**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

🞠◊🞠◊🞠◊🞠◊🞠



**BÁO CÁO ĐỀ TÀI**

**(Môn học: Hệ thống nhúng)**

**GIAO TIẾP CHUỘT VÀ BÀN PHÍM**

**MÃ HỌC PHẦN:  181ESYS431080**

**HỌC KỲ: 1 – NĂM HỌC: 2018-2019**

**GVHD:** **Th.S Đinh Công Đoan**

**Thành viên nhóm:**

**1. Bùi Phan Viết Cường 15110173**

**2. Huỳnh Thế Cương 15110172**

**3. Trần Văn Chinh 15110167**

**TP.HỒ CHÍ MINH – 12/2018**

**NHẬN XÉT CỦA GIẢNG VIÊN**

TPHCM, ngày 18 tháng 12 năm 2017

**Giáo viên hướng dẫn**

(Ký và ghi rõ họ tên)

**MỤC LỤC**

[I. MỞ ĐẦU 1](#_Toc532148727)

[**1. Tóm tắt 1**](#_Toc532148728)

[**2. Đặt vấn đề 1**](#_Toc532148729)

[**2.1. Những nghiên cứu trong và ngoài nước 1**](#_Toc532148730)

[**2.2. Tài liệu và kiến thức liên quan 3**](#_Toc532148731)

[**2.3. Tính cấp thiết cần nghiên cứu của đề tài 3**](#_Toc532148732)

[**2.4. Tính cấp thiết của đề tài (lý do chọn đề tài) 4**](#_Toc532148733)

[**2.5. Mục tiêu đề tài 4**](#_Toc532148734)

[**2.6. Đối tượng và phạm vi nghiên cứu 4**](#_Toc532148735)

[**2.7. Phương pháp nghiên cứu 5**](#_Toc532148736)

[**2.8. Nội dung đề tài 5**](#_Toc532148737)

[II. NỘI DUNG 6](#_Toc532148738)

[**1. Chương 1 : Giới thiệu vi xử lý 6**](#_Toc532148739)

[**1.1 Giới thiệu ARM và Dòng ARM Cortex 6**](#_Toc532148740)

[**1.2 Giới thiệu về ARM Cortex-M4 6**](#_Toc532148741)

[**1.3 Giới thiệu dòng vi xử lý được dùng trong đề tài 7**](#_Toc532148742)

[**2. Chương 2 : Cấu trúc chung của Kit 10**](#_Toc532148743)

[**2.1. Tổng quan về Kit STM32F429I-DISCOVERY 10**](#_Toc532148744)

[**2.1.1. Phần cứng của Kit 10**](#_Toc532148745)

[**2.1.2. Tính năng, giao thức kết nối của Kit 11**](#_Toc532148746)

[**2.1.3. Yêu cầu hệ thống 12**](#_Toc532148747)

[**2.1.4. Công cụ phát triển hỗ trợ Kit 12**](#_Toc532148748)

[**2.2. Vị trí các thành phần trên Kit và kích thước 12**](#_Toc532148749)

[**2.2.1. Sơ đồ khối của Kit 12**](#_Toc532148750)

[**2.2.2. Vị trí của các thành phần trên mạch 13**](#_Toc532148751)

[**2.2.3. Kích thước của Kit 14**](#_Toc532148752)

[**2.3. Chi tiết phần nhúng ST-LINK/V2 14**](#_Toc532148753)

[**2.3.1. Tổng quan về ST-LINK/V2 nhúng 14**](#_Toc532148754)

[**2.3.2. Drivers 15**](#_Toc532148755)

[**2.3.3. Sử dụng ST-Link/V2 hoặc V2-B để lập trình/gỡ lỗi tích hợp on-board 15**](#_Toc532148756)

[**2.3.4. Sử dụng ST-Link/V2 hoặc V2-B để lập trình/gỡ lỗi ứng dụng STM32 khác 16**](#_Toc532148757)

[**2.4. Chi tiết phần cứng 16**](#_Toc532148758)

[**2.4.1. Các chân trên Kit 16**](#_Toc532148759)

[**2.4.2. Nguồn và vấn đề cấp và chọn nguồn 19**](#_Toc532148760)

[**2.4.3. LEDs 20**](#_Toc532148761)

[**2.4.4. Nút bấm 21**](#_Toc532148762)

[**2.4.5. USB OTG 22**](#_Toc532148763)

[**2.4.6. Con quay hồi chuyển MEMS (ST-MEMS L3GD20) 22**](#_Toc532148764)

[**2.4.7. QVGA TFT LCD 2.4” 22**](#_Toc532148765)

[**2.4.8. 64-Mbit SDRAM 23**](#_Toc532148766)

[**2.4.9. JP3 (Idd) 23**](#_Toc532148767)

[**2.4.10. Xung nhịp OSC 23**](#_Toc532148768)

[**2.4.11. Kết nối mở rộng Kit có hỗ trợ 25**](#_Toc532148769)

[**2.5. Các mạch và sơ đồ liên quan đến Kit 28**](#_Toc532148770)

[**3. Chương 3: Những kiến thức liên quan 31**](#_Toc532148771)

[**3.1. Tổng quan về USB 31**](#_Toc532148772)

[**3.1.1. Định nghĩa 31**](#_Toc532148773)

[**3.1.2. Đặc điểm kỹ thuật BUS 32**](#_Toc532148774)

[**3.1.3. Nguồn cấp cho USB 33**](#_Toc532148775)

[**3.1.4. Phân loại USB 33**](#_Toc532148776)

[**3.1.5. Các thiết bị endpoints giao tiếp USB 34**](#_Toc532148777)

[**3.2. Giao tiếp chuẩn USB 35**](#_Toc532148778)

[**3.2.1. Chuẩn tín hiệu 35**](#_Toc532148779)

[**3.2.2. USB Host 35**](#_Toc532148780)

[**3.2.3. USB HID (Human Interface Device) 36**](#_Toc532148781)

[**3.2.4. Kịch bản hoạt động của chuẩn USB 36**](#_Toc532148782)

[**3.2.5. Chế độ truyền 38**](#_Toc532148783)

[**4. Chương 4: Ứng dụng 39**](#_Toc532148784)

[**4.1. Nạp nguồn vào Kit 39**](#_Toc532148785)

[**4.2. Nạp chương trình vào Kit 39**](#_Toc532148786)

[**4.3. Thư viện sử dụng và kỹ thuật 42**](#_Toc532148787)

[**4.4. Project trong Keil uVision 5 43**](#_Toc532148788)

[**4.4.1. Tạo Project mới 43**](#_Toc532148789)

[**4.4.2. Cấu hình Project 44**](#_Toc532148790)

[**4.4.3. Tổ chức chương trình 45**](#_Toc532148791)

[**4.5. Cách chương trình hoạt động 46**](#_Toc532148792)

[**4.5.1. Cấu hình GPIO 46**](#_Toc532148793)

[**4.5.2. Thư viện xử lý tín hiệu chuột và tín hiệu bàn phím 46**](#_Toc532148794)

[**4.6. Source code chương trình 48**](#_Toc532148795)

[**4.7. Ứng dụng minh họa ứng giao tiếp chuột và bàn phím sử dụng Kit STM32F429-DISCOVERY 54**](#_Toc532148796)

[**4.7.1. Giao tiếp với chuột 54**](#_Toc532148797)

[**4.7.2. Giao tiếp với bàn phím 57**](#_Toc532148798)

[III. PHẦN KẾT LUẬN 59](#_Toc532148799)

[**1. Kết quả sau khi hoàn thành đề tài 59**](#_Toc532148800)

[**2. Ưu, nhược điểm 59**](#_Toc532148801)

[**2.1. Ưu điểm 59**](#_Toc532148802)

[**2.2. Nhược điểm 59**](#_Toc532148803)

[**3. Những khó khăn 60**](#_Toc532148804)

[**4. Hướng phát triển của đề tài 60**](#_Toc532148805)

[IV. TÀI LIỆU THAM KHẢO 61](#_Toc532148806)

# MỞ ĐẦU

## Tóm tắt

Nhóm đã thực hiện đề tài là: Tìm hiểu Kit STM32F429I-DISCOVERY và làm ứng dụng minh họa giao tiếp chuột và bàn phím. Sau khi tham khảo tài liệu thì nhóm chọn Kit STM32F429I-DISCOVERY để làm host giao tiếp với chuột và bàn phím. Dùng linh kiện rời là chuột có dây và bàn phím liên kết qua cổng OTG mà Kit có hỗ trợ để tiến hành giao tiếp khi di chuyển chuột và nhấn nút, trái, phải, giữa đèn led trên Kit sẽ sáng theo mức độ tùy thao tác. Khi giao tiếp với bàn phím cũng tương tự nhấn phím ví dụ phím “A” để bật đèn led và bấm một phím khác như phím “N” để tắt đèn led trên Kit. Nhóm chọn Kit này vì Kit hỗ trợ chuẩn giao tiếp USB có thể chia làm 2 chế độ là USB Host và USB HID (Human Interface Devices) bên cạnh đó Kit còn nhiều chức năng có thể nâng cao và phát triển ứng dụng minh họa sau này.

Chức năng chính của ứng dụng minh họa chia làm 2 phần:

* Giao tiếp với Chuột qua cổng OTG trên Kit: khi nhấn chuột trái đèn lè trên Kit bật sáng, nhấn chuột phải đèn led tắt và nhấn chuột giữa led sẽ nhấp nháy. Khi di chuyển chuột nhanh chậm thì đèn led sáng theo mức độ, di chuyển chuột càng nhanh đèn càng sáng với mức độ cao.
* Giao tiếp với Bàn phím qua cổng OTG trên Kit: tiến hành chọn 2 phím là nút bật và tắt led, thay phiên nhấn phím “A” và “N” tương ứng là bật đèn led và tắt đèn led trên Kit

## 2. Đặt vấn đề

### 2.1. Những nghiên cứu trong và ngoài nước

* **Trong nước**

Hiện nay, lĩnh vực hệ thống nhúng tại Việt Nam tuy đã biết đến khá lâu nhưng tốc dộ phát triển vẫn còn khá chậm với rất ít sản phẩm có thể ứng dụng vào thực tế. Công việc chủ yếu vẫn là gia công nghệ phần mềm cho nước ngoài, trong đó chiếm tỉ lệ lớn là các thị trường Nhật, Mỹ và Châu Âu. Rất nhiều hãng sản xuất phần mềm lớn đã và đang đổ bộ vào thị trường Việt Nam để tận dụng lợi thế nguồn nhân lực giá rẻ và khai thác một một thị trường tiềm năng như IBM, Intel, CSC,… Đặc biệt là các khu công nghệ cao nơi chiếm phần lớn thị phần nguồn nhân công về ngành IOT, nhúng khá lớn.

Theo PGS.TS Nguyễn Hiếu Minh, Chủ nhiệm khoa Điện tử Viễn thông, Học viện Kỹ thuật mật mã, Việt Nam hiện nay mới chỉ vài trường có chuyên ngành sâu về Hệ thống nhúng và điều khiển tự động, trong khi nhu cầu nhân lực chuyên sâu về lĩnh vực này đang rất lớn và cấp bách.

* **Ngoài nước**

Thuật ngữ hệ thống nhúng hiện đã trở nên khá phổ biến. Thực tế, các sản phẩm như máy vi tính, thiết bị viễn thông, điện tử công nghiệp, điện tử dân dụng... đã và đang phát triển rất nhanh, trong đó các hệ thống nhúng điện tử và vi điện tử là cốt lõi của sự phát triển này. Theo các nhà thống kê trên thế giới, thị trường hệ thống nhúng lớn gấp khoảng 100 lần thị trường PC, trong đó số chip xử lý trong các hệ thống nhúng chiếm tới 99% số chip xử lý được sử dụng. Rất nhiều công ty, tập đoàn lớn trên thế giới đang tập trung phát triển vào lĩnh vực này và kéo theo nhu cầu khổng lồ về nhân lực với mức lương hấp dẫn như Intel, IBM hay mạch được sử dụng phổ biến nhất hiện nay là mạch ARM của công ty Acorn.

Hệ thống nhúng vốn rất đa dạng và phong phú, tuy nhiên có rất ít người biết được tầm quan trọng và sự hiện hữu của chúng trong thế giới quanh ta. Từ những hệ thống phức tạp như hàng không vũ trụ, phòng thủ quân sự, máy móc tự động trong công nghiệp, đến những phương tiện di chuyển thông thường như máy bay, xe điện, xe hơi, các trang thiết bị y tế trong bệnh viện, cho tới những thiết bị truyền hình và điện thoại di động chúng ta sử dụng hằng ngày, đâu đâu cũng có sự hiện diện của hệ thống nhúng.

Trong hơn 9 tỷ bộ vi xử lý được sản xuất hằng năm, chỉ có khoảng 150 triệu bộ (1,5%) được sử dụng cho máy vi tính cá nhân, phần còn lại (98,5%) là dành cho hệ thống nhúng (2).

Theo một thống kê khác của BCC Research Group (4) thì đến năm 2009, tổng doanh số của thị trường hệ thống nhúng trên toàn cầu sẽ đạt khoảng 88 tỷ đô-la Mỹ, với phần cứng chiếm 78 tỷ đô-la Mỹ và phần mềm chiếm 3,5 tỷ đô-la Mỹ, phần còn lại là các bo mạch nhúng. Tốc độ tăng trưởng trung bình hằng năm (AAGR) của phần mềm nhúng hiện đang đạt mức 16%.

### 2.2. Tài liệu và kiến thức liên quan

Về kiến thức 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, đặc biệt trong đề tài này cần tìm hiểu về chuẩn giao tiếp USB phân biệt USB Host và USB HID. 2 loại chuẩn USB này giao tiếp với nhau như thế nào.
* 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, các silde trên trang LMS và nguồn tư liệu từ Internet.

Tài liệu liên quan gồm có:

* 32F429IDISCOVERY Databrief
* Embedded Systems Architecture - A Comprehensive Guide for Engineers and Programmers.

### 2.3. Tính cấp thiết cần nghiên cứu của đề tài

Như đã nêu trong phần nghiên cứu trong và ngoài nước sự phát triển mạnh mẽ của khoa học kĩ thuật với nền kinh tế trí thức và xu hướng hội nhập toàn cầu như hiện nay, thế giới và Việt Nam đang thực hiện việc phát triển và ứng dụng các sản phẩm công nghệ thông tin, các sản phẩm công nghệ cao vào đời sống thực tế nhiều hơn với các xu hướng chính là IOT, nhúng, giao tiếp thông minh giữa con người và máy học. Hệ thống nhúng là một hệ thống được tích hợp cả phần cứng và phần mềm công nghệ cao phục vụ cho các bài toán chuyên dụng trong nhiều lĩnh vực công nghiệp, tự động hóa điều khiển và truyền thông. Vì thế việc nghiên cứu về các board mạch để giải quyết một số bài toán áp dụng cho đời sống được rất nhiều người quan tâm. Đề tài giao tiếp chuột và bàn phím mà nhóm thực hiện với Kit STM32F429I-DISCOVERY có liên quan đến hệ thống nhúng giao tiếp với các sản phẩm đời thường hàng ngày như bàn phím, chuột, loa có cổng USB mà từ đó ta có thể làm chủ và thao tác với các thiết bị này một cách thông minh hơn.

### 2.4. Tính cấp thiết của đề tài (lý do chọn đề tài)

Lý do chọn đề tài của nhóm là tập trung vào việc vận dụng kiến thức đã học trong hệ thống nhúng để giải quyết các bài toán giao tiếp giữa các thiết bị có cổng USB với board mạch, vi xử lý.

Việc kết hợp board giao tiếp với thời gian thực, đọc thời gian thực, hoặc việc đọc dữ liệu và xử lý tín hiệu từ các cảm biến, ở đây là cảm biến hồng ngoại hiện giờ thật sư rất cần thiết. Trong các nhà máy, việc quản lý số lượng sản phẩm trong dây chuyền mà không có sự hỗ trợ của máy móc, của cảm biến thì việc này vô cùng khó khăn để đạt được sự chính xác tuyệt đối, đảm bảo cho quá trình hoạt động hiệu quả của nhà máy. Hoặc về mảng thời gian thực cũng vô cùng quan trọng, một cái đồng hồ điện tử, đồng hồ số trên tường nhà các bạn cũng ra đời trên việc áp dụng mạch thời gian thực kết hợp với việc lập trình để hiển thị thời gian thực để chúng ta có thể thấy được cũng chính là nhờ hệ thống nhúng hỗ trợ chúng ta.

Vì thế tìm hiểu Kit STM32F429I-DISCOVERY và ví dụ minh họa giao tiếp chuột và bàn phím với Kit sẽ làm rõ các vấn đề liên quan về lý thuyết cổng USB, phân biệt USB Host và USB HID từ đó mở ra các ý tưởng sau này để làm việc chuyên sâu hơn với các cổng USB, OTG và các board mạch hỗ trợ.

### 2.5. Mục tiêu đề tài

Đề tài ra đời với mục đích giúp sinh viên tổng hợp lại các kiến thức đã tiếp thu được trong quá trình học môn Hệ thống nhúng, đồng thời có cái nhìn tổng quan về ngành lập trình nhúng.

Có được kiến thức cơ bản về board mạch, giao tiếp giữa các thiết bị mà ở đây là giao tiếp chuột và bàn phím hay nói chung là giao tiếp chuẩn USB, là bước đệm tốt hướng đến công việc sau khi ra trường.

Hiểu được lý thuyết và áp dụng lý thuyết đó vào một số thí nghiệm với Kit STM32F429I-DISCOVERY.

### 2.6. Đối tượng và phạm vi nghiên cứu

Đề tài hướng tới sinh viên thuộc các ngành điện tử viễn thông, công nghệ thông tin,… và các đối tượng có niềm đam mê, quan tâm đến nghiên cứu và phát triển hệ thống nhúng.

Phạm vi nghiên cứu chủ yếu là các giao tiếp cơ bản giữa kit STM32F429I-DISCOVERY với các thiết bị sau:

* Chuột có dây có chuẩn USB có thể kết nối OTG
* Bàn phím có chuẩn USB có thể kết nối OTG

### 2.7. Phương pháp nghiên cứu

* Đi từ những kiến thức cơ bản về board mạch, cách vận hành cấu hình đến lập trình và nhúng.
* Sử dụng các phần mềm Keil uVision từ đó biết debug và biên dịch ra file hex.
* Sử dụng phần mềm ST Link để nạp code lên Kit
* Chuẩn bị kit STM32F429I-DISCOVERY, các linh kiện cần thiết là chuột, bàn phím và các cable kết nối.
* Tìm hiểu về Kit STM32F429I-DISCOVERY, các linh kiện, thiết bị ngoại vi thông qua các datasheet của Kit và các tài liệu liên quan trên Internet và Youtube.

### 2.8. Nội dung đề tài

Nội dung đề tài sẽ đi theo các mục chính như sau:

* Giới thiệu vi xử lý ARM và dòng ARM Cortex.
* Giới thiệu về kit sử dụng trong đề tài là kit STM32F429I-DISCOVERY gồm các thành phần và cách hoạt động của mạch.
* Các kiến thức liên quan ở đây là phần lý thuyết về chuẩn giao tiếp USB, phân loại USB Host và USB HID, giao tiếp giữa 2 loại USB trên.
* Ứng dụng vào ví dụ minh họa giao tiếp chuột và bàn phím kết nối USB qua OTG với kit STM32F429I-DISCOVERY gồm:
  + Cách gắn mạch
  + Chi tiết code
  + Debug và biên dịch mã ra tập tin HEX
  + Nạp code vào Kit
  + Hình ảnh minh họa ứng dụng

# II. NỘI DUNG

## 1. Chương 1 : Giới thiệu vi xử lý

### 1.1 Giới thiệu ARM và Dòng ARM Cortex

Cấu trúc ARM (viết tắt từ tên gốc là Advanced RISC Machine) là một loại cấu trúc vi xử lý 32 bit và 64 bit kiểu RISC được sử dụng rộng rãi trong các thiết kế nhúng. Chúng có đặc điểm tiết kiệm năng lượng, vì vậy các bộ CPU ARM chiếm ưu thế trong các sản phẩm điện tử di động.

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.

Các đặc tính của ARM bao gồm:

* Cấu trúc nạp / lưu trữ.
* Hỗ trợ tập lệnh trực giao.
* Thanh ghi lớn.
* Hầu hết các lệnh được thực hiện trong 1 chu kỳ CPU.
* Chiều dài mã máy cố định, do đó dễ dàng thực hiện đường ống hóa (pipeline).

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, dòng Cortex là một lõi xử lí hoàn thiện, đưa ra một chuẩn CPU và kiến trúc hệ thống chung. Dòng Cortex gồm có 3 phân nhánh chính: dòng A dành cho các ứng dụng cao cấp, dòng R dành cho các ứng dụng thời gian thực như các đầu đọc và dòng M dành cho các ứng dụng vi điều khiển và chi phí thấp.

### 1.2 Giới thiệu về ARM Cortex-M4

Bộ vi xử lý ARM Cortex-M4 là thế hệ vi xử lý thứ 2 của dòng Cortex-M4 dựa trên kiến trúc ARMv7-M, được giới thiệu vào năm 2010. Nó được mở rộng thêm về tập lệnh và kiến trúc mới.

Cotex-M4 gồm 1 lõi CPU 32-bit, các thanh ghi 32-bit, đường dẫn dữ liệu nội bộ 32-bit và giao diện bus 32-bit, cùng một số thành phần khác. Ngoài ra, Cortex-M4 cũng đã hỗ trợ một số hoạt động liên quan đến dữ liệu 64-bit.

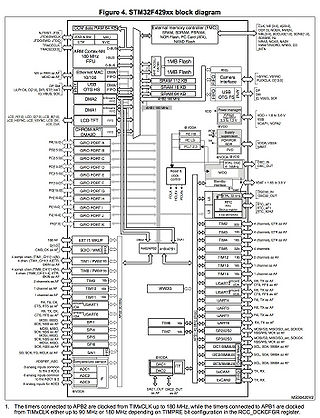
Lõi của Cortex-M4 là một CPU RISC 32-bit được thiết kế dữa trên kiến trúc Harvard (đặc trưng bằng sự tách biệt giữa vùng nhớ chứa dữ liệu và chương trình).

Nó là phiên bản đơn giản hóa từ mô hình lập trình của ARM7/9 nhưng có một tập lệnh phong phú và tối ưu hơn, hỗ trợ tốt cho các phép toán số nguyên, khả năng thao tác với bit và đáp ứng thời gian thực tốt hơn.

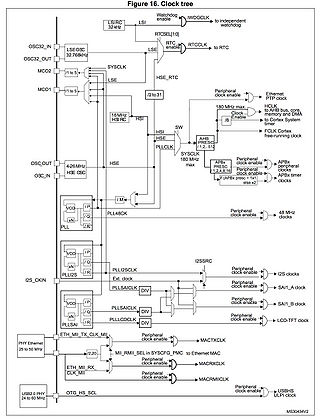
### 1.3 Giới thiệu dòng vi xử lý được dùng trong đề tài

Vi xử lý được dùng trong đề tại này là STM32F429ZIT6 được tích hợp trên Kit.

* STM32F429ZIT6 có bộ nhớ Flash 2MB được tổ chức 2 banks cho phép đọc ghi song song, 256 KB RAM (LQFP144)
* Bộ nhớ ngoài linh động với 32 bit bus dữ liệu: hỗ trợ bộ nhớ SRAM, PSRAM, SDRAM, Compact Flash/NOR/NAND
* On-board ST-LINK/V2 với lựa chọn chế độ chuyển đổi giữa debug và chạy mạch.
* ST-LINK/V2 ( có kết nối SWD để lập trình và gỡ lỗi) giúp cho việc nạp code vào mạch diễn ra nhanh chóng hơn từ việc code từ Keil uVision và đưa qua ST-Link Utility và nạp code vào mạch.
* Nguồn cấp cho Board kit là qua USB hoặc từ nguồn điện bên ngoài 3V or 5V tiện lợi vì có thể chuyển đổi từ cổng USB trên máy tính để cấp nguồn cho mạch.
* Màn hình TFT LCD QVGA 2.4”: Bộ điều khiển LCD-TFT có độ phân giải lên tới VGA với Bộ tăng tốc Chrom-ART chuyên dụng để tạo nội dung đồ họa nâng cao (DMA2D)
* Bộ cảm biến chuyển động thích hợp trong mạch L3GD20
* DMA đa năng: Bộ điều khiển DMA 16 luồng với hỗ trợ FIFO và hỗ trợ cụm
* RTC: đo độ chính xác của giây, lịch trong phần cứng
* Kết nối nâng cao
  + Thiết bị / máy chủ / bộ điều khiển OTG tốc độ cao USB 2.0 với PHY
  + Bộ điều khiển / máy chủ / OTG tốc độ cao / tốc độ cao USB 2.0 với DMA chuyên dụng, PHY và ULPI tốc độ cao
  + 10/100 Ethernet MAC với DMA chuyên dụng: hỗ trợ phần cứng IEEE 1588v2, MII / RMII



Hình 1: Sơ đồ khối STM32F429



Hình 2: Sơ đồ cây clock STM32F429

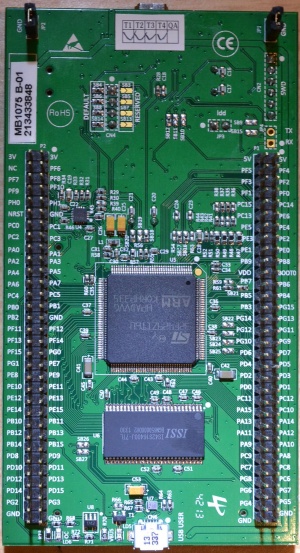
* Bản đồ bộ nhớ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Địa chỉ** | **Kích thước** | **Bus** | **Mô tả** |
| SRAM1 | 0x2000 0000 | 112KB | IDS | Bộ nhớ trong chính SRAM |
| SRAM2 | 0x2001 C000 | 16KB | S | Bộ nhớ trong phụ SRAM |
| SRAM3 | 0x2002 0000 | 64KB | S | Bộ nhớ trong phụ SRAM |
| CCM | 0x1000 0000 | 64KB | D | Bộ nhớ đôi ( chỉ có thể truy cập bởi CPU ) |
| BKPSRAM | 0x4002 4000 | 4KB | S | Backup SRAM |

Với viết tắt: I = Instruction bus, D = Data bus, S = System bus

## 2. Chương 2 : Cấu trúc chung của Kit

### 2.1. Tổng quan về Kit STM32F429I-DISCOVERY



Hình 3: Mặt trước và mặt sau của Kit

Bộ STM32F429-DISCOVERY (32F429IDISCOVERY) cho phép người dùng dễ dàng phát triển các ứng dụng với MCU hiệu suất cao STM32F429 với lõi ARM®Cortex®-M4.

Kit bao gồm một công cụ gỡ lỗi nhúng ST-LINK / V2-B, màn hình LCD TFT QVGA 2,4 ", hỗ trợ SDRAM 64 Mbit, con quay hồi chuyển ST MEMS hỗ trợ trong việc nhận diện góc quay, đầu nối micro-AB USB OTG, đèn LED và nút ấn.

#### 2.1.1. Phần cứng của Kit

* Kit đã bao gồm mạch nạp nên khi nạp không cần sử dụng đến các thiết bị như ST-Linker mà có thể nạp code qua ứng dụng ST-Link Utility
* Kit có tích hợp Micro Vi điều khiển (Micro Controller Unit – MCU) là đơn vị xử lý nhỏ, nó được tích hợp toàn bộ các bộ nhớ như ROM, RAM, các Port truy xuất, giao tiếp ngoại vi trực tiếp trên một con chip hết sức nhỏ gọn.
* Màn hình QVGA TFT LCD 2.4” hiển thị màu
* 6 Led bao gồm:
  + LD1 (đỏ/xanh) cho giao tiếp USB
  + LD2 (đỏ) dành cho cấp nguồn 3.3V
  + 2 Led người dùng: LD3 (xanh) và LD4 (đỏ)
  + 2 Led USB OTG: LD5 (xanh) VBUS và LD6 (đỏ) OC
* 2 nút bấm: User và Reset
* Cổng kết nối USB OTG với chuẩn micro-AB
* Cảm biến con quay hồi quy xoay 3 trục xyz
* Cổng USB cấp nguồn 3V hoặc 5V

#### 2.1.2. Tính năng, giao thức kết nối của Kit

Như đã đề cập trong phần cứng, phần này sẽ giải thích cụ thể về các giao thức kết nối và tính năng đặc biệt của Kit STM32F429-DISCOVERY

* Tính năng đầu tiên là tích hợp SD-Link/V2 hỗ trợ 2 chế độ nạp code chạy và debug
* Tính năng mbed-enabled : nhằm đảm bảo các nhà phát triển có thể sử dụng các công nghệ Mbed (Mbed OS, Mbed Linux và các công cụ khác) trên một loạt các phần cứng đủ điều kiện.
* Tính năng cảm biến con quay hồi quy xoay Kit theo 3 trục xyz
* Tích hợp màn hình Led có hỗ trợ cảm ứng
* Kết nối USB có các chức năng chính:
  + Cổng debug
  + Cổng COM ảo
  + Lưu trữ Mass
  + Nạp nguồn cho Kit
  + Giao tiếp USB OTG chuẩn micro-AB

#### 2.1.3. Yêu cầu hệ thống

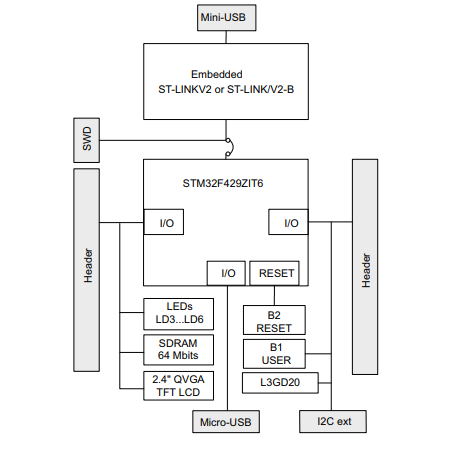
* Windows OS (XP, 7, 8, 10)
* Cáp USB Type-A tới Mini-B

#### 2.1.4. Công cụ phát triển hỗ trợ Kit

* IAR® EWARM (IAR Embedded Workbench®)
* Keil® MDK-ARM™
* GCC-based IDEs (free AC6: SW4STM32, Atollic® TrueSTUDIO®,...)
* ARM® mbed™ online

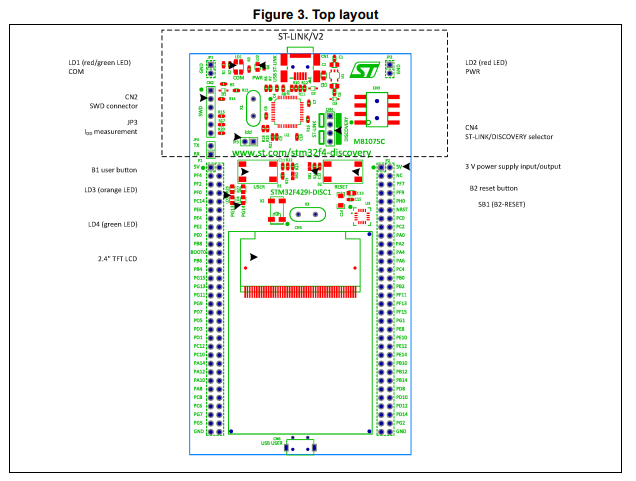
### 2.2. Vị trí các thành phần trên Kit và kích thước

#### 2.2.1. Sơ đồ khối của Kit

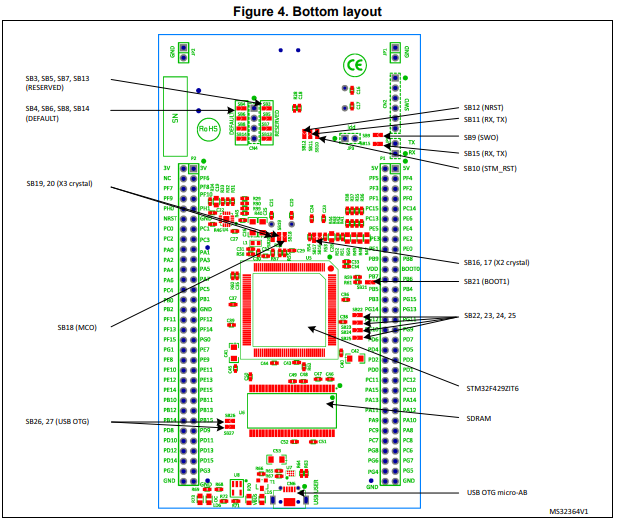


Hình 4: Sơ đồ khối các thành phần của Kit

#### 2.2.2. Vị trí của các thành phần trên mạch

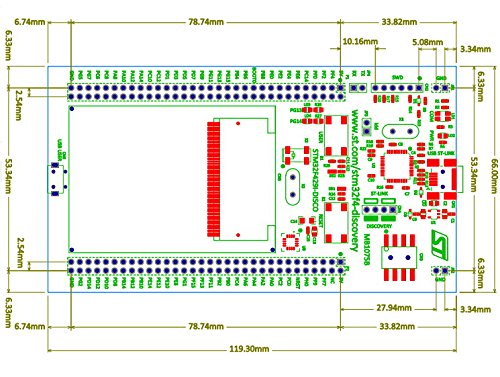


Hình 5: Vị trí các thành phần của Kit từ phía trên



Hình 6: Vị trí các thành phần của Kit từ phía dưới

#### 2.2.3. Kích thước của Kit



Hình 7: Kích thước của Kit (đơn vị: mm)

### 2.3. Chi tiết phần nhúng ST-LINK/V2

### 2.3.1. Tổng quan về ST-LINK/V2 nhúng

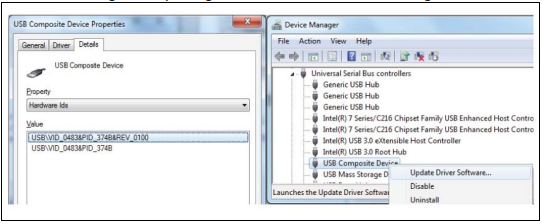
ST-LINK/V2 trên 32F429IDISCOVERY hoặc ST-LINK/V2-B trên STM32F429I-DISC1 được nhúng như công cụ lập trình và gỡ lỗi. Cổng COM ảo và bộ lưu trữ lớn USB các tính năng chỉ được ST-LINK / V2-B hỗ trợ để tương thích với Mbe

d.

ST-LINK/V2-B làm cho các Kit STM32F4429I-DISC1 được kích hoạt. Còn ST-LINK/V2 chỉ hỗ trợ SWD cho các thiết bị STM32.

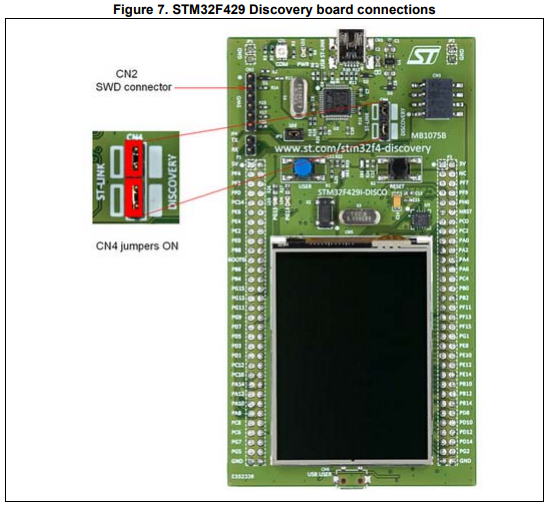
|  |  |
| --- | --- |
| Trạng thái chế độ | Mô tả |
| CN4 bật | ST-LINK/V2 (or V2-B) kích hoạt cho việc lập trình on-board (mặc định). |
| CN4 tắt | ST-LINK/V2 (or V2-B) kích hoạt cho chế độ chương trình thông qua kết nối CN3 (hỗ trợ SWD). |

#### 2.3.2. Drivers:



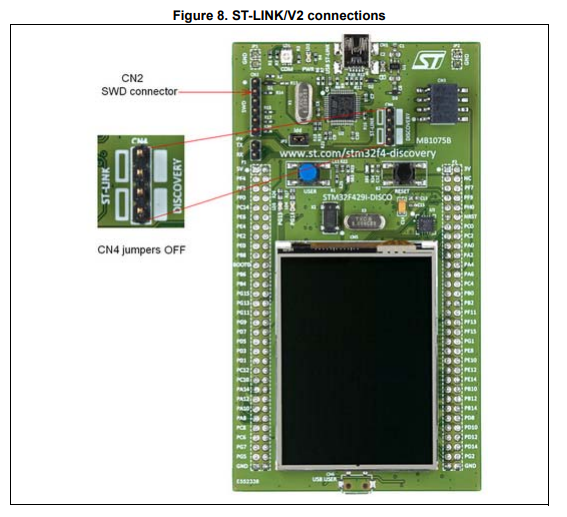
Hình 8: Cài đặt Driver để kết nối với ST-Link/V2 hoăc V2-B

#### 2.3.3. Sử dụng ST-Link/V2 hoặc V2-B để lập trình/gỡ lỗi tích hợp on-board



Hình 9: Kết nối ST-Link on-board trên Kit mặc định CN4 được bật

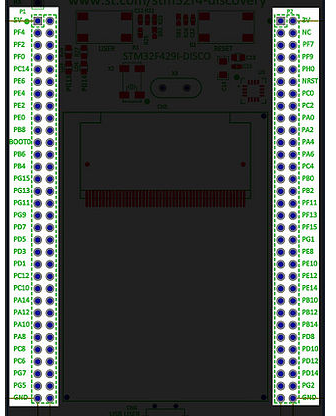
#### 2.3.4. Sử dụng ST-Link/V2 hoặc V2-B để lập trình/gỡ lỗi ứng dụng STM32 khác



Hình 10: Kết nối ST-Link với ứng dụng STM32 khác trên Kit với CN4 tắt

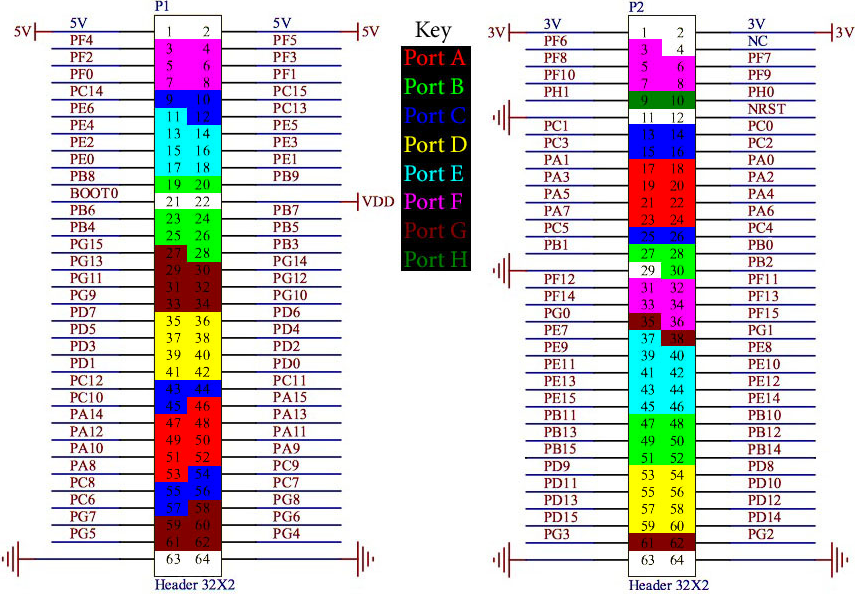
### 2.4. Chi tiết phần cứng

#### 2.4.1. Các chân trên Kit



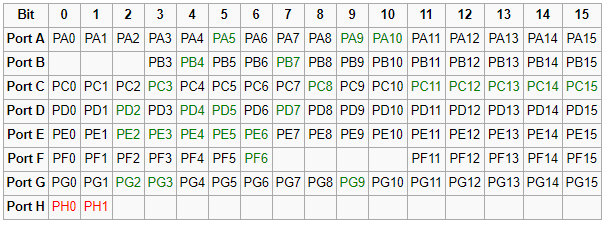
Hình 11: Tổng quan các chân trên Kit

Kit STM32F429-DISCOVERY có 143 chân chia làm 2 mức hoạt động tương ứng, mức 1 sẽ hoạt động với mức nguồn 3V và mức 2 hoạt động với mức nguồn là 5V. Và với mỗi mức sẽ được chia làm các Port A, B, C, D, E, F, G, H như hình bên dưới:

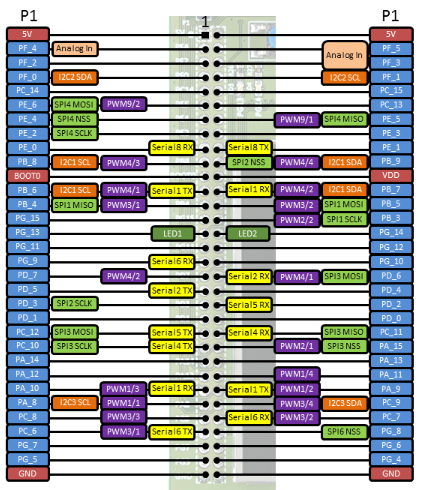


Hình 12: Mức nguồn hoạt động của các chân và các port tương ứng

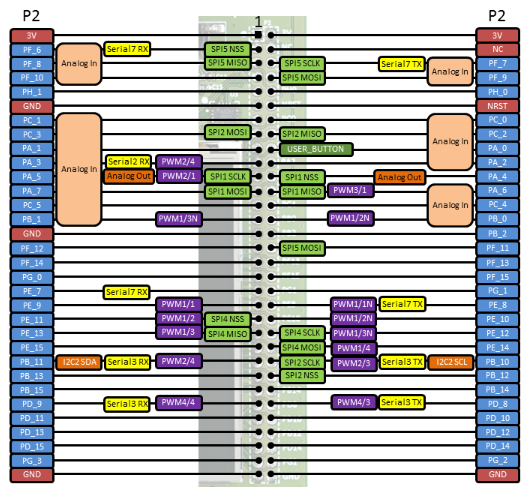
Các chân có sẵn trên Kit sẵn sàng cho hoạt động:



* Chân không được sử dụng. Có sẵn đầy đủ
* Chân được sử dụng. Có thể khả dụng nếu thiết bị ngoại vi không được sử dụng, chân đầu vào ngoại vi hoặc chân led
* Chân được sử dụng. Không sử dụng chân này.
* Chân không dung đến trong Kit



Hình 13: Chức năng của các chân P1 (3V)



Hình 13: Chức năng của các chân P1 (5V)

#### 2.4.2. Nguồn và vấn đề cấp và chọn nguồn

Nguồn điện được cung cấp bởi PC thông qua cáp USB hoặc bởi nguồn điện 5V bên ngoài như pin.

Các điốt D1 và D2 bảo vệ chân 5V và 3V khỏi nguồn điện bên ngoài:

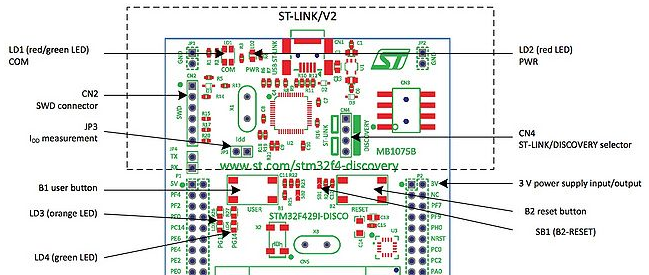
* 5V và 3V có thể được sử dụng làm nguồn cung cấp năng lượng đầu ra khi một bảng ứng dụng khác được kết nối với các chân P1 và P2. Trong trường hợp này, các chân 5V và 3V cung cấp nguồn 5 V hoặc 3 V và mức tiêu thụ điện phải thấp hơn 100 mA.
* 5V và 3V cũng có thể được sử dụng làm nguồn cung cấp năng lượng đầu vào, ví dụ: khi các đầu nối USB không được kết nối với PC. Trong trường hợp này, bo mạch STM32F429 Discovery phải được cung cấp năng lượng bởi một bộ cấp nguồn hoặc bởi một thiết bị phụ trợ tuân theo tiêu chuẩn EN-60950-1: 2006 + A11 / 2009 và phải là Điện áp cực thấp an toàn (SELV) với nguồn điện hạn chế năng lực.

Cấp nguồn bằng cáp USB và khi có nguồn điện được cấp đèn LED sẽ sáng:



Hình 14: Cấp nguồn 5V từ máy tính và đèn LED báo nguồn

#### 2.4.3. LEDs

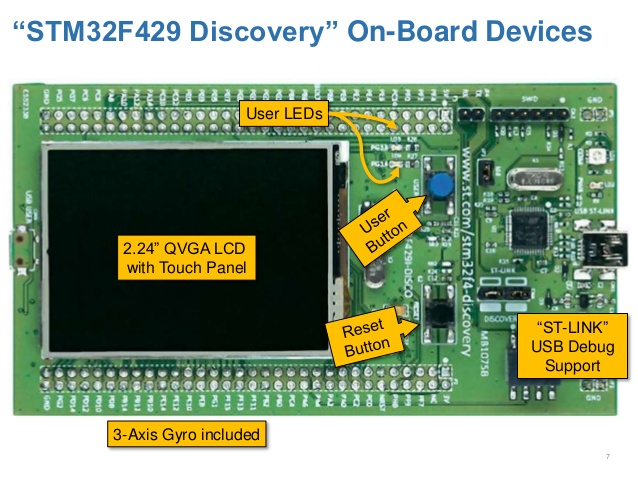


Hình 15: Vị trí Led trên Kit

Chức năng của 6 đèn Led:

* LD1 COM : LD1 mặc định sẽ màu đỏ. LD1 bật sang xanh trong quá trình giao tiếp giữa PC và ST-Link/V2
* LD2 PWR: đèn Led này mặc định màu đỏ báo mạch được cấp nguồn
* User LD3: Led xanh được bật khi có kết nối từ người dùng tới I/O PG13 ví dụ nhận tín hiệu từ USB
* User LD4: Led đỏ được bật khi có kết nooist từ người dùng tới I/O PG14 ví dụ nhận tín hiệu từ USB
* User LD5: Led xanh khi VBUS được bật trên CN6 và được kết nới tới PB13 của mạch
* User LD6: Led đỏ từ VBUS của CN6 và được kết nối tới I/O PC5 của mạch

#### 2.4.4. Nút bấm



Hình 16: Nút Reset và nút User trên Kit

Chức năng của 2 nút bấm:

* Nút User B1: Nút người dùng và Wake-Up được kết nối với I/O PA0 của STM32F429ZIT6.
* Nút Reset B2: Nút ấn được kết nối với NRST được sử dụng để reset vi xử lý STM32F429ZIT6.

#### 2.4.5. USB OTG

STM32F429ZIT6 hỗ trợ USB OTG với tốc độ tối đa trên bo mạch. Đầu nối USB Micro-AB (CN6) cho phép người dùng kết nối máy chủ hoặc thành phần thiết bị, chẳng hạn như phím USB, chuột, v.v. Hai đèn LED được dành riêng cho mô-đun này:

* LD5 (đèn LED màu xanh lá cây) cho biết khi VBUS hoạt động
* LD6 (đèn LED màu đỏ) biểu thị quá dòng từ thiết bị được kết nối

#### 2.4.6. Con quay hồi chuyển MEMS (ST-MEMS L3GD20)

L3GD20 là một cảm biến tốc độ góc ba trục siêu nhỏ gọn, công suất thấp. Nó bao gồm một yếu tố cảm biến và giao diện IC có thể cung cấp tốc độ góc đo được cho thế giới bên ngoài thông qua giao diện nối tiếp I2C / SPI.

L3GD20 có tỷ lệ đầy đủ là ± 245 / ± 500 / ± 2000 dps và có khả năng đo tốc độ với băng thông do người dùng lựa chọn. STM32F429ZIT6 điều khiển cảm biến chuyển động này thông qua giao diện SPI....

#### 2.4.7. QVGA TFT LCD 2.4”



Hình 17: Màn hình LCD 2.4” tích hợp trên Kit

Màn hình LCD có màn hình 2,41" với độ phân giải 262K màu sắc, loại của màn hình là QVGA (240 x 320) và được điều khiển trực tiếp bởi STM32F429ZIT6 bằng giao thức RGB. Vôn.

Nó bao gồm bộ điều khiển ILI9341LCD và có thể hoạt động với điện áp 2,8 ± 0,3V.

Đặc biệt màn hình hỗ trợ cảm ứng có thể sử dụng bút cảm ứng để thao tác với màn hình như ghi, vẽ,...

#### 2.4.8. 64-Mbit SDRAM

SDRAM 64 Mbit là một bộ nhớ truy cập ngẫu nhiên động, tốc độ cao được thiết kế để hoạt động trong các hệ thống bộ nhớ 3,3V chứa 67.108.864 bit. Nó được cấu hình bên trong như một DRAM bốn banks với giao diện đồng bộ. Mỗi bank 16.777.216 bit được tổ chức thành 4.096 hàng bằng 256 cột với 16 bits. Tất cả các tín hiệu được đăng ký trên cạnh dương của tín hiệu đồng hồ, CLK.

SDRAM 64 Mbit bao gồm các chức năng:

* Chế độ tự động làm mới
* Tiết kiệm năng lượng
* Tắt nguồn.

STM32F429ZIT6 đọc và ghi dữ liệu ở 80MHz.

#### 2.4.9. JP3 (Idd)

Trong phần giải thích về nhúng ST-Link/V2 vào Kit để lập trình và debug có nói đến trạng thái nhảy Jumper 3. Vậy Jumper JP3 là gì? Jumper JP3, được dán nhãn Idd, cho phép đo mức tiêu thụ STM32F429ZIT6 bằng cách loại bỏ jumper và kết nối một ampe kế. Jumper JP3 có 2 trạng thái:

* Jumper bật: STM32F429ZIT6 được cấp nguồn (mặc định).
* Jumper tắt: một ampe kế phải được kết nối để đo dòng STM32F429ZIT6, (nếu không có ampe kế, STM32F429ZIT6 không được cấp nguồn).

#### 2.4.10. Xung nhịp OSC

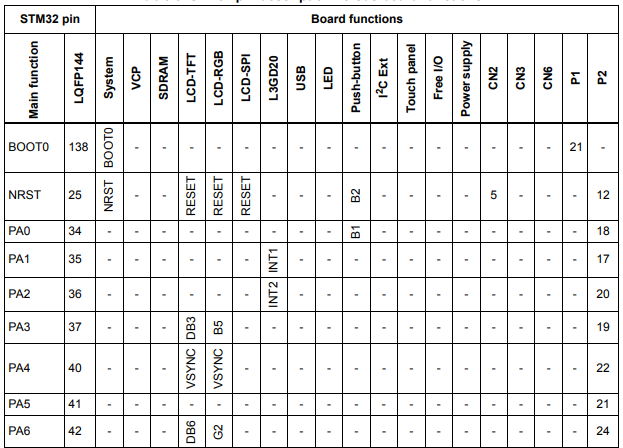
Cung cấp xung nhịp OSC:

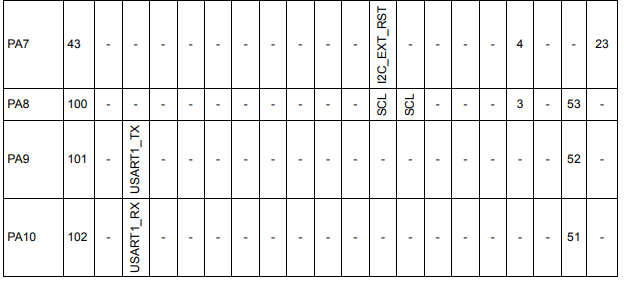
* MCO từ ST-LINK (từ MCO của STM32F429ZIT6): Tần số này không thể thay đổi, nó được cố định ở 8 MHz và được kết nối với PH0-OSC\_IN của STM32F429ZIT6. Cấu hình cần thiết là:
  + SB18 đóng, SB19 mở, R56 bị xóa
  + SB20, R57, C20, C21, X3 = không quá lớn
* Bộ tạo dao động on board (từ X3) Để biết tần số điển hình và các tụ điện và điện trở của nó, hãy tham khảo STM32F429ZIT6 bảng dữ liệu. Cấu hình cần thiết là:
  + SB18, SB19, SB20 mở
  + R56, R57, C20, C21, X3 hàn cứng
* Bộ tạo dao động từ PH0 bên ngoài (từ bộ tạo dao động ngoài qua chân 10 của đầu nối P2) Cấu hình cần có là:
  + SB19 đóng, SB18 mở, loại bỏ R56
  + SB20, R57, C20, C21, X3 = không quá lớn
* Không có bộ tạo dao động ngoài (chỉ từ bộ tạo dao động bên trong HSI). PH0 và PH1 có thể được sử dụng làm GPIO. Cấu hình cần thiết là:
  + SB18 mở, SB19 đóng, SB20 đóng, R56 bị xóa, R57 bị xóa
  + C20, C21, X3 = không quá lớn

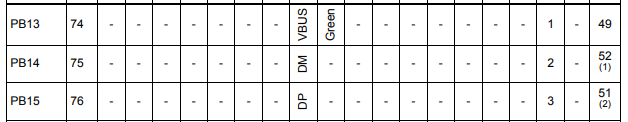
Cung cấp xung nhịp OSC 32KHz:

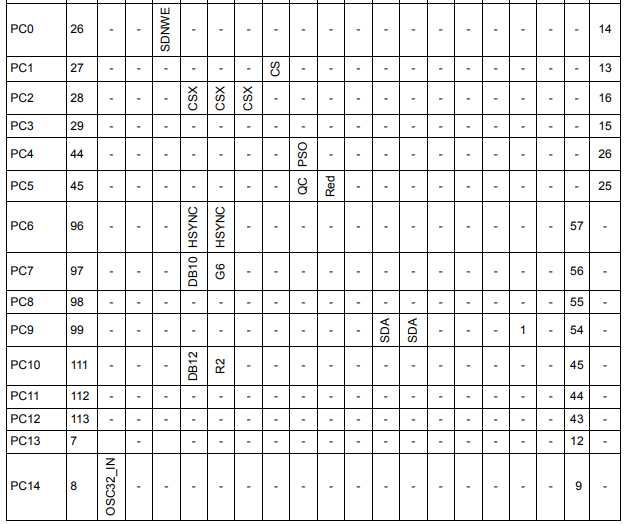
* Bộ dao động on board (từ X2, không được cung cấp). Cấu hình cần thiết là:
  + SB16 mở, SB17 mở
  + R53, R54, C23, C24, X2 hàn cứng trên Kit
* Bộ tạo dao động từ PC14 bên ngoài (từ bộ tạo dao động ngoài qua chân 9 của đầu nối P1) Cấu hình cần có là:
  + SB16 đã đóng, loại bỏ R53
  + SB17, R54, C23, C24, X2 = không quan tâm
* Không có bộ tạo dao động ngoài (PC14 và PC15 có thể được sử dụng làm GPIO. Cấu hình cần thiết là:
  + SB16 đóng, SB17 đóng, R53 bị xóa, R54 bị xóa
  + C23, C24, X2 = không quan tâm

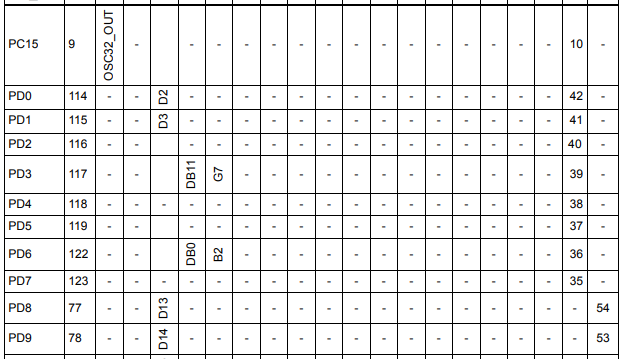
#### 2.4.11. Kết nối mở rộng Kit có hỗ trợ

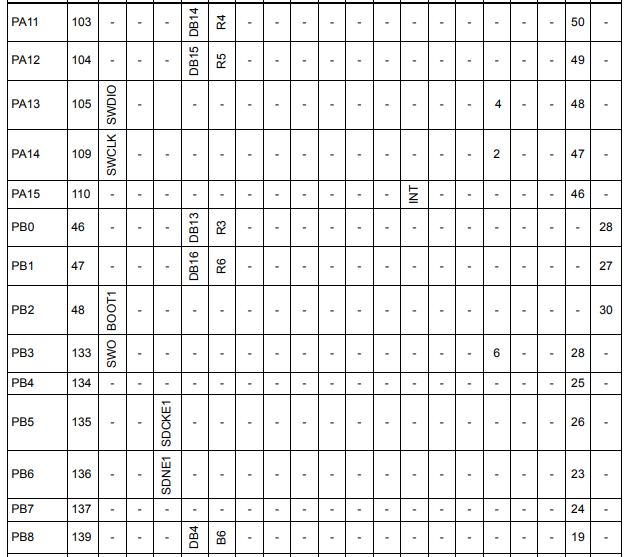


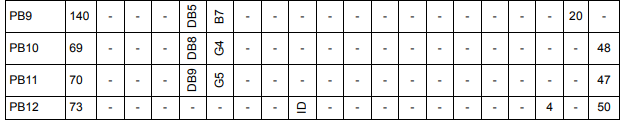






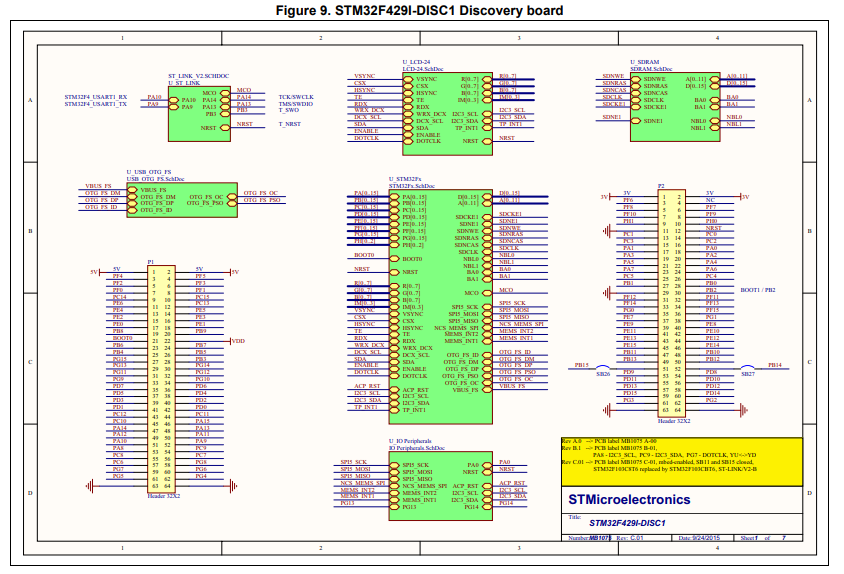




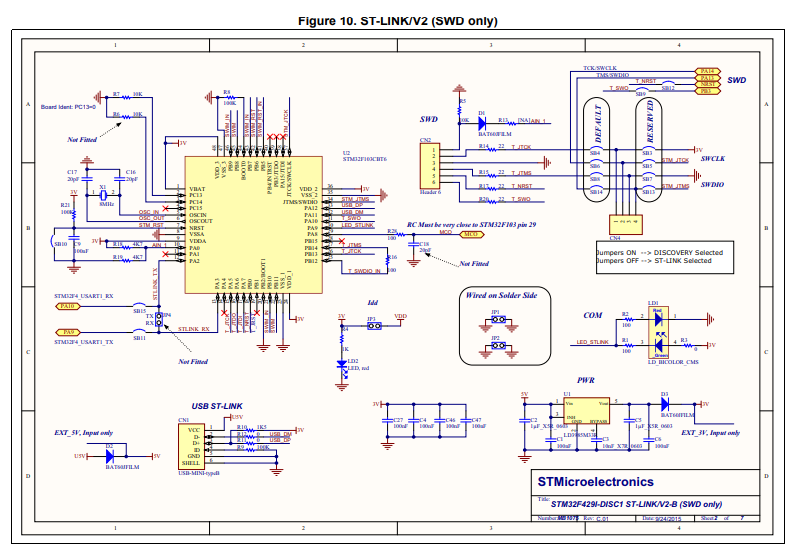


Hình 18: Danh sách các chuẩn kết nối Kit hỗ trợ

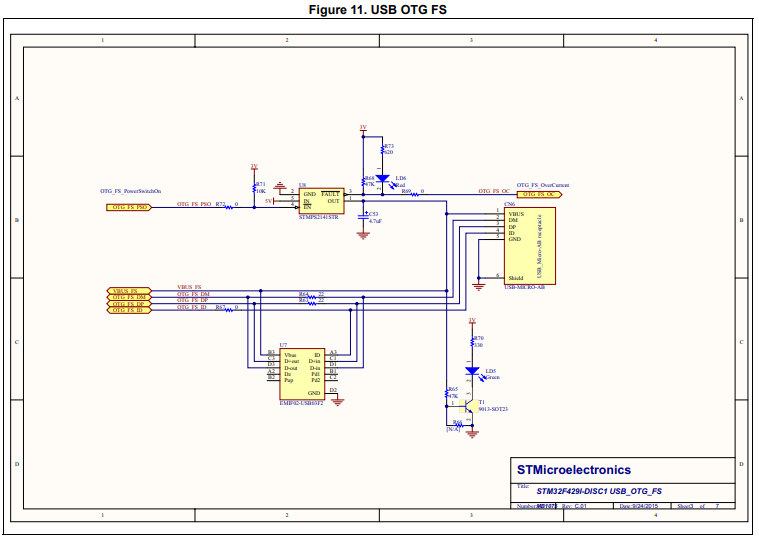
### 2.5. Các mạch và sơ đồ liên quan đến Kit



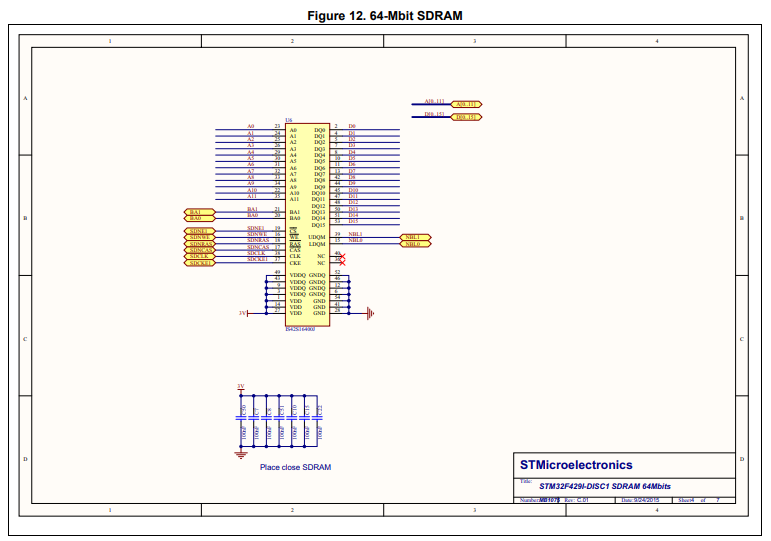
Hình 19: Mạch STM32F429I-DISC1 Discovery



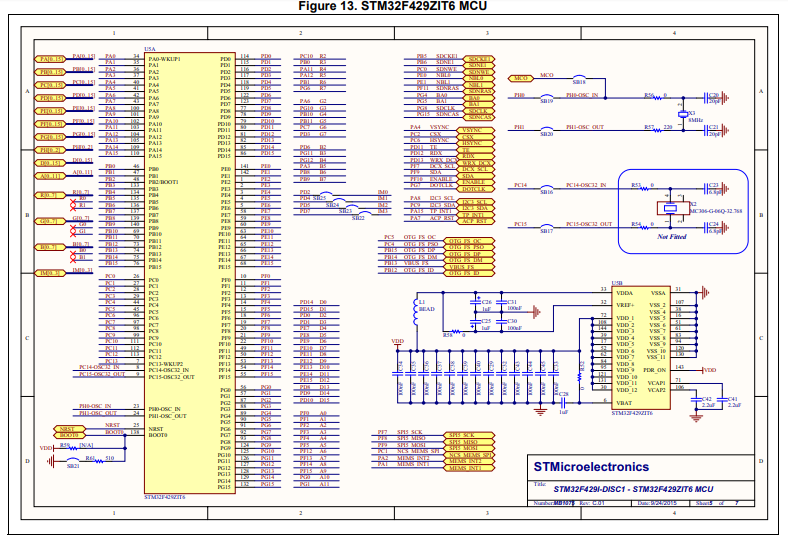
Hình 20: Mạch ST-Link/V2 nhúng trên Kit



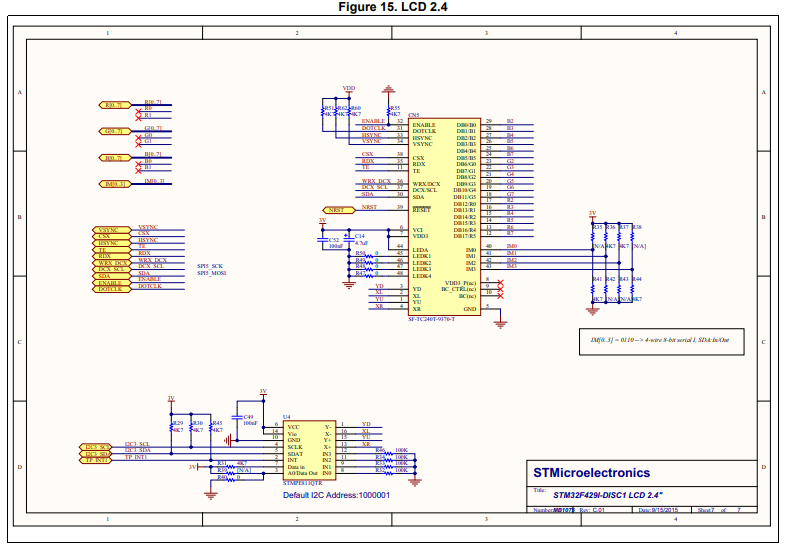
Hình 21: Giao tiếp USB OTG



Hình 22: Mạch 64-Mbit SDRAM trên Kit



Hình 23: Mạch STM32F429ZIT6 Micro Controller Unit



Hình 24: Mạch LCD 24” trên Kit

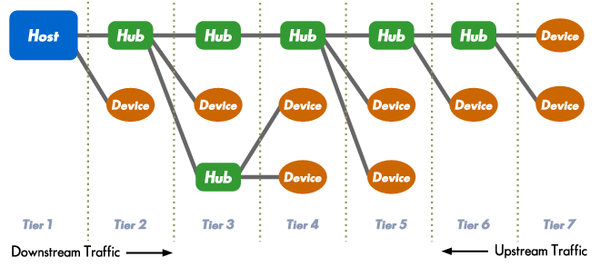
## 3. Chương 3: Những kiến thức liên quan

### 3.1. Tổng quan về USB

#### 3.1.1. Định nghĩa

Universal Serial Bus (USB) là một trong những giao thức phổ biến nhất được sử dụng trong sản phẩm điện tử tiêu dùng hiện nay, bao gồm máy tính, máy ảnh, thiết bị GPS, máy MP3, modem, máy in, máy scan,.... Các chuẩn USB ban đầu được phát triển bởi Compaq, Microsoft, Intel, và NEC, và sau đó bởi Hewlett-Packard, Lucent và Philips. Các công ty này cuối cùng hình thành tổ chức phi lợi nhuận USB Implementers Forum Inc. tổ chức này phát triển và công bố các thông số kỹ thuật của USB. Chương này mô tả các nguyên tắc cơ bản của bus USB và đi tìm hiểu các ứng dụng dựa trên USB với vi điều khiển PIC. Các bus USB là một giao thức phức tạp.

USB là một giao thức nối tiếp tốc độ cao, nó có thể cung cấp điện cho các thiết bị kết nối với nó. Một bus USB hỗ trợ kết nối lên đến 127 thiết bị (giới hạn bởi các trường địa chỉ 7-bit, lưu ý rằng địa chỉ 0 không được sử dụng vì nó có một mục đích đặc biệt) được kết nối thông qua dây cáp với chiều dài lên đến ba hoặc thậm chí năm mét. Nhiều thiết bị USB có thể được kết nối với bus thông qua Hub, Hub này có thể có 4, 8, hoặc thậm chí 16 cổng. Một thiết bị có thể được cắm vào một Hub mà Hub này kết nối đến các Hub khác, và như vậy. Số lượng tối đa cho phép là sáu tầng. Theo các đặc điểm kỹ thuật, khoảng cách tối đa của một thiết bị đến máy chủ của nó là khoảng ba mươi mét, thực hiện bằng cách sử dụng 5 hub. Đây là khoảng cách dài nhất. Nếu cho khoảng cách dài hơn ta có thể dùng chuẩn truyền Ethernet.



Hình 25: Sơ đồ Hub kết nối nhiều cổng USB

#### 3.1.2. Đặc điểm kỹ thuật BUS

Các đặc điểm kỹ thuật bus USB có hai phiên bản: phiên bản trước đó là USB 1.1, hỗ trợ tốc độ 11Mbps, trong khi phiên bản mới là USB 2.0, hỗ trợ tốc độ lên đến 480Mbps. Các đặc điểm kỹ thuật USB định nghĩa ba tốc độ dữ liệu:

* Low speed—1.5Mb/sec (Tốc độ thấp)
* Full speed—12Mb/sec (Tốc độ đầy đủ)
* High speed—480Mb/sec (Tốc độ cao)

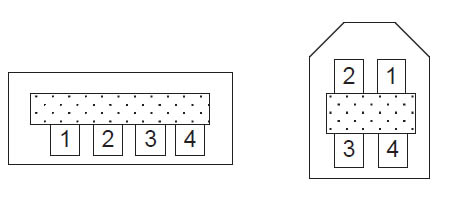
Tại thiết bị , chân dữ liệu D+ hoặc D- được kéo lên nguồn 3.3 V thông qua điện trở 1.5K. Khi thiết bị chạy ở tốc độ full-speed, chân D+ được kéo lên nguồn 3.3V, Khi thiết bị chạy ở tốc độ low speed chân D- được kéo lên nguồn 3.3V. Khi cả hai đường dữ liệu ở mức thấp, báo cho host rằng không có thiết bị nào được kết nối. Khi có thiết bị kết nối, khi đó đường dữ liệu D+ hoặc D- sẽ được kéo lên mức cao, nhờ đó mà host biết được có thiết bị đang được kết nối. Phụ thuộc đường dữ liệu nào được kéo lên mức cao host xác định được tốc độ truyền của thiết bị.

#### 3.1.3. Nguồn cấp cho USB

Nguồn tối đa cho một thiết bị bên ngoài kết nối sử dụng điện áp cấp bởi cổng USB được giới hạn trong khoảng 100mA tại điện áp 5.0V.

#### 3.1.4. Phân loại USB

Có hai loại đầu kết nối được sử dụng đó là kiểu A và kiểu B:



Hình 26: 2 loại đầu nối của USB (lần lượt đầu nối A và đầu nối B)

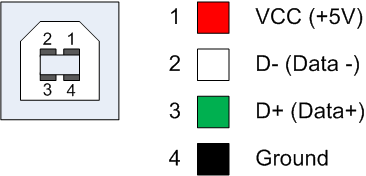
Quy định chân của đầu kết nối USB

|  |  |  |
| --- | --- | --- |
| **Chân số** | **Tên** | **Màu sắc** |
| 1 | + 5.0 V | Đỏ |
| 2 | - Data | Trắng |
| 3 | + Data | Xanh lá cây |
| 4 | Ground (đất) | Đen |

Thông số kỹ thuật còn đề cập loại đầu cắm mini kiểu B, sử dụng cho các thiết bị điện sách tay chẳng hạn như camera và các thiết bị cầm tay khác. Loại đầu cắm này có thêm chân thứ 5 gọi là chân ID, mặt dù chân này không được sử dụng.

Hai dây Data + và Data -, tạo thành 1 cặp xoắn và mang tín hiệu vi sai và trạm dữ liệu kết thúc đơn.

|  |  |  |
| --- | --- | --- |
| **Chân số** | **Tên** | **Màu sắc** |
| 1 | + 5.0 V | Đỏ |
| 2 | - Data | Trắng |
| 3 | + Data | Xanh lá cây |
| 4 | Không sử dụng | - |
| 5 | Ground (đất) | Đen |



Hình 27: USB type B với chân tương ứng

#### 3.1.5. Các thiết bị endpoints giao tiếp USB

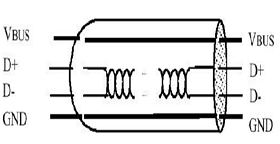
|  |  |  |
| --- | --- | --- |
| **Phân lớp thiết bị** | **Miêu tả** | **Thiết bị ví dụ** |
| 0x00 | Để dành | - |
| 0x01 | USB audio device (USB thiết bị âm thanh ) | Sound card (card âm thanh) |
| 0x02 | USB communications device ( USB Thiết bị truyền thông ) | Modem, fax |
| 0x03 | USB human interface device ( USB Thiết bị giao diện con người) | Keyboard, mouse (Bàn phím, chuột) |
| 0x07 | USB printer device (USB Thiết bị in) | Printer (Máy in) |
| 0x08 | USB mass storage device (USB Thiết bị lưu trữ ) | Memory card, flash drive |
| 0x09 | USB hub device ( USB Thiết bị hub) | Hubs |
| 0x0B | USB smart card reader device ( USB Thiết bị đọc thẻ thông minh) | Card reader (bộ đọc thẻ nhớ) |
| 0x0E | USB video device ( USB Thiết bị Video) | Webcam, scanner |
| 0xE0 | USB wireless device ( USB Thiết bị wireless) | Bluetooth |

### 3.2. Giao tiếp chuẩn USB

#### 3.2.1. Chuẩn tín hiệu

Chuẩn USB sử dụng 4 đường tín hiệu trong đó có 2 đường cấp nguồn DC (VBUS-5V và GND). 2 đường còn lại là một cặp tín hiệu vi sai (D+ và D-) cho phép truyền dữ liệu. Cặp dây tín hiệu này được nối xoắn ở bên trong nên có khả năng chống nhiễu tốt.

**Lưu ý:** cổng USB trên máy tính cho phép cấp nguồn nuôi ra bên ngoài với dòng lên tới 500mA. Như vậy, các thiết bị sử dụng ít điện năng như chuột, thẻ nhớ USB... đều có thể lấy trực tiếp nguồn từ cổng USB của máy tính mà không cần dùng thêm nguồn ngoài.

**[](https://sites.google.com/site/embedded247/embedded_system/usbprotocol/USB_signal.png?attredirects=0)**

Hình 28: Các cổng trên USB

#### 3.2.2. USB Host

USB Host là thiết bị đóng vai trò điều khiển toàn bộ mạng USB (có thể lên tới tối đa 126 thiết bị). Ví dụ như trên máy tính, USB Host được gắn trên mainboard. Để giao tiếp và điều khiển các USB device, USB Host controller cần được thiết kế tích hợp với USB RootHub (Hub mức cao nhất). Vai trò của thiết bị USB Host:

* Trao đổi dữ liệu với các USB Device
* Điều khiển USB Bus:
  + Quản lý các thiết bị cắm vào hay rút ra khỏi Bus USB qua quá trình điểm danh (Enumeration)
  + Phân xử, quản lý luồng dữ liệu trên Bus, đảm bảo các thiết bị đều có cơ hội trao đổi dữ liệu tùy thuộc vào cấu hình của mỗi thiết bị.

#### 3.2.3. USB HID (Human Interface Device)

USB Device: là các thiết bị đóng vai trò như các slave giao tiếp với USB Host. Xin lưu ý một điều hết sức quan trọng đó là các thiết bị này hoàn toàn đóng vai trò bị động, không bao giờ được tự ý gửi gói tin lên USB Host hay gửi gói tin giữa các USB Device với nhau, tất cả đều phải thông qua quá trình điều phối của USB Host. Các bạn sẽ hiểu cơ chế này rõ hơn trong phần truyền thông của chuẩn USB. Chức năng của thiết bị USB Device:

* Trao đổi dữ liệu với USB Host
* Phát hiện gói tin hay yêu cầu từ USB Host theo giao thức USB.

#### 3.2.4. Kịch bản hoạt động của chuẩn USB

Quá trình hoạt động của chuẩn USB có thể được chia làm hai giai đoạn chính

* Quá trình điểm danh: là quá trình USB Host phát hiện các thiết bị cắm vào và rút ra khỏi đường USB Bus. Mỗi khi một thiết bị tham gia vào Bus USB, USB Host sẽ tiến hành đọc các thông tin mô tả (Description) của USB Device, từ đó thiết lập địa chỉ (NodeID) và chế độ hoạt động tương ứng cho thiết bị USB Device. Các địa chỉ sẽ được đánh từ 1->126 nên về lý thuyết, chuẩn USB cho phép kết nối 126 thiết bị vào đường Bus. Khi thiết bị rút ra khỏi đường Bus, địa chỉ này sẽ được thu hồi.
* Quá trình truyền dữ liệu: để hiểu quá trình truyền dữ liệu này, chúng ta phải hiểu được hai khái niệm có thể nói là khó và quan trọng nhất trong chuẩn USB, đó là khái niệm Interface và Endpoint (Xin lưu ý là chỉ thiết bị USB device mới có Endpoint, USB Host không có Endpoint). Một thiết bị USB sẽ có thể có nhiều Interface, một Interface có thể sử dụng nhiều Endpoint. Tôi xin lấy một ví dụ sau để các bạn thấy:
  + Thẻ nhớ USB chỉ sử dụng 1 Interface theo chuẩn USB Mass storage, interface này sử dụng 3 Endpoint.
  + Bộ USB 3G sử dụng các Interface khác nhau như: CD Room, Mass storage và Communication, mỗi interface lại sử dụng nhiều Endpoint khác nhau.

Như vậy, đứng ở góc độ mức hệ thống, các Interface chính là các dịch vụ khác nhau mà thiết bị đó cung cấp còn các Endpoint chính là các cổng cần thiết cho mỗi dịch vụ. Tương ứng với khái niệm trong kiến trúc TCP/IP, ví dụ giao thức FTP là giao thức sử dụng để truyền file sẽ sử dụng hai cổng 20,21. Trong khi đó giao thức HTTP lại sử dụng port 80, giao thức Telnet sử dụng port 23.

Thực tế các Endpoint cũng như các Port trong chuẩn TCP/IP đóng vai trò như các bộ đệm truyền/nhận dữ liệu. Nhờ việc sử dụng nhiều bộ đệm mà các quá trình truyền thông được tiến hành song song và cho tốc độ cao hơn, bên cạnh đó giúp cho việc phân tách các dịch vụ khác nhau. Với chuẩn USB, các thiết bị được thiết kế với tối đa là 16 Enppoint. Các Endpoint được phân loại theo hướng truyền dữ liệu nhìn từ phía USB Host. Cụ thể:

* Các Endpoint truyền dữ liệu từ USB Device tới USB Host là endpoint IN
* Các Endoint truyền dữ liệu từ USB Host tới USB Device là endpoint OUT

Hình bên trái là kết quả chúng ta xem các thông tin cấu hình của một thiết bị USB hoạt động với 1 Interface, Interface này sử dụng hai Endpoint (0x01 và 0x82), 1 Endpoint In và một Endpoint Out, cả hai Endpoint hoạt động ở chế độ Bulk Transfer (Chi tiết chúng ta sẽ xem ở phần chế độ truyền ngay sau đây). Trên Linux, có thể dùng lệnh lsusb để xem các thông tin này với bất cứ thiết bị USB nào cắm vào Bus.

Để truyền được dữ liệu theo chuẩn USB, các thiết bị USB Device phải được kết nối với USB Host thông qua các Pipe (đường ống). Mỗi Pipe sẽ nối một Endpoint của USB Device với USB Host.

#### 3.2.5. Chế độ truyền

Chuẩn USB cung cấp cho chúng ta tổng cộng là 4 chế độ truyền, đáp ứng nhiều mục đích khác nhau tùy thuộc vào cơ chế truyền cũng như tốc độ mà người thiết kế mong muốn.

* Truyền điều khiển (Control transfer): là chế độ truyền được tất cả các thiết bị USB hỗ trợ để truyền các thông tin điều khiển với tốc độ tương đối chậm.
* Truyền ngắt (Interrupt transfer): sử dụng cho các thiết bị cần truyền một lượng dữ liệu nhỏ, tuần hoàn theo thời gian ví dụ như chuột, bàn phím. Khi đó, ví dụ cứ 10s một lần USB Host sẽ gửi request xuống và USB Device sẽ trả dữ liệu về cho USB Host (với trường hợp Interrupt In Endpoint).
* Truyền theo khối (Bulk transfer): sử dụng cho các thiết bị cần truyền một lượng dữ liệu lớn, yêu cầu độ chính xác tuyệt đối, không có ràng buộc quá chặt chẽ về thời gian thực ví dụ như thẻ nhớ USB, máy in. Cái này tương tự như giao thức TCP trong mạng Ethernet
* Truyền đẳng thời (Isochronos transfer): sử dụng cho các thiết bị cần truyền một lượng dữ liệu lớn với tốc độ rất nhanh, đảm bảo ràng buộc về thời gian thực tuy nhiên chấp nhận hy sinh độ chính xác ở một mức nhất định như các thiết bị nghe nhạc, xem phim kết nối theo chuẩn USB. Chuẩn này tương tự giao thức UDP trong mạng Ethernet.

## 4. Chương 4: Ứng dụng

### 4.1. Nạp nguồn vào Kit

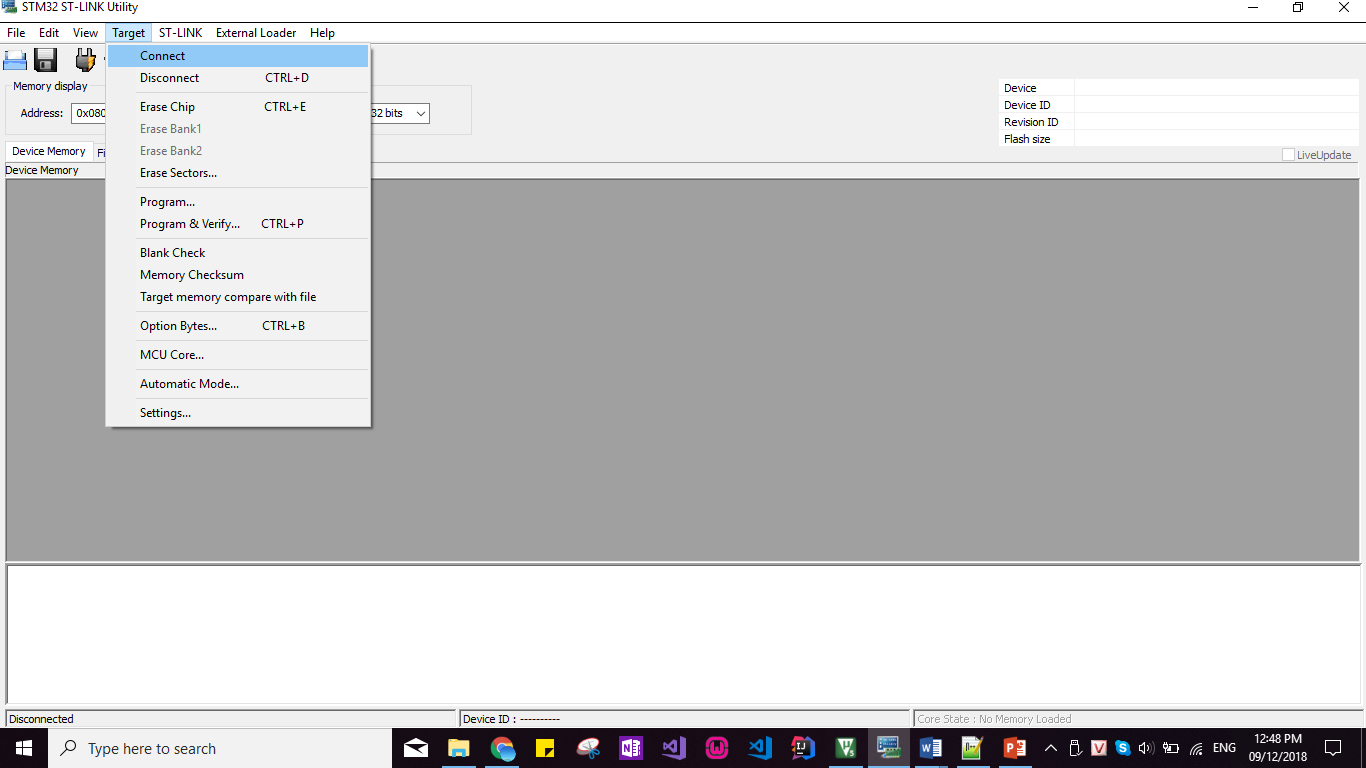
Điện áp cung cấp cho board từ 3V DC đến 5V DC nên ta chỉ cần 1 cable kết nối từ Kit sang cổng USB của PC. Từ đó chúng ta có thể thực hiện debug và nạp chương trình vào Kit. Khi Kit được cấp ngườn thì đèn led nguồn báo sáng chứng minh Kit đã có nguồn.



Hình 29: Cấp nguồn cho Kit và đèn led nguồn sáng

### 4.2. Nạp chương trình vào Kit

Vì CN3 được bật nên ST-Link/V2 được nhúng trên Kit chúng ta có thể nạp chương trình hoặc debug trực tiếp thông qua phần mềm ST-Link Utility mà không cần thêm thiết bị nạp code vật lý khác.

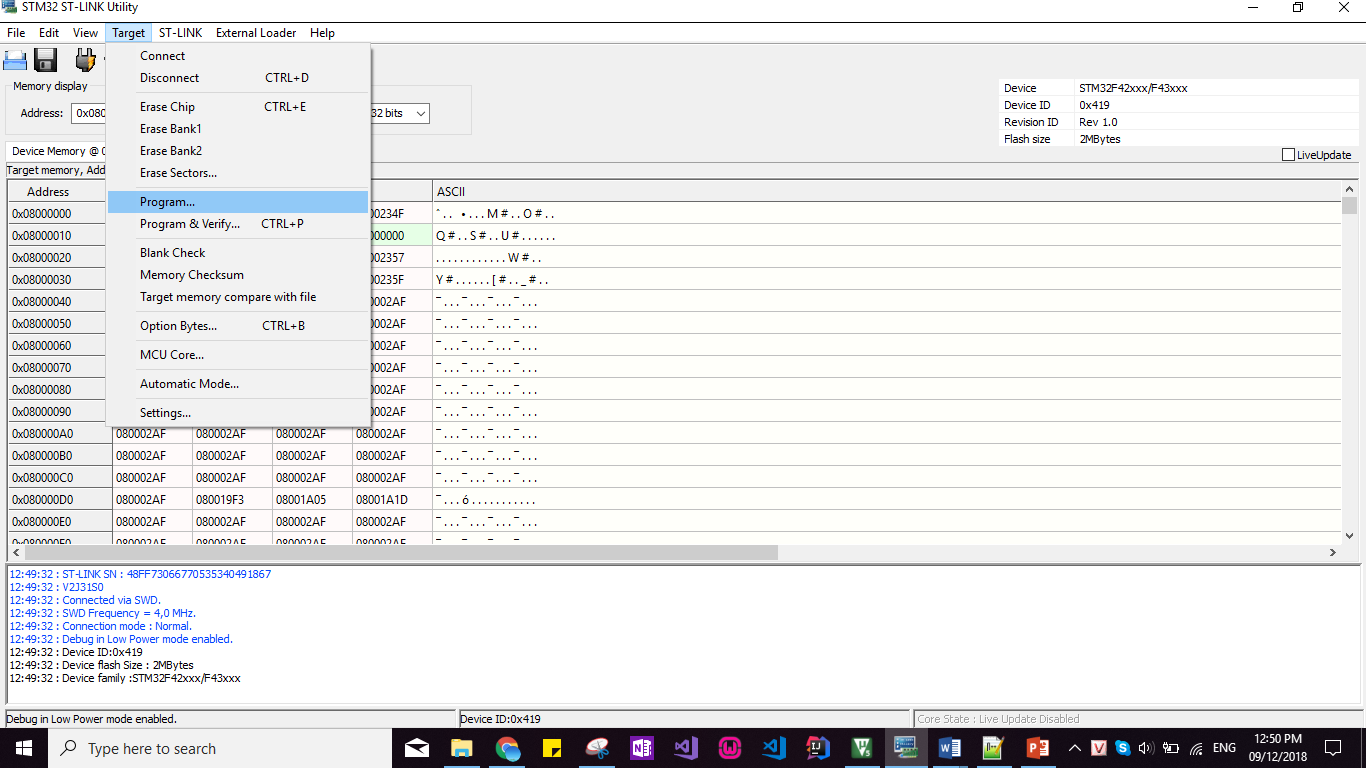


Hình 30: ST-Link Utility khi chưa kết nối tới Kit

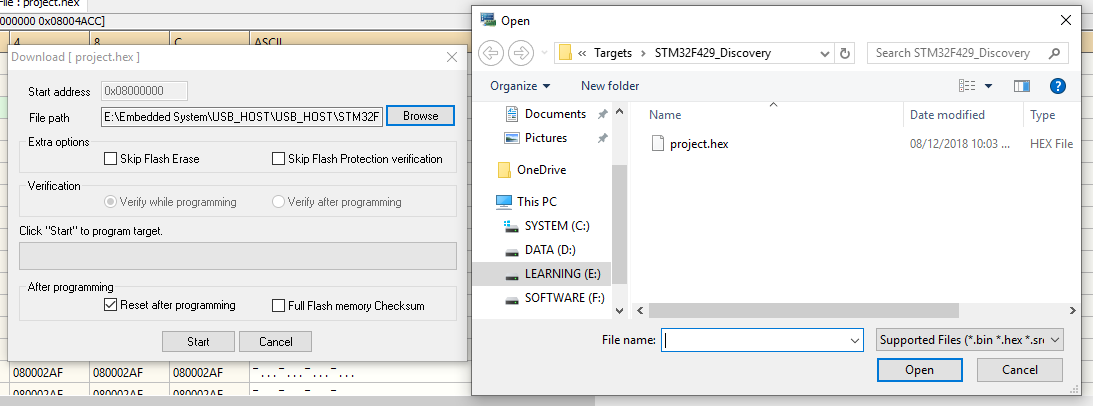
Sau khi vào Target => Connect ta sẽ có thông báo kết nối thành công như hình:



Hình 31: ST-Link Utility kết nối tới Kit thành công

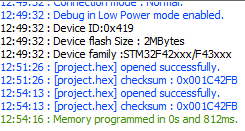


Hình 32: Chọn Target => Program... để mở cửa sổ nạp code vào Kit



Hình 33: Chọn tập tin HEX để nạp code vào Kit

Chọn vào nút Browse để tìm tới file HEX được xuất ra khi build bằng Keil uVision 5 sau đó nhấn nút Start để nạp code. Sau khi nạp code thành công vào mạch sẽ có thông báo như hình và ta chờ đèn led báo đang nạp code đến bao giờ led chuyển sang xanh là đã nạp code vò mạch thành công trên Kit và có thể chạy thử.



Hình 34: Thông báo nạp code thành công vào bộ nhớ Kit và chờ Kit chạy code và khởi động lại

### 4.3. Thư viện sử dụng và kỹ thuật

CMSIS

* STM32F4xx
* STM32F4xx RCC
* STM32F4xx GPIO
* STM32F4xx EXTI
* MISC

TM

* defines.h

USB: USB HID Host stack provided by STMicroelectronics (included in library)

|  |  |  |  |
| --- | --- | --- | --- |
| **USB** | **FS MODE** | **HS IN FS MODE** | **DESCRIPTION** |
| Data + | PA12 | PB15 | USB Data+ line |
| Data – | PA11 | PB14 | USB Data- line |
| ID | PA10 | PB12 | USB ID pin |
| VBUS | PA9 | PB13 | USB activate |

Về mặt kỹ thuật, Data + và Data - là đủ để giao tiếp USB theo bất kỳ cách nào. Nhưng, KIT STM32F429- Discovery sử dụng chân ID và VBUS để kích hoạt giao tiếp USB (chip STMPS2151 cho USB).

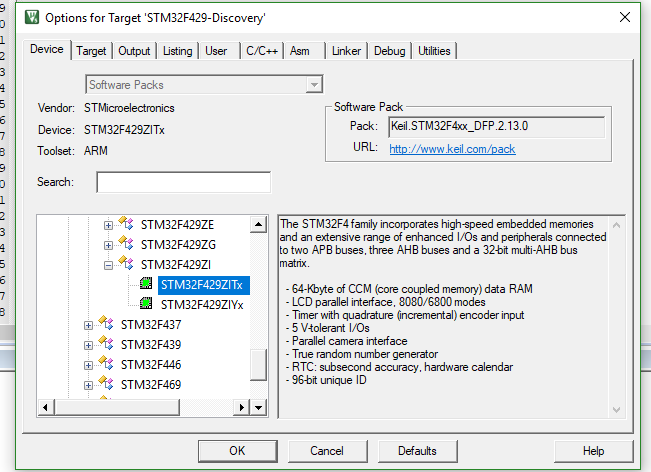
Theo mặc định, chế độ USB FS được sử dụng, cũng được sử dụng trên KIT STM32F4-Discovery.

Theo mặc định, định dạng bàn phím AZERTY đang được sử dụng. Nếu bạn sẽ làm việc với QWERTY format, hãy thêm các dòng bên dưới trong tệp định nghĩa: defines.h:

|  |
| --- |
| /\* Defines.h file \*/    /\* Keyboard supports for QWERTY mode \*/  /\* If you want to enable it, uncomment defines below \*/  #define QWERTY\_KEYBOARD |

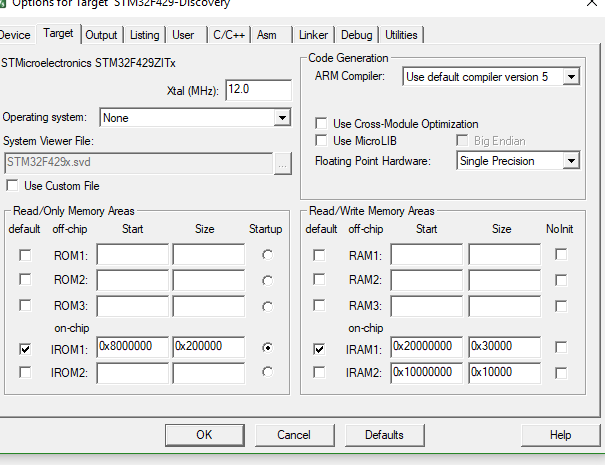
### 4.4. Project trong Keil uVision 5

#### 4.4.1. Tạo Project mới

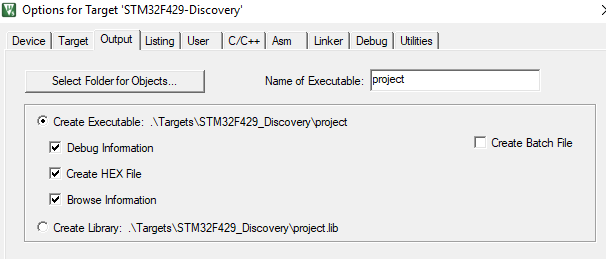


Hình 35: Chọn Device là STM32F429ZITx

#### 4.4.2. Cấu hình Project

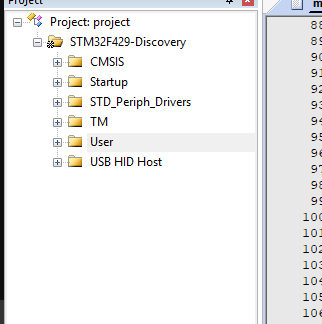


Hình 36: Cấu hình Target



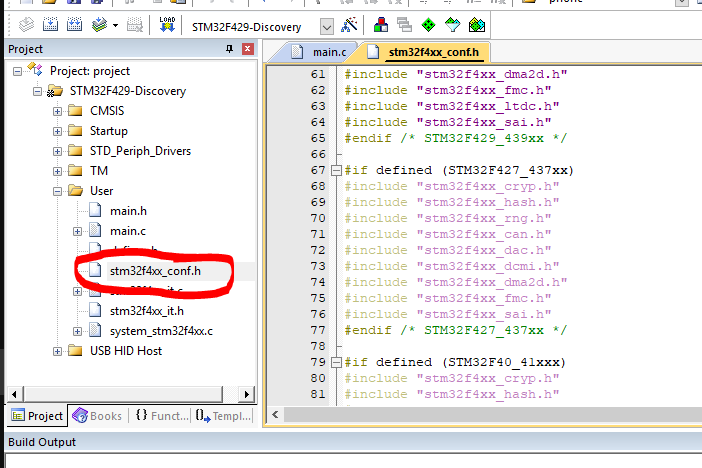
Hình 37: Cấu hình Output và chọn tạo tạp tin HEX

#### 4.4.3. Tổ chức chương trình



Hình 37: Tổ chức chương trình

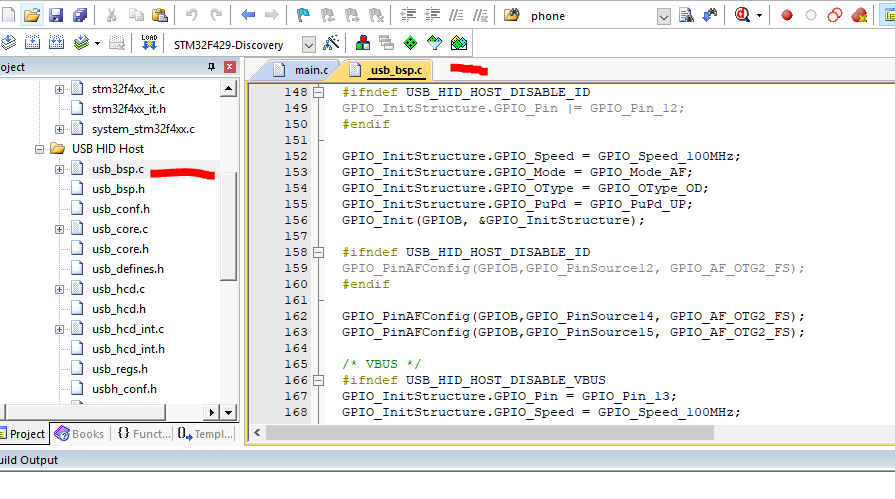
* SMSIS: sử dụng thư viện CMSIS để lập trình
* Starttup: các thư viện cấu hình cho chip STM32F429
* User: các file lập trình chính cho project
* USB HID Host : thư viện và chương trình con phục vụ USB HID



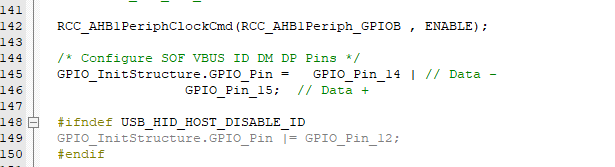
Hình 38: Chọn và khai báo thư viện

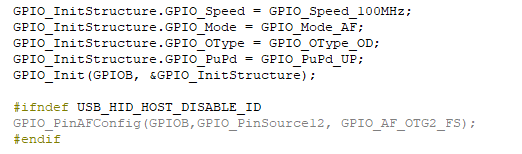
### 4.5. Cách chương trình hoạt động

#### 4.5.1. Cấu hình GPIO



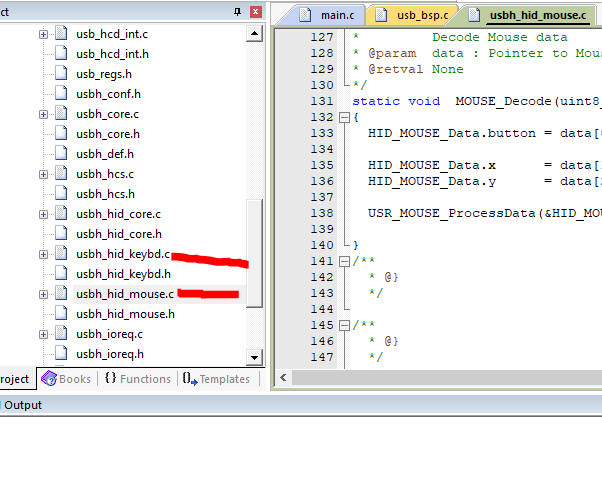
Hình 39: Cấu hình GPIO cho giao tiếp USB



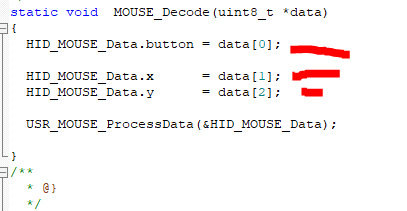


Hình 40: Khai báo GPIO cho chân D=, D- và chân ID

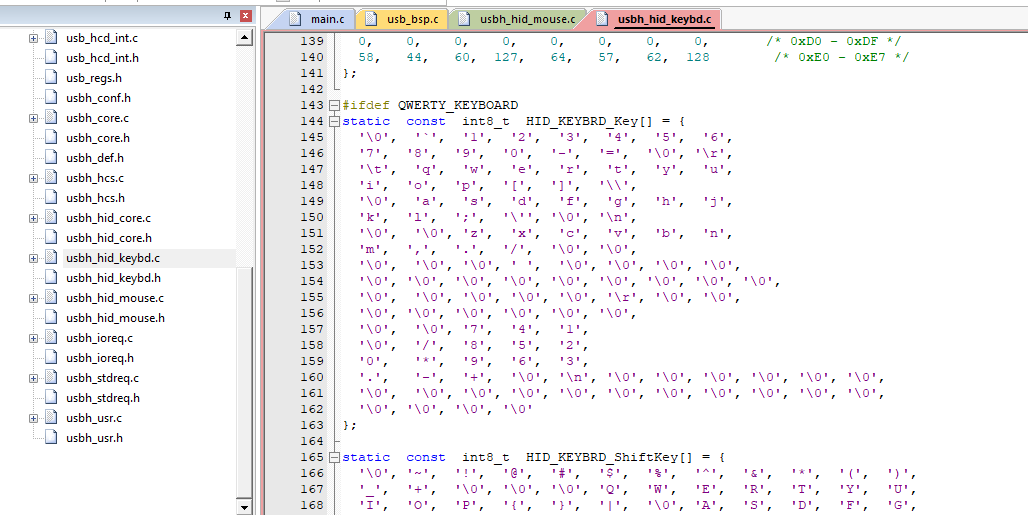
#### 4.5.2. Thư viện xử lý tín hiệu chuột và tín hiệu bàn phím



Hình 41: Thư viện usbh\_hid\_mouser và usbh\_hid\_keybd xử lý mouser và keyboarrd



Hình 42: Hàm đọc trạng thái nút nhấn và rê chuột



Hình 43: Định nghĩa bàn phím QWERTY và key tương ứng

### 4.6. Source code chương trình

/\*\*

\* Keil project for USB HID Host

\*

\* Before you start, select your target, on the right of the "Load" button

\*

\* @author Tilen Majerle

\* @email tilen@majerle.eu

\* @website http://stm32f4-discovery.com

\* @ide Keil uVision 5

\* @packs STM32F4xx Keil packs version 2.2.0 or greater required

\* @stdperiph STM32F4xx Standard peripheral drivers version 1.4.0 or greater required

\*/

/\* Include core modules \*/

#include "stm32f4xx.h"

/\* Include my libraries here \*/

#include "defines.h"

#include "tm\_stm32f4\_disco.h"

#include "tm\_stm32f4\_delay.h"

#include "tm\_stm32f4\_usb\_hid\_host.h"

#include <stdio.h>

int main(void) {

TM\_USB\_HIDHOST\_Result\_t USB\_HID\_Status; /\* USB HID Host status \*/

TM\_USB\_HIDHOST\_Keyboard\_t Keyboard\_Data; /\* Keyboard handler \*/

TM\_USB\_HIDHOST\_Mouse\_t Mouse\_Data; /\* Mouse handler \*/

/\* Initialize system \*/

SystemInit();

/\* Leds init \*/

TM\_DISCO\_LedInit();

/\* Delay init \*/

TM\_DELAY\_Init();

/\* Init USB HID \*/

TM\_USB\_HIDHOST\_Init();

while (1) {

/\* Process USB HID \*/

/\* This must be called periodically \*/

TM\_USB\_HIDHOST\_Process();

/\* Get connected device \*/

USB\_HID\_Status = TM\_USB\_HIDHOST\_Device();

/\* Switch status \*/

switch (USB\_HID\_Status) {

/\* Keyboard connected \*/

case TM\_USB\_HIDHOST\_Result\_KeyboardConnected:

/\* GREEN led ON \*/

TM\_DISCO\_LedOn(LED\_GREEN);

/\* Get keyboard data \*/

TM\_USB\_HIDHOST\_ReadKeyboard(&Keyboard\_Data);

/\* Check if keyboard button is pressed \*/

if (Keyboard\_Data.ButtonStatus == TM\_USB\_HIDHOST\_Button\_Pressed) {

if (Keyboard\_Data.Button == 'a') { /\* Turn on RED led on 'b' button \*/

TM\_DISCO\_LedOn(LED\_RED);

} else if (Keyboard\_Data.Button == 'm') { /\* Turn off RED led on 'm' button \*/

TM\_DISCO\_LedOff(LED\_RED);

}

}

break;

/\* Mouse connected \*/

case TM\_USB\_HIDHOST\_Result\_MouseConnected:

/\* RED led ON \*/

TM\_DISCO\_LedOn(LED\_RED);

/\* Get mouse data \*/

TM\_USB\_HIDHOST\_ReadMouse(&Mouse\_Data);

/\* Check buttons \*/

if (Mouse\_Data.LeftButton == TM\_USB\_HIDHOST\_Button\_Pressed) { /\* Left button \*/

TM\_DISCO\_LedOn(LED\_GREEN); /\* Toggle GREEN led \*/

}

if (Mouse\_Data.RightButton == TM\_USB\_HIDHOST\_Button\_Pressed) { /\* Right button \*/

TM\_DISCO\_LedOff(LED\_GREEN); /\* Toggle GREEN led \*/

}

if (Mouse\_Data.MiddleButton == TM\_USB\_HIDHOST\_Button\_Pressed) { /\* Middle button \*/

TM\_DISCO\_LedToggle(LED\_GREEN); /\* Toggle GREEN led \*/

}

if (Mouse\_Data.DiffX != 0 || Mouse\_Data.DiffY != 0) {

/\* If there is a difference for mouse movement at any coordinate \*/

TM\_DISCO\_LedToggle(LED\_GREEN); /\* Toggle RED led \*/

/\* If you need to move cursor now at fix position \*/

/\* Positions are stored like this \*/

//Mouse\_Data.AbsoluteX //Absolute X position for LCD

//Mouse\_Data.AbsoluteY //Absolute Y position for LCD

}

break;

/\* No device connected \*/

case TM\_USB\_HIDHOST\_Result\_Disconnected:

/\* Turn off leds \*/

TM\_DISCO\_LedOff(LED\_RED | LED\_GREEN);

break;

/\* Device is not supported \*/

case TM\_USB\_HIDHOST\_Result\_DeviceNotSupported:

/\* Device is not supported \*/

/\* Toggle leds forever \*/

TM\_DISCO\_LedToggle(LED\_RED | LED\_GREEN);

/\* Delay \*/

Delayms(50);

break;

/\* Error occurred somewhere \*/

case TM\_USB\_HIDHOST\_Result\_Error:

/\* Error occurred \*/

/\* Toggle RED LED forever \*/

TM\_DISCO\_LedToggle(LED\_RED);

/\* Delay \*/

Delayms(50);

break;

/\* Library is not initialized \*/

case TM\_USB\_HIDHOST\_Result\_LibraryNotInitialized:

/\* Library is not initialized \*/

/\* Toggle GREEN LED \*/

TM\_DISCO\_LedToggle(LED\_GREEN);

/\* Delay \*/

Delayms(50);

break;

}

}

}

### 4.7. Ứng dụng minh họa ứng giao tiếp chuột và bàn phím sử dụng Kit STM32F429-DISCOVERY

#### 4.7.1. Giao tiếp với chuột

Khi mới cắm chuột led đỏ sáng báo có tín hiệu kết nối USB từ chuột.



Hình 44: Kit nhận tín hiệu USB từ chuột có kết nối

Khi nhấn chuột trái thì Kit nhận tín hiệu chuột và có dữ liệu truyền vào là chuột trái được nhấn và xử lý bật led.



Hình 45: Kit nhận tín hiệu USB và sự kiện nhấn chuột trái

Khi nhấn chuột phải thì Kit nhận tín hiệu chuột và có dữ liệu truyền vào là chuột phải được nhấn và xử lý tắt led xanh.



Hình 46: Kit nhận tín hiệu USB và sự kiện nhấn chuột phải

Khi di chuyển chuột thì Kit nhận tín hiệu USB từ chuột và dữ liệu truyền vào là tọa độ di chuyển của chuột khác X, Y ban đầu khi mới kết nối. Khi đó đèn led sẽ toggle sáng theo mức độ di chuyển của chuột.



Hình 47: Kit nhận tín hiệu USB và sự kiện di chuyển chuột

#### 4.7.2. Giao tiếp với bàn phím



Hình 48: Kit nhận tín hiệu USB và sự kiện nhấn phím “A” sẽ bật led đỏ sáng



Hình 49: Kit nhận tín hiệu USB và sự kiện nhấn phím “M” sẽ bật led đỏ tắt

# III. PHẦN KẾT LUẬN

## 1. Kết quả sau khi hoàn thành đề tài

Nhóm đã hoàn thành đề tài “Giao tiếp chuột và bàn phím”, cũng như hệ thống lại những kiến thức về hệ thống nhúng bao gồm như dòng vi xử lý, dòng kit, những kiến thức liên quan và ứng dụng của đề tài.

Nghiên cứu và thực hành giao tiếp kit STM32F429-DISCOVERY với chuột và bàn phím làm việc giao tiếp với nhau qua cổng USB chuẩn tín hiệu USB, qua đó nắm thêm được nhiều kiến thức để làm tài liệu cho học tập và nghiên cứu sau này. Bên cạnh đó còn nắm được một phần lý thuyết về mô hình USB Host – USB HID và hiểu về cách thức hoạt động của chuột và bàn phím giao tiếp với các Kit.

Kết quả đạt được sau khi hoàn thành để tài thật sự rất có ý nghĩa trong việc phần nào đó định hình được những công việc liên quan đến hệ thống nhúng hiện tại, những công việc mà các robot công nghiệp, hay các robot, xe mô hình,.. hoạt động như thế nào. Điều đó sẽ thúc đẩy một phần đam mê về khoa học, về chế tạo robot của các thành viên trong nhóm cũng như trong lớp học Hệ thống nhúng.

## 2. Ưu, nhược điểm

### 2.1. Ưu điểm

* Kit có hỗ trợ nạp code trực tiếp hoặc debug qua chương trình ST-Link Utility nên không khó khăn trong việc vẽ mạch và nạp code vào Kit.
* Có thư viện hỗ trợ khá nhiều vì Kit có cộng đồng hỗ trợ lớn
* Thao tác với Kit không quá khó không mất nhiều thời gian để làm quen
* Kit dễ dàng tìm mua với các shop linh kiện offline và online

### 2.2. Nhược điểm

* Kiến thức của nhóm hạn hẹp. Không chuyên sâu nhiều vào phần cứng các thiết bị.
* Thông tin kỹ thuật và mô tả phần cứng viết bằng tiếng Anh mà nhóm tự dịch nên không thể không tránh khỏi những sai sót, không hiểu đúng nghĩa từ đó trong chuyên môn.
* Code sử dụng thư viện nên chưa làm rõ nhiều vào việc code thanh ghi

## 3. Những khó khăn

* Phần lý thuyết khá khó và phải tìm hiểu từ đầu 100% nên còn vướng mắc một số trong việc giao tiếp tín hiệu.
* Vì code có sử dụng thư viện từ cộng đồng nên mất nhiều thời gian để tìm hiểu và đọc các hàm để sử dụng được
* Không có người hướng dẫn cụ thể cũng như đây là ứng dụng mà nhóm tự tìm hiểu và làm theo thư viện trên diễn đàn.
* Kit hỗ trợ khá nhiều nên giá thành cao

## 4. Hướng phát triển của đề tài

* Không cần dùng nguồn của Laptop cũng như Máy tính bàn mà sẽ dùng nguồn 5V DC được cung cấp từ nguồn bên ngoài.
* Hiển thị vẽ chuột lên màn hình LCD
* Gõ chữ nhận tín hiệu và xuất ra chữ trên màn hình LCD
* Thao tác cảm ứng và cảm biến con quay quy hồi trên Kit để thao tác cảm ứng chạm và xoay màn hình
* Hỗ trợ Kit thành một máy tính thu nhỏ có thể giao tiếp USB với nhiều loại USB HID khác như loa, gamepad,...

# IV. TÀI LIỆU THAM KHẢO

1. <https://www.thesaigontimes.vn/Home/business/other/15662/> xu hướng phát triển của hệ thống nhúng

2. <https://www.st.com/en/evaluation-tools/32f429idiscovery.html?fbclid=IwAR17vsGR7GLkLQ5r6PwRQO_fwrcqepk8p-Iq68AxtrG6O1WjXNOt7C1ia4g#design-scroll>, thông tin kỹ thuật của Kit

3. <https://www.st.com/content/ccc/resource/technical/document/user_manual/6b/25/05/23/a9/45/4d/6a/DM00093903.pdf/files/DM00093903.pdf/jcr:content/translations/en.DM00093903.pdf> , chi tiết phần cứng của Kit

4. <https://stm32f4-discovery.net/2014/05/all-stm32f429-libraries-at-one-place/> , thư viện của Kit hỗ trợ SPI, USB, ...

5. <https://stm32f4-discovery.net/2014/08/library-31-usb-hid-host-for-stm32f4xx-devices/> , làm việc với USB HID như chuột và bàn phím

6. <https://www.sciencezero.org/index.php?title=STM32F429_Microcontroller> , thông tin vi xử lý STM32F429ZIT6

7. <https://www.youtube.com/watch?v=QGyErVUaerM&feature=youtu.be&fbclid=IwAR2fPlnwxuQFFr1__HrOxIV0RAnpjaf6b7rS50jQHezgw47i5C09hqYJKao&ab_channel=H%E1%BB%8CCIT-%C4%90I%E1%BB%86NT%E1%BB%AC> , lý thuyết giao tiếp chuẩn USB

8. <https://sites.google.com/site/embedded247/embedded_system/usbprotocol> , lý thuyết giao tiếp chuẩn USB và cách truyền dữ liệu