

LẬP TRÌNH ARM - STM32

STM32 - EXTI

Date: Mar 09, 2025



Polling & Interrupt

Main Program:

Repeat the following forever

{

if UART received data

Get the data and process it

if timer elapsed

Do the task

}

```
void uart_SendChar(char _chr) {
    USART_SendData(USART1, _chr);
    while(USART_GetFlagStatus(USART1, USART_FLAG_TXE)==RESET);
}
```

```
void USART1_IRQHandler(){
    if(USART_GetITStatus(USART1, USART_IT_RXNE)!=RESET){
        // Process received data
    }
}
```

Main Program:

Do a task

On UART receive interrupt:

Get the data and process it

On timer interrupt:

Do the task

Preemption Priority

Cách phân chia		PreemptionPriority		SubPriority	
Tỉ lệ số bit Preem: Sub	Lệnh trong lập trình NVIC_PriorityGroupConfig(A);	Số cấp ưu tiên	Phạm vi	Số cấp ưu tiên	Phạm vi
0 bit : 4 bit	A=NVIC_PriorityGroup_0	0	0	16	0-15
1 bit : 3 bit	A=NVIC_PriorityGroup_1	2	0-1	8	0-7
2 bit : 2 bit	A=NVIC_PriorityGroup_2	4	0-3	4	0-3
3 bit : 1 bit	A=NVIC_PriorityGroup_3	8	0-7	2	0-1
4 bit : 0 bit	A=NVIC_PriorityGroup_4	16	0-15	0	0

Bảng phân chia số cấp ưu tiên giữa 2 nhóm PreemptionPriority và SubPriority

Chú ý: Nhóm **PreemptionPriority** có mức độ ưu tiên cao hơn **SubPriority** và cấp độ ưu tiên càng nhỏ thì ưu tiên càng cao. Có nghĩa là cấp 0 của nhóm **PreemptionPriority** sẽ có mức độ ưu tiên cao nhất.

Preemption Priority

Interrupt	Preemption Priority	Subpriority
EXTI1	1 (cao)	0
EXTI2	2 (trung bình)	0
EXTI3	3 (thấp)	0

EXTI1 > EXTI2 > EXTI3

Kịch bản:

- Nếu EXTI3 đang chạy, mà EXTI1 xảy ra, thì EXTI1 sẽ ngắt EXTI3 ngay lập tức vì EXTI1 có Preemption Priority cao hơn.
- Nếu EXTI1 và EXTI2 cùng xảy ra, thì EXTI1 chạy trước vì có Preemption Priority cao hơn.
- Nếu EXTI2 và EXTI3 cùng xảy ra, thì EXTI2 chạy trước.

STM32F1 - NVIC Library

NVIC_SetVectorTable (A, B) ;
(Lệnh chọn nơi lưu bảng vector ngắn)

A:

NVIC_VectTab_RAM
NVIC_VectTab_FLASH

A: Nơi lưu

Lưu trong RAM

Lưu trong FLASH

B: Địa chỉ đầu vùng nhớ cần lưu. Địa chỉ này phải là bội số của 0x200

NVIC_PriorityGroupConfig (A) ;

(Lệnh chọn cách phân chia số cấp ưu tiên của 2 nhóm PreemptionPriority và SubPriority)

A:

NVIC_PriorityGroup_0
NVIC_PriorityGroup_1
NVIC_PriorityGroup_2
NVIC_PriorityGroup_3
NVIC_PriorityGroup_4

A: Nhóm ưu tiên được chọn

Nhóm 0: 0 Pre – 16 Sub
Nhóm 1: 2 Pre – 8 Sub
Nhóm 2: 4 Pre – 4 Sub
Nhóm 3: 8 Pre – 2 Sub
Nhóm 4: 16 Pre – 0 Sub

NVIC_InitTypeDef A;

A.NVIC_IRQChannel = B;

(Lệnh chọn ngoại vi cần cho phép ngắn)

B:

USART1_IRQHandler
SPI1_IRQHandler
EXTI4_IRQHandler
TIM2_IRQHandler

B: Ngoại vi cần ngắn

Ngắt USART1
Ngắt SPI1
Ngắt ngoài kênh số 4
Ngắt Timer2

STM32F1 - NVIC Library

A. `NVIC IRQChannelCmd = E;`
(Lệnh cho phép hoặc cấm ngắt ngoại vi đang
được cấu hình tại NVIC)

E:
ENABLE
DISABLE

E: Cho phép hoặc cấm
Cho phép ngắt
Cấm ngắt

`NVIC_Init(&A);`
(Lệnh cấu hình các thông số được lưu trong
biến A cho NVIC)

A. `NVIC IRQChannelPreemptionPriority = C;`
(Lệnh chọn thứ tự ưu tiên ngắt của ngoại vi đang cấu hình
trong nhóm chính)

C:
0
...
15

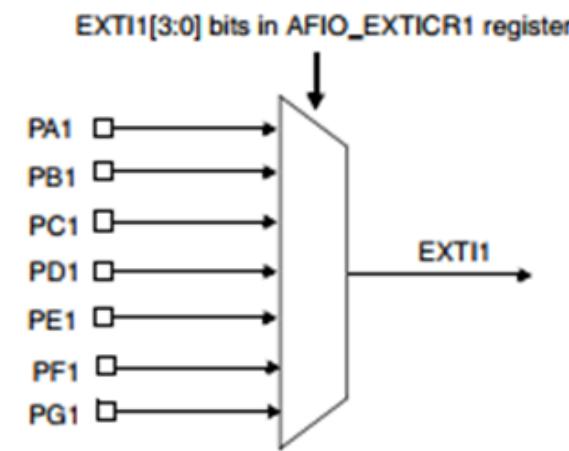
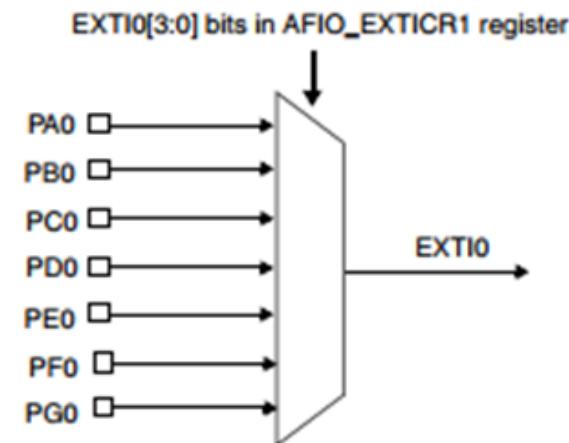
C: thứ tự ưu tiên
0: Ưu tiên cao nhất
...
15: Ưu tiên thấp nhất

A. `NVIC IRQChannelSubPriority = D;`
(Lệnh chọn thứ tự ưu tiên ngắt của ngoại vi đang cấu hình
trong nhóm phụ)

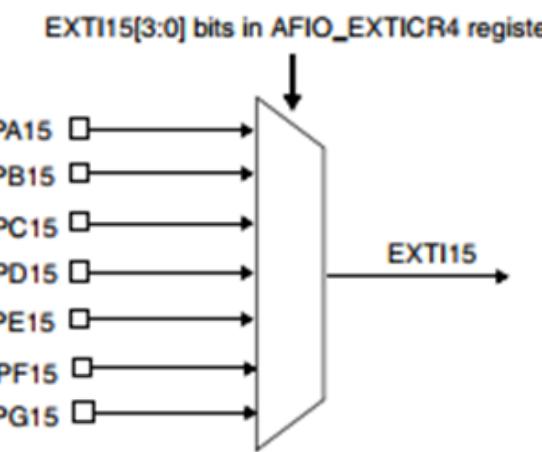
D:
0
...
15

D: thứ tự ưu tiên
0: Ưu tiên cao nhất
...
15: Ưu tiên thấp nhất

STM32F1 - EXTI Library



⋮



STM32F1 - EXTI Library

`EXTI_InitTypeDef A;`

`GPIO_EXTILineConfig(B, C);`

(Lệnh chọn chân nhận tín hiệu ngắt ngoài/sự kiện)

B:

`GPIO_PortSourceGPIOA`

...

`GPIO_PortSourceGPIOG`

C:

`GPIO_PinSource0`

...

`GPIO_PinSource15`

B: Chọn PORT

Chọn PORTA

...

Chọn PORTG

C: Chọn chân

Chọn chân 0

...

Chọn chân 15

`A.EXTI_Line = D;`

(Lệnh chọn kênh ngắt ngoài/sự kiện)

D:

`EXTI_Line0`

...

`EXTI_Line18`

D:

`EXTI_Line0`

...

`EXTI_Line18`

`A.EXTI_Mode = E ;`

(Lệnh chọn chế độ ngắt hoặc sự kiện)

E:

`EXTI_Mode_Interrupt`

`EXTI_Mode_Event`

E:

`EXTI_Mode_Interrupt`

`EXTI_Mode_Event`

`A.EXTI_Trigger = F;`

(Lệnh chọn cạnh tích cực)

F:

`EXTI_Trigger_Rising`

`EXTI_Trigger_Falling`

`EXTI_Trigger_Rising_Falling`

F:

`EXTI_Trigger_Rising`

`EXTI_Trigger_Falling`

`EXTI_Trigger_Rising_Falling`

`EXTI_Init(&A);`

(Lệnh cấu hình các thông số được lưu trong
biến A cho EXTI)

STM32F1 - EXTI Library

`EXTI_InitTypeDef A;`

`GPIO_EXTILineConfig(B, C);`

(Lệnh chọn chân nhận tín hiệu ngắt ngoài/sự kiện)

B:

`GPIO_PortSourceGPIOA`

...

`GPIO_PortSourceGPIOG`

C:

`GPIO_PinSource0`

...

`GPIO_PinSource15`

B: Chọn PORT

Chọn PORTA

...

Chọn PORTG

C: Chọn chân

Chọn chân 0

...

Chọn chân 15

`A.EXTI_Line = D;`

(Lệnh chọn kênh ngắt ngoài/sự kiện)

D:

`EXTI_Line0`

...

`EXTI_Line18`

D:

`EXTI_Line0`

...

`EXTI_Line18`

`A.EXTI_Mode = E ;`

(Lệnh chọn chế độ ngắt hoặc sự kiện)

E:

`EXTI_Mode_Interrupt`

`EXTI_Mode_Event`

E:

`EXTI_Mode_Interrupt`

`EXTI_Mode_Event`

`A.EXTI_Trigger = F;`

(Lệnh chọn cạnh tích cực)

F:

`EXTI_Trigger_Rising`

`EXTI_Trigger_Falling`

`EXTI_Trigger_Rising_Falling`

F:

`EXTI_Trigger_Rising`

`EXTI_Trigger_Falling`

`EXTI_Trigger_Rising_Falling`