

## *Chương 5*

# **CÁC BÀI THỰC HÀNH MODULE AUDIO VS1003 SD\_CARD, MP3**

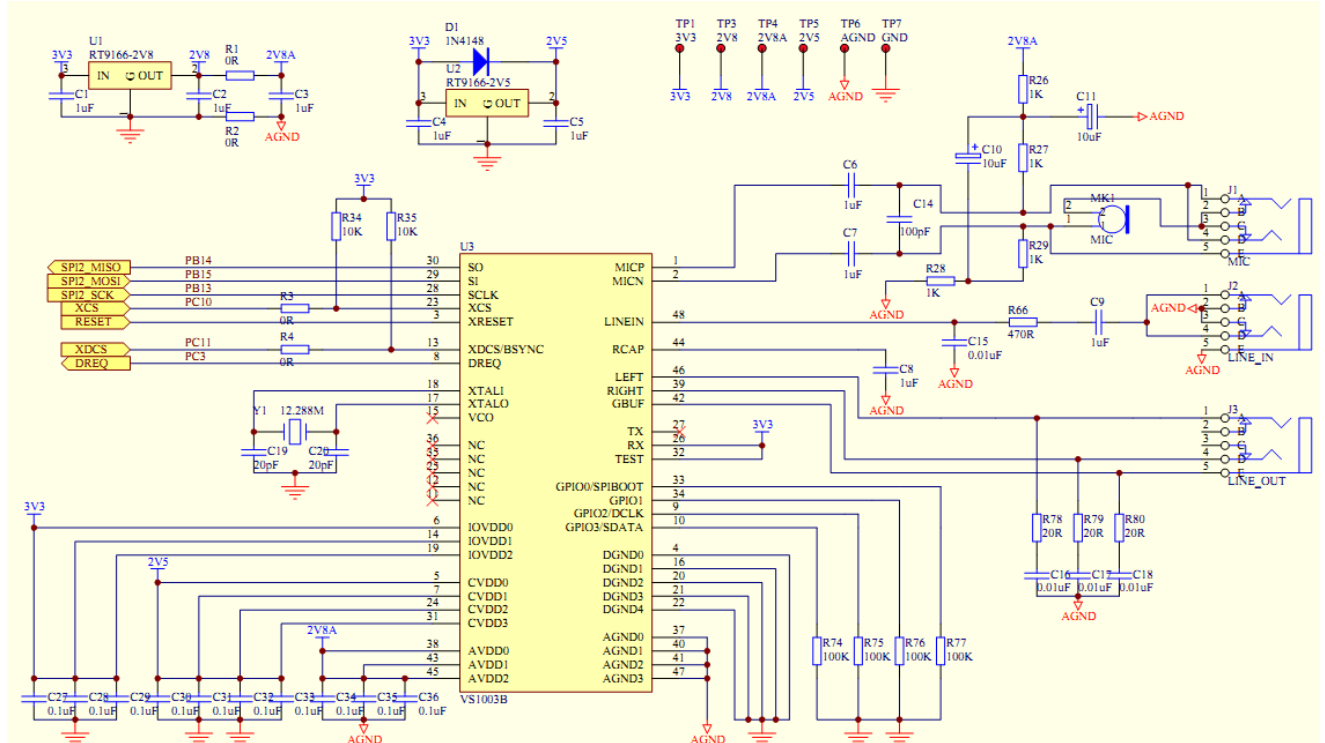
- **GIỚI THIỆU**
  - CÀI ĐẶT CHƯƠNG TRÌNH
  - CÁCH VIẾT CHƯƠNG TRÌNH
- **GIAO TIẾP VI ĐIỀU KHIỂN VỚI LED ĐƠN**
- **SỬ DỤNG PHẦN MỀM LẬP TRÌNH C CHO VI ĐIỀU KHIỂN**
- **CÁC CHƯƠNG TRÌNH ĐIỀU KHIỂN LED ĐƠN**
- **CÁC CHƯƠNG TRÌNH GIAO TIẾP NÚT NHẤN**

## I. GIỚI THIỆU

Trình bày phần giao tiếp âm thanh dùng chip VS1003, giao tiếp thẻ nhớ SD, các chương trình đọc dữ liệu từ thẻ nhớ và ghi dữ liệu ra IC giải mã âm thanh VS1003.

## II. SƠ ĐỒ MẠCH GIAO TIẾP VS1003

Sơ đồ mạch giao tiếp VS1003 với các micro-phone và vi điều khiển ARM như hình sau

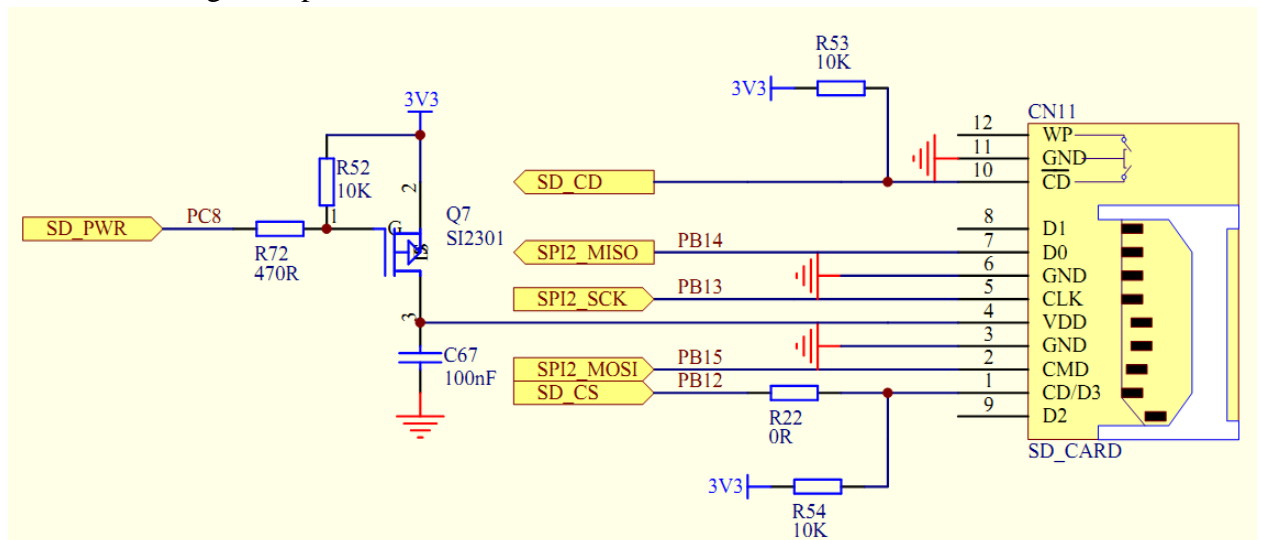


Hình 5-1. Sơ đồ mạch giao tiếp vi điều khiển IC đệm và led đơn.

Phần giao tiếp âm thanh ra gắn với tai nghe âm thanh stereo (Line\_out), có tín hiệu âm thanh vào từ micro (Mic) và từ tín hiệu âm thanh ngõ vào (Line\_in).

Các tín hiệu giao tiếp với vi điều khiển theo chuẩn SPI (SPI2\_MISO, SPI2\_MOSI, SPI2\_SCK, XCS) và tín hiệu Reset, yêu cầu ngắt DREQ.

Sơ đồ mạch giao tiếp thẻ nhớ SD:



Hình 5-2. Sơ đồ mạch vi điều khiển ARM giao tiếp SD\_CARD.

Các tín hiệu giao tiếp vi điều khiển ARM với SD gồm 5 tín hiệu:

Giao tiếp SPI : SPI2\_MISO dùng port PB14, SPI2\_MOSI dùng port PB15, SPI2\_SCK dùng port PB13, SD\_CS dùng port PB12, SD\_PWR dùng port PC8.

Tín hiệu SD\_CD dùng port PC6.

Thẻ nhớ SD và IC VS1003 cùng giao tiếp với ARM STM32 bằng chuẩn SPI2 và mỗi đối tượng có 1 tín hiệu cho phép khác nhau.

### III. CÁC BÀI THỰC HÀNH

#### CÁC BÀI ĐỌC BÀI HÁT TỪ THẺ NHỚ - NGHE NHẠC

**Bài mẫu 501.** Chương trình nghe nhạc từ thẻ nhớ SD card.

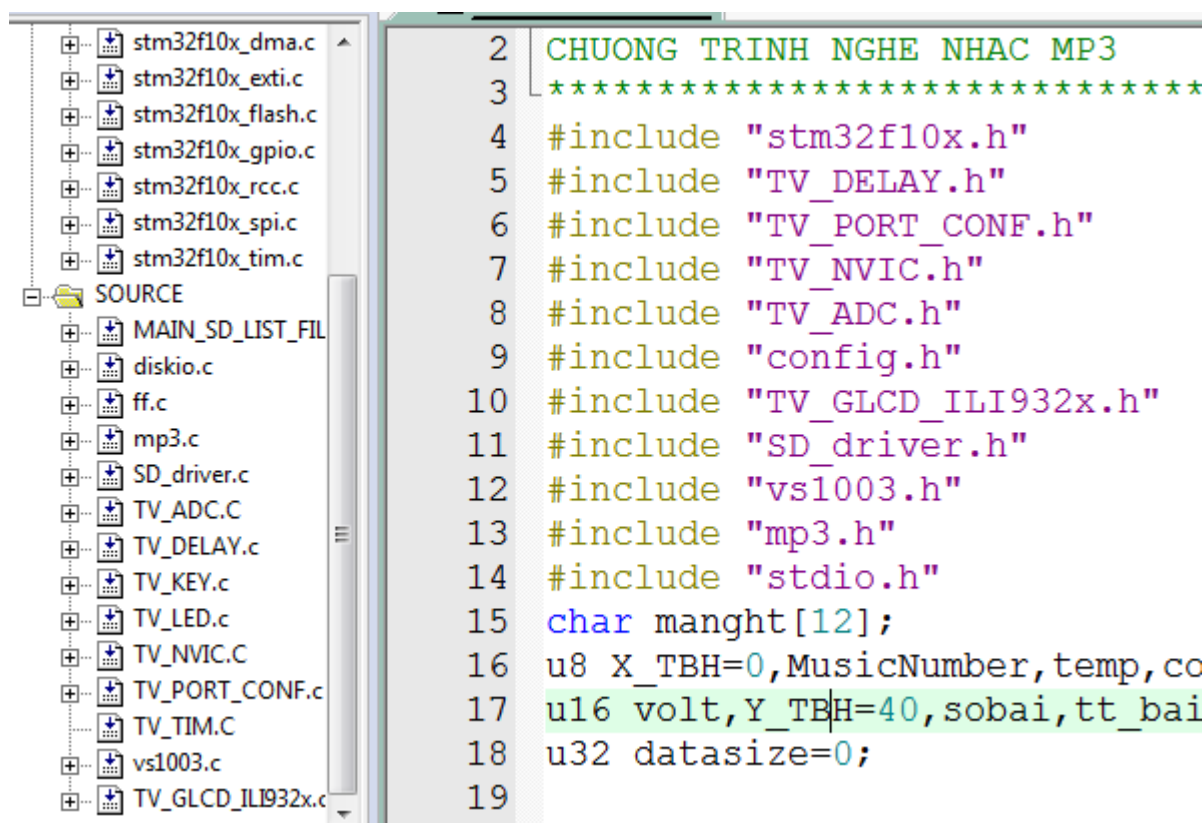
Hàng 1 hiển thị “NGHE NHAC TU THE NHO SD”

Hàng 3 hiển thị thông báo nếu thẻ nhớ không có bài hát, nếu có thì hiển thị tên bài hát, chỉ hiển thị được 8 ký tự, dung lượng bài hát, số byte đang đọc từ thẻ nhớ.

Điều chỉnh volume từ biến trở - gắn tai nghe hoặc loa để nghe.

Tạo thư mục “BAI\_501\_MUSIC\_SD\_MP3” để lưu project

- Mục đích: làm quen với giao tiếp thẻ nhớ SD, IC giải mã, các thư viện chứa các hàm đọc file và dữ liệu từ thẻ nhớ.
- Lưu đồ:
- Hình các file chính và thư viện:



Hình 5-3. Hình các file có liên quan và các đường dẫn file thư viện bài 501.

- Chương trình:

```
/******
```

## CHUONG TRINH NGHE NHAC MP3

```

***** /
#include "stm32f10x.h"
#include "TV_DELAY.h"
#include "TV_PORT_CONF.h"
#include "TV_NVIC.h"
#include "TV_ADC.h"
#include "config.h"
#include "TV_GLCD_ILI932x.h"
#include "SD_driver.h"
#include "vs1003.h"
#include "mp3.h"
#include "stdio.h"

char manght[12];
u8 X_TBH=0,MusicNumber,temp,count=0,i,vol ,buffer[1024];
u16 volt,Y_TBH=40,sobai,tt_baihat=0;
u32 datasize=0;
char listnhac[50][13];

FATFS fs;                // Work area (file system object) for logical drive
FIL fsrc, fdst,fap;//,file; // file objects
FRESULT res,re;          // FatFs function common result code
UINT br, bw,bt;;         // File R/W count
FILINFO finfo,fno;
DIR dirs;

extern u8 glImage_11[];
extern int AD_value;
u8 KTSD;

void CHINH_VOLUMN()
{
    if(count++==20) //chinh volumn
    {
        count=0;
        AD_value=ADC_GetConversionValue(ADC1);
        AD_value=(AD_value*330)/4096;
        vol=(AD_value*255)/330;
        vol=abs(vol-255);
        volt=vol;
        volt<=8;
        volt+=vol;
        Vs1003_CMD_Write(SPI_VOL,volt);
    }
}

int main(void)
{
    SystemInit();  PORT_CONF();      ADC_Configuration();

```

```

NVIC_CONFIGURATION_TIM2();
DELAY_INIT(72);
SPI_SD_CONFIGURATION(); SD_Init(); LCD_INIT();
Vs1003_Init(); Mp3Reset(); Vs1003SoftReset();
VsSineTest(); //tao am thanhkiem tra loa

LCD_CLEAR(WHITE);

f_mount(0,&fs);
sobai =Scanfile(listnhac,0,"MP3");

LCD_SHOW_STRING_XC(0,0,"NGHE NHAC TU THE NHO SD");
LCD_SHOW_NUM(200,00,sobai,3,16);
if(sobai==0)
{
    LCD_SHOW_STRING_XC(0,20,"KHONG CO NHAC TRONG THE NHO SD");
    while(1);
}

CHINH_VOLUME();
f_open(&fsrc, listnhac[tt_baihat],FA_OPEN_EXISTING | FA_READ);
while(1)
{
    LCD_SHOW_STRING_XC(X_TBH,Y_TBH,listnhac[tt_baihat]);
    LCD_SHOW_NUM(X_TBH+100,Y_TBH,fsrc.fsize,8,16);
    LCD_SHOW_NUM(X_TBH+180,Y_TBH,datasize,8,16);
    if(datasize>=fsrc.fsize) //datasize nhay de so sanh ket thuc bai hat
    {
        Y_TBH+=20; // tang len de hien thi hang ke
        tt_baihat++;
        if(tt_baihat>=sobai){tt_baihat=0; Y_TBH=60;}
        f_open(&fsrc, listnhac[tt_baihat],FA_OPEN_EXISTING | FA_READ);
        datasize=0;
    }
    datasize+=512;
    f_read(&fsrc,buffer,512,&br);
    for(i=0;i<16;i++)
    {
        while(!(GPIOC->IDR & MP3_DREQ));
        Vs1003_DATA_Write(buffer+i*32);
    }
}
}

```

- e. Tiến hành biên dịch và nạp.
- f. Quan sát kết quả:
- g. Giải thích chương trình: tiến hành khởi tạo hệ thống, khởi tạo port, khởi tạo ADC, khởi tạo ngắt, khởi SPI giao tiếp thẻ nhớ, khởi tạo thẻ nhớ SD, khởi tạo GLCD, khởi tạo IC âm thanh VS1003, reset MP3.

Hàm “f\_mount(0,&fs);” có chức năng đọc dung lượng bộ nhớ của thẻ nhớ (số 0 là địa chỉ của thẻ nhớ và chỉ có 1 thẻ nhớ nên có địa chỉ là 0).

Hàm “sobai =Scanfile(listnhac,0,"MP3");” tiến hành đọc số file bài hát lưu trong thẻ nhớ ở thư mục gốc, copy các tên nếu có vào mảng “listnhac”, bắt đầu từ vị trí 0, các file nhạc định dạng MP3.

Kết quả trả về là danh sách bài hát và số lượng bài.

Hiển thị số bài nếu không có bài nào thì hiển thị số 0.

Kiểm tra nếu không có bài hát thì hiển thị thông báo không có bài hát và ngừng.

Nếu có thì tiến hành gọi hàm chỉnh volume bằng biến trở đưa tín hiệu vào ADC, mở file bài hát theo listnhac và hiển thị tên bài hát.

Vòng lặp while:

- Tiến hành hiển thị tên bài hát, dung lượng và số byte đang đọc.
- Hàm “if(datasize>=fsrc.fsize)” tiến hành kiểm tra xem số lượng byte đã đọc có lớn hơn dung lượng bài hát hay không?
- Ban đầu thì biến “datasize” bằng 0 nên không thoả nên không làm các lệnh tương ứng. Khi thoả thì chuyển sang bài hát kế, nếu hết thì quay lại bài đầu tiên.
- Lệnh “datasize+=512;” cộng dồn 512 byte.
- Lệnh “f\_read(&fsrc,buffer,512,&br);” tiến hành đọc dữ liệu 1 mảng 512 byte từ thẻ nhớ của file đã mở ở trên, lưu vào vùng nhớ “buffer”.
- Vòng lặp for tiến hành kiểm tra tín hiệu ngắt từ IC VS1003 nếu sẵn sàng thì gửi dữ liệu ra IC VS1003 để giải mã dữ liệu nhạc về gửi ra loa để nghe.

Quá trình thực hiện cho đến khi dữ liệu đọc lớn hơn dữ liệu của file thì chuyển sang mở file tiếp theo, nếu hết bài hát thì quay lại bài đầu tiên.

h. Cho phép thay đổi:

### **Bài mẫu 2-2.** Chương trình nghe nhạc từ thẻ nhớ SD card.

Hàng 1 hiển thị “NGHE NHAC TU THE NHO SD”

Hàng 2 hiển thị “VOLUME:”

Hàng 3 đến hàng 7 hiển thị tên các bài hát.

Có 10 biểu tượng để điều khiển nghe nhạc.



Hình 5-4. Hình các biểu tượng điều khiển nghe nhạc.

Biểu tượng play dùng để mở nhạc, biểu tượng stop thì ngừng.

Biểu tượng chuyển sang bài kế và bài trước.

Bài nào đang hát thì hiển thị màu nền là màu vàng.

Điều chỉnh volume từ biến trở - gắn tai nghe hoặc loa để nghe.

Tạo thư mục “BAI\_502\_MUSIC\_SD\_MP3\_CONTROL” để lưu project.

- a. Mục đích: thêm phần điều khiển chọn bài hát.  
 b. Thiết lập hình ảnh cho menu:  
 Từ file ảnh của menu đã cho tạo ra file hex và phân chia khoảng cách.  
 Hình hiển thị ở vị trí bắt đầu từ (0, 220), kích thước ảnh là 240, 99.  
 Có tổng cộng 10 biểu tượng điều khiển :

Ta tiến hành tính toán kích thước tọa độ từng biểu tượng.

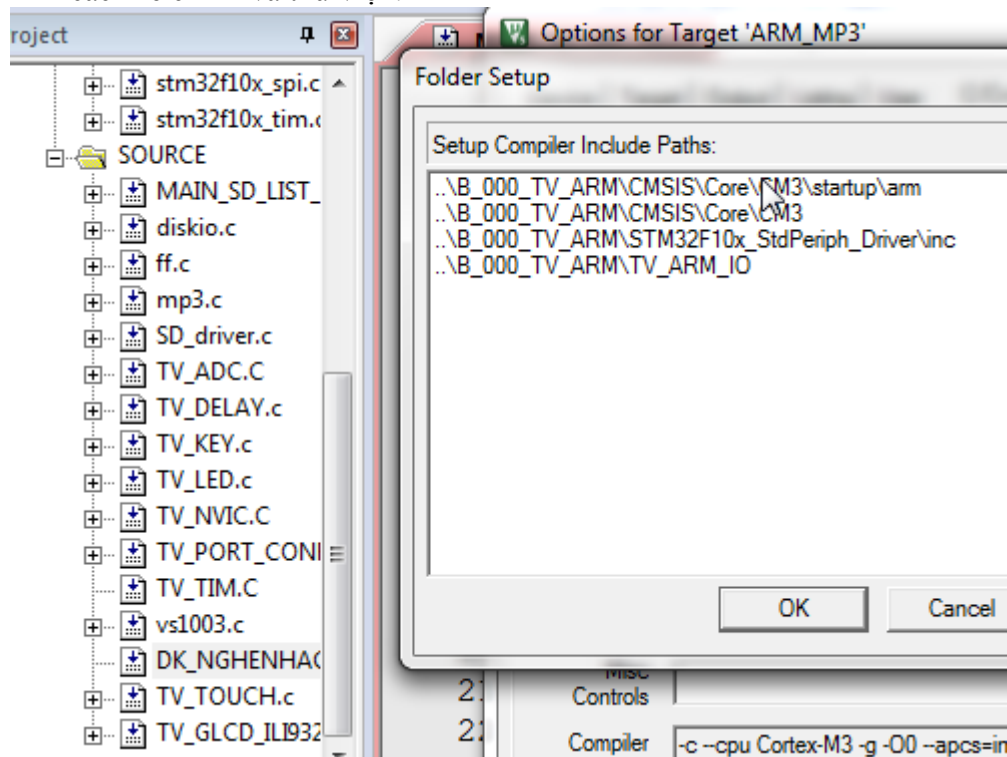
Chiều ngang là 240 trừ cho 4 khoảng trắng là 40 còn 200 chia cho 5 biểu tượng sẽ được 40.

Chiều dài là 99 cho bằng 100, trừ 20 còn 80, chia cho 2 bằng 40.

Vậy mỗi biểu tượng có kích đều bằng 40, 40.

TT	Biểu tượng	Tọa độ đầu		Tọa độ cuối	
				$X2 = X1 + 40$	$Y2 = Y1 + 40$
1	Ngừng	$X11 = 0,$	$Y11 = 220$	$0 + 40 = 40,$	$220 + 40 = 260$
2	Thu	$X12 = 50,$	$Y12 = 220$	$50 + 40 = 90,$	$220 + 40 = 260$
3	Ngừng	$X13 = 100,$	$Y13 = 220$	$100 + 40 = 140,$	$220 + 40 = 260$
4	Loa	$X14 = 150,$	$Y14 = 220$	$150 + 40 = 190,$	$220 + 40 = 260$
5	Lặp lại	$X13 = 200,$	$Y15 = 220$	$200 + 40 = 240,$	$220 + 40 = 260$
6	Bài trước	$X21 = 0,$	$Y11 = 280$	$0 + 40 = 40,$	$280 + 40 = 320$
7	Tua tới nhanh	$X22 = 50,$	$Y12 = 280$	$50 + 40 = 90,$	$280 + 40 = 320$
8	Mở	$X23 = 100,$	$Y13 = 280$	$100 + 40 = 140,$	$280 + 40 = 320$
9	Tua lại nhanh	$X24 = 150,$	$Y14 = 280$	$150 + 40 = 190,$	$280 + 40 = 320$
10	Bài sau	$X23 = 200,$	$Y15 = 280$	$200 + 40 = 240,$	$220 + 40 = 320$

- c. Lưu đồ:  
 d. Hình các file chính và thư viện:



Hình 5-5. Hình các file có liên quan và các đường dẫn file thư viện bài 502.

- e. Chương trình:

```

/*****
CHUONG TRINH NGHE NHAC MP3
*****/

#include "stm32f10x.h"
#include "TV_TOUCH.h"
#include "hardware_conf.h"
#include "TV_DELAY.h"
#include "TV_PORT_CONF.h"
#include "TV_NVIC.h"
#include "TV_ADC.h"
#include "config.h"
#include "TV_GLCD_ILI932x.h"
#include "SD_driver.h"
#include "vs1003.h"
#include "mp3.h"
#include "stdio.h"
#include "TV_LED.h"
u8 MusicNumber,temp,count=0,i,j,TTLED=0,ttbaihat=0,vol,flag=0,buffer[1024];

u16 volt;
u32 datasize=0;
extern u8 gImage_DK_NGHENHAC[];
char listnhac[50][13];
signed char tt_baihat=0;
unsigned char sobai;

FATFS fs;                // Work area (file system object) for logical drive
FIL fsrc, fdst,fap;      // file objects
FRESULT res,re;          // FatFs function common result code
UINT br, bw,bt;;         // File R/W count
FILINFO finfo,fno;
DIR dirs;

extern int AD_value;

uint8_t TT_LED=0,TT_TOUCH,TT_BAIHAT_ON_OFF=0,TT_LOA_ON_OFF=0;
uint8_t X13=100,X23=100,X25=200,KCX=40,X1=0,X_TBH=0;
uint16_t Y13=220,Y23=280,Y25=280,KCY=40,Y1=220,Y_TBH=20,
dem_5bai_ht=0;

uint8_t X21=0,X14=150;
uint16_t Y21=280,Y14=220;

void CHINH_VOLUMN()
{
    if((count++==20)&&((TT_LOA_ON_OFF==0))) //chinh volumn
    {
        count=0;
        AD_value=ADC_GetConversionValue(ADC1);
    }
}

```



```

        AD_value=(AD_value*330)/4096;
        vol=(AD_value*255)/330;
        vol=abs(vol-255);
        volt=vol;
        volt<=8;
        volt+=vol;
        Vs1003_CMD_Write(SPI_VOL,volt);
        LCD_SHOW_NUM(180,20,volt,5,16);
    }
}

void HIEN_THI_TEN_1BAI_HAT(u16 K)
{
    LCD_SHOW_STRING_XC(X_TBH,Y_TBH,listnhac[K]);
    LCD_SHOW_NUM(X_TBH+100,Y_TBH,fsrc.fsize,8,16);
    LCD_SHOW_NUM(X_TBH+180,Y_TBH,datasize,8,16);
    Y_TBH+=20;
}

void HIEN_THI_TEN_5BAI_HAT(u16 ttbhht)
{
    u16 i;
    X_TBH=0;Y_TBH=40;
    for(i=ttbhht;i<ttbhht+5;i++)
    {
        HIEN_THI_TEN_1BAI_HAT(i);
    }
}

void XU_LY_BAI_HAT_KE()
{
    POINT_COLOR = BLUE,BACK_COLOR = WHITE;
    HIEN_THI_TEN_1BAI_HAT(tt_baihat);

    tt_baihat++;
    dem_5bai_ht++;
    if(dem_5bai_ht==5)
    {
        HIEN_THI_TEN_5BAI_HAT(tt_baihat);
        dem_5bai_ht=0;Y_TBH=40;
    }
    if(tt_baihat>=sobai){tt_baihat=0; }
    f_open(&fsrc, listnhac[tt_baihat],FA_OPEN_EXISTING | FA_READ);
    datasize=0;

    POINT_COLOR = BLUE,BACK_COLOR = YELLOW;
    LCD_SHOW_STRING_XC(X_TBH,Y_TBH,listnhac[tt_baihat]);
    LCD_SHOW_NUM(X_TBH+100,Y_TBH,fsrc.fsize,8,16);

```

```

}

void XU_LY_BAI_HAT_TRUOC()
{
    if(dem_5bai_ht==0)
    {
        if (tt_baihat>0)
        {
            tt_baihat-=5;
            POINT_COLOR = BLUE,BACK_COLOR = WHITE;
            HIEN_THI_TEN_5BAI_HAT(tt_baihat);

            tt_baihat+=4;
            dem_5bai_ht=4;
            Y_TBH=120;
            POINT_COLOR = BLUE,BACK_COLOR = YELLOW;
            LCD_SHOW_STRING_XC(X_TBH,Y_TBH,listnhac[tt_baihat]);
            LCD_SHOW_NUM(X_TBH+100,Y_TBH,fsrc.fsize,8,16);
        }
    }
    else
    {
        POINT_COLOR = BLUE,BACK_COLOR = WHITE;
        LCD_SHOW_STRING_XC(X_TBH,Y_TBH,listnhac[tt_baihat]);
        LCD_SHOW_NUM(X_TBH+100,Y_TBH,fsrc.fsize,8,16);
        LCD_SHOW_NUM(X_TBH+180,Y_TBH,datasize,8,16);
        Y_TBH-=20;
        tt_baihat--;
        dem_5bai_ht--;

        f_open(&fsrc, listnhac[tt_baihat],FA_OPEN_EXISTING | FA_READ);
        datasize=0;

        POINT_COLOR = BLUE,BACK_COLOR = YELLOW;
        LCD_SHOW_STRING_XC(X_TBH,Y_TBH,listnhac[tt_baihat]);
        LCD_SHOW_NUM(X_TBH+100,Y_TBH,fsrc.fsize,8,16);
    }
}

void TOUCH_PLAY()
{
    TT_TOUCH=TOUCH_PRESS();
    if (TT_TOUCH==1)
    {
        if ((Pen_Point.X0>X23)&&(Pen_Point.X0<X23+KCX)&&
            (Pen_Point.Y0>Y23)&&(Pen_Point.Y0<Y23+KCY))
        {
            TT_BAIHAT_ON_OFF = 1;
        }
    }
}

```

```

}

void TOUCH_DIEUKHIE()
{
    TT_TOUCH=TOUCH_PRESS();
    if (TT_TOUCH==1)
    {
        if ((Pen_Point.X0>X13)&&(Pen_Point.X0<X13+KCX)&&
            (Pen_Point.Y0>Y13)&&(Pen_Point.Y0<Y13+KCY))
        {
            TT_BAIHAT_ON_OFF = 0;    TT_LED=~ TT_LED;
        }

        else
        if ((Pen_Point.X0>X25)&&(Pen_Point.X0<X25+KCX)&&
            (Pen_Point.Y0>Y25)&&(Pen_Point.Y0<Y25+KCY))
        {
            datasize = fsrc.fsize+100;
        }
        else
        if ((Pen_Point.X0>X21)&&(Pen_Point.X0<X21+KCX)&&
            (Pen_Point.Y0>Y21)&&(Pen_Point.Y0<Y21+KCY))
        {
            XU_LY_BAI_HAT_TRUOC();
        }

        else
        if ((Pen_Point.X0>X14)&&(Pen_Point.X0<X14+KCX)&&
            (Pen_Point.Y0>Y14)&&(Pen_Point.Y0<Y14+KCY))
        {
            TT_LOA_ON_OFF= ~TT_LOA_ON_OFF;
            if (TT_LOA_ON_OFF==1)
                Vs1003_CMD_Write(SPI_VOL,0xff00);
        }
        if (TT_LOA_ON_OFF ==0)    GPIO_ResetBits(GPIOD, GPIO_Pin_8);
        else                      GPIO_SetBits(GPIOD, GPIO_Pin_8);
    }
}

/*****
CHUONG TRINH CHINH
*****/
int main(void)
{
    SystemInit(); PORT_CONF();    ADC_Configuration();
    NVIC_CONFIGURATION_TIM2();
    NVIC_CONFIGURATION_TOUCH();
    TOUCH_INIT();

```

```

DELAY_INIT(72);
LED_INIT_2();

SPI_SD_CONFIGURATION();    SD_Init();    LCD_INIT();
Vs1003_Init();    Mp3Reset();    Vs1003SoftReset();
VsSineTest();    //tao am thanhkiem tra loa

LCD_CLEAR(WHITE);

f_mount(0,&fs);
sobai =Scanfile(listnhac,0,"MP3");

LCD_SHOW_STRING_XC(0,0,"NGHE NHAC TU THE NHO SD");
LCD_SHOW_NUM(200,00,sobai,3,16);
if(sobai==0)
{
    LCD_SHOW_STRING_XC(0,20," KHONG CO NHAC ");
    while(1);
}
LCD_SHOW_STRING_XC(0,20," VOLUME: ");
LCD_WRITE_BITMAP_Y(X1,Y1,240,99, gImage_DK_NGHENHAC);

f_open(&fsrc, listnhac[tt_baihat],FA_OPEN_EXISTING | FA_READ);

HIEN_THI_TEN_5BAI_HAT(tt_baihat);
Y_TBH=40;
POINT_COLOR = BLUE,BACK_COLOR = YELLOW;
LCD_SHOW_STRING_XC(X_TBH,Y_TBH,listnhac[tt_baihat]);
LCD_SHOW_NUM(X_TBH+100,Y_TBH,fsrc.fsize,8,16);

while(1)
{
    CHINH_VOLUMN();
    LCD_SHOW_NUM(X_TBH+180,Y_TBH,datasize,8,16);
    do
    {
        TOUCH_PLAY();
    }while(TT_BAIHAT_ON_OFF==0);

    if(datasize>=fsrc.fsize)    //datasize nhay de so sanh ket thuc bai hat
    {
        flag=1;
    }
    if(flag==1)
    {
        flag=0;
        XU_LY_BAI_HAT_KE();
    }

    if (TT_BAIHAT_ON_OFF==1)

```

```

        {
            datasize+=512;
            f_read(&fsrc,buffer,512,&br);
            for(i=0;i<16;i++)
            {
                while(!(GPIOC->IDR & MP3_DREQ));
                Vs1003_DATA_Write(buffer+i*32);
                TOUCH_DIEUKHIEN();
            }
        }
    }
}

```

- f. Tiến hành biên dịch và nạp.
- g. Quan sát kết quả:
- h. Giải thích chương trình: bạn hãy tự tìm hiểu.
- i. Cho phép thay đổi:

**Bài tập 503.** Hãy thêm phần hiển thị thanh trượt thời gian của bài hát và có thể điều chỉnh được.

Tạo thư mục “BAI\_503\_MUSIC\_HT\_THOIGIAN” để lưu project.

Gợi ý: lấy dung lượng của bài hát chia cho 1 frame là 32 byte sẽ được số frame chính là chiều dài của cả thanh trượt.

Khoảng cách trượt chính bằng chiều dài của cả thanh trượt chia cho số frame.

Mỗi lần gọi 1 frame 32 byte thì tăng thanh trượt đúng bằng khoảng cách đã tính.

**Bài tập 504.** Không sử dụng các biểu tượng điều khiển bằng hình ảnh mà hãy vẽ các biểu tượng đã vẽ ở chương trước để điều khiển.

Tạo thư mục “BAI\_504\_MUSIC\_ICON\_DK” để lưu project.

## IV. CÁC THƯ VIỆN

Phần này trình bày và giải thích các thư viện có liên quan nhưng quá nhiều nên không trình bày, các bạn có thể xem ở máy tính.