**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐH SƯ PHẠM KỸ THUẬT TP.HCM  
KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**HỆ THỐNG NHÚNG**

**HỆ THỐNG CHẤM CÔNG TỰ ĐỘNG BẰNG SINH TRẮC HỌC**

**GVHD: ThS Đinh Công Đoan**

|  |  |
| --- | --- |
| **Phan Hoàng Khải** | **20142004** |
| **Nguyễn Hoàng Thiên Bảo** | **20110440** |
| **Nguyễn Hồng Sơn** | **20110559** |

**Danh sách nhóm:**

# LỜI CẢM ƠN

Nhóm xin chân thành gửi lời cảm ơn đến quý Thầy Cô khoa Công nghệ thông tin,

nhất là quý Thầy Cô thuộc bộ môn Hệ thống nhúng đã tận tình chỉ dạy những kiến thức từ

cơ bản đến chuyên sâu để nhóm có thể tiến hành thực hiện và hoàn tất đồ án này.

Đặc biệt nhóm chúng em gửi lời cảm ơn sâu sắc nhất đến Thầy Đinh Công Đoan.

Thầy đã trực tiếp giảng dạy và tận tình hướng dẫn đồng thời tạo điều kiện tốt nhất cho

nhóm trong thời gian thực hiện đồ án. Đồng cảm ơn đến các anh chị, các bạn cùng khóa đã

cùng nhau san sẻ giúp đỡ và hợp tác cùng nhau trong quá trình thực hiện để đồ án, để đồ

án có thể hoàn thành nhanh nhất và đúng thời gian quy định.

Mặc dù trải qua và giải quyết những khó khăn và thử thách nhưng do kiến thức còn

hạn chế nên trong đồ án này chúng em còn nhiều thiếu sót về nội dung và hình thức. Nhóm

chúng em hy vọng Thầy thông cảm và tận tình đóng góp ý kiến quý báu để chúng em có

thể tiến hành cải tiến những mô hình về sau sao cho toàn diện nhất.

Một lần nữa chúng em xin chân thành cảm ơn!

MỤC LỤC

[LỜI CAM ĐOAN 1](#_Toc26283846)

[LỜI CẢM ƠN 2](#_Toc26283847)

[MỤC LỤC 3](#_Toc26283848)

[**LIỆT KÊ DANH SÁCH HÌNH** 6](#_Toc26283849)

[**LIỆT KÊ DANH SÁCH BẢNG** 7](#_Toc26283850)

[PHẦN MỞ ĐẦU 8](#_Toc26283851)

[**1.** **Lời nói đầu** 8](#_Toc26283852)

[**2.** **Đặt vấn đề** 8](#_Toc26283853)

[**2.1.** **Tóm lược những nghiên cứu trong và ngoài nước liên quan đến đề tài** 8](#_Toc26283854)

[**2.2.** **Tính cấp thiết cần nghiên cứu của đề tài** 9](#_Toc26283855)

[**2.3.** **Một số tài liệu có liên quan** 10](#_Toc26283856)

[**2.4.** **Lý do chọn đề tài** 11](#_Toc26283857)

[**2.5.** **Mục tiêu đề tài** 11](#_Toc26283858)

[**2.6.** **Đối tượng và phạm vi nghiên cứu** 11](#_Toc26283859)

[**2.7.** **Phương pháp nghiên cứu** 12](#_Toc26283860)

[**2.8.** **Nội dung đề tài** 12](#_Toc26283861)

[PHẦN NỘI DUNG 13](#_Toc26283862)

[CHƯƠNG 1: GIỚI THIỆU VI XỬ LÝ 13](#_Toc26283863)

[**1.1.** **Vi xử lý STM32F103XXXX** 13](#_Toc26283864)

[**1.2.** **Thông số kỹ thuật** 14](#_Toc26283865)

[CHƯƠNG 2: CẤU TRÚC CHUNG CỦA KIT 17](#_Toc26283866)

[**2.1. Cấu trúc Kit STM32F103C8T6** 17](#_Toc26283867)

[**2.2. Sơ đồ mạch chi tiết** 18](#_Toc26283868)

[CHƯƠNG 3: NHỮNG KIẾN THỨC LIÊN QUAN 19](#_Toc26283869)

[**3.1. Giao thức UART** 19](#_Toc26283870)

[**3.1.1. Giới thiệu** 19](#_Toc26283871)

[**3.1.2. Nguyên lý hoạt động** 21](#_Toc26283872)

[**3.2. Cảm biến vân tay R308** 26](#_Toc26283873)

[**3.2.1. Giới thiệu** 26](#_Toc26283874)

[**3.2.2. Thông số kỹ thuật** 27](#_Toc26283875)

[**3.2.3 Nguyên lý hoạt động** 27](#_Toc26283876)

[**3.2.4. Giao tiếp phần cứng** 28](#_Toc26283877)

[**3.2.5. Giao thức truyền gói dữ liệu** 29](#_Toc26283878)

[**3.3. LCD 16x2** 32](#_Toc26283879)

[**3.3.1. Giới thiệu** 32](#_Toc26283880)

[**3.3.2. Chức năng các chân** 32](#_Toc26283881)

[**3.4. Mạch nạp ST-LINK V2** 34](#_Toc26283882)

[**3.4.1. Công dụng** 34](#_Toc26283883)

[**3.4.2. Thông số kỹ thuật** 35](#_Toc26283884)

[**3.5. Module Joystick** 34](#_Toc26283882)

[**3.5.1. Công dụng** 34](#_Toc26283883)

[**3.5.2. Thông số kỹ thuật** 35](#_Toc26283884)

[**3.6. Module thời gian thực RTC DS1307** 34](#_Toc26283882)

[**3.6.1. Công dụng** 34](#_Toc26283883)

[**3.6.2. Thông số kỹ thuật** 35](#_Toc26283884)

[**3.7. KY-012 Mạch Buzzer 5V** 34](#_Toc26283882)

[**3.7.1. Công dụng** 34](#_Toc26283883)

[**3.7.2. Thông số kỹ thuật** 35](#_Toc26283884)

[**3.8. Mạch chuyển USB to TTL CP2102** 34](#_Toc26283882)

[**3.8.1. Công dụng** 34](#_Toc26283883)

[**3.8.2. Thông số kỹ thuật** 35](#_Toc26283884)

[**3.9. Module ghi đọc thẻ SD** 34](#_Toc26283882)

[**3.9.1. Công dụng** 34](#_Toc26283883)

[**3.9.2. Thông số kỹ thuật** 35](#_Toc26283884)

[CHƯƠNG 4: ỨNG DỤNG 36](#_Toc26283885)

[**4.1. Cách nạp chương trình vào Kit** 36](#_Toc26283886)

[**4.1.1. Cài đặt phần mềm Keil C** 36](#_Toc26283887)

[**4.1.2. Tạo project với Keli C** 38](#_Toc26283888)

[**4.1.3. Lập trình với Keli C** 41](#_Toc26283889)

[**4.1.4. Nạp chương trình vào Kit** 72](#_Toc26283890)

[**4.2. Mô tả ứng dụng** 72](#_Toc26283891)

[**4.2.1. Mô tả các chức năng** 72](#_Toc26283892)

[**4.2.2. Demo ứng dụng** 73](#_Toc26283893)

[CHƯƠNG 5: TỔNG KẾT 75](#_Toc26283894)

[**5.1. Kết quả** 75](#_Toc26283895)

[**5.2. Ưu nhược điểm** 75](#_Toc26283896)

[**5.2.1. Ưu điểm** 75](#_Toc26283897)

[**5.2.2. Nhược điểm** 75](#_Toc26283898)

[**5.3. Hướng phát triển** 75](#_Toc26283899)

[**TÀI LIỆU THAM KHẢO** 76](#_Toc26283900)

# PHẦN MỞ ĐẦU

1. **Lời nói đầu**

Ngày nay việc ứng dụng vi điều khiển, vi xử lý đang ngày càng phát triển rộng rãi và thâm nhập ngày càng nhiều vào các lĩnh vực kỹ thuật và đời sống xã hội. Tuy nhiên ứng dụng cho các hệ thống nhúng ngày nay không đơn giản chỉ dừng lại ở điều khiển đèn nhấp nháy, đếm số người vào/ra, hiển thị dòng thông báo trên matrix led hay điều khiển ON-OFF của động cơ... mà nó ngày càng trở nên phức tạp. Và với xu hướng tất yếu này cùng với sự phát triển mạnh mẽ của công nghệ chế tạo vi mạch, người ta đã tạo ra những vi điều khiển có cấu trúc mạnh hơn, đáp ứng thời gian thực tốt hơn, chuẩn hóa hơn so với các vi điều khiển 8-bit trước đây. Với sự phát triển mạnh mẽ của khoa học, đặc biệt là ngành điện tử, sự phát minh ra các linh kiện điện tử đã và đang ngày càng đáp ứng được yêu cầu của các hệ thống. Ưu điểm của việc sử dụng các linh kiện điện tử làm cho các hệ thống linh hoạt và đa dạng hơn, giá thành thấp hơn và độ chính xác cao hơn.

Sau nhiều năm học tập và nghiên cứu ở trường, chúng em đã được làm quen với các môn học chuyên ngành. Để áp dụng lý thuyết với thực tế học kỳ này chúng em đã chọn đồ án môn học hệ thống nhúng với đề tài “Xây dựng hệ thống chấm công tự động bằng sinh trắc học”. Tuy nhiên do kiến thức chuyên môn còn hạn chế, tài liệu tham khảo có giới hạn nên còn xảy ra nhiều sai sót. Chúng em rất mong mong thầy và các bạn góp ý bổ sung để đồ án của chúng em được hoàn thiện hơn và giúp chúng em hiểu biết hơn trong quá trình học tập.

1. **Đặt vấn đề**
   1. **Tóm lược những nghiên cứu trong và ngoài nước liên quan đến đề tài**

* Trong vài năm trở lại đây, một trong những xu hướng chủ yếu trong các thiết kế với vi điều khiển là sử dụng các chip ARM7 và ARM9 như một vi điều khiển đa dụng. Ngày nay các nhà sản xuất IC đưa ra thị trường hơn 240 dòng vi điều khiển sử dụng lõi ARM. Tập đoàn ST Microelectronic vừa cho ra mắt dòng STM32 vi điều khiển đầu tiên dựa trên nền lõi ARM Cortex-M3 thế hệ mới do hãng ARM thiết kế, lõi ARM Cortex-M3 là sự cải tiến của lõi ARM7 truyền thống. Dòng STM32 thiết lập các tiêu chuẩn mới về hiệu suất, chi phí, cũng như khả năng đáp ứng các ứng dụng tiêu thụ năng lượng thấp và tính điều khiển thời gian thực khắt khe.
* Khối trung tâm của STM32 là bộ xử lí Cortex-M3 là một vi điều khiển được tiêu chuẩn hóa gồm một CPU 32-bit, cấu trúc bus, đơn vị xử lí ngắt có hỗ trợ tính năng lồng ngắt vào nhau, hệ thống kiểm lỗi và tiêu chuẩn bố trí bộ nhớ.
* STM32 là một bước tiến quan trọng trên đường cong chi phí và hiệu suất của vi điều khiển nhúng (price/performance),. Với sự phát triển của công nghệ sản xuất, giá thành của các vi điều khiển STM32 ngày càng giảm, đồng thời hiệu suất và tính năng của chúng ngày càng được cải tiến và nâng cao giá chỉ gần 1 Euro với số lượng lớn, STM32 là sự thách thức thật sự với các vi điều khiển 8 và 16-bit truyền thống. STM32 đầu tiên gồm 14 biến thể khác nhau, được phân thành hai nhóm: dòng Performance có tần số hoạt động của CPU lên tới 72Mhz và dòng Access có tần số hoạt động của CPU lên tới 36Mhz. Các biến thể STM32 trong hai nhóm này tương thích hoàn toàn về cách bố trí chân (pin) và phần mềm, đồng thời kích thước bộ nhớ FLASH ROM có thể lên tới 128K và 20K SRAM.
* Dòng STM32 có hai nhánh, nhánh Performance hoạt động với xung nhịp lên đến 72Mhz và có đầy đủ các ngoại vi, nhánh Access hoạt động với xung nhịp tối đa 36Mhz và có ít ngoại vi hơn so với nhánh Performance.
* Về tình hình trong nước: ứng dụng hệ thống nhúng rất nhiều vào đời sống cũng như sản xuất như nhưng thiết bị tự động, thiết bị bảo mật bằng sinh trắc học…Tuy nhiên so với các nước phát triển thì chúng ta vẫn còn nhiều hạn chế.
  1. **Tính cấp thiết cần nghiên cứu của đề tài**
* Ngày nay với sự phát triển của ngành điện tử và ứng dụng điện tử đã giúp sự sáng tạo của con người trở thành hiện thực. Các lĩnh vực của cuộc sống đều áp dụng những thiết bị điện tử và dường như nhìn đâu trong gia đình chúng ta cũng có thiết bị điện tử. Ngành điện tử và ứng dụng điện tử đã tạo chỗ đứng và khẳng định được tầm quan trọng của mình đối với nhu cầu của con người.
* Với những ứng dụng cho các hệ thống nhúng ngày càng trở nên phổ biến: từ những ứng dụng đơn giản như điều khiển một chốt đèn giao thông định thời, đếm sản phẩm trong một dây chuyền sản xuất, điều khiển tốc độ động cơ điện một chiều, thiết kế một biển quảng cáo dùng Led ma trận, thiết bị xác thực bằng sinh trắc học. Đến các ứng dụng phức tạp như hệ thống điều khiển robot, bộ kiểm soát trong nhà máy hoặc hệ thống kiểm soát các máy năng lượng hạt nhân. Các hệ thống tự động trước đây sử dụng nhiều công nghệ khác nhau như các hệ thống tự động hoạt động bằng nguyên lý khí nén, thủy lực, rơle cơ điện, mạch điện tử số, các thiết bị máy móc tự động bằng các cam chốt cơ khí. Các thiết bị, hệ thống này có chức năng xử lý và mức độ tự động thấp so với các hệ thống tự động hiện đại được xây dựng trên nền tảng của các hệ thống nhúng.
* Trong nhiều năm trước, các dòng vi điều khiển 8051 được sinh viên dùng nhiều với tính năng đơn giản, dễ sử dụng; AVR được sử dụng nhiều trong các cuộc thi Robocon nhờ tốc độ sử lý khá cao, ổn định; PIC với ưu thế tốc độ cao, chi phí thấp hơn cũng được nghiên cứu, sử dụng nhiều, đặc biệt trong các cuộc thi lập trình tay nghề khu vực và thế giới. Nhưng trong một vài năm trở lại đây, có một dòng vi điều khiển mới, càng ngày càng nắm vị trí quan trọng trong các lĩnh vực đòi hỏi tốc độ xử lý cao như điện tử viễn thông, sản xuất các dòng diện thoại di động smartphone, giám sát, an ninh… Đó là họ vi điều khiển ARM. Với rất nhiều thế hệ ra đời, với nhiều tính năng , công dụng khác nhau. Với nhiều tính năng vượt trội của ARM và xu thế lựa chọn dòng vi điều khiển mới ở Việt Nam nên trong bài báo cáo cuối kỳ này, nhóm em thực hiện đề tài giao tiếp của STM32F103 với cảm ứng vân tay và màn hình LCD
  1. **Một số tài liệu có liên quan**

- Trước khi tìm hiểu đề tài, chúng ta cần trang bị những kiến thức căn bản liên quan đến hệ thống nhúng như board mạch, vi xử lý, vi điều khiển, các giao thức ở trên board nhúng

- Nắm vững nền tảng các môn học như Điện tử căn bản, Kiến trúc máy tính và hợp ngữ, sơ đồ nguyên lý và hoạt động của board mạch.

- Phần nội dung trong đề tài này có tham khảo kiến thức từ giáo trình chính thức bộ môn Hệ thống nhúng và nguồn tư liệu từ Internet.

* 1. **Lý do chọn đề tài.**

Hiện nay, với sự phát triển của các ngành công nghiệp thì nhu cầu về nguồn nhân lực là một điều không thể thiếu. Đặc biệt ở Việt Nam là một nước đang phát triển thì lượng dân số làm nhân viên và công nhân là rất lớn. Từ đó đặt ra bài toán lan giải là làm sao để quản lý nguồn nhân lực đó một cách hiệu quả cụ thể ở đây là việc chấm công nhân viên công nhân. Với việc chấm công bằng phương pháp điểm danh gọi tên hay ghi danh thì sẽ gây tốn rất nhiều thời gian đối với người quản lý lẫn nhân viên, công nhân. Để giải quyết bài toán này trên sự phát triển của ngành điện tử và công nghệ sinh trắc học thì người đã phát triển hệ thống chấm công nhân viên tự động bằng sinh trắc học như là vân tay, mẫu mắt. Đó là một đề tài rất hay và gây hứng thú với nhóm em, nên nhóm đã quyết định chọn đề tài “Hệ thống chấm công tự động bằng sinh trắc học” cụ thể ở đây nhóm chọn là dấu vân tay. Đây là ứng dụng cơ bản để tìm hiểu cách thức chấm công bằng vân tay và quá trình thực hiện.

* 1. **Mục tiêu đề tài**

Mục tiêu chính của đề tài này là khai thác một số chức năng có sẵn trong chip ARM STM32F103, cụ thể hơn là việc giao tiếp giữa cảm biến vân tay và màn hình LCD thông qua chip ARM STM32F103XXXX. Cấu trúc chung của KIT gồm thành phần, sơ đồ, demo... Giới thiệu những kiến thức liên quan được dùng trong đề tài. Chỉ ra các chức năng có sẵn trong chip được sử dụng trong đề tài và giải thích các chức năng. Trình bày ứng dụng, các bước tiến hành, cấu hình tham số để nạp chương trình vào kit. Tổng kết lại các những ưu nhược điểm của chip và chỉ ra các khó khăn trong quá trình làm đề tài và đưa ra hướng phát triển của đề tài.

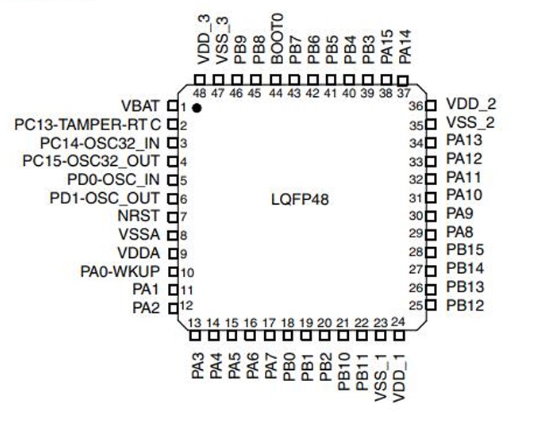
* 1. **Đối tượng và phạm vi nghiên cứu**
* **Đối tượng nghiên cứu:** 
  + Board STM32F103C8T6 của hãng ARM.
  + Các thiết bị hỗ trợ: cảm biến vân tay R308, màn hình LCD 16x2, Module Joystick, Module thời gian thực RTC DS1307, Module buzzer KY-012, Mạch chuyển USB to TTL CP2102, Module Ghi Đọc Thẻ SD
* **Phạm vi nghiên cứu:** 
  + Tìm hiểu cấu trúc chung của board STM32F103.
  + Nghiên cứu cách thức giao tiếp giữa board với cảm biến vân tay và màn hình LCD.
  + Viết code và nạp vào Kit.
  1. **Phương pháp nghiên cứu**
* Tìm hiểu lý thuyết về cách thức giao tiếp của board với cảm biến vân tay và màn hình LCD.
* Xây dựng thuật toán và thực hiện lập trình.
  1. **Nội dung đề tài**

Trong đề tài này, nhóm em sẽ giới thiệu về vi xử lý STM32F103C8T6 cũng như ứng dụng minh họa (chấm công bằng vân tay và hiển thị thông tin ra màn hình LCD). Và nhóm sẽ trình bày thêm phần nạp code vào KIT và demo ứng dụng.

# PHẦN NỘI DUNG

## CHƯƠNG 1: GIỚI THIỆU VI XỬ LÝ

* 1. **Vi xử lý STM32F103C8T6**

****

Hình 1:Sơ đồ chân STM32F103C8T6

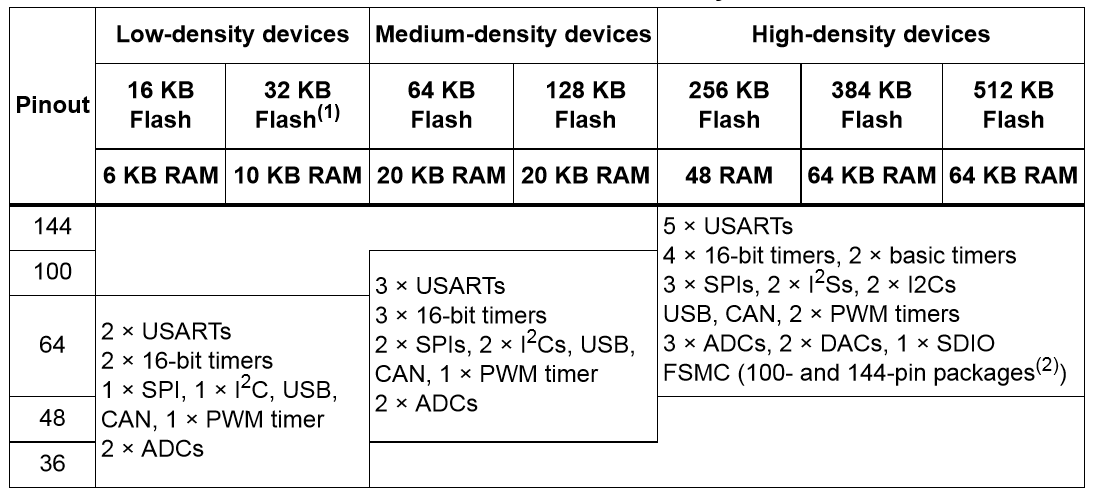


Hình 2: Hình ảnh thực tế của STM32F103C8T6

* 1. **Thông số kỹ thuật**

|  |  |
| --- | --- |
| Manufacturer Part Number | STM32F103C8T6 |
| Description | MCU ARM 64KB FLASH MEM LQFP-48 |
| Category | Integrated Circuits (ICs) |
| Family | Embedded - Microcontrollers |
| Series | STM32F1 |
| Core Processor | ARM® Cortex-M3™ |
| Core Size | 32-Bit |
| Speed | 72MHz |
| Connectivity | CAN, I²C, IrDA, LIN, SPI, UART/USART, USB |
| Peripherals | DMA, Motor Control PWM, PDR, POR, PVD, PWM, Temp Sensor, WDT |
| Number of I /O | 37 |
| Program Memory Size | 64KB (64K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 20K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2 V ~ 3.6 V |
| Data Converters | A/D 10x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C |
| Package / Case | 48-LQFP |

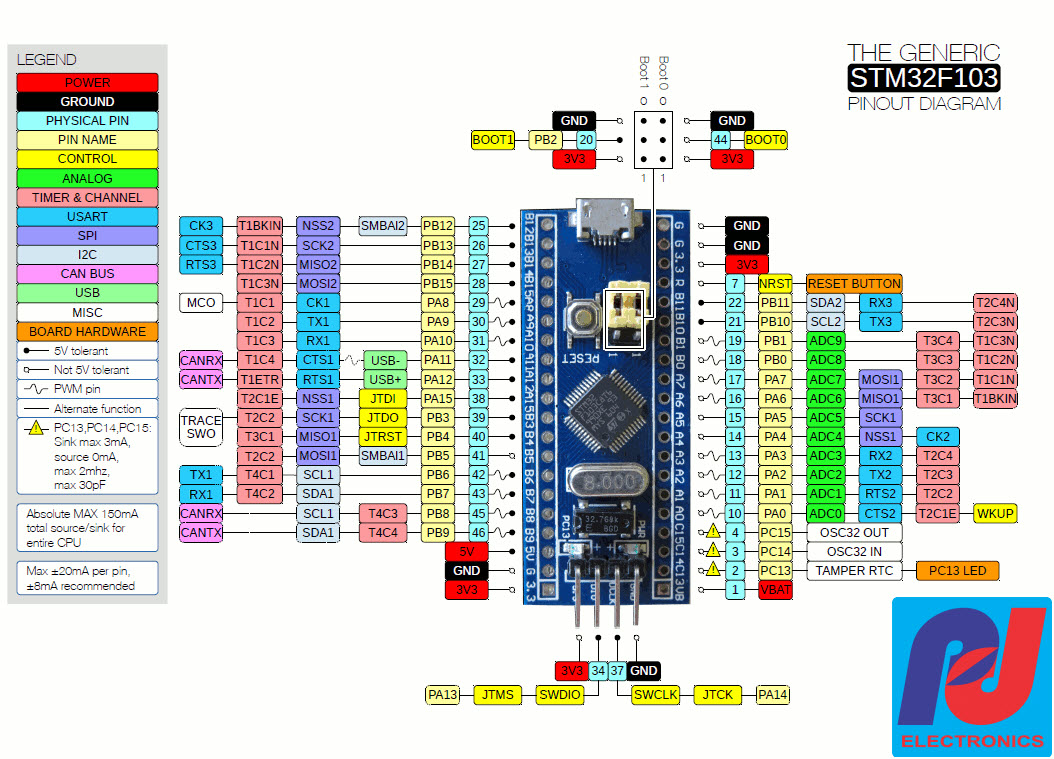
Hình 3:Thông số kỹ thuật của Kit STM32F103C8T6



* Dòng ARM Cortex là một bộ xử lí thế hệ mới đưa ra một kiến trúc chuẩn cho nhu cầu đa dạng về công nghệ. Không giống như các chip ARM khác, bộ xử lý Cortex là một lõi xử lí hoàn thiện, đưa ra một chuẩn CPU và kiến trúc hệ thống chung.
* Bộ vi xử lý Cortex-M3 là đơn vị xử lý trung tâm (CPU) của một chip vi điều khiển. Ngoài ra, còn có số lượng các thành phần khác được yêu cầu cho vi điều khiển hoàn toàn dựa vào vi xử lý Cortex-M3. Chip dựa vào vi xử lý Cortex-M3 từ các nhà sản xuất khác nhau sẽ có kích cỡ khác nhau về bộ nhớ, khác nhau về chủng loại, khác nhau về thiết bị ngoại vi, và khác nhau về các tính năng.
* Bộ vi xử lý có kiến trúc Harvard: có bus để giao tiếp bộ nhớ chương trình và bộ nhớ dữ liệu độc lập. Điều này cho phép truy cập dữ liệu và lệnh diễn ra cùng một lúc nên hiệu suất của bộ vi xử lý tăng lên.

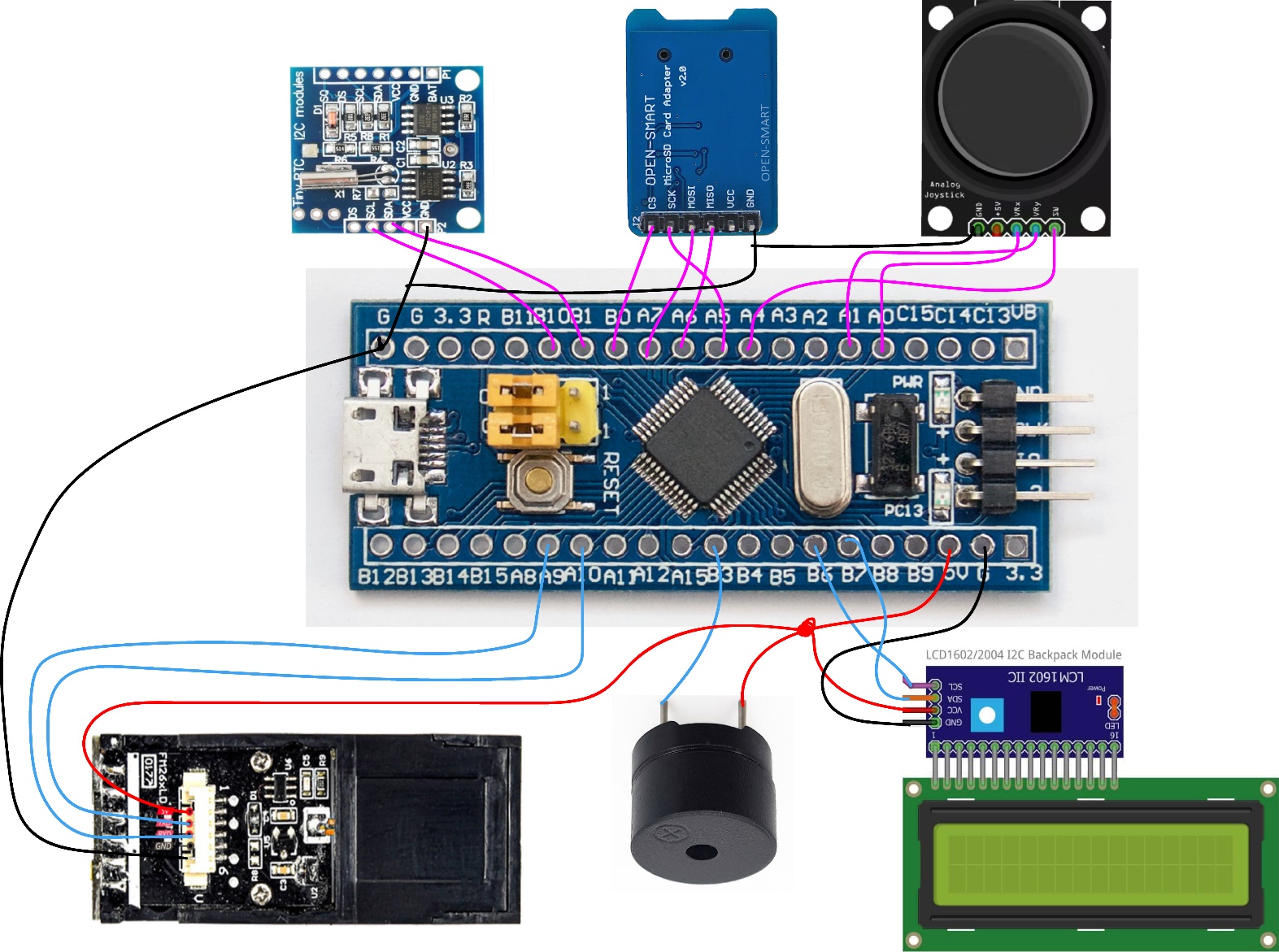
## CHƯƠNG 2: CẤU TRÚC CHUNG CỦA KIT

**2.1. Cấu trúc Kit STM32F103C8T6**



Hình 4:Cấu trúc Kit STM32F103C8T6

**2.2. Sơ đồ mạch chi tiết**



Hình 5: Sơ đồ mạch chi tiết

Graphical user interface, text

Description automatically generated

*Các giao thức sử dụng*

*A picture containing graphical user interface

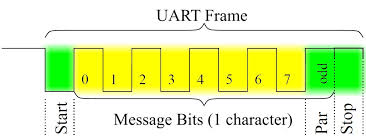
Description automatically generated*

## CHƯƠNG 3: NHỮNG KIẾN THỨC LIÊN QUAN

**3.1. Giao thức UART**

**3.1.1. Giới thiệu**

UART chuyển đổi giữa dữ liệu nối tiếp và song song. Một chiều, UART chuyển đổi dữ liệu song song bus hệ thống ra dữ liệu nối tiếp truyền đi. Một chiều khác, UART chuyển đổi dữ liệu nhận được, dạng dữ liệu nối tiếp thành dữ liệu song song cho CPU có thể đọc vào bus hệ thống.



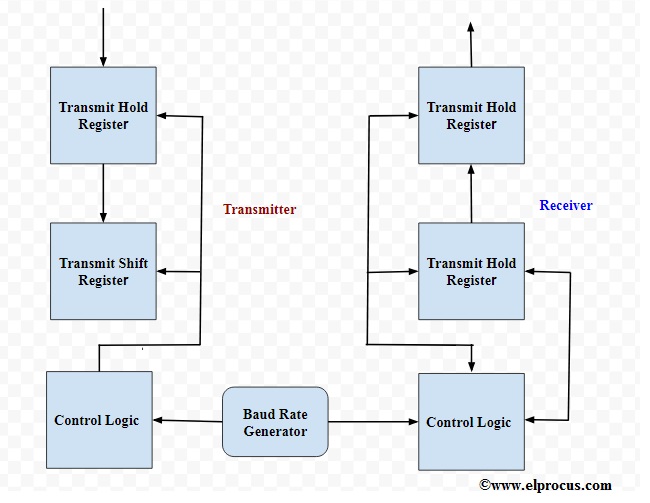
Hình 6:UART Frame

UART của PC hỗ trợ cả hai kiểu giao tiếp là giao tiếp đồng thời và giao tiếp không đồng thời. Giao tiếp đồng thời tức là UART có thể gửi và nhận dữ liệu vào cùng một thời điểm. Còn giao tiếp không đồng thời (không kép) là chỉ có một thiết bị có thể chuyển dữ liệu vào một thời điểm, với tín hiệu điều khiển hoặc mã sẽ quyết định bên nào có thể truyền dữ liệu. Giao tiếp không đồng thời được thực hiện khi mà cả 2 chiều chia sẻ một đường dẫn hoặc nếu có 2 đường nhưng cả 2 thiết bị chỉ giao tiếp qua một đường ở cùng một thời điểm. Thêm vào đường dữ liệu, UART hỗ trợ bắt tay chuẩn RS232 và tín hiêu điều khiển như RTS, CTS, DTR, DCR, RT và CD.

Sơ đồ khối UART bao gồm hai thành phần là máy phát và máy thu được hiển thị bên dưới. Phần máy phát bao gồm ba khối là thanh ghi giữ truyền, thanh ghi dịch chuyển và logic điều khiển. Tương tự, phần máy thu bao gồm một thanh ghi giữ, thanh ghi thay đổi và logic điều khiển. Hai phần này thường được cung cấp bởi một bộ tạo tốc độ baud. Trình tạo này được sử dụng để tạo tốc độ khi phần máy phát và phần máy thu phải truyền hoặc nhận dữ liệu.

Thanh ghi giữ trong máy phát bao gồm byte dữ liệu được truyền. Các thanh ghi thay đổi trong máy phát và máy thu di chuyển các bit sang phải hoặc trái cho đến khi một byte dữ liệu được truyền hoặc nhận. Một logic điều khiển đọc (hoặc) ghi được sử dụng để biết khi nào nên đọc hoặc viết.

Máy phát tốc độ baud giữa máy phát và máy thu tạo ra tốc độ dao động từ 110 bps đến 230400 bps. Thông thường, tốc độ truyền của vi điều khiển là 9600 đến 115200.



Hình 7: Sơ đồ khối UART

**3.1.2. Nguyên lý hoạt động**

Dưới đây giới thiệu về cách định dạng dữ liệu và giao thức dùng trong truyền thông nối tiếp. Chủ yếu là định dạng không đồng bộ dùng 2 chuẩn thông dụng là RS-232 và RS-485.

**Gửi dữ liệu nối tiếp**

Trong một liên kết nối tiếp, nơi gửi dữ liệu sẽ gửi từng bit một ở mỗi thời điểm nối tiếp nhau. Một liên kết nối tiếp chỉ có 2 thiết bị thì phải có đường dẫn dành cho mỗi chiều truyền hoặc là nó chỉ có 1 đường dẫn được chia sẻ bởi cả 2 thiết bị với thoả thuận của 2 thiết bị này. Khi mà có 3 hoặc nhiều thiết bị, tất cả các thiết bị này thường dùng chung một đường dẫn, và giao thức mạng quyết định xem thiết bị nào có quyền truyền nhận dữ liệu.

Một tín hiệu đòi hỏi bởi tất cả mọi liên kết nối tiếp là tín hiệu xung đồng hồ, hoặc là có sự tham khảo về thời gian để điều khiển đường truyền dữ liệu. Nơi truyền và nơi nhận dùng xung đồng hồ để quyết định khi nào gửi và khi nào đọc mỗi bít. Có hai dạng định dạng dữ liệu là đồng bộ và không đồng bộ, và mỗi định dạng này dùng các dạng xung đồng hồ khác nhau.

**Định dạng đồng bộ**

Trong truyền đồng bộ, mọi thiết bị dùng một xung đồng hồ được phát ra bởi một thiết bị hoặc từ một nguồn xung ngoài. Xung đồng hồ có thể có một tần số cố định hoặc cóc thể chốt tại những khoảng thời gian không đều. Mọi bít truyền đi được đồng bộ với đồng hồ. Nói cách khác, mỗi bít được truyền đi là dựa vào sự chuyển đổi của xung (như tăng hoặc giảm của sường xung). Nơi nhận dùng sự chuyển đổi xung để quyết định khi nào đọc mỗi bít truyền tới. Từ hình vẽ các bạn cũng có thể thấy là nơi truyền sẽ truyền các bit khi mà nhận thấy sự chuyển sườn xung từ cao xuống thấp, và nơi nhận thì ngược lại phát hiện khi nào có sự chuyển sườn xung từ thấp lên cao thì đọc các bit. Chi tiết chính xác của giao thức này có thể biến đổi khác đi. Ví dụ, nơi nhận có thể chốt dữ liệu nhận trong sườn xung tăng hoặc giảm, hoặc là phát hiện mức logic ở mức cao hoặc thấp. Định dạng đồng bộ dùng các cách khác nhau để bắt đầu và kết thúc việc truyền dữ liệu, bao gồm bít Start và bít Stop và tín hiệu lựa chọn chíp.

**Định dạng không đồng bộ**

Trong truyền không đồng bộ, liên kết không bao gồm đường xung đồng hồ, bởi vì mỗi điểm đầu cuối của liên kết đã có xung đồng hồ cho riêng từng cái. Mỗi điểm sẽ cần phải đồng ý cùng một tần số của đồng hồ và mọi đồng hồ chỉ khác nhau một vài %. Mỗi byte truyền đi bao gồm bít Start để đồng bộ đồng hồ và một hoặc nhiều bít Stop cho tín hiệu kết thúc việc truyền trong mỗi một từ được truyền đi. Cổng RS-232 trong PC dùng định dạng khoong đồng bộ để giao tiếp với modems (thiết bị mã hoá, giải mã dữ liệu) và các thiết bị khác. Dù RS-232 có thể truyền dữ liệu đồng bộ nhưng liên kết không đồng bộ vần được dùng phổ biến hơn. Phần lớn liên kết RS-485 dùng giao tiếp không đồng bộ.

Truyền không đồng bộ có thể dùng một trong vài cách định dạng phổ biến. Phổ biến nhất là kiểu 8-N-1, nơi truyền sẽ truyền mỗi byte dữ liệu một bít Start, tiếp theo là 8 bít dữ liệu bắt đầu với bít 0 (bít có trọng số nhỏ nhất Least Sìgnificant Bit) và kết thúc với 1 bít Stop. Chữ N trong định dạng 8-N-1 chỉ rằng truyền dữ liệu không dùng bít chẵn lẻ. Một dạng định dạng khác là bao gồm một bít chẵn lẻ giống như dạng đơn giản của kiểm soát lỗi.

Khi số các bit 1 trong byte là chẵn thì bít Odd Parity Bit = 1 và bít lẻ = 0, ...

Một số dạng khác không phổ biến là dùng một số khác nhau của số bít dữ liệu. Rất nhiều cổng nối tiếp hỗ trợ mọi nơi từ 5 ->8 bít dữ liệu, cộng với bít chẵn lẻ.

Tốc độ số bít là số bít một giây được truyền đi hoặc là nhận về trong một đơn vị thời gian. Tốc độ bus là số các sự kiện hình xảy ra hoặc truyền dữ liệu trên giây. Hai giá trị này thường đồng nhất với nhau trong nhiều liên kết. Trong đường dây điện thoại, modem tốc độ cao mã hoá nhiều bít trong mỗi chu kì dữ liệu vì thế tốc độ bus thực tế nhỏ hơn tốc độ bit (bit rate).

Mọi bít cần thiết cho truyền một giá trị từ bít Start đến bít Stop gọi là một Word. Mỗi bít trong dạng Word gọi là một Character. Trong vài liên kết, các bít là kí tự văn bản (dạng chữ hoặc số), trong khi các dạng kí tự khác lại là giá trị nhị phân. Thời gian truyền các các kí tự trong một giây bằng với tổng thời gian truyền từng bít trong word cộng lại. Thêm bít start và bít Stop làm tăng thời gian truyền mỗi byte lên 25% (vì có 10 bít cần truyền trong khi chỉ dùng có 8 bít). Với định dạng 8-N-1, một byte truyền với thời gian bằng 1/10 tần số bus: do đó 9600 bít/s truyền 960 byte/s. Nếu nơi nhận đòi hỏi phải có một thời gian kiểm tra dữ liệu nhận được, nơi truyền sẽ kéo dài độ rộng của bít Stop ra 1,5 hoặc 2 bít.

**Cơ chế chống mất dữ liệu**

Phần lớn các máy tính trong mạng nối tiếp có nhiều việc phải làm bên cạnh việc chờ nhận dữ liệu. Ví dụ, mỗi đơn vị dữ liệu có thể thu thập theo chu kì và lưu trữ dữ liệu tới khi một mắt xích khác trong mạng yêu cầu dữ liệu này. Hoặc một điều khiển có thể đáp ứng các điều kiện điều khiển và điều hành, thỉnh thoảng lại nhận thông tin hoặc nhận các yêu cầu từ trong mạng.

Một máy tính muốn truyền dữ liệu trong khi một máy nhận khác đang bận với các công việc khác. Việc thiết kế mạng phải đòi hỏi rằng mỗi nơi nhận có thể biết được dữ liệu nào chuyển đến nó và tất cả mọi dữ liệu đến máy nhận phải không có lỗi.

Có nhiều cách làm để thực hiện điều đó, bao gồm bắt tay (handshaking), bộ đệm (buffering), dùng dò (polling) và ngắt (interrupts) để phát hiện dữ liệu đã đến, kiểm soát lỗi (error checking), và thừa nhận dữ liệu đã tới( acknowledging). Mỗi liên kết có thể dùng một hoặc nhiều cách trong số những cách này.

**Bắt tay (handshaking)**

Với tín hiệu bắt tay, máy phát có thể xác định khi nào máy tính này phải truyền dữ liệu và máy nhận có thể biết khi nào nó sẵn sàng nhận dữ liệu. Tín hiệu có thể biến đổi qua RS-232 hoặc RS-485 theo giao thức chuẩn hoặc giao thức qui ước.

Một trong những dạng bắt tay về phần cứng, nơi nhận đưa ra dòng mức cao khi sẵn sàng nhận dữ liệu, và nơi truyền chờ tín hiệu này trước khi truyền dữ liệu. Nơi nhận có thể đưa ra dòng mức thấp trong mọi thời điểm, thậm chí cả trong quá trình chờ dòng phản hồi cao trước khi kết thúc quá trình truyền nhận. Một số dạng liên kết khác hoạt động giống nguyên tắc ở trên nhưng với bắt tay bằng phần mềm, bằng cách nơi nhận gửi một mã để báo nó sẵn sàng nhận dữ liệu, và một mã khác để báo báo cho nơi truyền dừng quá trình gửi dữ liệu.

**Bộ đệm (Buffer)**

Bộ đệm là một dạng khác để nơi nhận có thể chắc chắn là không mất một dữ liệu nào gửi đến chúng. Bộ đệm có thể có ích cho phía truyền, nơi cho phép ứng dụng làm việc có hiệu quả bằng cách lưu trữ dữ liệu để gửi khi liên kết sẵn sàng để truyền nhận dữ liệu.

Bộ đệm có thể là bộ đệm phần cứng, phần mềm hoặc cả hai. Cổng nối tiếp dùng tất cả các dạng này nhưng máy tính cổ nhất có 16 byte bộ đệm phần cứng được tích hợp trong những UART. Trong chiều nhận, điều đó có nghĩa rằng UART có thể lưu trữ 16 byte trước khi phần mềm cần đọc chúng. Trong chiều nhận, UART có thể lưu trữ 16 byte và UART sẽ cẩn thận truyền mỗi byte theo từng bít theo giao thức lựa chọn.

Khi bộ đệm phần cứng không đủ rộng, một máy tính cá nhân có thể dùng bộ đệm phần mềm, bộ đệm này có thể lập trình được kích thước và kích thước tối đa cho phép bởi bộ nhớ hệ thống. Các thiết bị phần mềm của cổng truyền nhận dữ liệu giữa bộ đệm phần cứng và phần mềm.

Trong các vi điều khiển, bộ đệm có xu hướng trở nên nhỏ hơn, và một số chip không có bộ đệm phần cứng. Việc làm hẹp bộ nhớ đệm điều quan trọng hơn ở đây là các chíp này dùng các công nghệ khác để chắc chắn là không dữ liệu nào bị mất.

**Thăm dò và ngắt**

Sự kiện gây ra ở cổng nối tiếp bao gồm khi truyền và nhận dữ liệu, thay đổi tín hiệu bắt tay và gửi, nhận thông điệp lỗi. Có hai cách cho ứng dụng phát hiện và gây ra những sự kiện này.

Các thứ nhất là có chương trình tự động nhảy tới các chuỗi sự kiện được sắp xếp trước (như bảng vector ngắt) khi một sự kiện xảy ra. Ứng dụng phản ứng nhanh và tự động hoạt động ở cổng mà không lãng phí thời gian kiểm tra, chỉ cần biết như không có hoạt động nào xảy ra.

Dạng lập trình này gọi là chạy đua sự kiện (event-driven) bởi vì một sự kiện bên ngoài có thể xảy ra trong bất kì thời điểm nào và chương trình chạy tới một bảng đặc biệt.

Cách thứ hai là thăm dò bằng cách đọc theo từng chu kì hoặc phát ra tín hiệu tìm kiếm khi nào một sự kiện xảy ra. Dạng lập trình này gọi là lập trình thủ tục, và không dùng ngắt phần cứng. Ứng dụng phải chắc chắn thăm dò cổng một cách đầy đủ để không mất bất kì một dữ liệu nào hoặc sự kiện nào. Tần số thăm dò phụ thuộc vào kích thước bộ đệm và tổng dữ liệu cần lấy (cần cho phản ứng nhanh). Ví dụ, nếu một thiết bị có 16 byte bộ đệm và dò cổng 1 lần/1 giây, thiết bị này chỉ có thể nhận không thể lớn hơn 16 byte/ 1 giây hoặc là bộ đệm sẽ bị tràn hoặc là dữ liệu sẽ bị mất.

Phương pháp thăm dò thường áp dụng cho truyền dữ liệu ngắn, đột ngột hoặc khi máy tính gửi dữ liệu và chớ đợi tín hiệu phản hồi nhanh. Một giao diện thăm dò không yêu cầu ngắt phần cứng, và bạn có thể chạy dạng lập trình này ở trên cổng mà không có đường ngắt. Nhiều giao diện thăm dò dùng ngắt timer của hệ thống để có kế hoạch đọc cổng sau một khoảng thời gian cố định.

**3.2. Cảm biến vân tay R308**

#### **3.2.1. Giới thiệu**



Hình 8: Hình ảnh thực tế cảm biến vân tay R308

Cảm biến nhận dạng vân tay R308 sử dụng giao tiếp UART TTL hoặc USB 1.1 để giao tiếp với Vi điều khiển hoặc kết nối trực tiếp với máy tính (thông qua mạch chuyển USB-UART hoặc giao tiếp USB 1.1).

Cảm biến nhận dạng vân tay R308 được tích hợp nhân xử lý nhận dạng vân tay phía trong, tự động gán vân tay với 1 chuỗi data và truyền qua giao tiếp UART ra ngoài nên hoàn toàn không cần các thao tác xử lý hình ảnh, đơn giản chỉ là phát lệnh đọc/ghi và so sánh chuỗi UART nên rất dễ sử dụng và lập trình.

**3.2.2. Thông số kỹ thuật**

- Điện áp nguồn: DC 4.4~6.0V (khuyên dùng 5v)

- Dòng làm việc: 110 mA

- Dòng cực đại: 140 mA

- Thời gian nhận vân tay: <0,3 giây

- Phương thức đối soát vân tay:

+ Phương pháp so sánh (1:1)

+ Phương pháp tìm kiếm (1:N)

- Tệp tính năng: 256 byte

- Tệp mẫu: 512 byte

- Dung lượng lưu trữ: 500 vân tay

- Cấp độ bảo mật: Cấp độ 3

- Tỷ lệ nhận dạng sai (FAR): <0,0001%

- Tỷ lệ từ chối (FRR): <1,0%

- Thời gian tìm kiếm: <1,0 giây (tại 1:1000, giá trị trung bình)

- Giao diện: UART

- Tốc độ truyền thông (UART): 57600bps, không thể sửa đổi.

- Môi trường làm việc:

+ Nhiệt độ: -20℃－+40℃

+ Độ ẩm tương đối: 40%RH－85%RH (không ngưng tụ)

- Môi trường lưu trữ:

+ Nhiệt độ: -40℃－+85℃

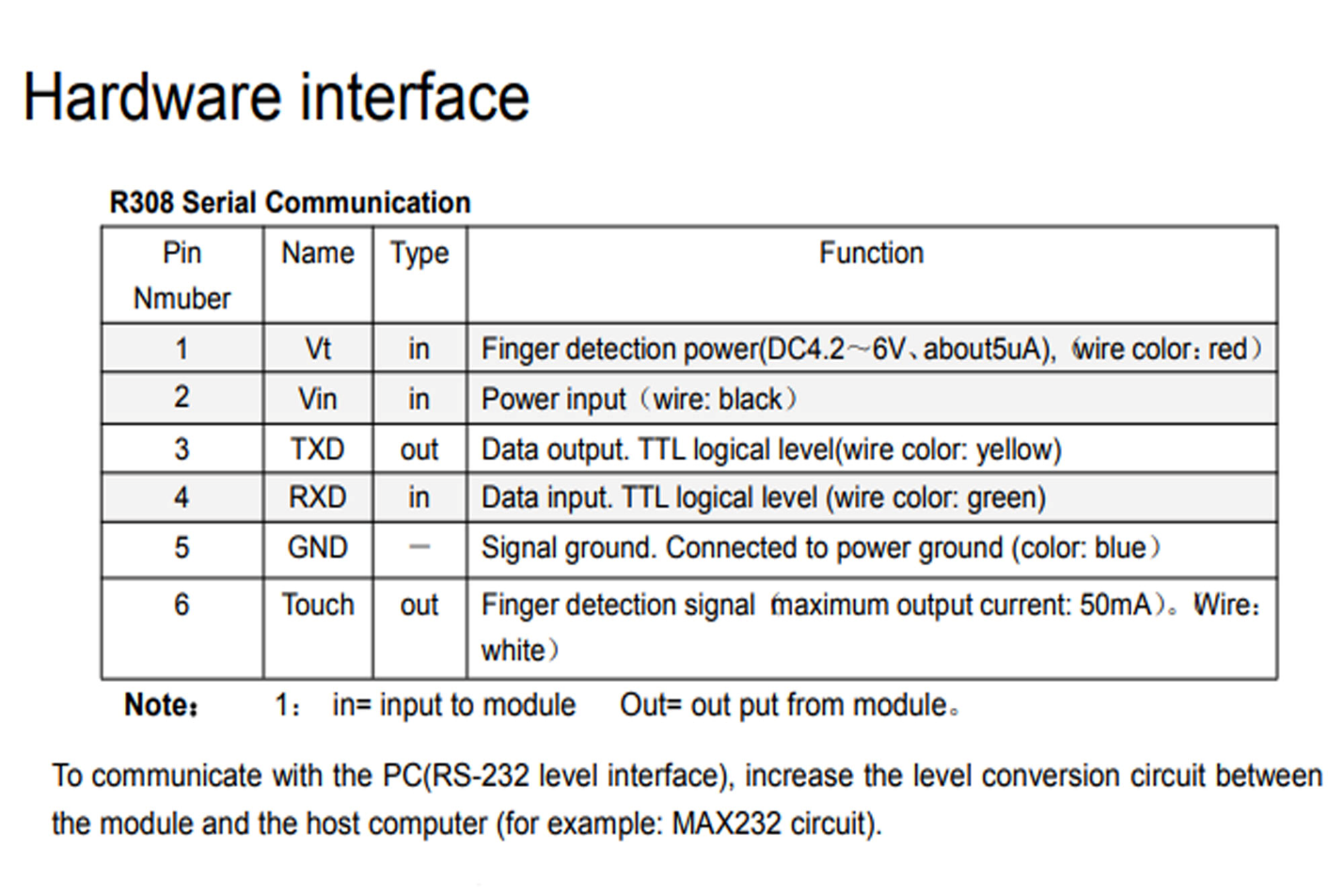
+ Độ ẩm tương đối: <85%H (không ngưng tụ)

- Trọng lượng: 20g

**3.2.3 Nguyên lý hoạt động**

* Nguyên lí hoạt động của module cảm biến vân tay cơ bản gồm 2 phần:
  + Lấy dữ liệu hình ảnh vân tay: Khi lấy dữ liệu, module sẽ lấy dữ liệu hình ảnh vân tay 2 lần thông qua cảm biến quang học và xử lí 2 hình ảnh này để tạo ra một mẫu.
  + Kết hợp các dữ liệu để tạo ra mẫu vân tay: Quá trình này sảy ra sau khi đã có hình ảnh vân tay. Đây là quá trình kết hợp 2 mẫu hình ảnh vân tay để tạo ra một mẫu. Hệ thống sẽ xử lí để lưu chữ vào thư viện vân tay của module.
* Khi kiểm tra, người dùng vẫn đặt ngón tay vào vị trí của cảm biến quang học, hệ thống sẽ tạo ra một bản mẫu của ngón tay và so sánh với những mẫu được lưu trữ trong thư viện. Đối với kiểu so sánh 1:1, hệ thống sẽ so sánh trực tiếp vân tay với những mẫu được chỉ định trong Module. Đối với kiểu so sánh 1: N hay tìm kiếm, hệ thống sẽ quét toàn bộ tất cả những vân tay có trong thư viện vân tay.
* Trong cả 2 trường hợp, hệ thống sẽ trả về kết quả phù hợp, có hoặc không có vân tay.

**3.2.4. Giao tiếp phần cứng**

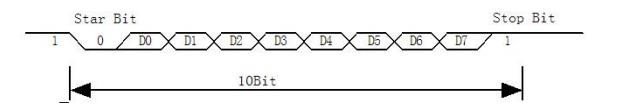


Hình 9: Hình ảnh các ngõ giao tiếp của cảm biến R305

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên** | **Chức năng** |
| 1 | VCC | Ngõ vào 3.6-6V |
| 2 | GND | Ngõ vào 0V |
| 3 | Tx | Ngõ ra truyền dữ liệu |
| 4 | Rx | Ngõ vào nhận dữ liệu |
| 5 | VCC | 5 VDC |
| 6 | D- | Data - |
| 7 | D+ | Data + |
| 8 | GND | 0V |

Bảng 2: Thông tin các ngõ giao tiếp của cảm biến R305

* Giao thức truyền thông nối tiếp:
  + Đó là chế độ truyền thông nối tiếp bán song công không đồng bộ. Tốc độ bauds mặc định của FG là 57600bps. Người dùng có thể tùy chỉnh tốc độ từ 9600-115200bps.
  + Mẫu khung truyền là 10bits. Start bit là bit 0, sau đó là 8 bits dữ liệu truyền theo kiểu LSB, và cuối cùng là stop bit là bit 1. Không có bit kiểm tra.



Hình 10: Khung dữ liệu truyền đi của cảm biến R305

* Khi cấp nguồn, cảm biến vân tay sẽ mất khoảng 500ms để khởi tạo. Trong thời gian này cảm biến vân tay không thể thực hiện các lệnh từ máy tính.

**3.2.5. Giao thức truyền gói dữ liệu**

* **Định dạng gói dữ liệu**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Header*** | ***Adder*** | ***Package***  ***identifier*** | ***Package length*** | ***Package content (instuction/data/Parameter*）** | ***Checksum*** |

Bảng 3: Định dạng gói dữ liệu của R305

* **Ý nghĩa gói dữ liệu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Tên*** | ***Kí hiệu*** | ***Độ dài*** | ***Ý nghĩa*** | |
| Header | Start | 2 bytes | Mặc định là 0xEF01, là byte truyền đầu tiên. | |
| Adder | ADDER | 4 bytes | Mặc đinh là 0xFFFFFFFF, có thể thay đổi bằng hàm SetAdder, là byte truyền kế tiếp. Nếu địa chỉ thiết bị sai, thiết bị sẽ từ chối truyền. | |
| Package  identifier | PID | 1 bytes | 01H | Gói lệnh. |
| 02H | Gói dữ liệu. |
| 07H | Gói xác nhận. |
| 08H | Kết thúc gói dữ liệu. |
| Package length | LENGTH | 2 bytes | Chiều dài nội dung gói (gói lệnh và gói dữ liệu) cộng với chiều dài checksum. Đơn vị là byte. Chiều dài tối đa là 256 bytes. Byte cao được truyền đầu tiên. | |
| Package content | DATA | - | Lệnh, dữ liệu hoặc các thông số của lệnh. | |
| Checksum | SUM | 2 bytes | Tổng của PID, LENGTH và các gói dữ liệu đi kèm, bit tràn được bỏ qua, byte cao truyền trước. | |

Bảng 4: Ý nghĩa của gói dữ liệu truyền cảu cảm biến R305

* **Kiểm tra và phản hồi khi nhận dữ liệu**

**Chú ý:** các lệnh này chỉ được gửi từ máy tính tới các Module cảm biến vân tay. Sau khi nhận được lệnh, Module sẽ phản hồi tình hình thực hiện lệnh và kết quả để báo cho máy tính là đã nhận được các gói. Các gói phản hồi sẽ có các thông số hoặc các gói kèm theo. Máy tính sẽ không thể xác định tình trạng của gói Module nhận được hoặc kết quả thực hiện các lệnh trong tập tin nếu không có các tập tin phản hồi từ Module. Gói phản hồi 1 byte gồm mã xác nhận hay có thể kèm theo những thông số trả về cho máy tính.

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tên gói*** | ***Mô tả*** | ***Tên gói*** | ***Mô tả*** |
| 00h | Lệnh thực hiện hoàn toàn. | 0Dh | Lỗi khi tải mẫu lên. |
| 01h | Lỗi khi nhận gói dữ liệu. | 0Eh | Module không thể nhận các gói kèm theo. |
| 02h | Không có ngón tay trên cảm biến. | 0Fh | Lỗi khi tải ảnh lên. |
| 03h | Không thể lấy mẫu vân tay. | 10h | Không thể xóa các mẫu. |
| 06h | Không thể tạo file kí tự do mẫu vân tay không trùng khớp. | 11h | Không thể xóa thư viện vân tay. |
| 07h | Không thể tạo file do thiếu đối tượng hoặc do vân tay quá bé. | 13h | Sai mật khẩu. |
| 08h | Ngón tay không phù hợp. | 15h | Không thể tạo hình ảnh đầy đủ của các hình ảnh hợp lệ. |
| 09h | Không thể tìm thấy vân tay trùng khớp. | 18h | Lỗi khi viết thông tin trên Flash. |
| 0Ah | Không kết hợp các tập tin. | 19h | Không thể xác định lỗi. |
| 0Bh | Địa chỉ PageID quá lớn. | 1Ah | Số thanh ghi không hợp lệ. |
| 0Ch | Lỗi khi đọc các mẫu từ thư viện hoặc các mẫu không hợp lệ. | 1Bh | Cấu hình thanh ghi không đúng. |
| 1Ch | Sai trang ghi chú. | 1Dh | Các cổng giao tiếp không hoạt động hoặc hệ thống không thể sử dụng. |

Bảng 5: Mã xác nhận gửi về từng Module khi tiến hành giao tiếp của R305

**3.3. LCD 16x2**

**3.3.1. Giới thiệu**

Ngày nay, thiết bị hiển thị LCD (Liquid Crystal Display) được sử dụng trong rất nhiều các ứng dụng của VĐK. LCD có rất nhiều ưu điểm so với các dạng hiển thị khác: Nó có khả năng hiển thị kí tự đa dạng, trực quan (chữ, số và kí tự đồ họa), dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn rất ít tài nguyên hệ thống và giá thành rẽ …

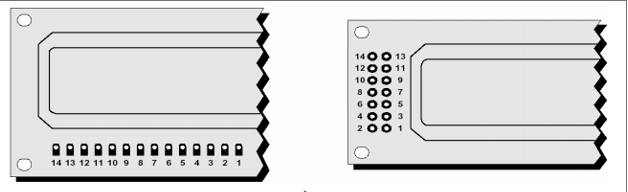


Hình 11: Hình ản thực tế LCD 16x2

**3.3.2. Chức năng các chân**

Khi sản xuất LCD, nhà sản xuất đã tích hợp chíp điều khiển (HD44780) bên trong lớp vỏ và chỉ đưa các chân giao tiếp cần thiết. Các chân này được đánh số thứ tự và đặt tên như hình sau:

**Sử dụng giao thức I2C**



Hình 12: Sơ đồ chân của LCD

Chức năng các chân:

|  |  |  |
| --- | --- | --- |
| Chân số | Tên | Chức năng |
| 1 | Vss | Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển |
| 2 | Vdd | Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với VCC=5V của mạch điều khiển |
| 3 | Vcc | Chân này dùng để điều chỉnh độ tương phản của LCD. |
| 4 | RS | Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi.  + Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read)  + Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD. |
| 5 | R/W | Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc. |
| 6 | E | Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E.  + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào(chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E.  + Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (lowto-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức  thấp. |
| 7-14 | DB0-DB7 | Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này :  + Chế độ 8 bit : Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7.  + Chế độ 4 bit : Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7  Chi tiết sử dụng 2 giao thức này được đề cập ở phần sau. |
| 15 | - | Nguồn dương cho đèn nền |
| 16 | - | GND cho đèn nền |

**3.4. Mạch nạp ST-LINK V2**

**3.4.1. Công dụng**

****

Hình 13: Hình ảnh thực tế mạch nạp ST Link V2

ST Link-V2 sử dụng làm mạch nạp STM8 STM32 chuẩn SWD, giao tiếp dễ dàng, nhanh và ổn định, chỉ cần 4 dây

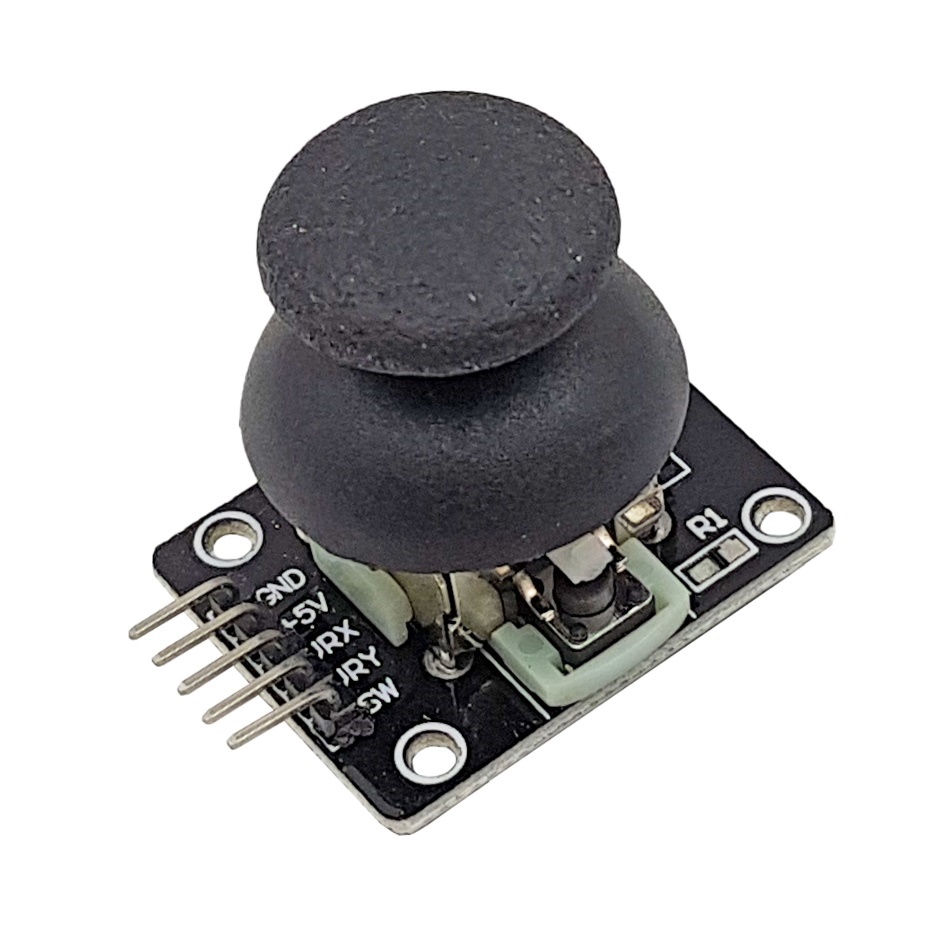
Hỗ trợ tất cả các dòng STM8 chuẩn SWIM

**3.4.2. Thông số kỹ thuật**

* Cổng giao tiếp: USB
* Giao diện tương thích tốc độ cao USB 2.0
* Có mạch bảo vệ khi điện áp tăng, không sợ lỗi mạch ST Link-V2
* STM8
* Sơ đồ chân kết nối mạch nạp với STM32
* GND -- GND
* SWCLK -- SWCLK
* SWDIO -- SWIO
* 3.3V -- 3V3
* Điện áp sử dụng từ 1,65 V đến 5,5 V được hỗ trợ trên SWIM
* Tốc độ lập trình SWIM: 9,7 Kbytes / s ở tốc độ thấp và 12,8 Kbytes / giây ở tốc độ cao

**3.5. Module Joystick**

**3.5.1. Công dụng**



Hình 14:Hình ảnh thực tế Module Joystick

[**Module Joystick đơn**](https://nshopvn.com/product/module-joystick-don/) thường được sử dụng để làm tay cầm hoặc cần điều khiển, mạch có kích thước nhỏ gọn với bốn lỗ ốc trên mạch rất dễ gá bắt.

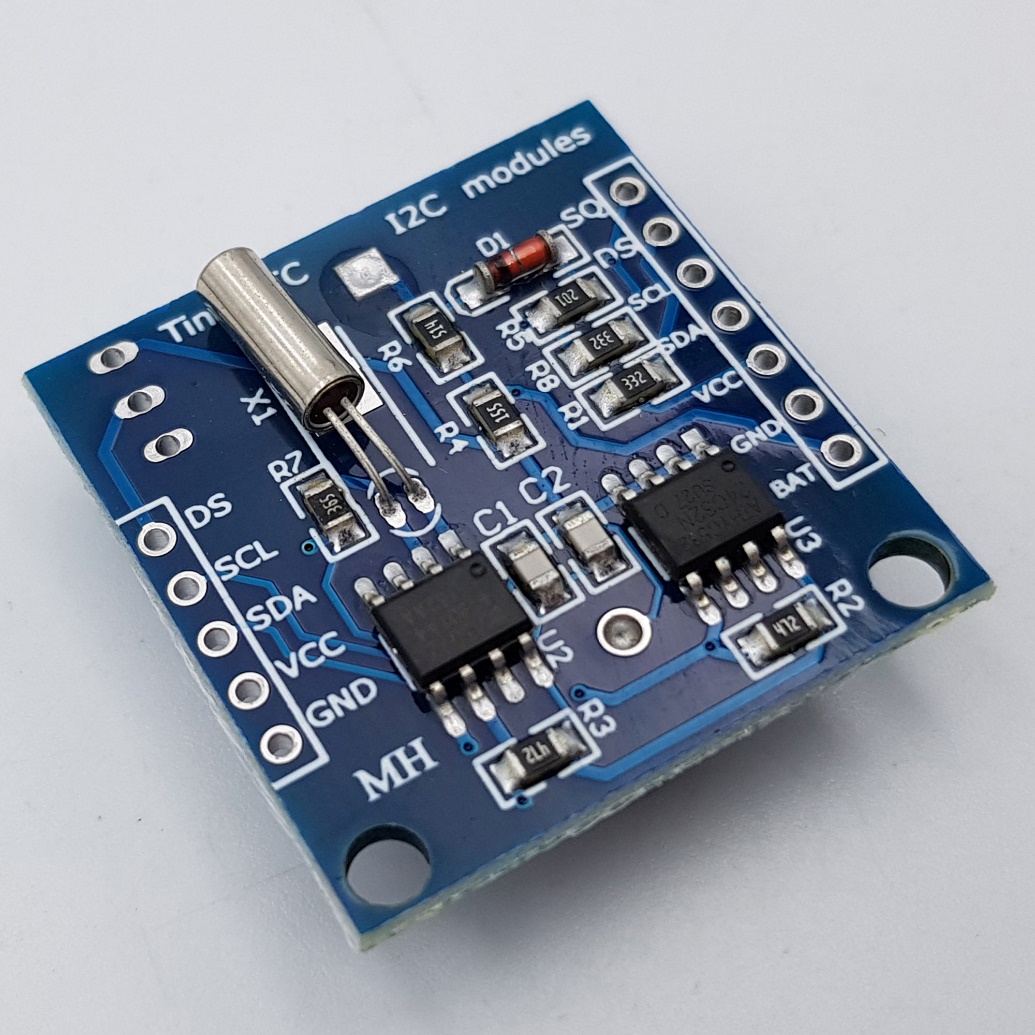
**3.5.2. Thông số kỹ thuật**

* Nguồn cấp: Tùy chọn, thường cấp 3.3 hoặc 5VDC.
* Kiểu dạng tín hiệu ngõ ra 1 Digital và 2 Analog ( 1 nút nhấn và hai trục X, Y), mức tín hiệu theo nguồn cấp vào.
* Kích thước: 4.0cm x 2.6cm x 3.2cm
* Trọng lượng: 12g

**Sử dụng giao thức ADC**

**3.6. Module thời gian thực RTC DS1307**

**3.6.1. Công dụng**



Hình 15:Hình ảnh thực tế Module RTC DS1307

[**Module thời gian thực RTC DS1307**](https://nshopvn.com/product/module-thoi-gian-thuc-rtc-ds1307/) có chức năng lưu trữ thông tin ngày tháng năm cũng như giờ phút giây, nó sẽ hoạt động như một chiếc đồng hồ và có thể xuất dữ liệu ra ngoài qua giao thức I2C.

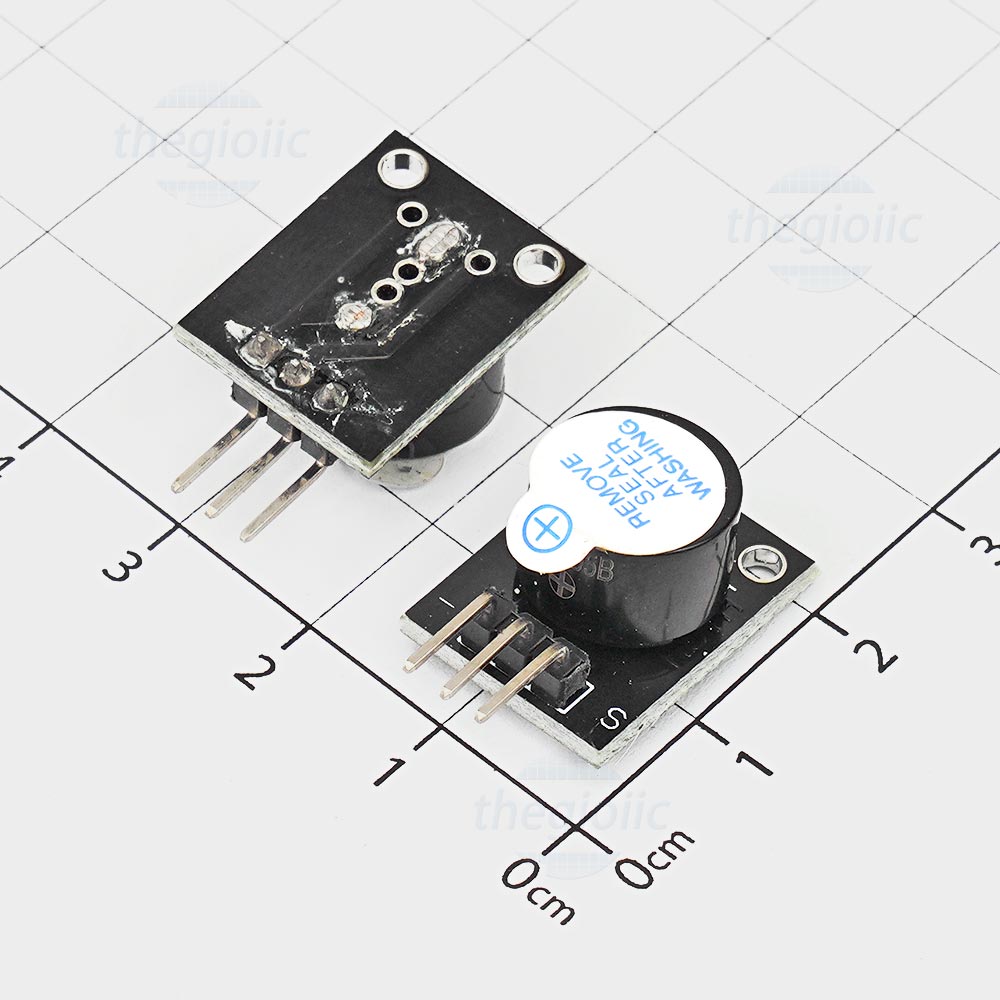
**3.6.2. Thông số kỹ thuật**

* Điện áp làm việc: 3.3V đến 5V
* Bao gồm 1 IC thời gian thực DS1307
* Các thành phần cần thiết như thạch anh 32768kHz, điện trở pull-up và tụ lọc nguồn đều được tích hợp trên board
* LED báo nguồn
* Có sẵn pin dự phòng duy trì thời gian khi mất điện
* 5-pin bao gồm giao thức I2C sẵn sàng giao tiếp: INT (QWO), SCL, SDA, VCC và GND
* Dễ dàng thêm một đồng hồ thời gian thực để dự án của bạn
* Nhỏ gọn và dễ dàng để lắp thêm vào bo mạch hoặc test board

**Sử dụng giao thức I2C**

**3.7. KY-012 Mạch Buzzer 5V**

**3.7.1. Công dụng**



Hình 16: Hình ảnh thực tế KY-012 Mạch Buzzer 5V

[KY-012](https://www.thegioiic.com/products/ky-012-mach-buzzer-5v) là một mạch buzzer lý tưởng để tăng thêm âm thanh cho dự án của bạn và tương thích khi hoạt động với các bộ vi điều khiển như [Arduino](https://www.thegioiic.com/product/arduino). Mạch buzzer này bao gồm một bộ rung áp điện hoạt động, nó tạo ra âm thanh có tần số gần 2.5kHz khi tín hiệu ở mức cao.

**3.7.2. Thông số kỹ thuật**

* Điện áp hoạt động : 3.5V ~ 5.5V
* Dòng hoạt động: 30mA / 5VDC
* Tần số cộng hưởng 2500Hz ± 300Hz
* Đầu ra âm thanh tối thiểu 85Db @ 10cm
* Nhiệt độ làm việc: -20°C ~ 70°C
* Kích thước: 18.5 x 15 mm

**Sử dụng giao thức TIM**

**3.8. Mạch chuyển USB to TTL CP2102**

**3.8.1. Công dụng**



Hình 17:Hình ảnh thực tế Mạch chuyển USB to TTL CP2102

[**Mạch chuyển USB to TTL CP2102**](https://nshopvn.com/product/mach-chuyen-usb-to-ttl-cp2102/) là module chuyển đổi tín hiệu USB sang tín hiệu Serial tuần tự theo chuẩn TTL. Mạch sử dụng chip CP2102

**3.8.2. Thông số kỹ thuật**

* Phạm vi nhiệt độ: -40Cto + 85C
* Hỗ trợ windows vista / xp / server 2003/200, Mac OS-X / OS-9, Linux
* Màu chính: đỏ
* Kích thước: chiều dài (Không bao gồm USB): 30 mm
* USB để lấy nguồn, dẫn đến giao diện bao gồm 3,3V (<40mA), 5V, GND, TX, RX, mức pin tín hiệu là 3,3V, logic dương
* Hỗ trợ tốc độ truyền trong khoảng 300bps ~ 1Mbps

**Sử dụng giao thức UART**

**3.9. Module ghi đọc thẻ SD**

**3.9.1. Công dụng**



Hình 18:Hình ảnh thực tế Module ghi đọc thẻ SD

[**Module ghi đọc thẻ SD**](https://nshopvn.com/product/module-ghi-doc-the-sd/) sẽ giúp ứng dụng của bạn được nhanh hơn và dễ dàng hơn khi cần đến đọc/ ghi dữ liệu từ thẻ nhớ. Module SD card có thể sử dụng với bất cứ loại vi điều khiển nào.

**3.9.2. Thông số kỹ thuật**

* Tương thích với cả nguồn 5V và 3.3V
* Hỗ trợ khe cắm mở rộng
* SD Card hỗ trợ định dạng FAT16
* Hỗ trợ việc recording và playback cho lượng âm thanh lớn
* Giao tiếp thông qua chuẩn SPI (4 Pin): MOSI, SCK, MISO và CS.

**Sử dụng giao thức SPI**

## CHƯƠNG 4: ỨNG DỤNG

**4.1. Cách nạp chương trình vào Kit**

**4.2. Mô tả ứng dụng**

**4.2.1. Mô tả các chức năng**

Chức năng chấm công

Để sử dụng chức năng chấm công trên máy, người dùng cần dùng joystick để chuyển đến chế độ chấm công và sau đó bấm nút để chọn. Khi người dùng đặt tay vào máy, hệ thống sẽ tự động xác định liệu họ đang đi vào hoặc ra khỏi nơi làm việc và ghi lại thời gian hiện tại. Nếu đây là lần đầu tiên sử dụng máy, giờ ra sẽ không được ghi nhận.

Khi người dùng đặt vân tay lần thứ hai, hệ thống sẽ ghi nhận thời gian vào và ra, và sau đó lưu vào thẻ nhớ dưới dạng file nhị phân. Việc này giúp cho quản lý chấm công và tính lương được thực hiện một cách chính xác và nhanh chóng.

Chức năng thêm vân tay

Người dùng điều chỉnh cần gạt joystick để chọn chức năng thêm vân tay. Hệ thống sẽ đọc trong thẻ nhớ xem còn ID nào trống sẽ cấp mới cho vân tay mới

Chức năng xem giờ

Điều chỉnh joystick đến chức năng xem giờ sẽ xem được thời gian hiện tại

Chức năng xóa toàn bộ

Đây là chức năng reset toàn bộ ứng dụng về số 0

**4.2.2. Demo ứng dụng**

Để thực hiện chạy ứng dụng ta cấp nguồn cho ứng dụng bằng cáp sạc USB





## CHƯƠNG 5: TỔNG KẾT

**5.1. Kết quả**

* Hiểu được cấu trúc, cách thức hoạt động của board STM32F103C8T6.
* Hiểu được cấu trúc và cách hoạt động của cảm biến vân tay R308 và LCD 16x2.
* Hiểu được cơ chế truyền data từ cảm biến vân tay với STM32F103C8T6.
* Tạo được project với Keli C và nạp chương trình vào Kit.

**5.2. Ưu nhược điểm**

**5.2.1. Ưu điểm**

* Xác định được từng vân tay của từng người khác nhau.
* Thêm được vân tay
* In thông tin của người có cân tay đã lưu ra màn hình.
* Bước đầu sử dụng thư viện thời gian thực(RTC) để xác định điểm danh đúng giờ hay trễ giờ.
* Có lưu trữ thẻ nhớ, và cấu trúc dữ liệu minh bạch, dễ dàng nâng cấp sau này
* Có còi buzzer phản hồi
* Sử dụng joystick dễ dàng điều khiển, thân thiện

**5.2.2. Nhược điểm**

- Cần phải cấp 2 nguồn 5V cho thẻ nhớ và cho mạch

**5.3. Hướng phát triển**

**-** Nâng cấp bảo mật cho hệ thống

Phát triển ứng dụng web dễ dàng quản lí

**TÀI LIỆU THAM KHẢO**

* Giáo trình môn Hệ thống nhúng, ThS Đinh Công Đoan, 2015.
* Các nguồn tài liệu tham khảo trên internet:
  + Giao thức UART

<https://advancecad.edu.vn/khai-niem-co-ban-ve-truyen-thong-uart-so-do-khoi-ung-dung/>

* + Cảm biến vân tay R305:

<https://dientutuonglai.com/module-nhan-dang-van-tay-r305.html>

* + Giao tiếp với cảm biến vân tay R305:

<http://hocdientu.vn/threads/giao-tiep-module-cam-bien-van-tay-r305-part-1-gioi-thieu-chung.2973/>

* + Giới thiệu về LCD 16x2

<http://www.echipkool.com/2012/09/gioi-thieu-co-ban-ve-lcd-16x2.html>