

DCD	SPI1_IRQHandler	;SPI1 single interrupt vector for all sources
DCD	UART0_IRQHandler	;UART0 status and error
DCD	UART1_IRQHandler	;UART1 status and error
DCD	UART2_IRQHandler	;UART2 status and error
DCD	ADC0_IRQHandler	;ADC0 interrupt
DCD	CMF0_IRQHandler	;CMF0 interrupt
DCD	TPM0_IRQHandler	;TPM0 single interrupt vector for all sources
DCD	TPM1_IRQHandler	;TPM1 single interrupt vector for all sources
DCD	TPM2_IRQHandler	;TPM2 single interrupt vector for all sources
DCD	RTC_IRQHandler	;RTC alarm interrupt
DCD	RTC_Seconds_IRQHandler	;RTC seconds interrupt
DCD	PIT_IRQHandler	;PIT single interrupt vector for all channels
DCD	I2S0_IRQHandler	;I2S0 Single interrupt vector for all sources
DCD	USB0_IRQHandler	;USB0 OTG
DCD	DAC0_IRQHandler	;DAC0 interrupt
DCD	TSIO_IRQHandler	;TSIO interrupt
DCD	MCG_IRQHandler	;MCG interrupt
DCD	LPTMR0_IRQHandler	;LPTMR0 interrupt
DCD	LCD_IRQHandler	;Segment LCD interrupt
DCD	PORTA_IRQHandler NB	;PORTA pin detect
DCD	PORTC_PORTD_IRQHandler	;Single interrupt vector for PORTC and PORTD pin detect
Vectors_End		

Nếu bạn thay set interrupt mà không cài interrupt handler cho nó thì giá trị của bảng vector ban đầu trong chương trình sẽ có những giá trị linh tinh, dựa vào những giá trị ở những vùng không phải chip và dữ liệu treo, những giá trị này ta không muốn vì có thể ảnh hưởng đến chương trình vector mà chúng ta cần, thì khi nhúng vào nó sẽ giúp DMA chạy bình thường.

```

PUBWEAK DMA0_IRQHandler
PUBWEAK DMA0_DriverIRQHandler
SECTION .text:CODE:REORDER:NOROOT(2)
DMA0_IRQHandler
    LDR    R0, =DMA0_DriverIRQHandler
    BX     R0

PUBWEAK DMA1_IRQHandler
PUBWEAK DMA1_DriverIRQHandler
SECTION .text:CODE:REORDER:NOROOT(2)
DMA1_IRQHandler
    LDR    R0, =DMA1_DriverIRQHandler
    BX     R0

PUBWEAK DMA2_IRQHandler
PUBWEAK DMA2_DriverIRQHandler

```

chúng ta cần cài handler

trong đó,

DCD	SPI0_IRQHandler	;SPI0 single interrupt vector for all sources
DCD	SPI1_IRQHandler	;SPI1 single interrupt vector for all sources
DCD	UART0_IRQHandler	;UART0 status and error
DCD	UART1_IRQHandler	;UART1 status and error
DCD	UART2_IRQHandler	;UART2 status and error
DCD	ADC0_IRQHandler	;ADC0 interrupt
DCD	CMF0_IRQHandler	;CMF0 interrupt
DCD	TPM0_IRQHandler	;TPM0 single interrupt vector for all sources
DCD	TPM1_IRQHandler	;TPM1 single interrupt vector for all sources
DCD	TPM2_IRQHandler	;TPM2 single interrupt vector for all sources
DCD	RTC_IRQHandler NB	;RTC alarm interrupt
DCD	RTC_Seconds_IRQHandler	;RTC seconds interrupt
DCD	PIT_IRQHandler	;PIT single interrupt vector for all channels
DCD	I2S0_IRQHandler	;I2S0 Single interrupt vector for all sources
DCD	USB0_IRQHandler	;USB0 OTG
DCD	DAC0_IRQHandler	;DAC0 interrupt
DCD	TSIO_IRQHandler	;TSIO interrupt
DCD	MCG_IRQHandler	;MCG interrupt
DCD	LPTMR0_IRQHandler	;LPTMR0 interrupt
DCD	LCD_IRQHandler	;Segment LCD interrupt
DCD	PORTA_IRQHandler	;PORTA pin detect
DCD	PORTC_PORTD_IRQHandler	;Single interrupt vector for PORTC and PORTD pin detect
Vectors_End		

Trên đây là chúng ta cần cài handler

```

THUMB

PUBWEAK Reset_Handler
SECTION .text:CODE:REORDER:NOROOT(2)
Reset_Handler
CPSIE I ; Mask interrupts
LDR R0, =0xE000ED08
LDR R1, =__vector_table
STR R1, [R0]
LDR R0, =SystemInit
BLX R0
CPSIE I ; Unmask interrupts
LDR R0, =__iar_program_start
BX R0

PUBWEAK NMI_Handler
SECTION .text:CODE:REORDER:NOROOT(1)
NMI_Handler
B .

PUBWEAK HardFault_Handler
SECTION .text:CODE:REORDER:NOROOT(1)

```

đây là handler, nó là function

```

PUBWEAK SysTick_Handler
SECTION .text:CODE:REORDER:NOROOT(1)
SysTick_Handler
B .

PUBWEAK DMA0_IRQHandler
PUBWEAK DMA0_DriverIRQHandler
SECTION .text:CODE:REORDER:NOROOT(2)
DMA0_IRQHandler
LDR R0, =DMA0_DriverIRQHandler
BX R0

PUBWEAK DMA1_IRQHandler
PUBWEAK DMA1_DriverIRQHandler
SECTION .text:CODE:REORDER:NOROOT(2)
DMA1_IRQHandler
LDR R0, =DMA1_DriverIRQHandler
BX R0

```

Ví dụ systick chỉ cần bật, thì enable interrupt mà không cài vector table interrupt cho nó chừa? Thì nó sẽ bị treo trong handler này (bôi xanh)

Câu lệnh B . thực hiện vòng lặp vô hạn vì câu lệnh while (1), đó là lý do đèn sáng suốt vì không có gì ngắt khi không cần GPIO nữa, hoặc là tắt đèn.

Vậy làm sao để ngắt khi cần dùng này

```

__FlashConfig
DCD 0xFFFFFFFF
DCD 0xFFFFFFFF
DCD 0xFFFFFFFF
DCD 0xFFFFFFFF
__FlashConfig_End

__Vectors      EQU    __vector_table
__Vectors_Size EQU    __Vectors_End - __Vectors

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; Default interrupt handlers.
;;

        THUMB

        PUBWEAK Reset_Handler
        SECTION .text:CODE:REORDER:NOROOT(2)
Reset_Handler
        CPSID   I          ; Mask interrupts
        LDR     R0, =0xE000ED08
        LDR     R1, =__vector_table
        STR     R1, [R0]
        LDR     R0, =SystemInit
        BLX     R0
        CPSIE   I          ; Unmask interrupts

```

Khi b t vector interrupt lên thì hàm ph c v ng t c a mình ây, cho nên b ng

```

        I
DCD DMA0_IRQHandler ;DMA channel 0 transfer complete and error interrupt
DCD DMA1_IRQHandler ;DMA channel 1 transfer complete and error interrupt
DCD DMA2_IRQHandler ;DMA channel 2 transfer complete and error interrupt
DCD DMA3_IRQHandler ;DMA channel 3 transfer complete and error interrupt
DCD Reserved20_IRQHandler ;Reserved interrupt

```

Này t ng ng v i th ng systick

```

DCD SysTick_Handler ;SysTick Handler

```

Thì gi s hàm c a mình ko ph i systick handler mà là systick isr ch ng h n thì c n ph i i n tên c a hàm ó vào ây

Công vi c này có th do compiler làm

C th là systick handler m c nh c a file này, còn khi nh ngh a hàm c a mình, systick iar thì ko c

Ph i l p trình c b ng interrupt này nó ghi vào trong ph n u c a b nh Flash, khi g p interrupt thì nó s tham chi u t i interrupt ID bao nhiêu thì nó s t ng ng v i a ch ây

```

DCD PendSV_Handler ;PendSV Handler
DCD SysTick_isr ;SysTick Handler

```

ó vào ây ghi a ch c a hàm

a ch ghi d i mã assembly mà thôi, vì t b ng ngôn ng C thì vì t ki u khác

```
DCD SysTick_isr;SysTick_Handler
```

Giả sử hàm của mình là systick_isr này thì cách của nó sẽ bày bưng cách ghi tên hàm của nó này. Thì đó là 1 cách.

Cách thứ 2 là trong hàm handler này gọi hàm của mình

```

    .
    .
    .
    PUBWEAK SysTick_Handler
    SECTION .text:CODE:REORDER:NOROOT(1)
SysTick_Handler
    B .

    NB
    PUBWEAK DMA0_IRQHandler
    PUBWEAK DMA0_DriverIRQHandler
    SECTION .text:CODE:REORDER:NOROOT(2)
DMA0_IRQHandler
    LDR    R0, =DMA0_DriverIRQHandler
    BX     R0

```

Thì nó sẽ bày bưng cách sử dụng hàm này bằng systick_isr

```

    SECTION .text:CODE:REORDER:NOROOT(1)
SysTick_Handler
    B sys_tick_isr

    NB
    PUBWEAK DMA0_IRQHandler
    PUBWEAK DMA0_DriverIRQHandler
    SECTION .text:CODE:REORDER:NOROOT(2)
DMA0_IRQHandler
    LDR    R0, =DMA0_DriverIRQHandler
    BX     R0

```

Này là câu lệnh này nên nó sẽ nhảy vào symbol như thế này, tuy nhiên nếu systick handler xa vị trí systick handler quá thì không được, lúc đó sẽ thành BX

```
BX sys_tick_isr
```

Cách thứ 3, cách mà người ta hay sử dụng là người ta viết thành 1 hàm trong file C tên gì tùy họ đặt cái này

```
DCD SysTick_Handler;SysTick_Handler
```

Chuyển gì xảy ra, vì thế thì có thể conflict vì thế nên thì người ta đã không

```

SysTick_Handler
    B .

    PUBWEAK DMA0_IRQHandler
    PUBWEAK DMA0_DriverIRQHandler
    SECTION .text:CODE:REORDER:NOROOT(2)
DMA0_IRQHandler
    LDR    R0, =DMA0_DriverIRQHandler
    BX     R0
    NB

```

Mình có thể xóa, nhưng mà mình không muốn xóa.

Câu trả lời khác: có thể có 2 symbol giống nhau sẽ có conflict, nó nhắc lại 2 biến global giống nhau conflict. Lúc link với nhau nó sẽ phát hiện ra các symbol trùng lặp, lỗi ngay quá trình link

Vậy nên có 1 từ khóa để biến

PUBWEAK

Từ khóa này cho phép ghi đè, tức là có thể người nào define giống nó thì nó sẽ ghi đè, không phải là ghi đè mà symbol này sẽ biến mất

```
SysTick_Handler 1
B .
NB
```

Nó overwrite luôn

Người dùng khi sử dụng nên thì người ta không muốn sử dụng file asm này vì khó hiểu cho nên người ta chỉ định hàm handler giống như vậy

Chỉ có 2 cách ưu tiên, có 1 vấn đề là chỉ có 1 function này

```
DCD SysTick_Handler ;SysTick_Handler
```

Thì nó đang không hiển thị ra symbol này là gì

```
DCD SysTick_isr ;SysTick_Handler
```

Nó sẽ báo lỗi là unresolved symbol tức là vì mình chưa declaration của nó

File.s này đang giống file.c là nó cần hiểu như symbol này nghĩa là gì

```
_vector_table
DCD sfe(CSTACK)
DCD Reset_Handler

DCD NMI_Handler ;NMI_Handler
DCD HardFault_Handler ;Hard Fault Handler
DCD 0 ;Reserved
DCD 0 ;Reserved
DCD 0 ;Reserved

_vector_table_0x1c
DCD 0 ;Reserved
DCD 0 ;Reserved
DCD 0 ;Reserved
DCD 0 ;Reserved
DCD SVC_Handler ;SVCall_Handler
DCD 0 ;Reserved
DCD 0 ;Reserved
DCD PendSV_Handler ;PendSV_Handler
DCD SysTick_isr ;SysTick_Handler
NR
```

Cho nên cần phải extern nó (symbol ngoài) vào

```

MODULE ?cstartup

;; Forward declaration of sections.
SECTION CSTACK:DATA:NOROOT(3)

SECTION .intvec:CODE:NOROOT(2)

EXTERN __iar_program_start
EXTERN SystemInit
PUBLIC __vector_table
PUBLIC __vector_table_0: NB
PUBLIC __Vectors
PUBLIC __Vectors_End
PUBLIC __Vectors_Size

```

Câu hỏi: khi xảy ra ngắt thì nó chuyển vào bảng vector ngắt này nó tìm cái handler của mình sau đó thì handler của mình phải ko?

Có thể có và có thể không, trường hợp không là nó sẽ có 1 phép init riêng nó thực hiện ngay lập tức của chip luôn mà không yêu cầu bảng vector ngắt này, tu vào 1 số dòng vì x lý có của con tr PC của bảng vector ngắt này sau đó thì yêu cầu bảng handler này của mình, tuy nhiên các dòng nó có như phép get init thì nó sẽ get như ngắt này trong lúc nó đang thực hiện các thủ tục hàm main thì nó sẽ khác nó cách ngắt này, tức là ngay lập tức nó của con tr PC sang thủ tục kia luôn chứ không yêu cầu này. Tu vào thì t k

Ví dụ ngắt của systick là nó có ID, ví dụ systick của n i v i đây thì 5 trong con chip chúng ta

DCD	0	;Reserved
DCD	PendSV_Handler	;PendSV Handler
DCD	SysTick_Handler	;SysTick Handler
	NB	
		;External Interrupts
DCD	DMA0_IRQHandler	;DMA channel 0 transfer complete and error interrupt
DCD	DMA1_IRQHandler	;DMA channel 1 transfer complete and error interrupt
DCD	DMA2_IRQHandler	;DMA channel 2 transfer complete and error interrupt
DCD	DMA3_IRQHandler	;DMA channel 3 transfer complete and error interrupt
DCD	Reserved20_IRQHandler	;Reserved interrupt
DCD	FTFA_IRQHandler	;FTFA command complete and read collision
DCD	LVD_LVW_IRQHandler	;Low-voltage detect, low-voltage warning
DCD	LLWU_IRQHandler	;Low Leakage Wakeup
DCD	I2C0_IRQHandler	;I2C0 interrupt
DCD	I2C1_IRQHandler	;I2C1 interrupt
DCD	SPI0_IRQHandler	;SPI0 single interrupt vector for all sources
DCD	SPI1_IRQHandler	;SPI1 single interrupt vector for all sources
DCD	UART0_IRQHandler	;UART0 status and error
DCD	UART1_IRQHandler	;UART1 status and error
DCD	UART2_IRQHandler	;UART2 status and error
DCD	ADC0_IRQHandler	;ADC0 interrupt
DCD	CMP0_IRQHandler	;CMP0 interrupt

thì mình có thể coi nó là dây thứ m i m y này từ 1 hay 10 đến 15 chúng ta

Tín hiệu của nó gửi vào dây thứ 15 thì nó sẽ báo bus nào phải fetch giá trị thứ 15 này chúng ta (tổng số vị trí dòng) và nó sẽ của con tr PC từ ngắt này (vector thứ 15 này)

Vấn đề của nó là chúng ta

ng v i m i interrupt thì nó ch có m i 1 vector ng t, 1 cái interrupt handler duy nh t,
mu n thay i cái ó thì ph i program b ng interrupt này nh ng mà b ng interrupt này
hi n t i ang n m trên flash, mu n thay i b ng interrupt thì ph i a nó lên RAM