**TÀI LIỆU HƯỚNG DẪN THỰC HÀNH**

**Tên bài thực hành:**

**Buổi 2: Thực hành chuyên sâu: lập trình ngắt vi điều khiển STM32**

**1/2023**

**MỤC LỤC**

[**1. CƠ SỞ LÝ THUYẾT 2**](#_heading=h.35nkun2)

[**2. BÀI THÍ NGHIỆM THỰC HÀNH 8**](#_heading=h.1ksv4uv)

[**2.1. Mục đích - Yêu cầu 8**](#_heading=h.44sinio)

[**2.1.1. Mục đích 8**](#_heading=h.2jxsxqh)

[**2.1.2. Yêu cầu 8**](#_heading=h.z337ya)

[**2.1.3. Thời gian thực hiện và chia nhóm 8**](#_heading=h.3j2qqm3)

[**2.2. Chuẩn bị 8**](#_heading=h.1y810tw)

[**2.2.1. Danh mục thiết bị thực hành 8**](#_heading=h.4i7ojhp)

[**2.2.2. Sơ đồ đấu nối trang thiết bị 9**](#_heading=h.2xcytpi)

[**2.3. Nội dung 10**](#_heading=h.1ci93xb)

[**2.3.1. Các bước thực hiện 10**](#_heading=h.3whwml4)

[**2.3.2. Code chương trình đầy đủ 19**](#_heading=h.2bn6wsx)

[**2.3.3. Kết quả đạt được 22**](#_heading=h.qsh70q)

[**3. PHỤ LỤC 23**](#_heading=h.3as4poj)

# 1. CƠ SỞ LÝ THUYẾT

**Nguyên lý** **lập trình ngắt cho vi điều khiển STM32**

*a)* *Khái niệm ngắt (Interrupts)*

+ Là tín hiệu được tạo ra bởi các ngoại vi của vi điều khiển và được gửi đến vi xử lý. Các tín hiệu này được gọi là Interrupt Request (IRQ).

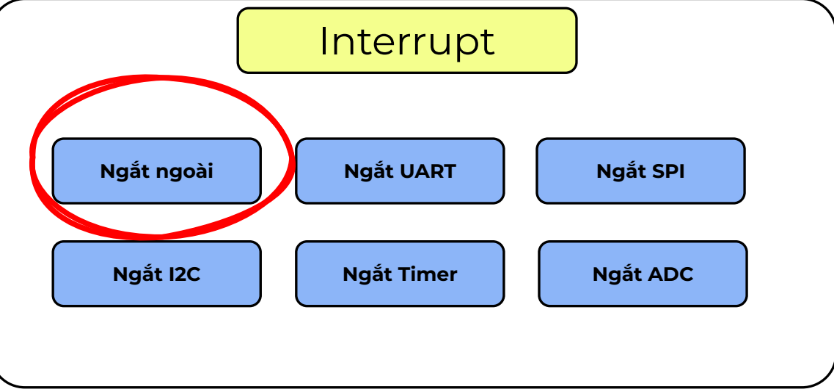
+ Khi gặp ngắt vi xử lý sẽ tạm ngưng thực thi chương trình bình thường đểthực thi một đoạn chương trình đặc biệt là: Interrupt Service Routine (ISR) – Interrupt Handler  
+ Các tín hiệu yêu cầu Interrupt Request (IRQ) được gửi vào khối xử lý ngắt (NVIC) của vi xử lý

*b) Phân loại ngắt*

+ Bao gồm: System Exception và Interrupts

+ System Exception là các điều kiện, các ngoại lệ được sinh ra bởi vi xử lý => Bên trong nội bộ vi xử lý sinh ra các Exceptions.

+ Interrupts là một phần của Exception, những điều kiện, ngoại lệ từ bên ngoài (Ngoại vi) được gửi đến vi xử lý.



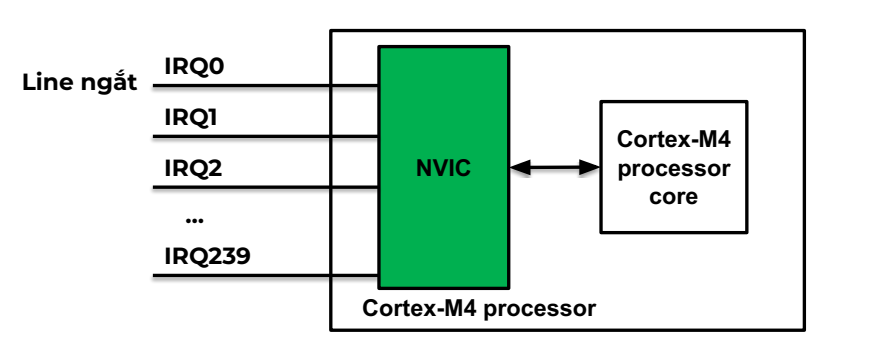
*Hình 1: Một số ngắt phổ biến*

*c) NVIC – Bộ xử lý ngắt*

+ Là một ngoại vi của lõi xử lý ARM-Cortex M

**+** Cấu hình:

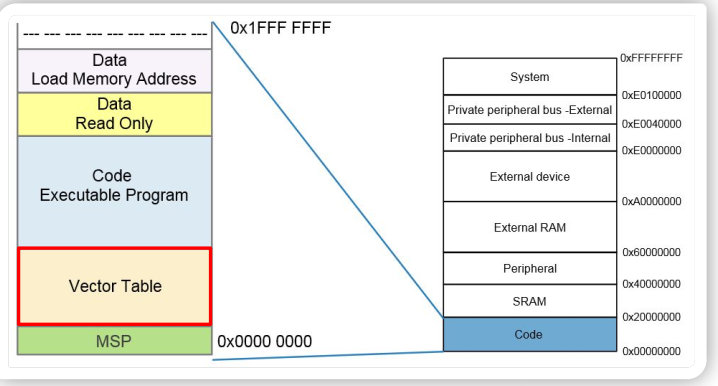
* Enable: Cho phép nhận yêu cầu ngắt từ ngoại vi
* Disable: Không cho phép nhận yêu cầu ngắt từ ngoại vi
* Pend: Thay đổi trạng thái chờ được xử lý
* Cấu hình độ ưu tiên các ngắt
* Đọc trạng thái các ngắt (active/pending)



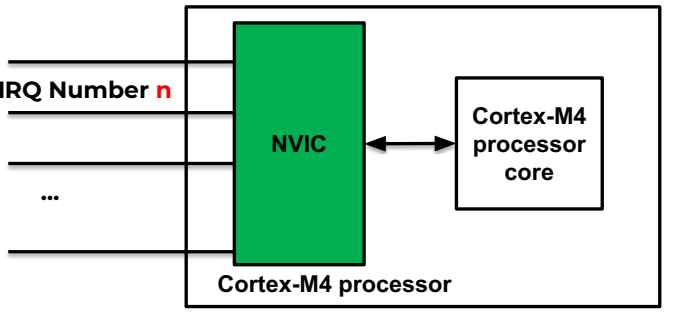
*Hình 2: Các tín hiệu Interrupts từ các ngoại được gửi vào khối NVIC  
thông qua các Line ngắt*

*d) Nguyên lý hoạt động ngắt*

+ Vector Table nằm ởvùng nhớ cố định trong memory map ở phân vùng Code



+ Với một Interrupt Number n thì NVIC sẽ sử dụng n để tìm ra địa chỉ của Interrupt Service Routine trong Vector Table



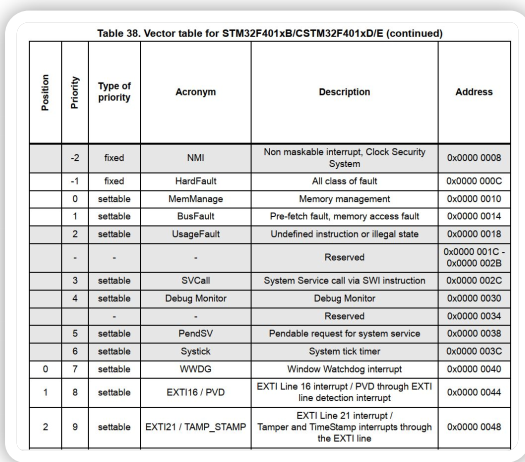
+ Bảng vector ngắt trong STM32F401RE

● Type of priority:

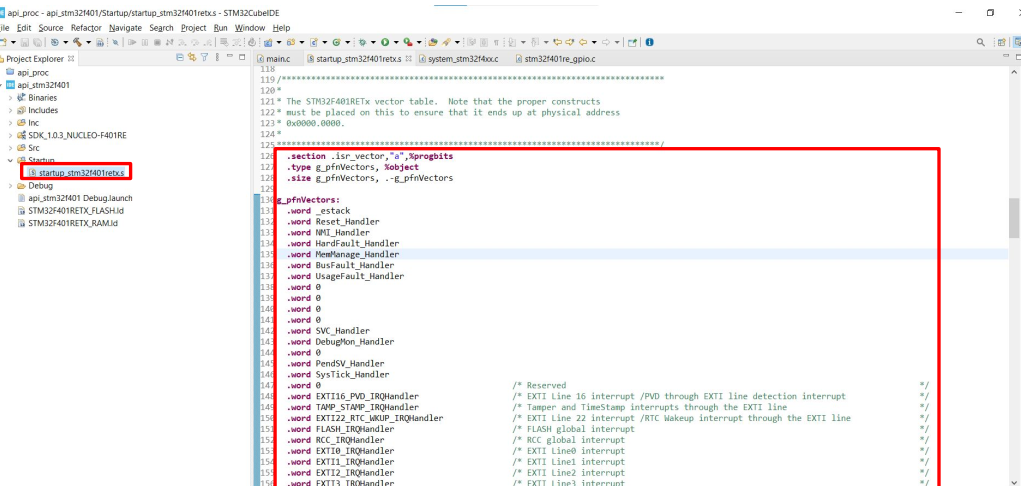
○ fixed: cố định

○ setable: có thể thiết lập.  
● Address: địa chỉ của ngắt trong  
memory map

Lưu ý: Mức độ ưu tiên của ngắt xếp từ trên xuống dưới.

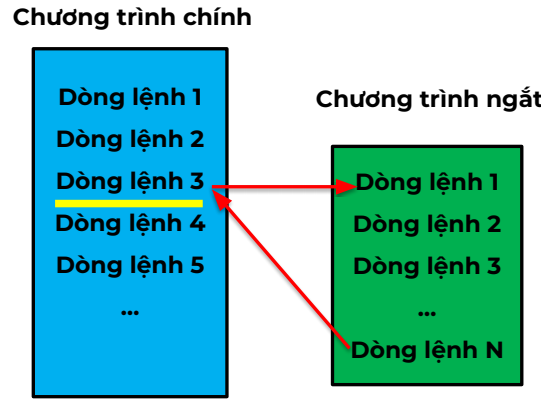
**

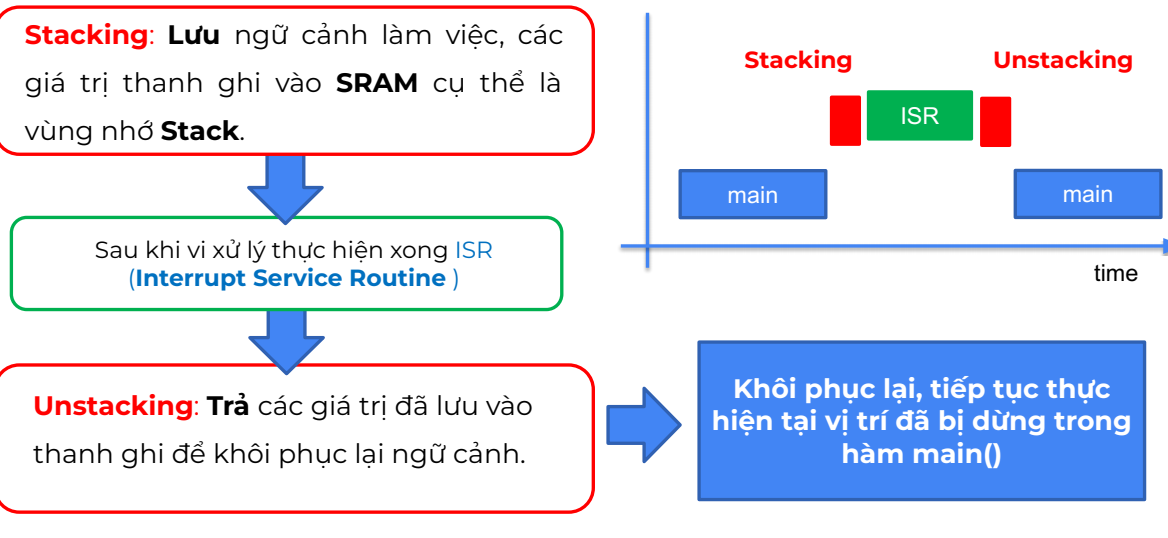
*Hình 3:* *BẢNG VETOR NGẮT TRONG STM32CubeIDE*

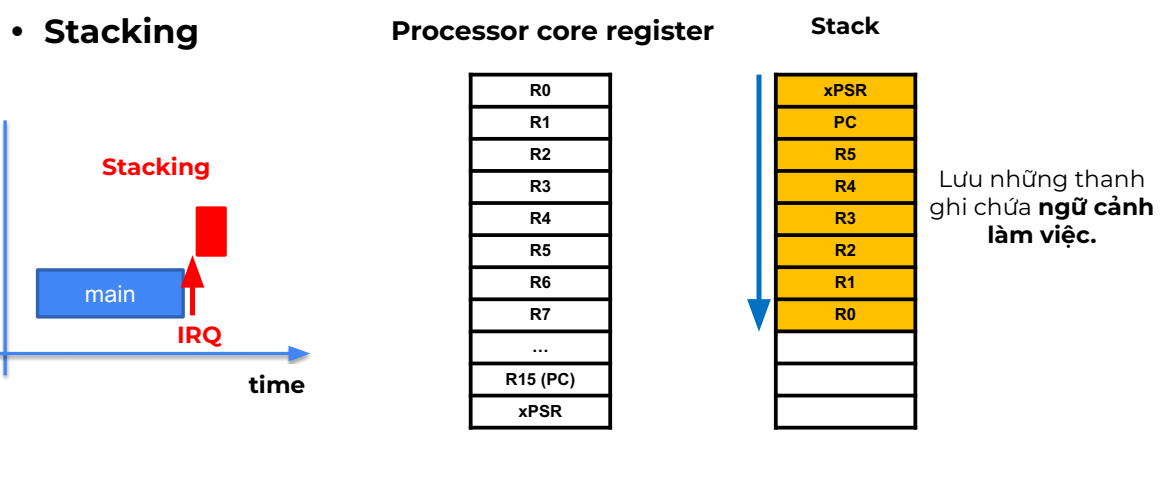
**

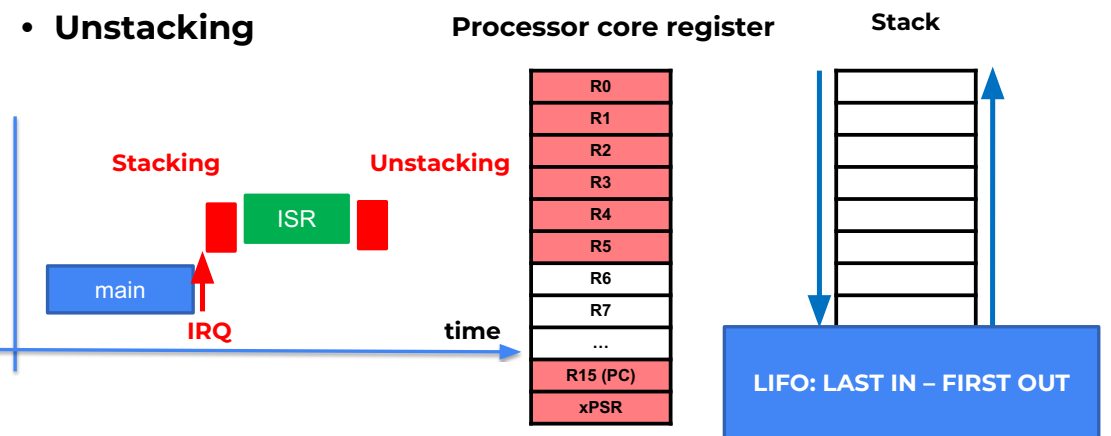
+ Cách thưc hoạt động khi có ngắt:

* Thực hiện nốt lệnh đang thực hiện (lệnh Assembly)
* Dừng chương trình chính
* Thực thi các lệnh trong chương trình ngắt
* Quay trở về thực hiện lệnh tiếp theo trong chương trình

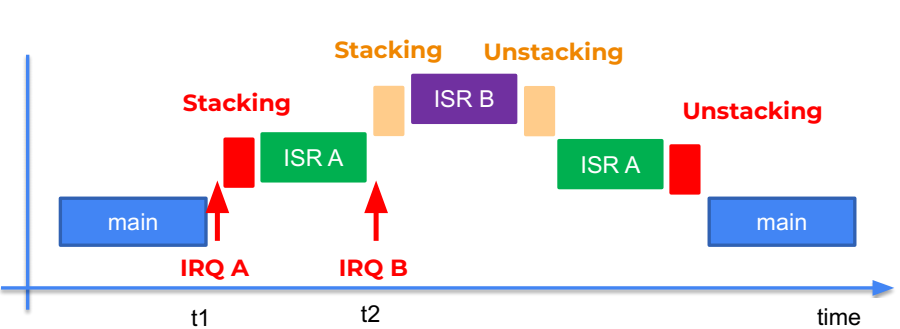




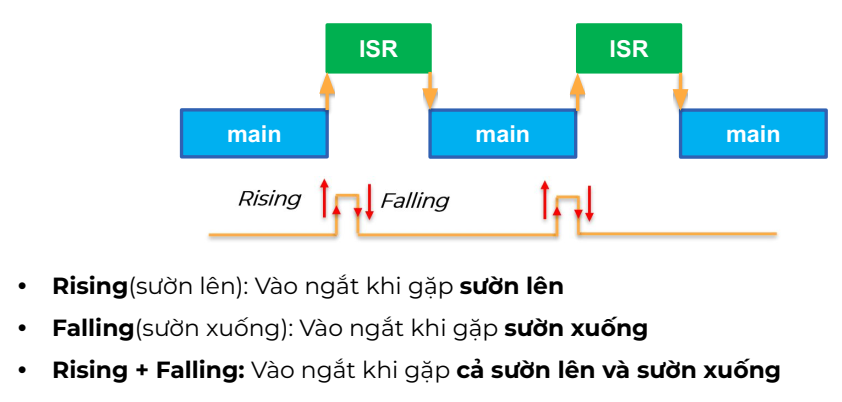


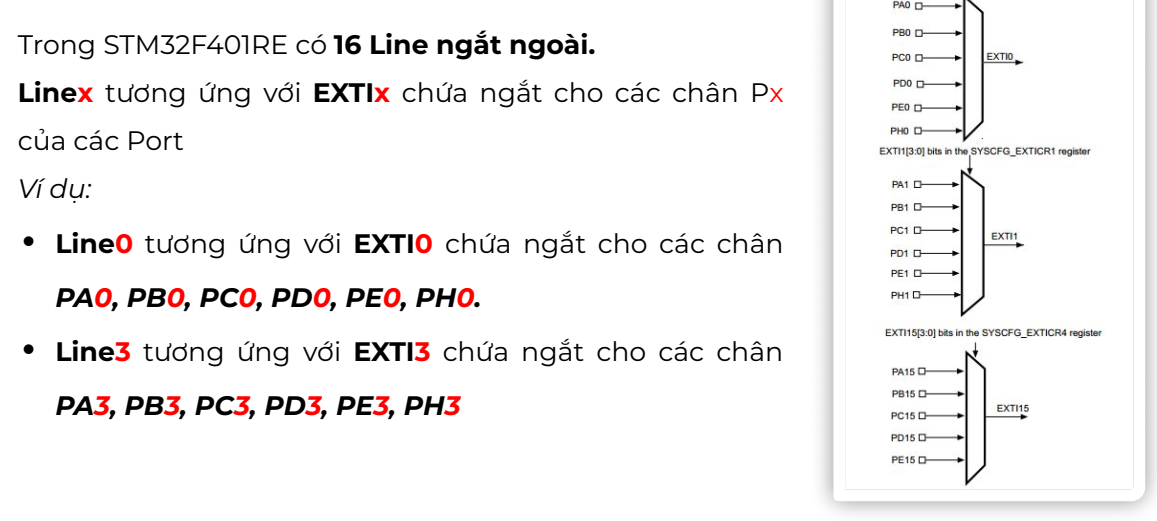


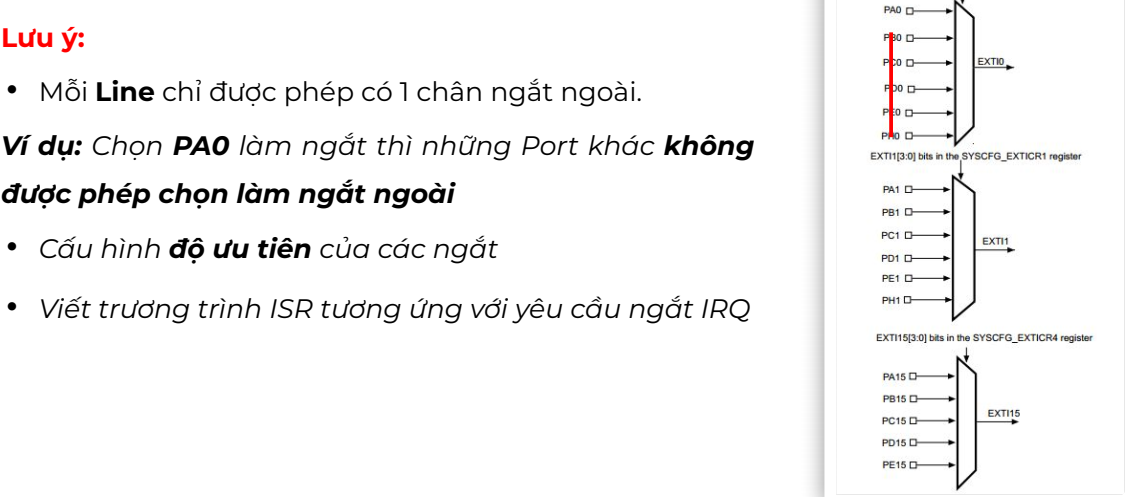
+ **Lưu ý:** Khi có nhiều ngắt thì **mức độ ưu tiên** rất quan trọng quyết định ngắt nào sẽ được phục vụ trước



+ **Ngắt ngoài**: Khi có sự chuyển đổi mức logic trên chân ngắt thì một sự kiện ngắt sẽ sinh ra. Khi đó bộ NVIC sẽ kiểm tra xem ngắt ở Line nào để chạy vào chương trình ngắt tương ứng







# 2. BÀI THÍ NGHIỆM THỰC HÀNH

## 2.1. Mục đích - Yêu cầu

### 2.1.1. Mục đích

* Sinh viên nắm rõ nguyên lý lập trình ngắt cho vi điều khiển STM32
* Sinh viên nắm rõ cách thức xử lí giao diện dòng lệnh vi điều khiển STM32
* Cách lập trình cấu hình ngắt ngoài cho nút nhấn trên Kit mở rộng

### 2.1.2. Yêu cầu

* Yêu cầu sinh viên đã nắm rõ kiến thức lập trình nhúng cơ bản cho vi điều khiển STM32 như tạo mới một chương trình nhúng,dịch & gỡ rối chương trình vi điều khiển STM32, thêm thư viện SDK và sử dụng các hàm API của thư viện.

### 2.1.3. Thời gian thực hiện và chia nhóm

* Thời gian mỗi buổi thực hành là từ 3-4 giờ, chia làm 10-12 nhóm nhỏ, mỗi nhóm 2 sinh viên/1 bộ kit/1 máy tính, 8 bộ kit dự phòng. Sinh viên tìm hiểu cơ sở lý thuyết ngắn gọn và các bước thực hiện có thể thao tác dễ dàng.
* Trước mỗi buổi thực hành, mỗi nhóm sinh viên được cung cấp: 1 phiếu bàn giao thiết bị thực hành bao gồm danh mục các thiết bị thực hành, cuối buổi thực hành sinh viên kiểm tra và nộp lại phiếu bàn giao + thiết bị thực hành

## 2.2. Chuẩn bị

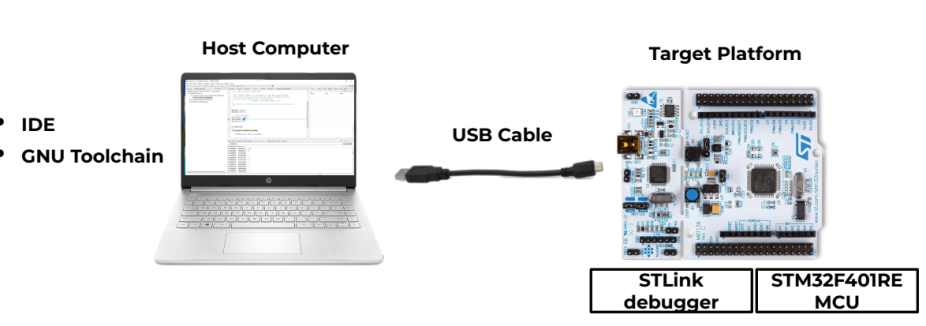
### 2.2.1. Danh mục thiết bị thực hành

* Phần cứng: Các thiết bị chính trong buổi thực hành bao gồm 01 bộ KIT STM32STM32, 01 mạch nạp Kit, 01 máy tính, 01 bộ cáp nối chuyên dụng. Tất cả được được đồng bộ theo số thứ tự từ 1-20
* Phần mềm: TM32CubeIDE, Simplicity Studio

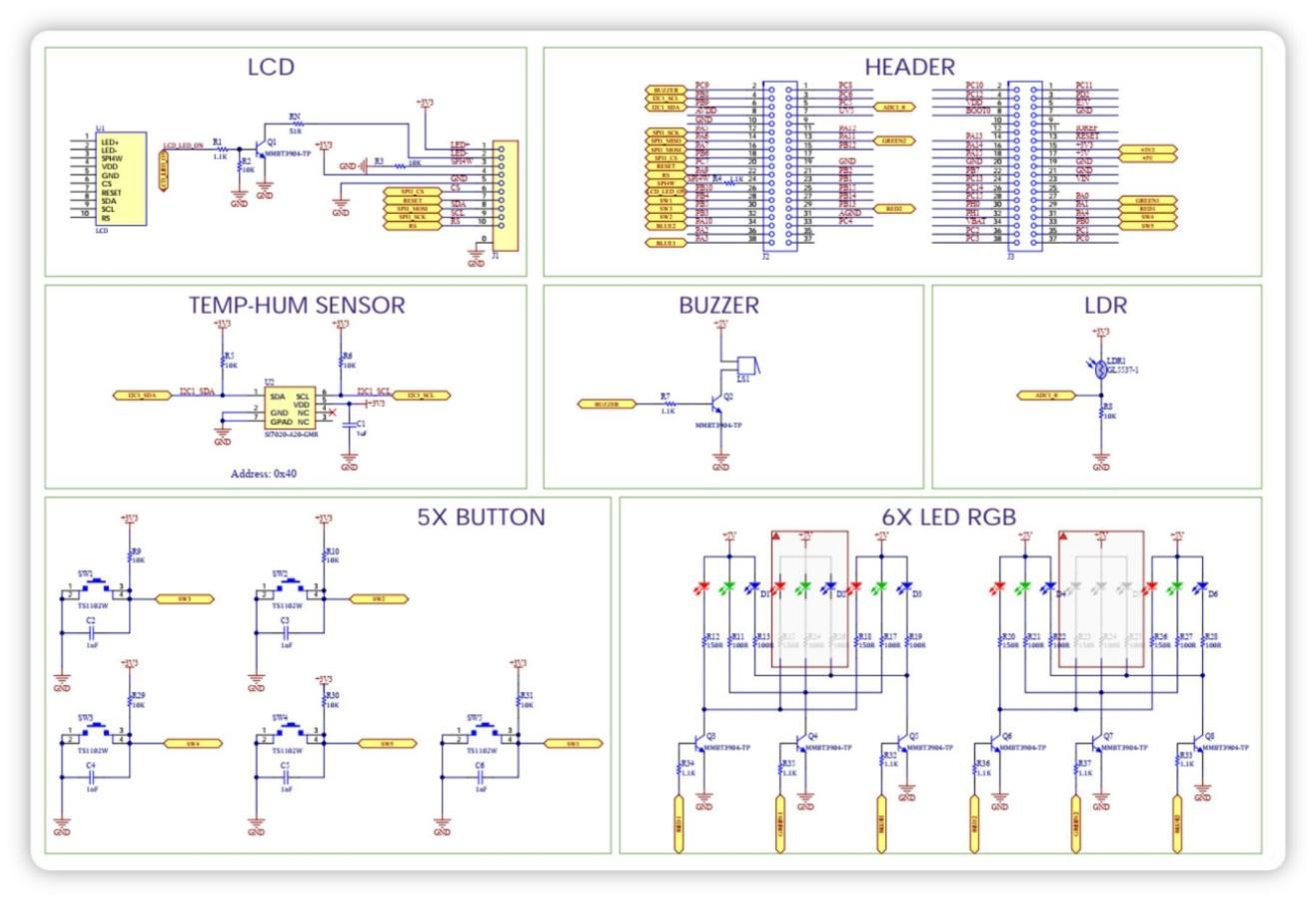
**Danh mục thiết bị thực hành phòng lab**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stt | Tên thiết bị | Số lượng | Thông số  kỹ thuật | Vai trò |
| 1 | Máy tính để bàn | 01 | - Intel Core i3-2100 3.1 Ghz/3M Cache  - 4GB DDR3 (nâng cấp lên ram 8GB thêm 300K)  -SSD 120G Gb chuẩn SATA 3 - 6Gb/s Ổ SSD | -Tính toán, viết chương trình, nạp code vào Kit |
| 2 | KIT STM32STM32 | 01 | -Vi điều khiển STM32  -Màn hình LCD  -Còi (Loa)  -Nút nhấn  -LED RGB  -Cảm biến nhiệt độ, độ ẩm  -Cảm biến ánh sáng  -Ngoại vi & cổng kết nối | -Thực hiện chức năng thu thập, xử lý dữ liệu, điều khiển và kết nối thiết bị  -Tính toán & hiển thị các tham số  - Báo hiệu led |
| 3 | Mạch nạp Kit | 01 |  | -Nạp code vào Kit |
| 4 | Bộ cáp nối chuyên dụng | 01 | -Type C, D, USB | -Đấu nối thiết bị KIT STM32-máy tính  -Đấu nối thiết bị KIT STM32-ngoại vi |

### 2.2.2. Sơ đồ đấu nối trang thiết bị

**

*Hình 4: Sơ đồ đấu nối thiết bị KIT STM32-máy tính*



*Hình 5: Sơ đồ đấu nối chân thiết bị trên KIT STM32*

## 2.3. Nội dung

### 2.3.1. Các bước thực hiện

***Nội dung 1****:* ***Lập trình cấu hình ngắt ngoài cho vi điều khiển STM32***

Viết chương trình cấu hình ngắt ngoài cho nút nhấn trên Board STM32F401  
Nucleo. Khi có ngắt xảy ra trên chân PC13 thì led trên chân PA5 sẽ được bật và sau đó sẽ trở lại trạng thái tắt như ban đầu.

**Bước 1: Tạo một project “Interrupts” như nội dung 1 buổi 1.**

**Bước 2: Thêm các thư viện cần thiết như "stm32f401re\_rcc.h", "stm32f401re\_gpio.h", "misc.h", "stm32f4xx\_exti.h", "stm32f4xx\_syscfg.h" bằng câu lệnh #include như nội dung 2 buổi 1**

**Bước 3: Cấu hình chân PA5 của led trên Board STM32 ở chế độ output**

Khai báo biến thuộc kiểu dữ liệu struct GPIO

+ Cấp clock cho Port A.

+ Chọn chân sử dụng với chức năng điều khiển Led.

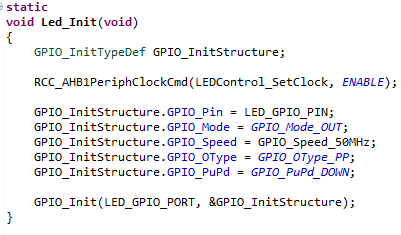
+ Chọn chân điều khiển led ở chế độ Output.

+ Tốc độ xử lý trên chân là 50MHz.

+ Chọn chế độ là đẩy kéo Push Pull.

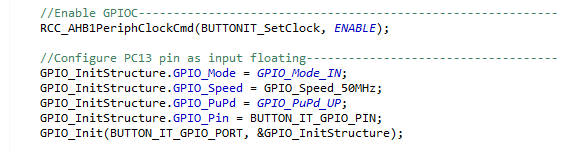
+ Trạng thái ban đầu trên chân là kéo xuống GND Pull Down.

+ Khởi tạo tất cả các giá trị bên trên bằng cách sử dụng hàm **GPIO\_Init** với  
đối số truyền vào là **GPIOA** và **GPIO\_InitStructure**



**Bước 4: Xây dựng hàm Interruppt\_Init cấu hình cho phép ngắt khi nút ấn kết nối với chân PC13 khi được ấn như sau**

**4.1. Khai báo các biến thuộc kiểu dữ liệu cấu trúc của GPIO, EXTI và  
NVIC**



**4.2. Khởi tạo chân GPIO PC13**

+ Cấp clock cho Port C.

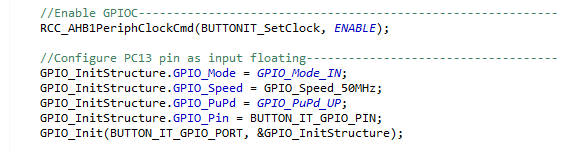
+ Chọn chế độ trên chân PC13 là Input.

+ Chọn tốc tốc trên chân PC13 là 50MHz.

+ Chọn chế độ kéo trở lên dương nguồn Pull Up.

+ Chọn chân để sử dụng là PC13

+ Sử dụng hàm **GPIO\_Init** với đối số là **GPIOC, &GPIO\_InitStructure**



**4.3. Cấu hình sử dụng Line ngắt 13**

+ Cấp clock cho ngoại vi **Syscfg**.

+ Sử dụng hàm **SYSCFG\_EXTILineConfig** với tham số truyền vào là ngắt.  
ở GPIO nào và Line ngắt nào để kết nối Line ngắt 3 với PC13.



**4.4. Cấu hình sử dụng ngắt**

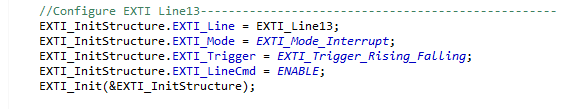
+ Khởi tạo Line13 được sử dụng chức năng ngắt.

+ Chọn chế độ ngắt.

+ Chọn sườn xuống phát hiện sự kiện ngắt Falling.

+ Cho phép ngắt được hoạt động.

+ Sử dụng hàm **EXTI\_Init** để khởi tạo tất cả các giá trị ở trên với đối số là  
**&EXTI\_InitStructure**



**4.5. Cấu hình trình phục vụ ngắt NVIC**.

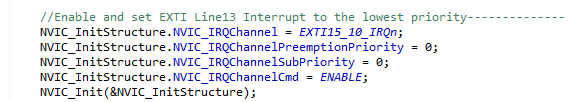
+ Sử dụng trình phục vụ ngắt EXTI15\_10\_IRQn.

+ Cấu hình ưu tiên ngắt PreemptionPriority mức 0.

+ Cấu hình ưu tiên ngắt SubPriority mức 0.

+ Cho phép sử dụng trình phục vụ ngắt.

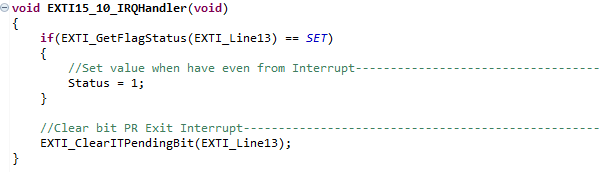
+ Sử dụng hàm NVIC\_Init để cấu hình các giá trị ở trên với đối số là  
&NVIC\_InitStructure



**Bước 5: Khai báo hàm phục vụ ngắt trên chân PC13 như sau**

+ **EXTI\_GetFlagStatus(EXTI\_Line13)** kiểm tra sự kiện ngắt có đúng trên  
chân PC13 hay không.

+ **EXTI\_ClearITPendingBit(EXTI\_Line13)** xóa cờ ngắt sau khi thực hiện  
xong chương trình ngắt.

 **Bước 6: Xây dựng hàm LedControl\_SetStatus**

**Bước 7: Trước hàm while(1), sử dụng các hàm**

+ SystemCoreClockUpdate

+ Led\_Init

+ Interruppt\_Init

**Bước 8: Trong hàm while(1), sử dụng biến Status và hàm LedControl\_SetStatus để thực hiện yêu cầu đề bài**

**Bước 9: Kết quả**



***Nội dung 2:*** ***Thực hành cấu hình ngắt ngoài cho nút nhấn trên Kit mở rộng***

Viết chương trình cấu hình ngắt ngoài cho nút nhấn trên Kit mở rộng. Khi ấn nút B2 trên kit mở rộng thì Led Green trên kit mở rộng sẽ đảo trạng thái sau đó Led sẽ quay về trạng thái ban đầu

**Bước 1: Tạo một project “Interrupts” như nội dung 1 buổi 1.**

**Bước 2: Thêm các thư viện cần thiết như "stm32f401re\_rcc.h",  
"stm32f401re\_gpio.h", "misc.h", "stm32f4xx\_exti.h", "stm32f4xx\_syscfg.h"  
bằng câu lệnh #include như nội dung 2 buổi 1**

**Bước 3: Cấu hình chân PA0 của led trên Kit STM32 ở chế độ output**

+Khai báo biến thuộc kiểu dữ liệu struct GPIO.

+ Cấp xung clock cho PortA.

+ Chọn chân sử dụng với chức năng điều khiển Led.

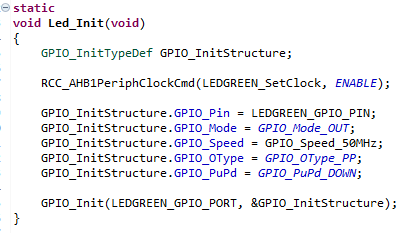
+ Chọn chân điều khiển led ở chế độ Output.

+ Tốc độ xử lý trên chân là 50MHz.

+ Chọn chế độ là đẩy kéo Push Pull.

+ Trạng thái ban đầu trên chân là kéo xuống GND Pull Down.

+ Khởi tạo tất cả các giá trị bên trên bằng cách sử dụng hàm **GPIO\_Init** với  
đối số truyền vào là **GPIOA** và **GPIO\_InitStructure.**



**Bước 4: Xây dựng hàm Interruppt\_Init cấu hình cho phép ngắt khi nút nhấn kết nối với chân B2 trên Kit STM32F401**

**4.1. Khai báo các biến thuộc kiểu dữ liệu cấu trúc của GPIO, EXTI và  
NVIC**



**4.2. Khởi tạo chân GPIO PB3**

+ Cấp clock cho GPIOB.

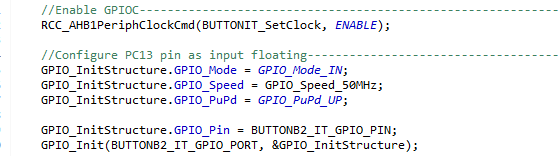
+ Chọn chế độ trên chân PB3 là Input.

+ Chọn tốc tốc trên chân PB3 là 50MHz.

+ Chọn chế độ kéo trở lên dương nguồn Pull Up.

+ Chọn chân để sử dụng là PB3

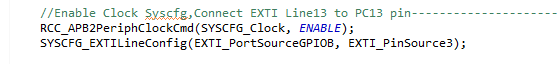
+ Sử dụng hàm GPIO\_Init với đối số là GPIOB, &GPIO\_InitStructure



**4.3. Cấu hình sử dụng Line ngắt 3**

+ Cấp clock cho ngoại vi **Syscfg**.

+ Sử dụng hàm **SYSCFG\_EXTILineConfig** với tham số truyền vào là ngắt.  
ở GPIO nào và Line ngắt nào để kết nối Line ngắt 3 của nút B2



**4.4. Cấu hình sử dụng ngắt**

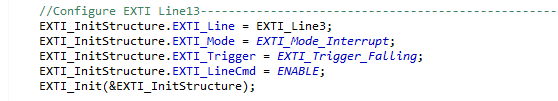
+ Khởi tạo Line3 được sử dụng chức năng ngắt.

+ Chọn chế độ ngắt.

+ Chọn sườn xuống phát hiện sự kiện ngắt Falling.

+ Cho phép ngắt được hoạt động.

+ Sử dụng hàm EXTI\_Init để khởi tạo tất cả các giá trị ở trên với đối số là  
&EXTI\_InitStructure.



**4.5. Cấu hình trình phục vụ ngắt NVIC**

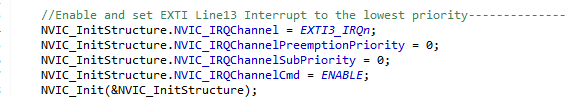
+ Sử dụng trình phục vụ ngắt EXTI3\_IRQn.

+ Cấu hình ưu tiên ngắt PreemptionPriority mức 0.

+ Cấu hình ưu tiên ngắt SubPriority mức 0.

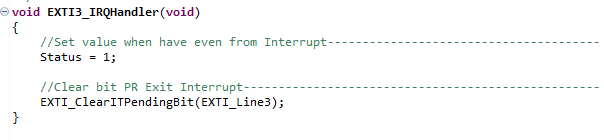
+ Cho phép sử dụng trình phục vụ ngắt.

+ Sử dụng hàm NVIC\_Init để cấu hình các giá trị ở trên với đối số là  
**&NVIC\_InitStructure.**



**Bước 5: Khai báo hàm phục vụ ngắt trên chân PB3 như sau**

**EXTI\_ClearITPendingBit(EXTI\_Line3)** xóa cờ ngắt sau khi thực hiện  
xong chương trình ngắt.



**Bước 6: Xây dựng hàm LedControl\_SetStatus**

**Bước 7: Trước hàm while(1), sử dụng các hàm**

+ SystemCoreClockUpdate

+ Led\_Init

+ Interruppt\_Init

**Bước 8: Trong hàm while(1),sử dụng biến Status và hàm LedControl\_SetStatus  
để thực hiện yêu cầu đề bài**

**Bước 9: Kết quả**



### 2.3.2. Code chương trình đầy đủ

|  |
| --- |
|  |

### 2.3.3. Kết quả đạt được

+ Trước mỗi buổi thực hành mỗi nhóm sinh viên được cung cấp 1 phiếu báo cáo thực hành tóm tắt nội dung và kết quả thực hành.

+ Giáo viên đánh giá Bản báo cáo kết quả thực hành của sinh viên dựa trên bảng tiêu chí đánh giá thực hành. Trong mỗi buổi thực hành, tùy theo năng lực sinh viên có thể thực hiện hết các nội dung hoặc 2/3 nội dung là đạt yêu cầu.

+ Phiếu báo cáo và tiêu chí đánh giá kết quả thực hành xem chi tiết tại Phụ lục

# 3. PHỤ LỤC

**Phiếu báo cáo kết quả thực hành (Sinh viên)**

Tên bài: ……………………………………………………………………………...

Họ và tên sinh viên……………………………Mã sinh viên……………………...

………………………………………………………………..……………………...

Nhóm…………………..Lớp……………… ..Ngày…..tháng…..năm…………..

Giáo viên hướng dẫn…………………………Ca thực tập……………………….

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stt | Nội dung  thực hành | Mức độ  hoàn thành (%) | Thời gian  hoàn thành | Đánh giá kết quả (100) |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| Thảo luận sinh viên: | | | | |

**Phiếu đánh giá kết quả thực hành (Giảng viên)**

Tên bài: ……………………………………………………………………………...

Họ và tên sinh viên……………………………Mã sinh viên……………………...

Nhóm…………………..Lớp…………………Ngày…..tháng…..năm…………..

Giáo viên hướng dẫn………………………….Ca thực tập……………………….

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thứ tự** | **Nội dung đánh giá** | **Điểm chuẩn** | **Yêu cầu** | **Ghi chú** |
| 1 | - Tạo 1 chương trình nhúngtrên TM32CubeIDE | 10 | -Thành thạo |  |
| 2 | - Lập trình cấu hình ngắt ngoài cho vi điều khiển STM32 | 20 | -Thành thạo |  |
| 3 | - Lập trình cấu hình ngắt ngoài cho nút nhấn trên Kit mở rộng | 20 | -Thành thạo |  |
| 4 | -Thiết lập thông số | 10 | -Thành thạo  -Đúng trình tự nguyên tắc |  |
| 5 | -Thay đổi thông số | 10 | - Đúng trình tự nguyên tắc |  |
| 6 | -Đọc và phân tích kết quả | 10 | - Áp dụng các công thức  - Tính toán đúng kết quả |  |
| 7 | -Viết báo cáo | 20 | - Gọn gàng khoa học |  |
| Tổng điểm: | | | | |
| Nhận xét giảng viên: | | | | |