**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN ĐIỆN TỬ**

**-----🙡🕮🙣---**



**BÁO CÁO**

**ĐỀ TÀI:**

**GVHD:**

**Sinh viên thực hiện**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ và tên** | **Ghi chú** |
| **1** | **2111412** | **Phan Kế Vĩnh Hưng** |  |
| **2** | **2112588** | **Phạm Nguyễn Hoàng Tuấn** |  |
| **3** | **2112100** | **Phạm Nhật Quang** |  |

*TP.Hồ Chí Minh, 30 tháng 11 năm 2024*

**BẢNG PHÂN CÔNG NHIỆM VỤ**

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ và tên** | **Nhiệm vụ đảm nhận** | **Mức độ hoàn thành** |
| **2111412** | **Phan Kế Vĩnh Hưng** |  | 100% |
| **2112588** | **Phạm Nguyễn Hoàng Tuấn** |  | 100% |
| **2112100** | **Phạm Nhật Quang** |  | 100% |

**MỤC LỤC**

[1. GIỚI THIỆU 2](#_Toc183971162)

[1.1. Tổng quan 2](#_Toc183971163)

[1.2. Nhiệm vụ của đề tài 2](#_Toc183971164)

[2. LÝ THUYẾT 3](#_Toc183971165)

[2.1. Giới thiệu về FreeRTOS 3](#_Toc183971166)

[2.1.1 Giới thiệu chung 3](#_Toc183971167)

[2.1.2 Các Tính Năng Của FreeRTOS 3](#_Toc183971168)

[2.2. Giới thiệu về Kit phát triển STM32F103C8T6 Blue Pill ARM Cortex-M3: 5](#_Toc183971169)

[2.3. Giới thiệu màn hình TFT ILI9341 9](#_Toc183971170)

[2.3.1 Giao thức SPI (Serial Peripheral Interface) 9](#_Toc183971171)

[2.3.2 Giao tiếp màn hình ILI9341 9](#_Toc183971172)

[3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG 14](#_Toc183971173)

[3.1. Yêu cầu thiết kế 14](#_Toc183971174)

[3.2. Phân tích thiết kế 15](#_Toc183971175)

[4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM 16](#_Toc183971176)

[4.1. Lưu đồ giải thuật chi tiết 16](#_Toc183971177)

[4.2. Phân tích lưu đồ 17](#_Toc183971178)

[5. KẾT QUẢ THỰC HIỆN 18](#_Toc183971179)

[5.1. Sản phẩm thực hiện 18](#_Toc183971180)

[5.2. Số liệu đo đạc 18](#_Toc183971181)

[6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 19](#_Toc183971182)

[6.1. Kết luận: 19](#_Toc183971183)

[6.2. Hướng phát triển và nâng cao 19](#_Toc183971184)

[7. TÀI LIỆU THAM KHẢO 20](#_Toc183971185)

**TÓM TẮT SẢN PHẨM**

# 1. GIỚI THIỆU

## 1.1. Tổng quan

***Mục tiêu cần nghiên cứu:***

## 1.2. Nhiệm vụ của đề tài

# 2. LÝ THUYẾT

## 2.1. Giới thiệu về FreeRTOS

### 2.1.1 Giới thiệu chung

FreeRTOS là một hệ điều hành thời gian thực mã nguồn mở, được phát triển lần đầu tiên bởi Richard Barry vào năm 2003 và hiện nay được duy trì và phát triển bởi Amazon Web Services (AWS). FreeRTOS đặc biệt thích hợp cho các hệ thống nhúng với tài nguyên hạn chế như bộ nhớ và bộ xử lý, được sử dụng rộng rãi trong các ứng dụng yêu cầu xử lý các tác vụ thời gian thực như điều khiển robot, hệ thống tự động hóa, và Internet of Things (IoT).

FreeRTOS cung cấp một nền tảng ổn định và hiệu quả cho việc phát triển các ứng dụng nhúng, với các tính năng quản lý đa nhiệm và đồng bộ hóa các tác vụ, giao tiếp giữa các thiết bị và quản lý bộ nhớ. Với mã nguồn mở và miễn phí, FreeRTOS mang lại lợi thế về tính linh hoạt và khả năng tùy biến cao cho người sử dụng.

### 2.1.2 Các Tính Năng Của FreeRTOS

FreeRTOS cung cấp một loạt các tính năng quan trọng giúp tối ưu hóa việc phát triển phần mềm cho các hệ thống nhúng yêu cầu tính thời gian thực.

**2.1.2.1 Quản Lý Tác Vụ (Task Management)**

Quản lý tác vụ là tính năng cơ bản của FreeRTOS. Hệ điều hành này hỗ trợ cơ chế đa nhiệm, cho phép nhiều tác vụ (tasks) chạy đồng thời, mỗi tác vụ có thể có mức độ ưu tiên khác nhau. Mỗi tác vụ được cấu hình để thực hiện một công việc cụ thể và có thể được lập trình để chạy theo chu kỳ hoặc theo yêu cầu. FreeRTOS có thể chuyển đổi giữa các tác vụ một cách nhanh chóng (context switch) dựa trên sự kiện hoặc thời gian trôi qua, đảm bảo rằng các tác vụ quan trọng được xử lý đúng hạn.

**2.2.2.2 Quản Lý Bộ Nhớ (Memory Management)**

FreeRTOS cung cấp các công cụ quản lý bộ nhớ động và tĩnh. Bộ nhớ động cho phép người dùng cấp phát và giải phóng bộ nhớ trong quá trình chạy, trong khi bộ nhớ tĩnh có thể được cấu hình ngay từ đầu và không thay đổi trong suốt quá trình thực thi. FreeRTOS cung cấp các API để người dùng dễ dàng quản lý bộ nhớ, đồng thời bảo vệ hệ thống khỏi các lỗi như tràn bộ nhớ.

**2.3.2.3 Cơ Chế Đồng Bộ (Synchronization)**

FreeRTOS cung cấp các công cụ đồng bộ hóa mạnh mẽ, bao gồm semaphore, mutex, và queue, để đồng bộ hóa các tác vụ và quản lý việc chia sẻ tài nguyên giữa các tác vụ. Những công cụ này giúp tránh các vấn đề như deadlock, race condition trong các hệ thống đa nhiệm và đảm bảo rằng dữ liệu giữa các tác vụ được đồng bộ hóa một cách an toàn.

**2.4.2.4 Hỗ Trợ Nhiều Giao Thức và Thiết Bị Ngoại Vi**

FreeRTOS hỗ trợ nhiều giao thức giao tiếp và thiết bị ngoại vi, bao gồm UART, SPI, I2C, CAN, và nhiều chuẩn khác. Điều này cho phép hệ thống nhúng có thể dễ dàng giao tiếp với các thiết bị ngoại vi như cảm biến, bộ điều khiển, bộ nhớ, và các thiết bị kết nối khác trong một hệ thống phân tán.

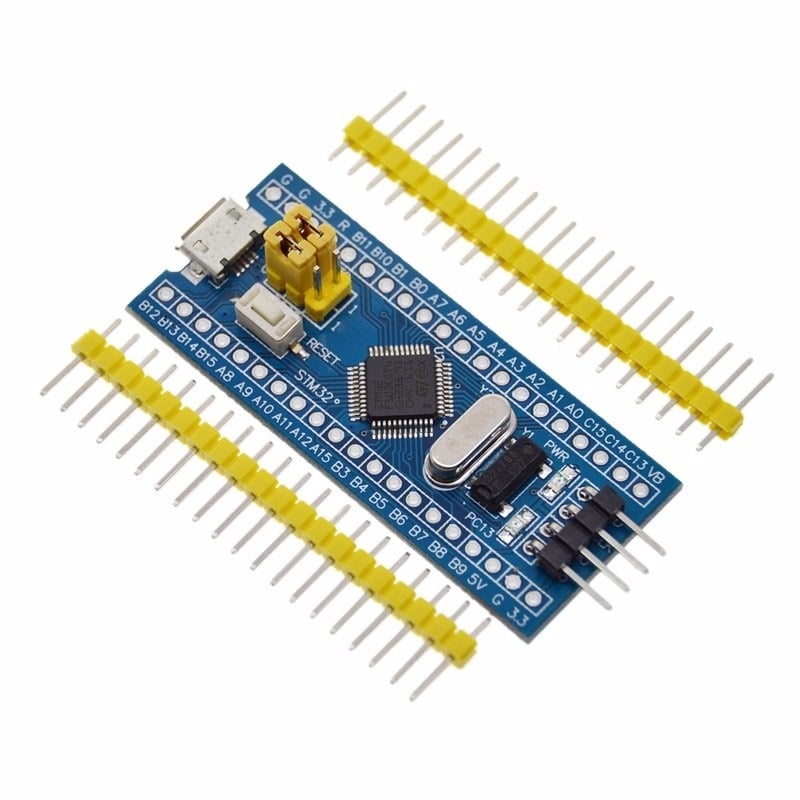
**2.5.2.5 Tính Mở và Miễn Phí (Open Source)**

FreeRTOS hoàn toàn mã nguồn mở và miễn phí, cho phép các nhà phát triển tùy chỉnh và mở rộng hệ điều hành để phù hợp với các yêu cầu cụ thể của dự án. Điều này giúp FreeRTOS trở thành một công cụ lý tưởng cho các ứng dụng nhúng có ngân sách hạn chế hoặc yêu cầu tính linh hoạt cao.

## 2.2. Giới thiệu về Kit phát triển STM32F103C8T6 Blue Pill ARM Cortex-M3:

STM32F103C8T6 Blue Pill được thiết kế để cung cấp một nền tảng phát triển ổn định cho các ứng dụng vi điều khiển, đặc biệt là trong các dự án yêu cầu hiệu suất cao, như điều khiển động cơ, cảm biến, robot, và các hệ thống tự động hóa. Bo mạch này sử dụng vi xử lý ARM Cortex-M3 với tốc độ xung nhịp lên đến 72 MHz, bộ nhớ Flash 64KB và RAM 20KB, cung cấp đủ tài nguyên cho nhiều ứng dụng phức tạp.

Một trong những điểm mạnh của STM32F103C8T6 Blue Pill là khả năng lập trình linh hoạt, hỗ trợ nhiều môi trường lập trình phổ biến như STM32CubeIDE và Arduino IDE. Điều này cho phép các lập trình viên dễ dàng phát triển ứng dụng, từ những dự án cơ bản đến những ứng dụng yêu cầu tính toán phức tạp.

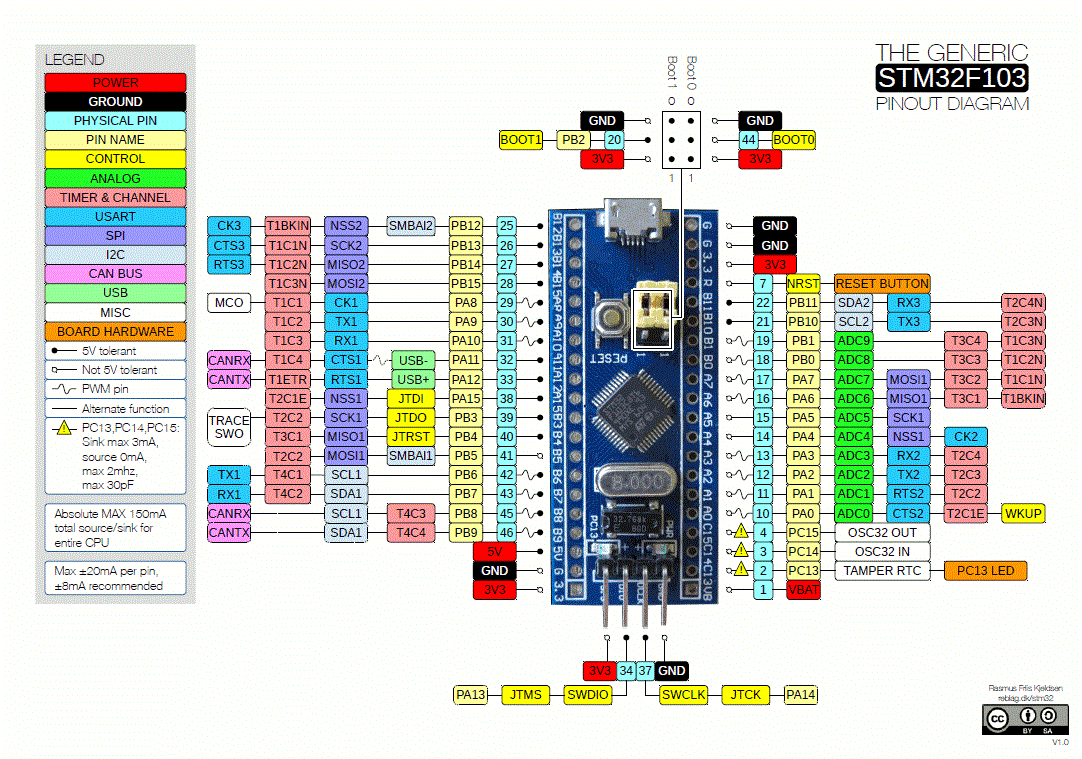


Hình 1: Kit phát triển STM32F103C8T6 Blue Pill ARM Cortex-M3

**Thông Số Kỹ Thuật STM32F103C8T6 Blue Pill**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Thông số** |  |  | | --- | |  | | | **Chi tiết** | | --- |  |  | | --- | |  | |
| |  | | --- | | **Kích thước bo mạch** |  |  | | --- | |  | | |  | | --- | | 53mm x 32mm |  |  | | --- | |  | |
| |  | | --- | | **Tốc độ xung nhịp** |  |  | | --- | |  | | |  | | --- | | 72 MHz |  |  | | --- | |  | |
| |  | | --- | | **Điện áp hoạt động** |  |  | | --- | |  | | 3.3V |
| |  | | --- | | **Điện áp vào** |  |  | | --- | |  | | |  | | --- | | 5V (chân 5V vào) |  |  | | --- | |  | |
| |  | | --- | | **Bộ nhớ Flash** |  |  | | --- | |  | | |  | | --- | | 64 KB |  |  | | --- | |  | |
| |  | | --- | | **Bộ nhớ RAM** |  |  | | --- | |  | | |  | | --- | | 20 KB |  |  | | --- | |  | |
| |  | | --- | | **Cổng I/O** |  |  | | --- | |  | | |  | | --- | | 37 chân GPIO, hỗ trợ nhiều giao thức |  |  | | --- | |  | |
| |  | | --- | | **Bộ chuyển đổi ADC** |  |  | | --- | |  | | |  | | --- | | 12-bit, 10 kênh ADC |  |  | | --- | |  | |
| |  | | --- | | **Tốc độ truyền** |  |  | | --- | |  | | |  | | --- | | Tùy thuộc vào giao thức (UART, SPI, I2C) |  |  | | --- | |  | |
| |  | | --- | | **Nhiệt độ hoạt động** |  |  | | --- | |  | | -40°C đến 85°C |

**Các Chân Cơ Bản Trên STM32F103C8T6 Blue Pill**



Hình 2: Sơ đồ chân của STM32F103

* **Chân nguồn (Power pins)**: STM32F103C8T6 Blue Pill có 2 chân cấp nguồn, bao gồm một chân 5V và một chân 3.3V, cung cấp điện áp cho bo mạch và các thiết bị ngoại vi.
* **Chân GND (Ground)**: Các chân nối đất giúp hoàn thiện mạch điện và ổn định các tín hiệu.
* **Chân GPIO (General Purpose I/O)**: Bo mạch này có 37 chân GPIO có thể lập trình được. Các chân này có thể được cấu hình cho các chức năng như I2C, SPI, UART, PWM, hoặc đơn giản là điều khiển các thiết bị ngoại vi như LED, nút nhấn, cảm biến, và động cơ.
* **Chân ADC (Analog to Digital Converter)**: STM32F103C8T6 hỗ trợ một bộ chuyển đổi ADC 12-bit với 10 kênh đầu vào, cho phép thu thập và xử lý tín hiệu analog từ các cảm biến với độ chính xác cao.
* **Chân UART**: Bo mạch có 3 kênh UART (USART1, USART2, và USART3), giúp dễ dàng giao tiếp với các thiết bị ngoại vi qua cổng serial với tốc độ lên đến 115200 bps.
* **Chân SPI và I2C**: Cung cấp giao tiếp tốc độ cao với các thiết bị ngoại vi khác như bộ nhớ EEPROM, cảm biến, và các module giao tiếp khác.
* **Chân PWM**: Các chân PWM trên STM32F103C8T6 có thể được sử dụng để điều khiển động cơ, đèn LED hoặc các thiết bị khác bằng cách điều chế độ rộng xung (PWM).
* **Chân Reset (RST)**: Được sử dụng để khởi động lại bo mạch và đưa vi điều khiển trở về trạng thái khởi tạo.

## 2.3. Giới thiệu màn hình TFT ILI9341

### 2.3.1 Giao thức SPI (Serial Peripheral Interface)

Giao thức SPI được phát triển vào những năm 1980 và được sử dụng rộng rãi để giao tiếp với các IC, đặc biệt là trong các ứng dụng hệ thống nhúng. Mặc dù việc thiết kế và triển khai giao thức SPI trên FPGA không đòi hỏi thời gian thiết kế dài, nhưng các thiết kế SPI sử dụng ít tài nguyên hơn là điều cần thiết đối với các FPGA chi phí thấp có lượng tài nguyên logic nhỏ. SPI bus được dùng để gửi dữ liệu giữa các vi điều khiển và các thiết bị ngoại vi ví dụ như EEPROM, ADC, DAC, RTC, SENSORS, SD CARD, LCD, RFID, CARD MODULE, WIRELESS RX/TX,…

SPI thường dùng trong internet, ethernet vì giảm được độ thiết kế phần cứng. Cứ tưởng tượng cả hơn 100 signal phải truyền đi mà phần cứng quá nhiều thì ảnh hưởng đến tiền bạc và có thể tạo nhiễu nên chúng ta sẽ dùng SPI với 4 dây. SPI thường dùng cho việc giao tiếp khoảng cách gần, ko yêu cầu high speed data. Có một điểm thuận lợi là chúng ta không giới hạn số lượng bit tùy chọn để có thể truyền và nhận.

SPI là loại bus đồng bộ data vì master hay slave sẽ được đồng bộ theo cạnh của clock. Tốc độ tối đa có thể lên 10Mbps.

Giao thức này bao gồm 4 dây giao tiếp: SCLK(serial clock used for all data comm), MOSI(master out slave in-output data line from master), MISO (master in slave out-input data line for master coming from slave), SS/CS (slave select/chip select – used to select a slave).

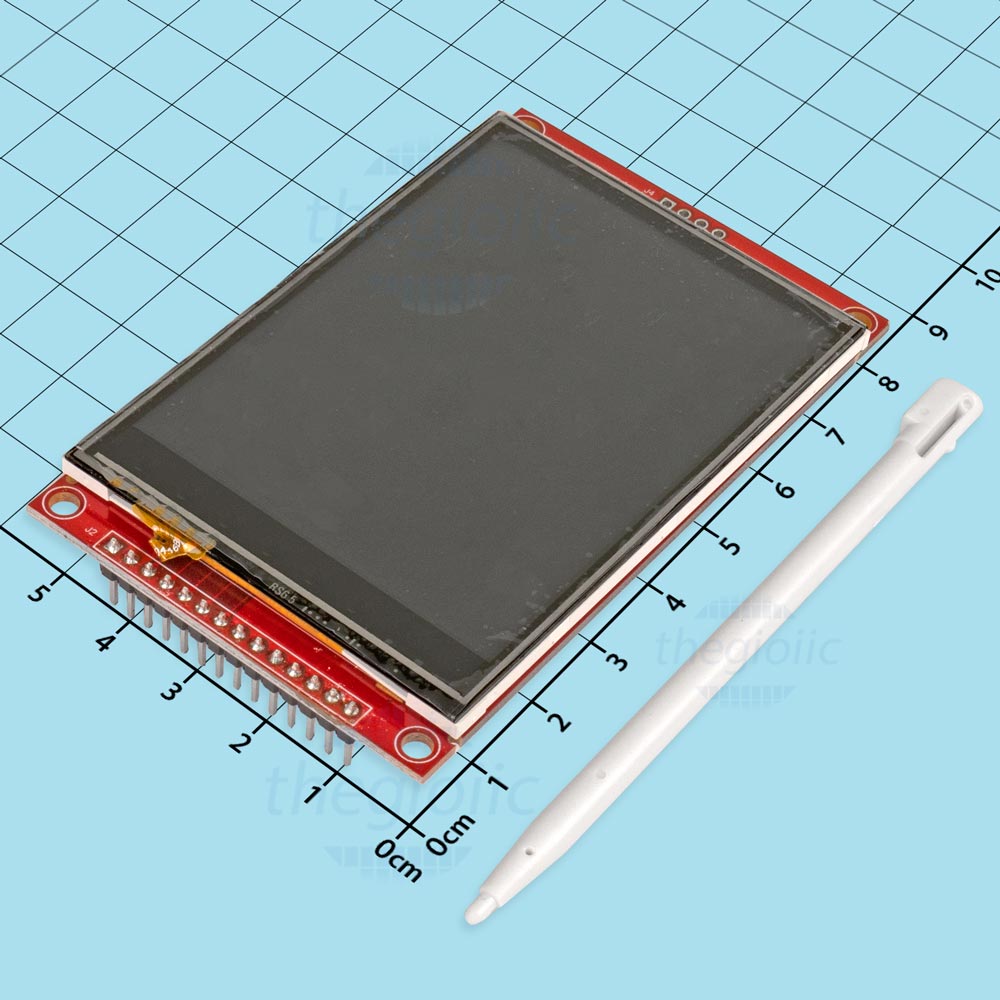
Đây là giao thức master cơ bản với một thiết bị bắt đầu tương tác với các slave còn lại. Đây là giao thức song công (cả master và slave có thể gửi datta cùng một lúc thông qua dây MOSI và MISO lần lượt).

### 2.3.2 Giao tiếp màn hình ILI9341

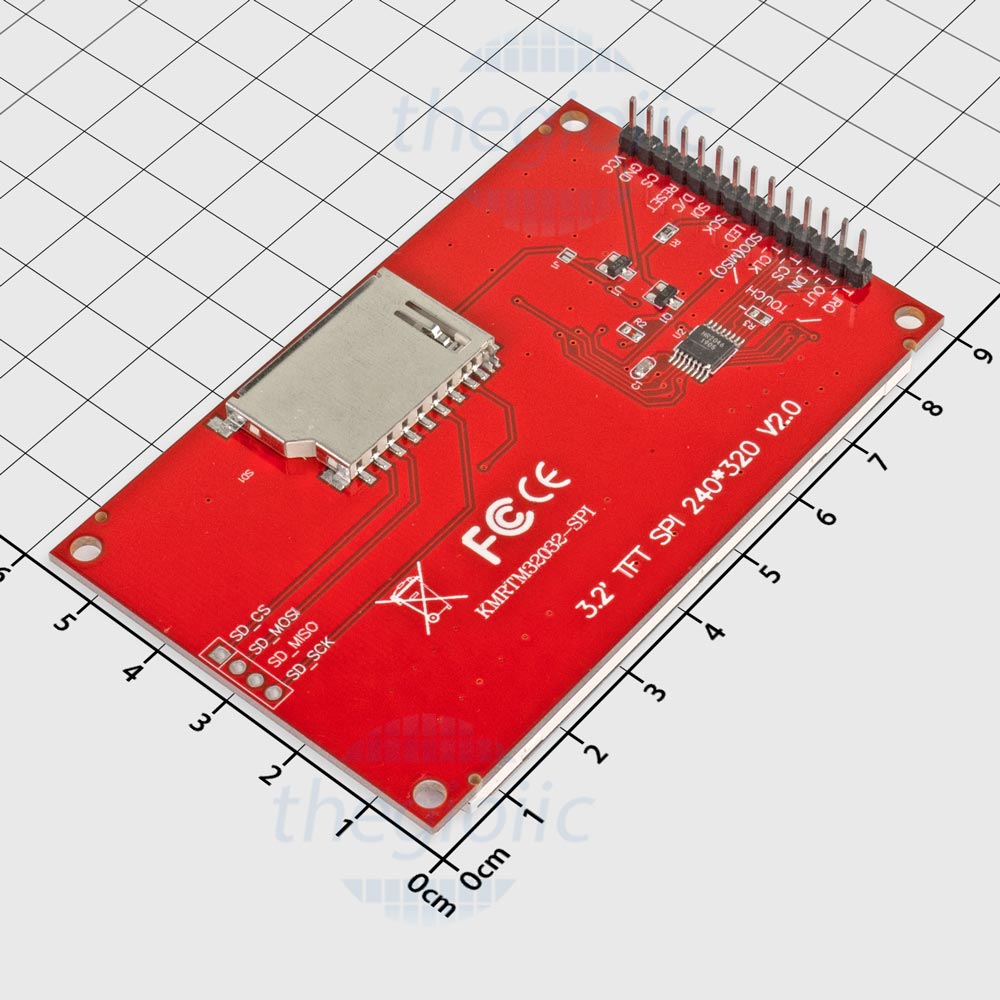
Màn hình cảm ứng LCD TFT Touch Screen 2.8 inch ILI9341 SPI Interface được sử dụng trong các ứng dụng điều khiển cảm ứng và hiển thị, màn hình sử dụng giao tiếp SPI nên rất dễ giao tiếp, giúp bạn xây dựng giao diện điều kiển cảm ứng trên màn hình 1 cách chuyên nghiệp.

**Thông số kỹ thuật:**

* Model: LCD TFT Touch Screen 2.8 inch ILI9341 SPI Interface
* Điện áp sử dụng: 3.3~5VDC
* Điện áp giao tiếp: TTL 3.3~5VDC
* IC Driver hiển thị: ILI9341 giao tiếp SPI.
* Cỡ màn hình: 2.8 inch
* Độ phân giải: 240 x 320 pixels
* IC Driver cảm ứng: XPT2046 giao tiếp SPI
* Tích hợp khe thẻ nhớ SD giao tiếp SPI.
* Kích thước hiển thị: 46(W) X 65(H)mm
* Kích thước toàn màn hình: 86 x 50 mm



Hình 3: Mặt trước của màn hình ILI9341

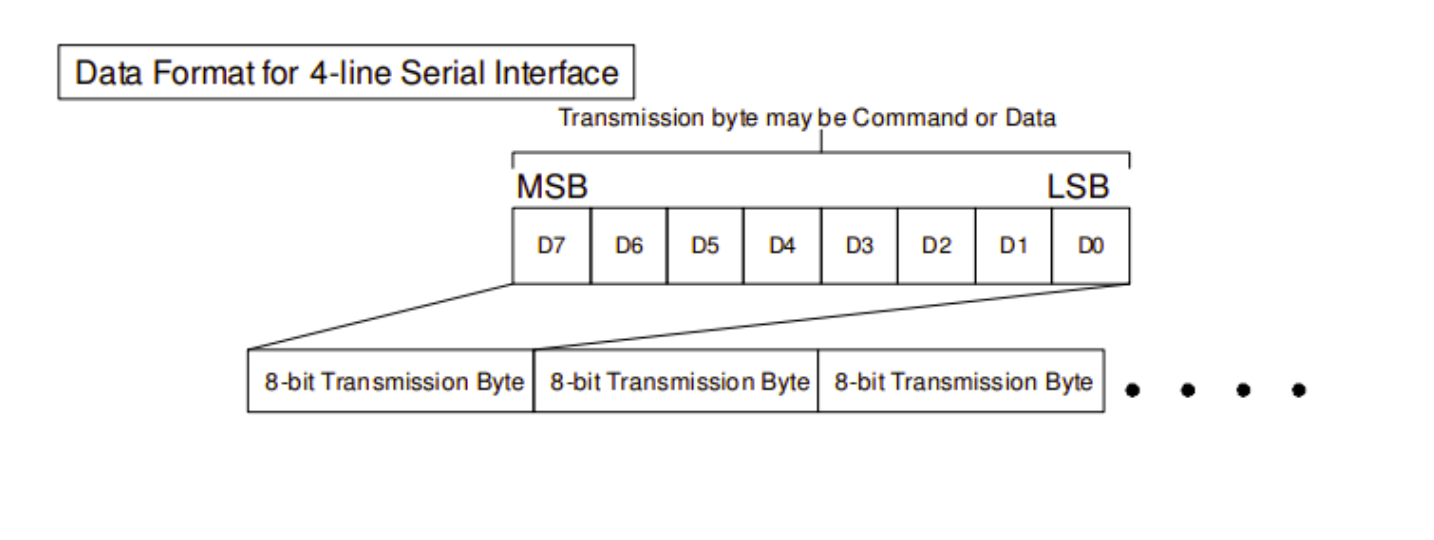


Hình 4: Mặt sau của màn hình ILI9341

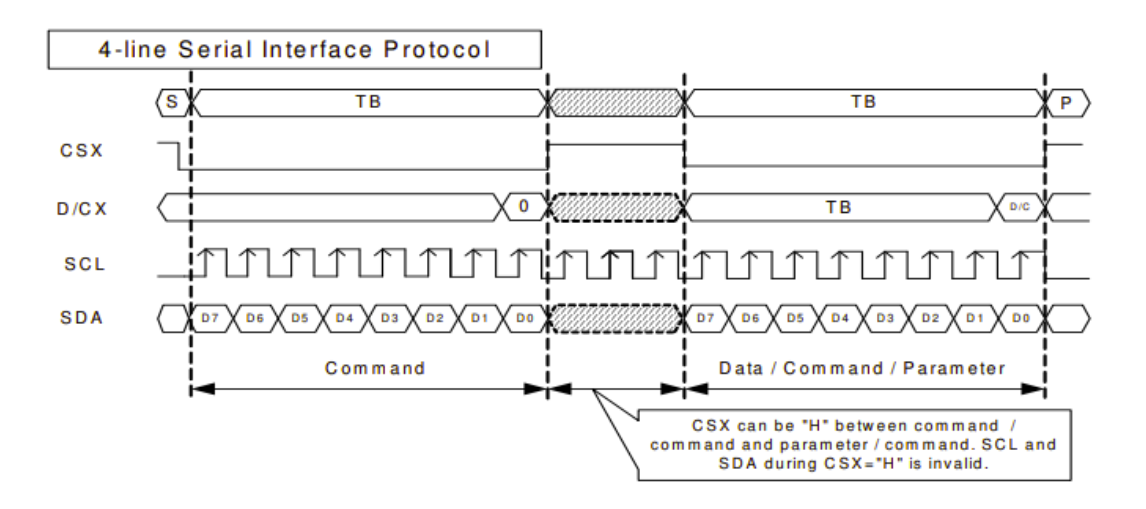
Để có thể tìm hiểu sâu hơn về việc điều khiển giao tiếp màn hình và xa hơn là kết hợp các ngoại vi khác, ví dụ là camera, cảm biến,… ta sẽ tập trung phân tích các chức năng chính sau: MCU interfaces, RGB interfaces, Display Data Format.

**2.3.2.1 MCU interfaces**

Màn hình ILI9341 cung cấp 4 loại MCU inteface, bao gồm là 8080 – I, 8080 – II giao tiếp chuỗi song song (series parallel interface), và giao tiếp nối tiếp bằng 3/4 đường (3-/4-line serial interface). Trong đề tài này, em sẽ chọn 4-line serial interface, bao gồm CSX, SDA, SCL, D/CX để giao tiếp với chuỗi dữ liệu đầu vào nối tiếp



Hình 5: Sơ đồ truyền dữ liệu của 4-line serial interface



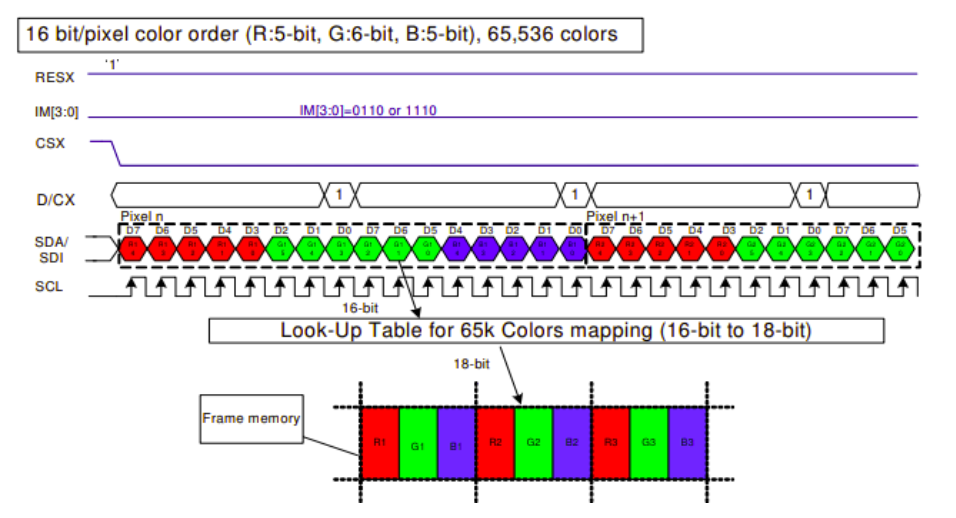
Hình 6: Sơ đồ timing của giao thức truyền 4-line serial interface

**2.3.2.2 RGB interfaces**

RGB interface là giao thức để cấu hình loại pixel để màn hình có thể phân loại màu ra. Giao thức này có các chế độ màu đa dạng, nếu chia theo số bit giao tiếp ta có thể có các chế độ như 18-/16-/6- bit RGB. Trong đề tài này, em sẽ chọn giao tiếp 16-bit RGB interface (65K màu).

**2.3.2.3 Display Data Format**

Sau khi đã chọn các thông số giao tiếp trên kia, tiếp theo chúng ta sẽ để ý đến cấu trúc hiển thị màn ảnh.



Như có thể thấy, 1 pixel ảnh sẽ cần 16 bit data, tương đương với 2 lần gửi dữ liệu từ master. Khi nhận dữ liệu này chân D/CX sẽ ở mức 1 vì đang nhận tín hiệu nhận dữ liệu. Sau khi nhận đủ dữ liệu 1 pixel thì dữ liệu sẽ được tra trong bản look-up để tra ra đúng màu và sau đó xuất ra 1 pixel trên màn hình.

# 3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

## 3.1. Yêu cầu thiết kế

Mục đích của đề tài yêu cầu là thiết kế một sản phẩm áp dụng RTOS để đo thông tin tọa độ thu được từ chức năng chạm cảm ứng và hiển thị thông tin lên LCD TFT ILI9341.

***Input and output:***

* Input (Đầu vào):

- Nguồn cung cấp:  3.3 - 10 V

- Dữ liệu pixel từ driver cảm ứng

* Output (Đầu ra):

- Hiển thị dữ liệu tọa độ lên màn hình LCD.

***Use Cases:***Người dùng cấp nguồn điện 3.3V cho sản phẩm. Ấn nút reset để bắt đầu lại quá trình đọc dữ liệu từ môi trường.

***Functions:***

* Thu thập dữ liệu pixel từ driver cảm ứng.
* Hiển thị kết quả đo lường trên màn hình LCD.

***Performance:***

* Độ chính xác của đo lường: Độ cảm ứng: ±5% g/cm.
* Tần suất đo lường: Chu kì lấy mẫu cảm ứng: ± 0.028ms/pixel.
* Thời gian đáp ứng: Dữ liệu từ cảm ứng sẽ được cập nhật và hiển thị qua LCD trong vòng 0.9µs.

***Manufacturing****:*

* Sản phẩm sẽ được lắp ráp từ các thành phần như STM32, màn hình LCD ILI9341, và các linh kiện hỗ trợ khác.
* Thiết bị cần được thiết kế để dễ dàng lắp ráp, bảo trì, và thay thế các linh kiện khi cần thiết.

***Power:***

* Thiết bị được cấp nguồn từ nguồn 3.3V DC.
* Tiêu thụ điện năng thấp, phù hợp để hoạt động liên tục.
* Có thể cân nhắc sử dụng pin dự phòng để tăng tính linh hoạt.

***Physical Size/ Weight:***

* Kích thước: Nhỏ gọn, phù hợp để đặt trên bàn hoặc gắn tường.
* Trọng lượng: Nhẹ, dễ dàng di chuyển và lắp đặt.

***Installation:***

* Sản phẩm đã được lắp ráp hoàn chỉnh. Có thể dễ dàng tháo lắp khi hư hỏng linh kiện và cần thay thế.
* Có hướng dẫn sử dụng kèm theo.

***Cerfication:***

* • Hệ thống phải được chứng nhận an toàn và đáng tin cậy để sử dụng có hiệu quả. Nó phải đáp ứng các tiêu chuẩn an toàn cần thiết.

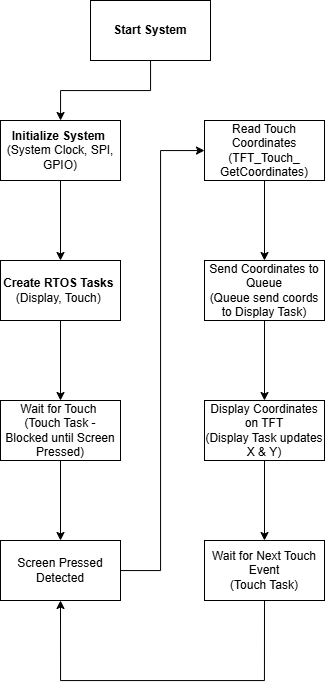
## 3.2. Phân tích thiết kế

A computer drawing of a red square

Description automatically generated

# 4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

## 4.1. Lưu đồ giải thuật chi tiết



## 4.2. Phân tích lưu đồ

**Start System**: Hệ thống bắt đầu hoạt động.

**Initialize System (System Clock, SPI, GPIO)**: Hệ thống khởi tạo các thành phần cơ bản như đồng hồ hệ thống, giao tiếp SPI (Serial Peripheral Interface), và GPIO (General Purpose Input/Output) để kết nối các phần cứng.

**Create RTOS Tasks (Display, Touch)**: Hệ thống tạo các tác vụ RTOS cho các chức năng khác nhau, bao gồm tác vụ hiển thị (Display) và tác vụ cảm ứng (Touch).

**Wait for Touch (Touch Task - Blocked until Screen Pressed)**: Tác vụ cảm ứng bị chặn (blocked), chờ đợi sự kiện màn hình cảm ứng được nhấn.

**Screen Pressed Detected**: Khi màn hình cảm ứng được nhấn, sự kiện được phát hiện.

**Read Touch Coordinates (TFT\_Touch\_GetCoordinates)**: Đọc tọa độ cảm ứng từ màn hình TFT để xác định vị trí chạm (X và Y).

**Send Coordinates to Queue (Queue send coords to Display Task)**: Tọa độ vừa đọc được gửi vào hàng đợi (queue) để truyền cho tác vụ hiển thị (Display Task).

**Display Coordinates on TFT (Display Task updates X & Y)**: Tác vụ hiển thị nhận tọa độ từ hàng đợi và cập nhật màn hình TFT, hiển thị các tọa độ chạm (X và Y) trên màn hình.

**Wait for Next Touch Event (Touch Task)**: Sau khi hoàn thành việc hiển thị, tác vụ cảm ứng tiếp tục chờ đợi sự kiện chạm tiếp theo.

# 5. KẾT QUẢ THỰC HIỆN

## 5.1. Sản phẩm thực hiện

## 5.2. Số liệu đo đạc

# 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 6.1. Kết luận:

Sau quá trình nghiên cứu, thiết kế và triển khai thực hiện, sản phẩm đã hoàn thành và đạt được các mục tiêu đề ra. Thiết bị có độ nhạy màn hình cao, và đáp ứng hiển thị ra màn hình một cách nhanh chóng. Tuy tính ứng dụng chưa có tính thực tế cao nhưng thông qua quá trình thực hiện, nhóm chúng em đã tích lũy một số kinh nghiệm về việc lựa chọn và điều khiển driver màn hình, để thực hiện giao tiếp với STM32F103C8T6. Thực hành làm việc nhóm và cùng nhau phân tích giải quyết các vấn đề phát sinh như sự cố mất kết nối, thông tin hiển thị sai, lỗi phần cứng.

Các vấn đề chính đã giải quyết trong dự án này:

- Thiết kế lắp ráp màn hình ILI9341 với STM32F103C8T6.

- Phát triển phần mềm với ứng dụng RTOS để có thể nâng cao hiệu quả đáp ứng độ nhạy của màn hình.

- Hiển thị thông tin tọa độ của điểm được nhấn ra màn hình LCD.

## 6.2. Hướng phát triển và nâng cao

Để đáp ứng nhu cầu ngày càng cao của người dùng và mở rộng khả năng ứng dụng của sản phẩm, các hướng phát triển và nâng cao chức năng của sản phẩm có thể được xem xét như sau:

***Bổ sung tính năng lưu trữ và phân tích dữ liệu:*** Tích hợp bộ nhớ trong hoặc khả năng kết nối với dịch vụ lưu trữ đám mây để lưu trữ dữ liệu đo lường trong thời gian dài. Kèm theo đó là các công cụ phân tích dữ liệu giúp người dùng dễ dàng nhận biết các xu hướng thay đổi của môi trường và đưa ra các quyết định hợp lý.

**Thiết kế một phần mềm có thể tính ứng dụng cao hơn**: Phát triển thêm các chương trình có tính thời gian thực, kết hợp RTOS như một trò chơi, hoặc nút bấm thông minh.

***Tích hợp các cảm biến khác:*** Phát triển thêm các phiên bản có tích hợp các cảm biến khác như cảm biến CO2, ánh sáng, áp suất, để cung cấp giải pháp đo lường môi trường toàn diện, đáp ứng nhu cầu của các lĩnh vực cụ thể như nông nghiệp công nghệ cao, quản lý tòa nhà, hoặc nghiên cứu khí tượng; module camera OV7670 để xử lý ảnh và video theo thời gian thực.

# 7. TÀI LIỆU THAM KHẢO