CLASE 5: NVIC

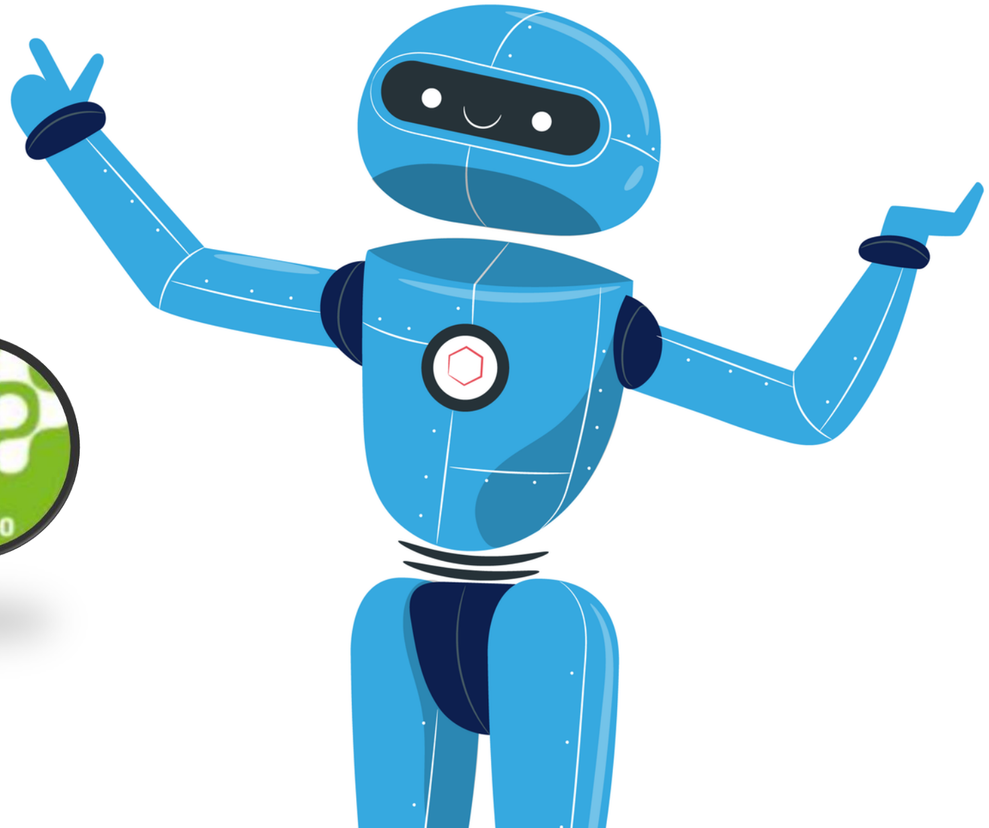
MICROCONTROLADORES

ARM

life.augmented

STM32 L4

STM32 Nucleo



TEXAS INSTRUMENTS

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

DESCRIPCION DEL NVIC

arm

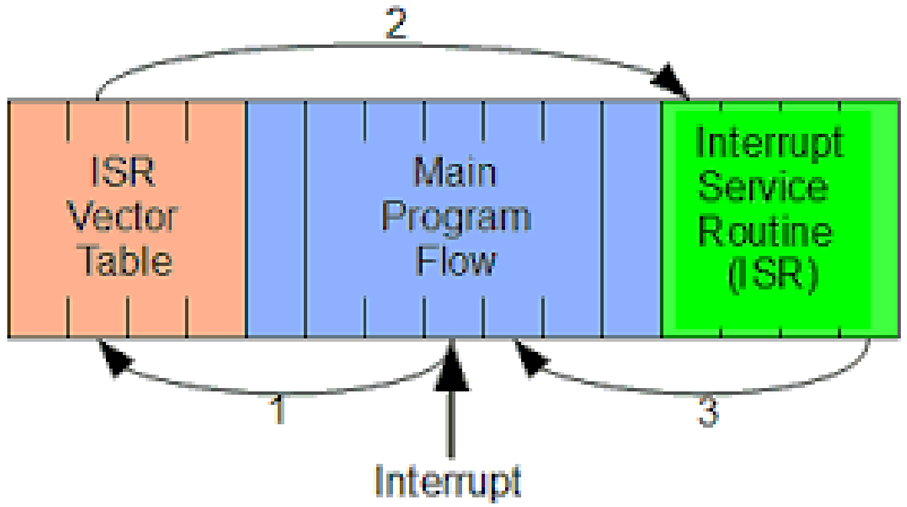
MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

INTERRUPCIONES

Interrumpe el flujo normal del programa principal.



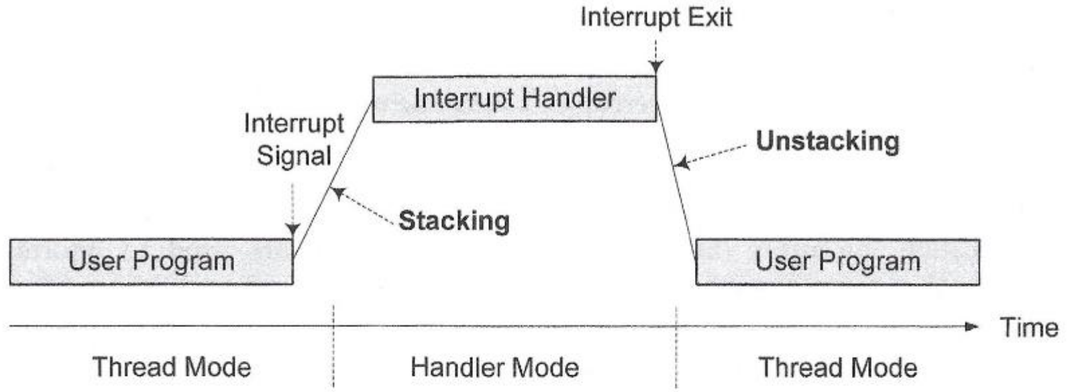
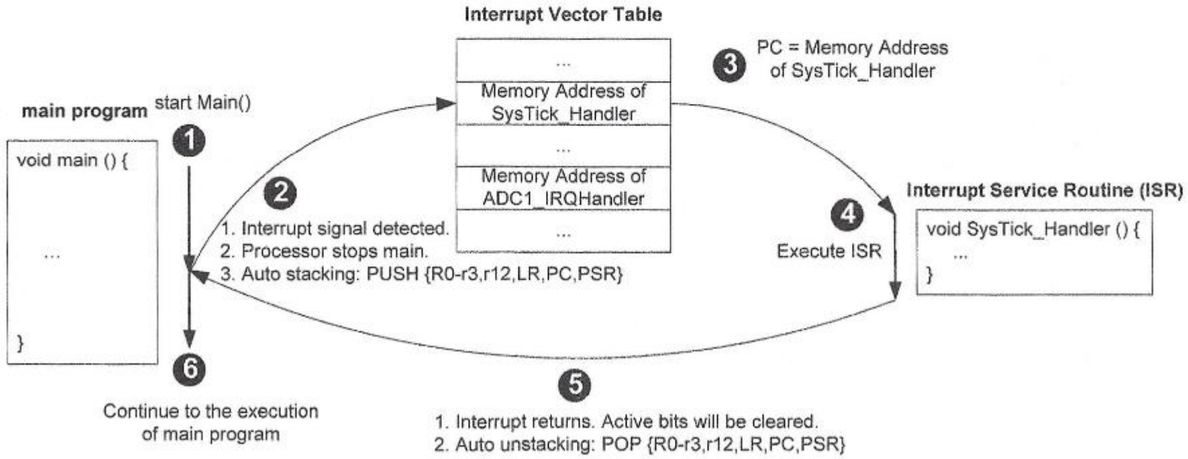


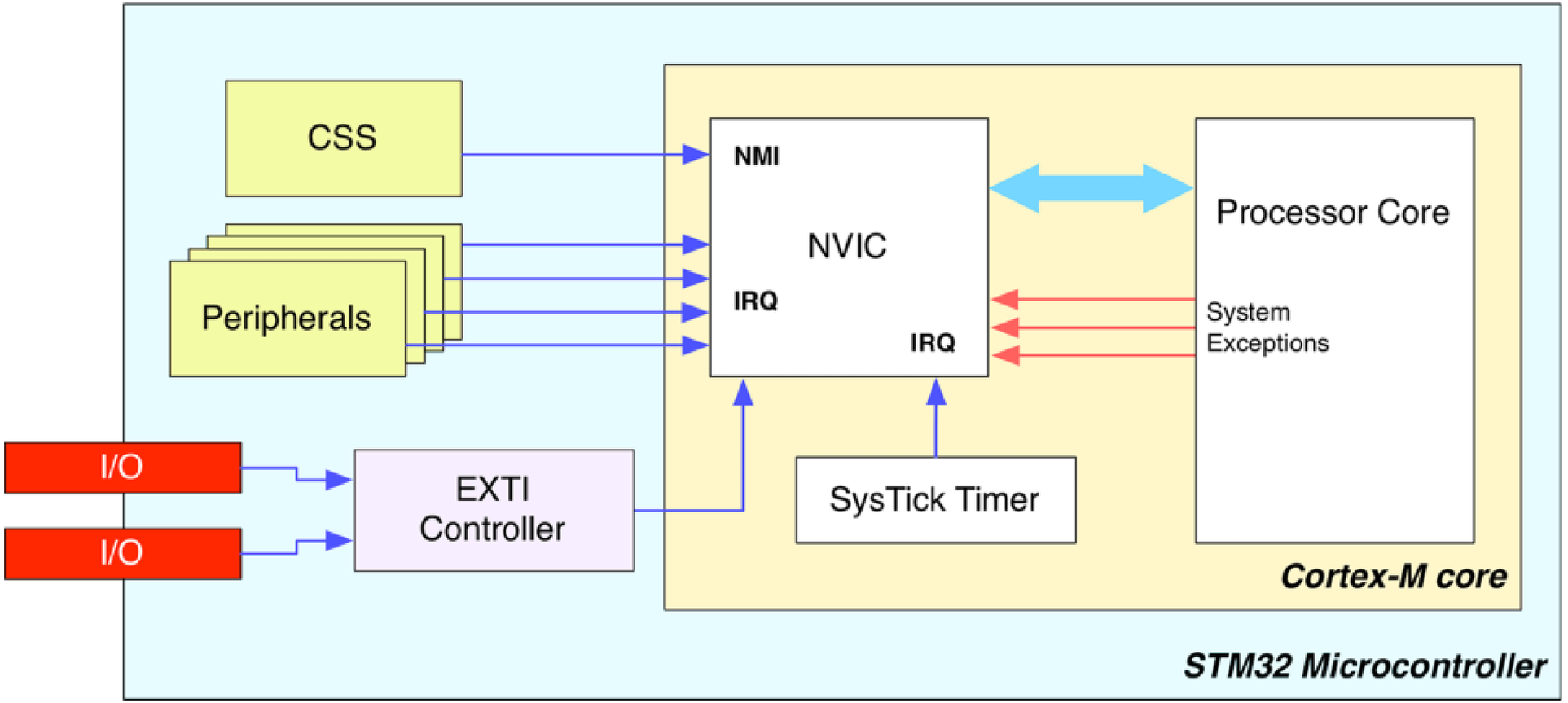
Figure 11-3. Automatic stacking and unstacking for interrupt handler



arm
MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M



INTERRUPCIONES CORTEX-M

INTERRUPCIONES

- **NVIC**

La NVIC es un periférico interno del CORE ARM CORTEX-M y es el encargado de manejar todas las interrupciones. Realiza estas tres funciones:

1. Habilita y deshabilita las interrupciones.
2. Configura la prioridad y sub-prioridad de una determinada interrupción.
3. Establece y limpia el handling bit de una interrupción.

Interrupt control bit	Corresponding register (32 bits)
Enable bit	Interrupt set enable register (ISER)
Disable bit	Interrupt clear enable register (ICER)
Pending bit	Interrupt set pending register (ISPR)
Un-pending bit	Interrupt clear pending register (ICPR)
Active bit	Interrupt active bit register (IABR)
Software trigger bit	Software trigger interrupt register (STIR)

INTERRUPCIONES CORTEX-M

ESTADOS DE EXEPCIONES

- **Inactivo:** La excepción no está activa ni pendiente.
- **Pendiente:** La excepción está a la espera de que la procese el procesador.
- **Activo:** Una excepción que está siendo atendida por el procesador pero que no se ha completado.
- **Activo y Pendiente:** la excepción está siendo atendida por el procesador y hay un pendiente de excepción de la misma fuente.

INTERRUPCIONES CORTEX-M

INTERRUPCIONES

- **NUMERO DE INTERRUPCION (IRQn)**

Los procesadores ARM CORTEX-M soportan hasta 240 tipos de interrupciones, excluyendo la interrupción del reset, son identificados con un único número en el rango de -15 a 240.

```
/******  Cortex-M4 Processor Exceptions Numbers ***********************************************************/
  NonMaskableInt_IRQn        = -14,    /*!< 2 Non Maskable Interrupt
  MemoryManagement_IRQn      = -12,    /*!< 4 Cortex-M4 Memory Management Interrupt
  BusFault_IRQn              = -11,    /*!< 5 Cortex-M4 Bus Fault Interrupt
  UsageFault_IRQn            = -10,    /*!< 6 Cortex-M4 Usage Fault Interrupt
  SVCall_IRQn                = -5,     /*!< 11 Cortex-M4 SV Call Interrupt
  DebugMonitor_IRQn          = -4,     /*!< 12 Cortex-M4 Debug Monitor Interrupt
  PendSV_IRQn                = -2,     /*!< 14 Cortex-M4 Pend SV Interrupt
  SysTick_IRQn               = -1,     /*!< 15 Cortex-M4 System Tick Interrupt
/******  STM32 specific Interrupt Numbers *****************************************************************/
  WWDG_IRQn                  = 0,      /*!< Window WatchDog Interrupt
  PVD_IRQn                   = 1,      /*!< PVD through EXTI Line detection Interrupt
  TAMP_STAMP_IRQn            = 2,      /*!< Tamper and TimeStamp interrupts through the EXTI line
  RTC_WKUP_IRQn              = 3,      /*!< RTC Wakeup interrupt through the EXTI line
  FLASH_IRQn                 = 4,      /*!< FLASH global Interrupt
  RCC_IRQn                   = 5,      /*!< RCC global Interrupt
  EXTI0_IRQn                 = 6,      /*!< EXTI Line0 Interrupt
  EXTI1_IRQn                 = 7,      /*!< EXTI Line1 Interrupt
  EXTI2_IRQn                 = 8,      /*!< EXTI Line2 Interrupt
  EXTI3_IRQn                 = 9,      /*!< EXTI Line3 Interrupt
  EXTI4_IRQn                 = 10,     /*!< EXTI Line4 Interrupt
  DMA1_Stream0_IRQn          = 11,     /*!< DMA1 Stream 0 global Interrupt
  DMA1_Stream1_IRQn          = 12,     /*!< DMA1 Stream 1 global Interrupt
  DMA1_Stream2_IRQn          = 13,     /*!< DMA1 Stream 2 global Interrupt
  DMA1_Stream3_IRQn          = 14,     /*!< DMA1 Stream 3 global Interrupt
  DMA1_Stream4_IRQn          = 15,     /*!< DMA1 Stream 4 global Interrupt
  DMA1_Stream5_IRQn          = 16,     /*!< DMA1 Stream 5 global Interrupt
  DMA1_Stream6_IRQn          = 17,     /*!< DMA1 Stream 6 global Interrupt
  ADC_IRQn                   = 18,     /*!< ADC1, ADC2 and ADC3 global Interrupts
  EXTI9_5_IRQn               = 23,     /*!< External Line[9:5] Interrupts
```

DEFINIDOS POR LA ARQUITECTURA

STM32F401RE
Definidos por el fabricante

MICRO-CONTRO-LADORES ARM

AKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

INTERRUPCIONES

- **RUTINAS DE SERVICIO DE INTERRUPCION (ISR)**

Son rutinas especiales que son invocados automáticamente en respuesta a una interrupción. Cada rutina de servicio de interrupción esta definido por defecto en el system startup. (startup_stm32f401xe.s).

```
SysTick_Handler PROC
                EXPORT  SysTick_Handler            [WEAK]
                B       .
                ENDP
```

```
void EXTI15_10_IRQHandler(void){
    if(EXTI->PR & EXTI_PR_PR13){
        EXTI->PR |= EXTI_PR_PR13;
        GPIOA->ODR ^= GPIO_ODR_OD5;
    }

}
```

arm
MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

TABLA DE INTERRUPCIONES

arm

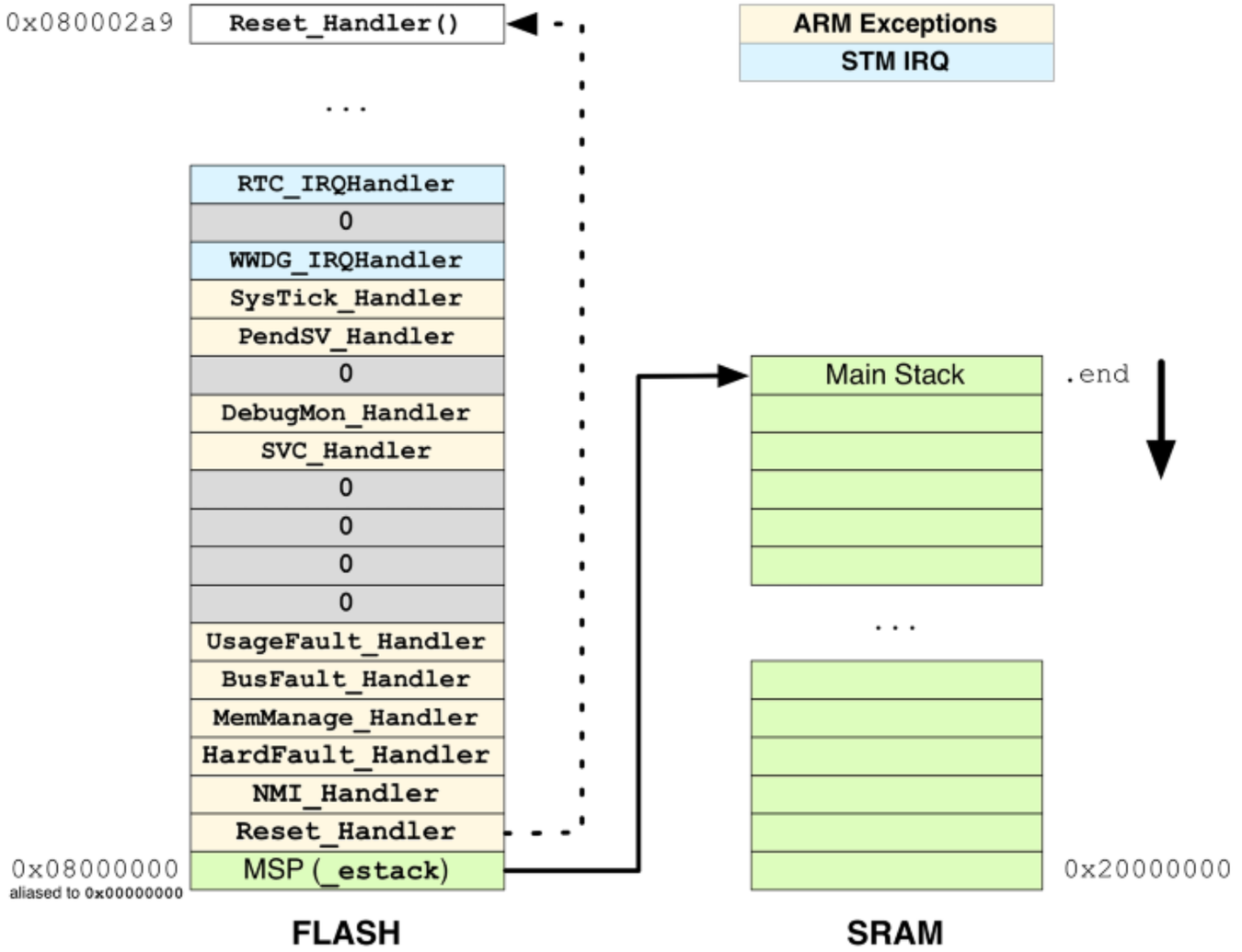
MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

INTERRUPCIONES

- TABLA DE VECTOR DE INTERRUPCIONES



PRIORIDAD Y SUBPRIORIDAD

arm

MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

PRIORIDAD

Configurable priority values are in the range 0-15.

- System handler priority registers (SHPRx)

Handler	Field	Register description
Memory management fault	PRI_4	*System handler priority register 1 (SHPR1)*
Bus fault	PRI_5	
Usage fault	PRI_6	
SVCall	PRI_11	*System handler priority register 2 (SHPR2) on page 233*
PendSV	PRI_14	*System handler priority register 3 (SHPR3) on page 234*
SysTick	PRI_15	

- Interrupt priority register x (NVIC_IPRx)

INTERRUPCIONES CORTEX-M

PRIORIDAD Y SUBPRIORIDAD

Para aumentar el control de prioridad en sistemas con interrupciones, el NVIC admite la agrupación de prioridades.

- Un campo superior que define la prioridad del grupo.
- Un campo inferior que define una subprioridad dentro del grupo.

Application interrupt and reset control register (AIRCR)

Address offset: 0x0C
Reset value: 0xFA05 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VECTKEYSTAT[15:0](read)/ VECTKEY[15:0](write)															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDIANESS	Reserved				PRIGROUP			Reserved					SYS RESET REQ	VECT CLR ACTIVE	VECT RESET
r					rw	rw	rw						w	w	w

Bits 10:8 **PRIGROUP**: Interrupt priority grouping field
This field determines the split of group priority from subpriority, see *Binary point on page 228*.

PRIGROUP [2:0]	Interrupt priority level value, PRI_N[7:4]			Number of	
	Binary point(1)	Group priority bits	Subpriority bits	Group priorities	Sub priorities
0b0xx	0bxxxx	[7:4]	None	16	None
0b100	0bxxx.y	[7:5]	[4]	8	2
0b101	0bxx.yy	[7:6]	[5:4]	4	4
0b110	0bx.yyy	[7]	[6:4]	2	8
0b111	0b.yyyy	None	[7:4]	None	16

MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

REGISTROS PARA PROGRAMAR EL NVIC

arm

MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

REGISTROS DE LA NVIC

Address	Name	Type	Required privilege	Reset value	Description
0xE000E100-0xE000E11F	NVIC_ISER0-NVIC_ISER7	RW	Privileged	0x00000000	*Table 4.3.2: Interrupt set-enable register x (NVIC_ISERx) on page 210*
0XE000E180-0xE000E19F	NVIC_ICER0-NVIC_ICER7	RW	Privileged	0x00000000	*Table 4.3.3: Interrupt clear-enable register x (NVIC_ICERx) on page 211*
0XE000E200-0xE000E21F	NVIC_ISPR0-NVIC_ISPR7	RW	Privileged	0x00000000	*Table 4.3.4: Interrupt set-pending register x (NVIC_ISPRx) on page 212*
0XE000E280-0xE000E29F	NVIC_ICPR0-NVIC_ICPR7	RW	Privileged	0x00000000	*Table 4.3.5: Interrupt clear-pending register x (NVIC_ICPRx) on page 213*
0xE000E300-0xE000E31F	NVIC_IABR0-NVIC_IABR7	RW	Privileged	0x00000000	*Table 4.3.6: Interrupt active bit register x (NVIC_IABRx) on page 214*
0xE000E400-0xE000E4EF	NVIC_IPR0-NVIC_IPR59	RW	Privileged	0x00000000	*Table 4.3.7: Interrupt priority register x (NVIC_IPRx) on page 215*
0xE000EF00	STIR	WO	Configurable	0x00000000	*Table 4.3.8: Software trigger interrupt register (NVIC_STIR) on page 216*

PDF: Programming manual STM32L4

INTERRUPCIONES CORTEX-M

REGISTROS DE LA NVIC

Interrupt set-enable register x (NVIC_ISERx)

```
Address offset: 0x100 + 0x04 * x, (x = 0 to 7)
Reset value: 0x0000 0000
Required privilege: Privileged
NVIC_ISER0 bits 0 to 31 are for interrupt 0 to 31, respectively
NVIC_ISER1 bits 0 to 31 are for interrupt 32 to 63, respectively
....
NVIC_ISER6 bits 0 to 31 are for interrupt 192 to 223, respectively
NVIC_ISER7 bits 0 to 15 are for interrupt 224 to 239, respectively
```

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SETENA[31:16]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SETENA[15:0]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

Bits 31:0 **SETENA**: Interrupt set-enable bits.

Write:

0: No effect

1: Enable interrupt

Read:

0: Interrupt disabled

1: Interrupt enabled.

If a pending interrupt is enabled, the NVIC activates the interrupt based on its priority. If an interrupt is not enabled, asserting its interrupt signal changes the interrupt state to pending, but the NVIC never activates the interrupt, regardless of its priority.

Bits 16 to 31 of the NVIC_ISER7 register are reserved.

INTERRUPCIONES CORTEX-M

REGISTROS DE LA NVIC

Interrupt clear-enable register x (NVIC_ICERx))

```
Address offset: 0x180 + 0x04 * x, (x = 0 to 7)
Reset value: 0x0000 0000
Required privilege: Privileged
NVIC_ICER0 bits 0 to 31 are for interrupt 0 to 31, respectively
NVIC_ICER1 bits 0 to 31 are for interrupt 32 to 63, respectively
....
NVIC_ICER6 bits 0 to 31 are for interrupt 192 to 223, respectively
NVIC_ICER7 bits 0 to 15 are for interrupt 224 to 239, respectively
```

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRENA[31:16]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRENA[15:0]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits 31:0 **CLRENA**: Interrupt clear-enable bits.

Write:

0: No effect

1: Disable interrupt

Read:

0: Interrupt disabled

1: Interrupt enabled.

Bits 16 to 31 of the NVIC_ICER7 register are reserved.

INTERRUPCIONES CORTEX-M

REGISTROS DE LA NVIC

Interrupt priority register x (NVIC_IPRx)

Address offset: 0x400 + 0x04 * x, (x = 0 to 59)
Reset value: 0x0000 0000
Required privilege: Privileged

	31 – 24	23 – 16	15 – 8	7 – 0
NVIC_IPR59	IP[239]	IP[238]	IP[237]	IP[236]
NVIC_IPRx	IP[4x+3]	IP[4x+2]	IP[4x+1]	IP[4x]
NVIC_IPR0	IP[3]	IP[2]	IP[1]	IP[0]

arm
MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

- **HABILITAR/DESHABILITAR INTERRUPCIONES DE PERIFERICOS**

Escribir un 1 en el registro **ISER** habilita una interrupción y escribir un 1 en el registro **ICER.**

$$byteOffset = floor\left(\frac{\text{Numero de Interrupción (IRQn)}}{32}\right)$$

$$byteOffset = \text{Numero de Interrupción (IRQn)} >> 5U$$

$$BitOffset = \text{Numero de Interrupción (IRQn)} \ \& \ 0x1F$$

Habilitar la interrupcion para el numero de interrupcion 77:

$$byteOffset = floor\left(\frac{77}{32}\right) = 2$$

$$BitOffset = 77 \ \& \ 31$$

$$BitOffset = 13$$

$$NVIC->ISER[byteOffset] = 1 << bitOffset;$$

$$NVIC->ISER[2] = 1 << 13;$$

en general:

$$NVIC->ISER[IRQn >> 5U] = 1U << (XXX\_IRQn \ \& \ 0x1F);$$

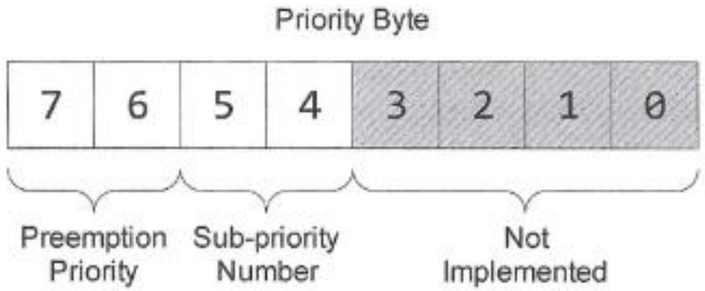
MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

- **ESTABLECER LA PRIORIDAD DE LAS INTERRUPCIONES DE LOS PERIFERICOS**

Cada interrupción cuenta con su registro de interrupción (IP), El cual tiene mapeado 8bits por interrupción. **Un valor bajo de un numero de prioridad representa una prioridad.**



$$NVIC->IP[XXXX\_IRQn] = priority << (4\ \&\ 0xFF);$$

prioridad de exepciones del sistema

```
SCB->SHP[(((uint32_t)IRQn) & 0xFUL)-4UL] = (uint8_t)((priority << (8U - __NVIC_PRIO_BITS)) & (uint32_t)0xFFUL);
SCB->SHP[(((uint32_t)IRQn) & 0xFUL)-4UL] = (uint8_t)((priority << (4)) & (uint32_t)0xFFUL);
```

arm
MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

FUNCIONES CMSIS PARA EL MANEJO DE INTERRUPCIONES

El software usa las instrucciones **CPSEI** y **CPSID** para habilitar y deshabilitar las interrupciones.

```
void __disable_irq(void) // Disable Interrupts
void __enable_irq(void) // Enable Interrupts
```

En adición la CMSIS incluye las funciones:

CMSIS interrupt control function	Description
void NVIC_SetPriorityGrouping(uint32_t priority_grouping)	Set the priority grouping
void NVIC_EnableIRQ(IRQn_t IRQn)	Enable IRQn
void NVIC_DisableIRQ(IRQn_t IRQn)	Disable IRQn
uint32_t NVIC_GetPendingIRQ (IRQn_t IRQn)	Return true (IRQ-Number) if IRQn is pending
void NVIC_SetPendingIRQ (IRQn_t IRQn)	Set IRQn pending
void NVIC_ClearPendingIRQ (IRQn_t IRQn)	Clear IRQn pending status
uint32_t NVIC_GetActive (IRQn_t IRQn)	Return the IRQ number of the active interrupt
void NVIC_SetPriority (IRQn_t IRQn, uint32_t priority)	Set priority for IRQn
uint32_t NVIC_GetPriority (IRQn_t IRQn)	Read priority of IRQn
void NVIC_SystemReset (void)	Reset the system

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO