

## M480 NAND Data Logger

Example Code Introduction for 32-bit NuMicro® Family

### Information

Application	This sample code accesses files in external EBI NAND Flash.
BSP Version	M480 Series BSP CMSIS v3.03.000
Hardware	M483 Nand Flash v1.0

*The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.*

*Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.*

*All data and specifications are subject to change without notice.*

*For additional information or questions, please contact: Nuvoton Technology Corporation.*

[www.nuvoton.com](http://www.nuvoton.com)

## 1 Overview

This sample code accesses files in external EBI NAND Flash. User can store the data in NAND Flash as a data logger.

### 1.1 Demo Result

The demo result is shown below.

```

Serial-COM3 - SecureCRT
File Edit View Options Transfer Script Tools Help
Serial-COM3
=====
M480 NAND FATFS DEMO
=====
format..
[NANDLIB] NANDLIB format version on NAND chip: 01000000
rc=0 FR_OK
fopen..
rc=0 FR_OK
fwrite..
fclose..
rc=0 FR_OK
write speed: 16 KB/sec <61024>
fopen..
rc=0 FR_OK
fread..
fclose..
rc=0 FR_OK
read speed: 1084 KB/sec <944>
done ...
█
Ready Serial: COM3 23, 1 23 Rows, 57 Cols VT100 NUM

```

Figure 1-1 Demo Result

## 2 Code Description

EBI 8-bit configuration.

```
void Configure_EBI_8BIT_Pins(void)
{
    /* EBI AD0~5 pins on PC.0~5 */
    SYS->GPC_MFPL |= SYS_GPC_MFPL_PC0MFP_EBI_AD0 | SYS_GPC_MFPL_PC1MFP_EBI_AD1 |
                   SYS_GPC_MFPL_PC2MFP_EBI_AD2 | SYS_GPC_MFPL_PC3MFP_EBI_AD3 |
                   SYS_GPC_MFPL_PC4MFP_EBI_AD4 | SYS_GPC_MFPL_PC5MFP_EBI_AD5;

    /* EBI AD6, AD7 pins on PA.6, PA.7 */
    SYS->GPA_MFPL |= SYS_GPA_MFPL_PA6MFP_EBI_AD6 | SYS_GPA_MFPL_PA7MFP_EBI_AD7;

    /* EBI ADR2~3 pins on PB.2~3 */
    SYS->GPB_MFPL |= SYS_GPB_MFPL_PB2MFP_EBI_ADR3 | SYS_GPB_MFPL_PB3MFP_EBI_ADR2;

    /* EBI nWR and nRD pins on PA.10 and PA.11 */
    SYS->GPA_MFPH |= SYS_GPA_MFPH_PA10MFP_EBI_nWR | SYS_GPA_MFPH_PA11MFP_EBI_nRD;

    /* EBI CS0 pin on PF.6 */
    SYS->GPF_MFPL |= SYS_GPF_MFPL_PF6MFP_EBI_nCS0;

    /* Set WP (PB0) output high to enable write/erase */
    GPIO_SetMode(PB, BIT0, GPIO_MODE_OUTPUT);
    PB0 = 1;
}
```

NAND Flash configuration. Hook the NAND information and drivers.

```
NDRV_T NandDiskDriver = {
    .vid = 0x98,      /* vendor id: Toshiba */
    .did = 0xD3,      /* device id */
    .mcycle = 1,      /* 5 address cycles */
    .pblock = 4096,   /* total physical block counts */
    .lblock = 4000,   /* total logical block counts */
    .ppb = 64,        /* pages per block */
    .pagesize = 4096, /* page size */
    .oobsize = 256,   /* spare area size */
    .ecc = 8,         /* correctable error bit */
    .eccbyte = 13,    /* ECC byte per 512 byte data */
    .init = Init,
    .pread = ReadPage,
    .pwrite = WritePage,
    .isDirtyPage = IsDirtyPage,
    .isValidBlock = IsValidBlock,
    .erase = EraseBlock,
    .markBad=MarkBadBlock
};
```

In main.c, initialize system, configure the EBI, and mount the NAND disk. Then, do file access (open/ read/ write/ close).

```
int32_t main(void)
{
    long          p1;
    TCHAR         nand_path[] = { '0', ':', 0 }; /* NAND drive started from 0 */
    uint32_t s1, s2, cnt;
```

```

static FIL file1;          /* File objects */

Buff = (BYTE *)Buff_Pool;

/* Initialize System, peripheral clock and multi-function I/O */
SYS_Init();
UART_Open(UART0, 115200);
timer_init();

/* Configure multi-function pins for EBI 8-bit application */
Configure_EBI_8BIT_Pins();

/* Initialize EBI bank0 to access NAND flash */
EBI_Open(EBI_BANK0, EBI_BUSWIDTH_8BIT, EBI_TIMING_NORMAL, EBI_OPMODE_ADSEPARATE,
EBI_CS_ACTIVE_LOW);

printf("\n===== \n");
printf("\n  M480 NAND FATFS DEMO \n");
printf("\n===== \n");

/* Mount NAND disk */
f_mount(&gFatfsVol, nand_path, 0);
printf("format.. \n");
put_rc(f_mkfs("", FM_FAT32, 0, Buff, FF_MAX_SS));
f_mount(&gFatfsVol, nand_path, 1);

/* Write file test: fopen -> fwrite -> fclose */
printf("fopen.. \n");
put_rc(f_open(&file1, "hello.txt", FA_OPEN_APPEND | FA_WRITE));
p1 = 10 * 1024 * 1024;
printf("fwrite.. \n");
s1 = gSec;
while(p1 > 0)
{
    f_write(&file1, Buff, 8192, &cnt);
    p1 -= cnt;
}
s2 = gSec;

printf("fclose.. \n");
put_rc(f_close(&file1));
printf("write speed: %d KB/sec <%d> \n", 10*1024*100/(s2-s1), s2-s1);

/* Read file test: fopen -> fread -> fclose */
printf("fopen.. \n");
put_rc(f_open(&file1, "hello.txt", FA_READ));
p1 = 10 * 1024 * 1024;
printf("fread.. \n");
s1 = gSec;
while(p1 > 0)
{
    f_read(&file1, Buff, 8192, &cnt);
    p1 -= cnt;
}
s2 = gSec;

printf("fclose.. \n");
put_rc(f_close(&file1));
printf("read speed: %d KB/sec <%d> \n", 10*1024*100/(s2-s1), s2-s1);

printf("done ... \n");
}

```

## **3 Software and Hardware Requirements**

### **3.1 Software Requirements**

- BSP version
  - ◆ M480 Series BSP CMSIS v3.03.000
- IDE version
  - ◆ Keil uVersion 5.26

### **3.2 Hardware Requirements**

- Circuit components
  - ◆ M483 Nand Flash v1.0
  - ◆ NAND FLASH – TH58NVG3S0HTAI0

● Pin Connect

V <sub>DD</sub>	↔	V <sub>DD</sub>
GND	↔	GND
PC.0 (AD0)	↔	D0
PC.1 (AD1)	↔	D1
PC.2 (AD2)	↔	D2
PC.3 (AD3)	↔	D3
PC.4 (AD4)	↔	D4
PC.5 (AD5)	↔	D5
PA.6 (AD6)	↔	D6
PA.7 (AD7)	↔	D7
PB.2 (ADR3)	↔	ALE
PB.3 (ADR2)	↔	CLE
PA.10 (nWR)	↔	WE#
PA.11 (nRD)	↔	RE#
PF.6 (nCS0)	↔	CE0#
EBI		
PB.1	↔	RB#
PB.0	↔	WP#
	↔	
M480		NAND Flash

Figure 3-1 Pin Connect

## 4 Directory Information

The directory structure is shown below.












 EC_M480_NAND_Data_Logger_V1.00	
 Library	Sample code header and source files
 CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
 Device	CMSIS compliant device header file
 NandLib	NAND Flash logic to physical mapping
 StdDriver	All peripheral driver header and source files
 SampleCode	
 ExampleCode	Source file of example code
 ThirdParty	
 bch	Bose-Chaudhuri-Hocquenghem (BCH) codes used by NAND Error Correction Code
 FatFs	Generic FAT File system

Figure 4-1 Directory Structure

## 5 Example Code Execution

1. Browse the sample code folder as described in the Directory Information section and double-click *NAND\_Data\_Logger.uvproj*.
2. Enter Keil compile mode.
  - Build
  - Download
  - Start/Stop debug session
3. Enter debug mode.
  - Run

## 6 Revision History

Date	Revision	Description
2021.07.18	1.00	1. Initially issued.

### **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.**

---

*Please note that all data and specifications are subject to change without notice.  
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*