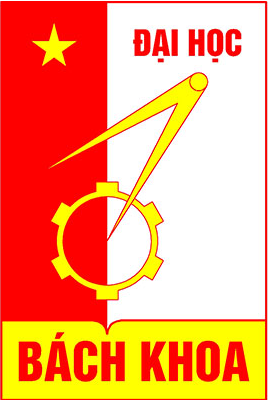
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**Viện Công Nghệ Thông Tin & Truyền Thông**



**BÁO CÁO BÀI TẬP LỚN MÔN HỌC**

**Project 2**

**Đề tài:** **Tìm hiểu và lập trình trên Kit STM32F4 Discovery, ghép nối màn hình TFT LCD 320 QVT.**

Sinh viên thực hiện : Đinh Đắc Cường MSSV 20150481

Giảng viên hướng dẫn : TS. Ngô Lam Trung

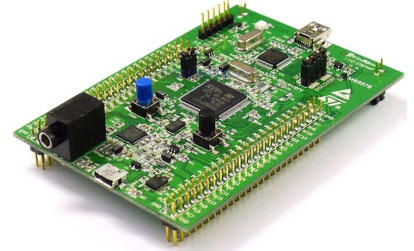
**Hà Nội, tháng 1 năm 2019**

**GIỚI THIỆU VỀ PROJECT**

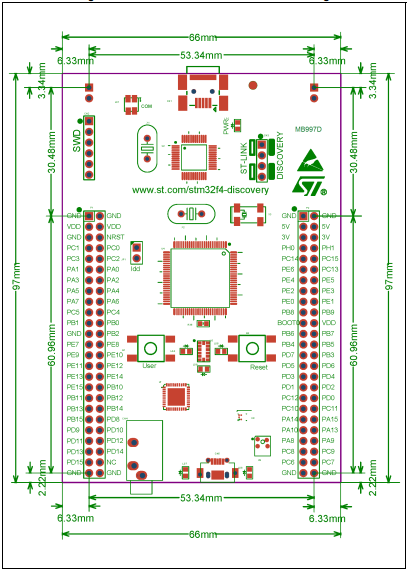
* Lập trình thư viện cho KIT STM32F4 Discovery để ghép nối màn hình TFT LCD 320 QVT. Sử dụng các thư viện vừa viết lập trình game Flappy Bird.
* Công cụ phát triển: Cube MX, Keil v5, TrueStudio.
* Link mã nguồn và các thư viện của project: https://github.com/dinhdaccuong/Project2\_STM32F4DIS\_TFTLCD/tree/master/STM32F4DIS\_LCD\_Final

1. **Giới thiệu về Kit STM32F4 Discovery.**

* Vi điều khiển được trang bị chip 32-bit ARM Cortext – M4, xung nhịp tối đa 168 Mhz.
* Bộ nhớ flash 1M
* Bộ nhớ RAM 192Kb
* Mạch nạp chương trình ST-LINK/V2 được tích hợp trực tiếp trên board, giúp nạp và debug chương trình một cách thuận tiện và hiệu quả.
* Nguồn cung cấp cho board thông qua cáp USB hoặc nguồn 5V.
* Nguồn điện có thể cung cấp cho kit các ứng dụng bên ngoài: 5V hoặc 3V.
* Từ bảng mạch có thể cung cấp nguồn 5V hoặc 3.3V cho các thiết bị ngoại vi.
* Được trang bị cảm biến chuyển động LIS3DSH, ST MEMS 3 trục.
* Cảm biến âm thanh MP45DT02 ST-MEMS, mic cảm biến âm thanh vô hướng kĩ thuật số.
* Bộ chuyển đổi DAC âm thanh CS43L22.
* Tám đèn LED:
* LD1(đỏ/xanh lá cây) – tín hiệu giao tiếp ST-LINK/V2
* LD2(đỏ) báo hiệu nguồn 3V on.
* Bốn đèn LED: LD3(màu cam), LD4(xanh lá cây), LD5(đỏ), LD6(xanh dương). Người sử dụng tùy chỉnh.
* Hai LED USB OTG, LD7(xanh lá cây) và LD8(màu đỏ).
* Hai nút bấm User (màu xanh) và Reset (màu đen).
* USB OTG với cổng micro A-B.
* Header mở rộng cho tất cả LQFP100 I/O.
* Zack cắm 3.5mm lấy tín hiệu âm thanh.

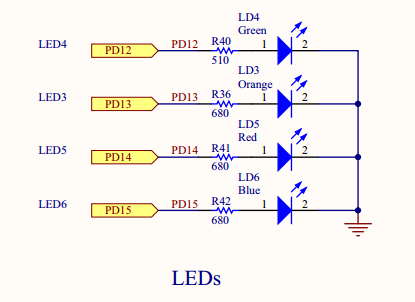


***Hình 1.1 Hình ảnh thực tế của Kit STM32F4 Discovery***

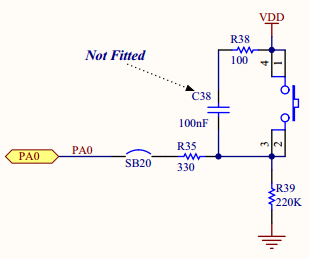


***Hình 1.2 Sơ đồ chân thực tế trên Kit STM32F4 Discovery***

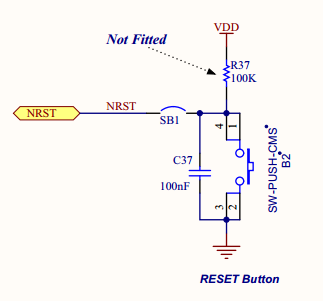
**Một số ghép nối cơ bản trên Kit STM32F4 – Discovery:**

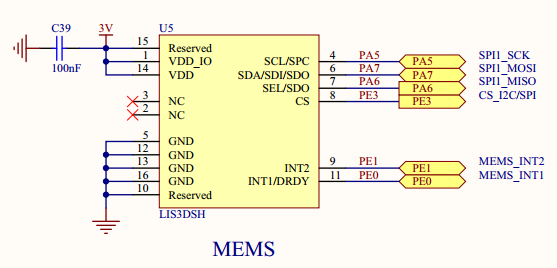


***Hình 1.3 Các LED 4-6 được nối trực tiếp với chân PD12-15 của vi điều khiển.***



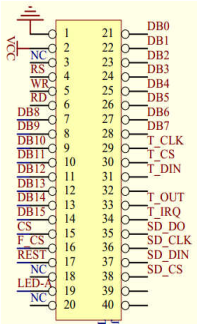
***Hình 1.4 Nút bấm User(màu xanh) được nối với chân PA0, nút bấm ở trạng thái mở thì PA0 được nối đất.***



***Hình 1.5 Ghép nối nút bấm RESET. Tích cực mức thấp.***

***Hình 1.6 Ghép nối cảm biến gia tốc LIS3DSH.***

1. **Giới thiệu màn hình TFT LCD 320 QVT.**



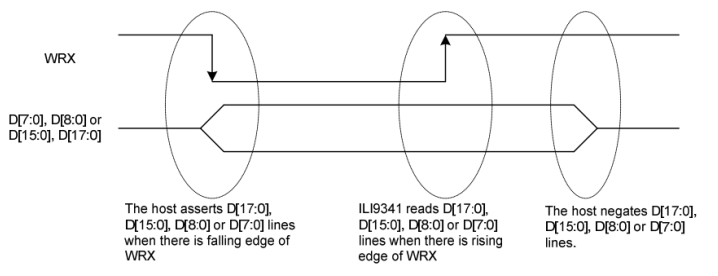


***Hình 2.1 Sơ đồ chân và hình ảnh thực tế của màn hình TFT LCD 320 QVT***

* **Đặc điểm kĩ thuật:**
* Điện áp hoạt động 3.3V.
* Độ phân giải 240x320 với 262k màu (16bit màu).
* Chuẩn giao tiếp 2 chế độ 8 bit và 16 bit.
* Cảm ứng điện trở.
* Hỗ trợ khe cắm thẻ nhớ bằng giao tiếp SPI.
* IC điều khiển hiển thị màn hình SSD1289.
* IC điều khiển cảm ứng XPT2046.
* **Mô tả các chân giao tiếp của màn hình TFT LCD.**

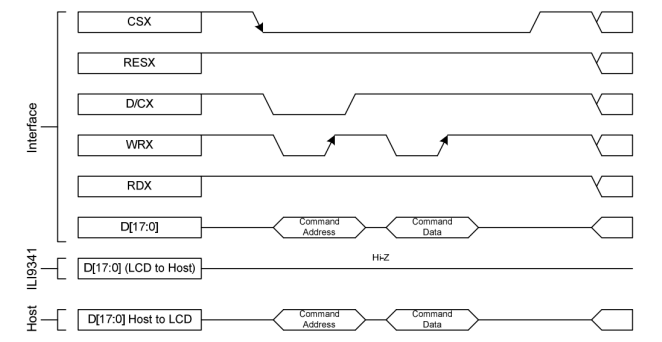
|  |  |  |
| --- | --- | --- |
| **Thứ tự chân** | **Tên chân** | **Chức năng** |
| 1 | GND | Nối GND |
| 2 | VCC | Nối nguồn 3.3V |
| 3 | NC | Not connect |
| 4 | RS | RS = 0 gửi địa chỉ thanh ghi LCD  RS = 1 ghi dữ liệu vào thanh ghi LCD |
| 5 | WR | Ghi dữ liệu, sường lên của xung trên chân WR sẽ ghi data lên LCD. |
| 6 | RD | Đọc dữ liệu, sườn lên của xung trên chân RD đọc trạng thái LCD. |
| 7-14 | DB8 – DB15 | 8 bit cao dữ liệu |
| 15 | CS | Chipselect – tích cực mức thấp |
| 17 | REST | Reset LCD – tích cực mức thấp |
| 18 | NC | Not connect |
| 19 | LED\_A | Chân anot đèn nền. |
| 20 | NC | Not connect |
| 21-27 | DB0 – DB7 | 8 bit thấp dữ liệu |
| 28 | T\_CLK | Xung đồng bộ giao tiếp SPI IC XPT2046 |
| 29 | T\_CS | Slave Select giao tiếp SPI IC XPT2046 |
| 30 | T\_DIN | Slave In giao tiếp SPI IC XPT2046 |
| 31 | NC | Not connect |
| 32 | T\_OUT | Slave Out giao tiếp SPI IC XPT2046 |
| 33 | T\_IRQ | Được kéo xuống mức thấp khi có tín hiệu cảm ứng từ màn hình. |
| 34 | SD\_DO | Slave Out giao tiếp SPI với thẻ nhớ |
| 35 | SD\_CLK | Xung đồng bộ giao tiếp SPI với thẻ nhớ. |
| 36 | SD\_DIN | Slave In giao tiếp SPI với thẻ nhớ. |
| 37 | SD\_CS | Slave Select giao tiếp SPI với thẻ nhớ |

***Bảng 2.1 Mô tả các chân giao tiếp của màn hình LCD***

* **Ghi dữ liệu vào thanh ghi IC SSD1289**

***Hình 2.2 Chu kì ghi dữ liệu của IC SSD1289***

* Chân WR điều khiển ghi dữ liệu, ở sườn xuống, khi tín hiệu WRX xuống mức thấp, vi xử lí chốt dữ liệu lên bus data, ở sườn lên khi tín hiệu WRX kéo lên mức cao , SSD1289 đọc dữ liệu trên bus data.
* SSD1289 có 2 chế độ ghi: Ghi lệnh và ghi dữ liệu được điều khiển bằng chân RS.
* Nếu RS = 0 : Ghi lệnh (gửi địa chỉ thanh ghi của IC SSD1289)
* Nếu RS = 1 : Ghi dữ liệu vào thanh ghi IC SSD1289.



***Hình 2.3 Quá trình ghi dữ liệu của IC SSD1289***

**RESX – RESET**

**D/CX – RS**

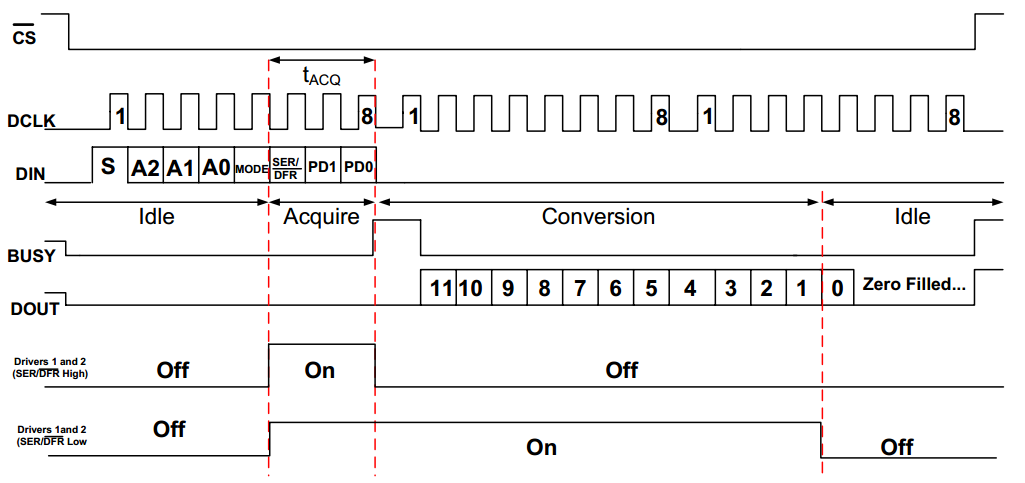
**WRX – WR**

* Quá trình ghi lệnh:
* Chân CS tư mức cao xuống mức thấp.
* Chân RESET giữ ở mức cao.
* Chân RS từ mức cao xuống mức thấp.
* Chân WR từ mức cao xuống mức thấp.
* Chốt địa chỉ thanh ghi trên bus dữ liệu D[15:0].
* Chân WR từ mức thấp lên mức cao để ghi lệnh và LCD.
* RS từ mức thấp lên mức cao để kết thúc ghi lệnh.
* Quá trình ghi dữ liệu:
* Chân CS từ mức cao xuống mức thấp.
* Chân RESX (RESET) giữ mức cao.
* Chân D/CX (RS) giữ mức cao.
* Chân WRX (WR) từ mức cao xuống mức thấp.
* Chốt data trên bus dữ liệu D[15:0].
* Chân WRX (WR) từ mức thấp lên mức cao để ghi dữ liệu vào LCD.
* Một số thanh ghi thường được sử dụng để lập trình cho màn hình TFT LCD.

|  |  |  |
| --- | --- | --- |
| **Tên thanh ghi** | **Địa chỉ thanh ghi** | **Chức năng** |
| Horizontal RAM address position | 44h | Đặt giới hạn hiển thị theo chiều ngang từ cột x1 đến cột x2. |
| Vertical RAM address start position | 45h | Đặt giới hạn hiển thị theo chiều dọc từ hàng y1. |
| Vertical RAM address end position | 46h | Đặt giới hạn hiển thị theo chiều dọc đến hàng y2. |
| Set GDDRAM X address counter | 4Eh | Tọa độ x của pixel |
| Set GDDRAM Y address counter | 4Fh | Tọa độ y của pixel |
| Driver Output Control | 01h | Điều khiển chiều quét hình ảnh từ trái sang phải, từ trên xuống dưới và ngược lại. Có thể lật gương, xoay ngang/dọc màn hình từ thiết lập trên thanh ghi này. |
| Ram Data Write | 22h | Ghi dữ liệu vào RAM |

***Bảng 2.2 Một số thanh ghi thường sử dụng của IC SSD1289***

* IC điều khiển chức năng cảm ứng màn hình XPT2046.
* Khi có cảm ứng (chạm) trên màn hình thì tín hiệu trên chân T\_IRQ của màn hình được kéo xuống mức thấp, đồng thời IC sẽ tính được tọa độ x, y của điểm chạm. Sau khi nhận biết được tín hiệu thay đổi trên chân T\_IRQ, vi điều khiển sẽ kéo chân T\_CS xuống thấp, tiếp hành chuyển đổi DAC và truyền nhận dữ liệu với màn hình bằng giao tiếp SPI.
* Mỗi giao tiếp của bộ vi xử lí với IC mất 24 chu kì xung clock (Để đọc được cả 2 tọa độ x, y thì cần 2 lần giao tiếp, nghĩa là sẽ tốn 48 chu kì xung clock). Với mỗi giao tiếp, 8 xung đầu tiên truyền byte điều khiển thể hiện mode chuyển đổi cho IC, 12 xung tiếp theo thực hiện chuyển đổi ADC. Xung thứ 21 biểu thị kết thúc chuyển đổi và 3 xung còn lại không được sử dụng.



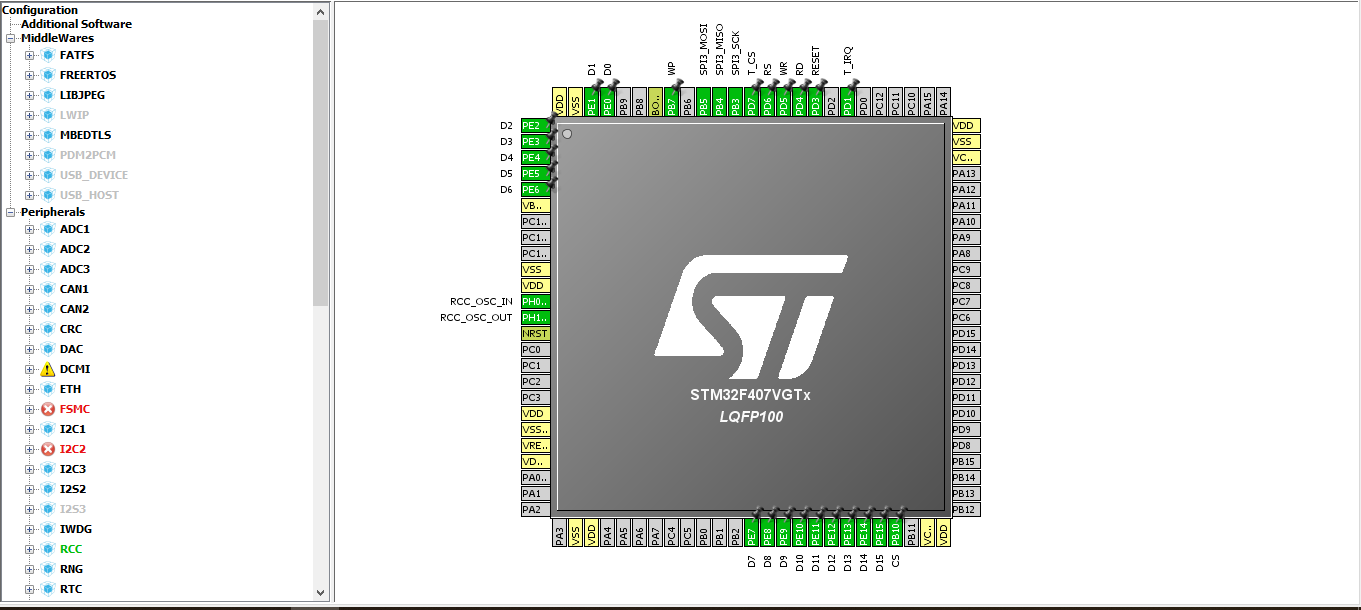
***Hình 2.4 Quá trình giao tiếp của vi điều khiển và IC XPT2046.***

1. **Nối ghép màn hình TFT LCD và Kit STM32F4 Discovery.**

|  |  |
| --- | --- |
| **STM32F4 Discovery** | **TFT LCD** |
| PE0 – PE15 | DB0 – DB15 |
| PD1 | T\_IRQ |
| PD3 | RESET |
| PD4 | WD |
| PD5 | WR |
| PD6 | RS |
| PD7 | T\_CS |
| PB3 – SPI3\_SCK | T\_CLK |
| PB4 – SPI3\_MISO | T\_OUT |
| PB5 – SPI3\_MOSI | T\_IN |
| PB10 | CS |
| 5V | VCC |
| GND | GND |

***Bảng 3.1 Sơ đồ ghép nối STM32F4 và TFT LCD***

* **Cấu hình các các chân giao tiếp với màn hình của Kit bằng Cube MX.**



***Hình 3.1 Cấu hình các chân giao tiếp màn hình của Kit bằng CubeMX***

* Chân PD1 – T\_IRQ chế độ input.
* Chọn bộ SPI3 giao tiếp IC XPT2046. Với hệ số chia xung clock là 16
* Các chân còn lại đều ở chế độ output no pull up – no pull down.
* Cấu hình thạch anh ngoài cho Chip.

****

***Hình 3.2 Hình ảnh thực tế của nối ghép.***

1. **Mô tả các hàm của thư viện đã lập trình.**

**Các thư viện đã lập trình hoạt động theo logic màn hình nằm ngang.**

**(0, 0) x**

****

**y**

***yyyyyyy Hình 4.1 Tọa độ quy định trên màn hình***

1. **Các hàm định nghĩa trong thư viện “ssd1289.h”.**

**Dsdsd**

* void SSD1289\_Init()

Hàm khởi tạo màn hình. Trước lập trình cho màn hình phải gọi hàm này trước tiên.

* void SSD1289\_Reset()

Hàm reset màn hình, xóa các thanh ghi của IC SSD1289

* void SSD1289\_Write\_Com(uint16\_t DH)

Hàm ghi lệnh (gửi địa chỉ thanh ghi)

Tham số: DH – Địa chỉ thanh ghi.

* void SSD1289\_Write\_Data(uint16\_t DH)

Hàm ghi dữ liệu vào thanh ghi.

**Tham số:** DH – Dữ liệu cần ghi.

* void SSD1289\_Write\_Com\_Data(uint16\_t com,uint16\_t data)

Hàm gửi đồng thời địa chỉ thanh ghi và dữ liệu thanh ghi

**Tham số:**

com: Địa chỉ thanh ghi.

data: Dữ liệu thanh ghi.

* void SSD1289\_Address\_Set(uint16\_t x1,uint16\_t y1,uint16\_t x2,uint16\_t y2)

Hàm thiết lập tọa độ sẽ vẽ lên màn hình (là một hình chữ nhật)

**Tham số:**

x1, y1: Tọa độ góc trên cùng bên trái màn hình.

x2, y2: Tọa độ góc trên cùng bên phải màn hình.

* void SSD1289\_Set\_Cursor(uint16\_t x\_pos, uint16\_t y\_pos)

Hàm chuyển vị trí trí sắp vẽ đến tọa độ x\_pos, y\_pos.

* void SSD1289\_Write\_Pixel(uint16\_t x, uint16\_t y, uint16\_t color)

Hàm vẽ 1 pixel có tọa độ x, y và màu color.

* void SSD1289\_Write\_Pic(const uint16\_t \*pic)

Vẽ ra toàn màn hình, ảnh cần có kích thước 240x320

**Tham số:** \*pic trỏ đến mảng lưu hình ảnh.

* void SSD1289\_Clear\_Screen()

Hàm xóa màn hình thành màu trắng

* void SSD1289\_Write\_Back\_Ground(uint16\_t color)

Hàm xóa màn hình thành màu color

* void SSD1289\_Print\_Char(uint16\_t x\_pos, uint16\_t y\_pos, uint16\_t font\_size, uint16\_t color, uint16\_t back\_color, char ch)

Hàm in một kí tự ra màn hình.

**Tham số:** x\_pos , y\_pos tọa độ in.

font\_size: Kích thước font chữ

color: màu chữ

back\_color: màu nền chữ.

ch: kí tự muốn vẽ.

* void SSD1289\_Print\_String(uint16\_t x\_pos, uint16\_t y\_pos, uint16\_t dis\_char, uint16\_t dis\_line, uint16\_t font\_size, uint16\_t color, uint16\_t back\_color, char\* string)

Hàm in xâu kí tự ra màn hình.

**Tham số:**

X\_pos, y\_pos: Tọa độ cần in chữ.

Dis\_char: Khoảng cách giữa 2 chữ.

Dis\_line: Khoảng cách giữa 2 dòng

Color: màu chữ.

Back\_color: Màu nền chữ.

String: Chuỗi cần in.

* void SSD1289\_Draw\_H\_Line(uint16\_t x1, uint16\_t x2, uint16\_t y1, uint16\_t color)

Vẽ một đoạn thẳng màu color song song với trục x tại điểm y1 từ x1 đến x2

* void SSD1289\_Draw\_V\_Line(uint16\_t x1, uint16\_t y1, uint16\_t y2, uint16\_t color)

Vẽ một đoạn thẳng màu color song song với trục y tại điểm x1 từ y1 đến y2.

* void SSD1289\_Draw\_Rectangle(uint16\_t x1, uint16\_t y1, uint16\_t x2, uint16\_t y2, uint16\_t fill, uint16\_t color)

Vẽ hình chữ nhật có tọa độ góc trên bên trái (x1, y1), tọa độ góc dưới bên phải là (x2, y2).

**Tham số:**

Fill: bằng 1 tô hình chữ nhật. Khác 1 không tô.

Color: Màu tô và màu vẽ viền hình chữ nhật.

* void SSD1289\_Write\_Pic\_Coor(uint16\_t x1, uint16\_t y1, uint16\_t width, uint16\_t heigh, uint16\_t \*pic)

Vẽ một bức ảnh được trỏ bởi \*pic.

**Tham số:**

(x1, y1): Vị trí bắt đầu vẽ bức ảnh.

width: chiều rộng bức ảnh.

heigh: chiều cao bức ảnh.

* void SSD1289\_Print(uint16\_t x, uint16\_t y, char \*string)

Hàm in chuỗi kí tự được trỏ bới \*string tại tọa độ (x, y) với màu chữ trăng và nền chữ màu đen. Hàm này nên sử dụng trong trường hợp màu background là màu đen.

1. **Các hàm định nghĩa trong thư viện “xpt2046.h”**

* void Read\_IRQ(): Đọc tín hiệu từ chân T\_IRQ của màn hình.
* uint16\_t XPT2046\_Read(uint8\_t cmd): Đọc tọa độ điểm chạm từ màn hình.

cmd = 0x90: Đọc tọa độ x.

cmd = 0xD0: Đọc tọa độ y.

* uint16\_t XPT2046\_Calibrate(uint16\_t coor, uint16\_t r): Hàm tính tọa độ của điểm chạm trên màn hình hiển thị.

**Tham số:**

coor : tọa độ điểm chạm của màn hình cảm ứng được tính bằng hàm XPT2046\_Read.

r : bằng 1, tính tọa độ x, bằng 0 tính tọa độ y.