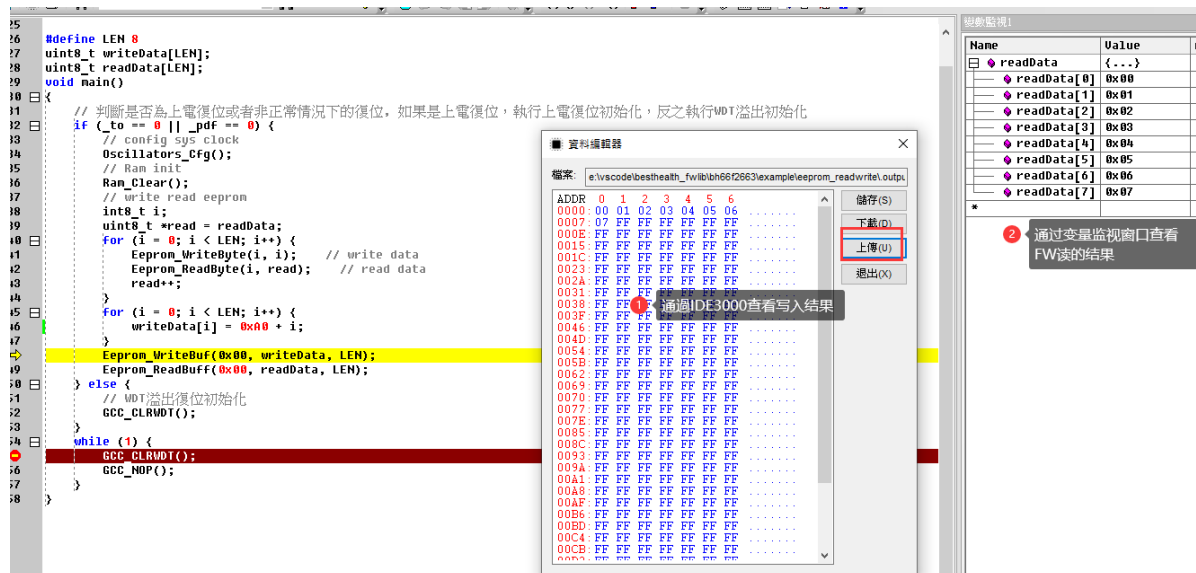
1. 连接elink与开发板
2. 编译并下载程序
3. 执行

现象

- 执行前通过IDE3000->工具->编辑器->dataEdit 参看当前的EEPROM值



- 先通过EEPROM byte 读写函数将EEPROM地址0-7的数据依次写入为0-7



- 继续运行，使用 EEPROM 连续读写函数将EEPROM地址0~7的数据修改为0xA0-A7

```

25
26 #define LEN 8
27 uint8_t writeData[LEN];
28 uint8_t readData[LEN];
29 void main()
30 {
31     // 判断是否为上电复位或者非正常情况下的复位，如果是上电复位，执行上电复位初始化，反之执行WDT溢出初始化
32     if (_to == 0 || _pdf == 0) {
33         // config sys clock
34         Oscillators_Cfg();
35         // Ram init
36         Ram_Clear();
37         // write read eeprom
38         uint8_t i;
39         uint8_t *read = readData;
40         for (i = 0; i < LEN; i++) {
41             Eeprom_WriteByte(i, i); // write data
42             Eeprom_ReadByte(i, read); // read data
43             read++;
44         }
45         for (i = 0; i < LEN; i++) {
46             writeData[i] = 0xA0 + i;
47         }
48         Eeprom_WriteBuf(0x00, writeData, LEN);
49         Eeprom_ReadBuf(0x00, readData, LEN);
50     } else {
51         // WDT溢出复位初始化
52         GCC_CLRWDVT();
53     }
54     while (1) {
55         GCC_CLRWDVT();
56         GCC_NOP();
57     }
58 }

```

资料存储器

檔案: e:\ivscodelib\eshealth_fw\libh66f2663\example\eepram_readwrite.outp...

ADDR 0 1 2 3 4 5 6

0000: A0 A1 A2 A3 A4 A5 A6

0007: A7 FF FF FF FF FF FF

000E: FF FF FF FF FF FF FF

0015: FF FF FF FF FF FF FF

001C: FF FF FF FF FF FF FF

0023: FF FF FF FF FF FF FF

002A: FF FF FF FF FF FF FF

0031: FF FF FF FF FF FF FF

0038: FF FF FF FF FF FF FF

003F: FF FF FF FF FF FF FF

0046: FF FF FF FF FF FF FF

004D: FF FF FF FF FF FF FF

0054: FF FF FF FF FF FF FF

005B: FF FF FF FF FF FF FF

0062: FF FF FF FF FF FF FF

0069: FF FF FF FF FF FF FF

0070: FF FF FF FF FF FF FF

0077: FF FF FF FF FF FF FF

007E: FF FF FF FF FF FF FF

0085: FF FF FF FF FF FF FF

008C: FF FF FF FF FF FF FF

0093: FF FF FF FF FF FF FF

009A: FF FF FF FF FF FF FF

00A1: FF FF FF FF FF FF FF

00A8: FF FF FF FF FF FF FF

00AF: FF FF FF FF FF FF FF

00B6: FF FF FF FF FF FF FF

00BD: FF FF FF FF FF FF FF

00C4: FF FF FF FF FF FF FF

00CB: FF FF FF FF FF FF FF

00D2: FF FF FF FF FF FF FF

儲存(S)

下載(D)

上傳(U)

退出(X)

变量监视1

Name	Value
readData {...}	
readData[0]	0xA0
readData[1]	0xA1
readData[2]	0xA2
readData[3]	0xA3
readData[4]	0xA4
readData[5]	0xA5
readData[6]	0xA6
readData[7]	0xA7

通过变量窗口查看FW读出的结果

FAQ

1. EEPROM 写的时候为什么无法效应中断

- 为防止EEPROM误操作，EEPROM写操作时有一定的时序，为了保证时序正确，在写EEPROM时会暂时关闭EMI，在数据操作完成后再恢复EMI状态