

BÁO CÁO THỰC HÀNH BÀI 5

Môn học: **CHUYÊN ĐỀ THIẾT KẾ HỆ THỐNG NHÚNG 1**- Mã lớp: **CE437.N11**
Giảng viên hướng dẫn thực hành: **Phạm Minh Quân**

Thông tin sinh viên	MSSV 20520211 20520219 20520597	Họ và tên Trương Hữu Khang Nguyễn Linh Anh Khoa Phan Duy Thông
Link các tài liệu tham khảo (nếu có)		
Đánh giá của giảng viên: + Nhận xét + Các lỗi trong chương trình + Gợi ý		

[Báo cáo chi tiết các thao tác, quy trình sinh viên đã thực hiện trong quá trình làm bài thực hành. Chụp lại hình ảnh màn hình hoặc hình ảnh kết quả chạy trên sản phẩm. Mô tả và giải thích chương trình tương ứng để cho ra kết quả như hình ảnh đã trình bày. Sinh viên xuất ra file .pdf và đặt tên theo cấu trúc: MSSV_HoTen_Labx_Report.pdf (Trong đó: MSSV là mã số sinh viên, HoTen là họ và tên, x trong Labx là chỉ số của bài thực hành tương ứng)]

MỤC LỤC

BÁO CÁO THỰC HÀNH BÀI 5 1

Câu 1. Viết chương trình ghi 100 dòng dữ liệu “Hello, We are access Flash Memory in the course
CE437\r\n” vào bộ nhớ Flash, sau đó đọc toàn bộ dữ liệu trên vùng nhớ đó và xuất ra UART 3

Câu 1. Viết chương trình ghi 100 dòng dữ liệu “Hello, We are access Flash Memory in the course CE437\r\n” vào bộ nhớ Flash, sau đó đọc toàn bộ dữ liệu trên vùng nhớ đó và xuất ra UART

Hàm dùng để lấy phần Sector cần ghi dữ liệu vào

```
uint32_t GetSector(uint32_t Address)
{
    uint32_t sector = 0;

    if (Address >= 0x08000000 && Address <= 0x08007FFF) {
        sector = FLASH_SECTOR_0;
    } else if (Address >= 0x08008000 && Address <= 0x0800FFFF) {
        sector = FLASH_SECTOR_1;
    } else if (Address >= 0x08010000 && Address <= 0x08017FFF) {
        sector = FLASH_SECTOR_2;
    } else if (Address >= 0x08018000 && Address <= 0x0801FFFF) {
        sector = FLASH_SECTOR_3;
    } else if (Address >= 0x08020000 && Address <= 0x0803FFFF) {
        sector = FLASH_SECTOR_4;
    } else if (Address >= 0x08040000 && Address <= 0x0807FFFF) {
        sector = FLASH_SECTOR_5;
    } else if (Address >= 0x08080000 && Address <= 0x080BFFFF) {
        sector = FLASH_SECTOR_6;
    } else if (Address >= 0x080C0000 && Address <= 0x080FFFFF) {
        sector = FLASH_SECTOR_7;
    } else {
        sector = 0;
    }
    return sector;
}
```

Block	Name	Bloc base address on AXIM interface	Block base address on ICTM interface	Sector size
Main memory block	Sector 0	0x0800 0000 - 0x0800 7FFF	0x0020 0000 - 0x0020 7FFF	32 Kbytes
	Sector 1	0x0800 8000 - 0x0800 FFFF	0x0020 8000 - 0x0020 FFFF	32 Kbytes
	Sector 2	0x0801 0000 - 0x0801 7FFF	0x0021 0000 - 0x0021 7FFF	32 Kbytes
	Sector 3	0x0801 8000 - 0x0801 FFFF	0x0021 8000 - 0x0021 FFFF	32 Kbytes
	Sector 4	0x0802 0000 - 0x0803 FFFF	0x0022 0000 - 0x0023 FFFF	128 Kbytes
	Sector 5	0x0804 0000 - 0x0807 FFFF	0x0024 0000 - 0x0027 FFFF	256 Kbytes
	Sector 6	0x0808 0000 - 0x080B FFFF	0x0028 0000 - 0x002B FFFF	256 Kbytes
	Sector 7	0x080C 0000 - 0x080F FFFF	0x002C 0000 - 0x002F FFFF	256 Kbytes
Information block	System memory	0x1FF0 0000 - 0x1FF0 EDBF	0x0010 0000 - 0x0010 EDBF	60 Kbytes
	OTP	0x1FF0 F000 - 0x1FF0 F41F	0x0010 F000 - 0x0010 F41F	1024 bytes
	Option bytes	0x1FFF 0000 - 0x1FFF 001F	-	32 bytes

Giá trị của Sector sẽ phụ thuộc vào datasheet của MCU

Phần chương trình ghi 100 dòng dữ liệu vào vùng nhớ

```
uint32_t Address = START_ADDRESS;
char *data      = "Hello, We are access Flash Memory in the course
CE437\r\n";
int numofwords  = (strlen(data) / 4) + ((strlen(data) % 4) != 0);
Flash_Write_Data(Address, (uint32_t *)data, numofwords, 100);
```

Phần chương trình đọc toàn bộ dữ liệu đã ghi vào Sector

```
while (1) {
    /* USER CODE END WHILE */
    /* USER CODE BEGIN 3 */
    Address = START_ADDRESS;
    for (int i = 0; i < 100; i++) {
        Flash_Read_Data(Address, (uint32_t *)read_data, numofwords, 100);
        Address += numofwords * 4;
        Convert_To_Str(read_data, string);
        printf("Address: 0x%08lX, String: %s", Address, string);
    }
    printf("%s\n", "Done");
}
```

Giải thích:

Để ghi được 100 dòng dữ liệu thì ta cần để phần code ghi vào Flash vào một vòng lặp, mỗi lần gọi lại hàm ghi vào Flash, vùng nhớ cần được ghi vào sẽ bị xóa sạch. Khi dữ liệu được đưa vào để tiến hành ghi vào vùng nhớ Flash, nó sẽ được chuyển sang dạng uint32_t, nên với mỗi phần dữ liệu ta cộng địa chỉ lên 4 (4 bytes do kiểu dữ liệu là uint32_t).

```
for (int i = 0; i < numberoflines; i++) {
    while (sofar < numberofwords) {
        if (HAL_FLASH_Program(FLASH_TYPEPROGRAM_WORD, StartSectorAddress,
Data[sofar]) == HAL_OK) {
            StartSectorAddress += 4; // use StartPageAddress += 2 for half word
and 8 for double word
            sofar++;
        } else {
            /* Error occurred while writing data in Flash memory*/
            return HAL_FLASH_GetError();
        }
    }
    sofar = 0;
}
```

Đối với việc đọc dữ liệu từ vùng nhớ đã ghi, ta sẽ đọc từ phần đầu của phần vùng nhớ đã ghi dữ liệu, đọc ra từng dòng dữ liệu sau đó tăng địa chỉ lên.

Kết quả thu được:

Address	0	4	8	C	ASCII
0x080C0000	6C6C6548	57202C6F	72612065	63612065	Hello, We are ac
0x080C0010	73736563	616C4620	4D206873	726F6D65	cess Flash Memor
0x080C0020	6E692079	65687420	756F6320	20657372	y in the course
0x080C0030	33344543	000A0D37	6C6C6548	57202C6F	CE437...Hello, w
0x080C0040	72612065	63612065	73736563	616C4620	e are access Fla
0x080C0050	4D206873	726F6D65	6E692079	65687420	sh Memory in the
0x080C0060	756F6320	20657372	33344543	000A0D37	course CE437...
0x080C0070	6C6C6548	57202C6F	72612065	63612065	Hello, We are ac
0x080C0080	73736563	616C4620	4D206873	726F6D65	cess Flash Memor
0x080C0090	6E692079	65687420	756F6320	20657372	y in the course

```

Address: 0x080C1298, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C12D0, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C1308, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C1340, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C1378, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C13B0, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C13E8, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C1420, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C1458, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C1490, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C14C8, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C1500, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C1538, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C1570, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C15A8, String: Hello, We are access Flash Memory in the course CE437
Address: 0x080C15E0, String: Hello, We are access Flash Memory in the course CE437
Done

```