

02 Interrupt

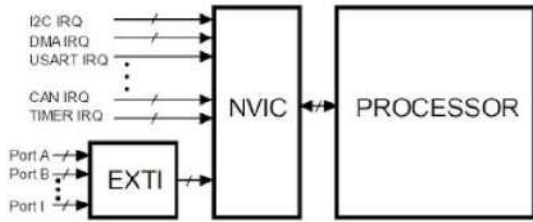
5 Mayıs 2021 Çarşamba

08:02

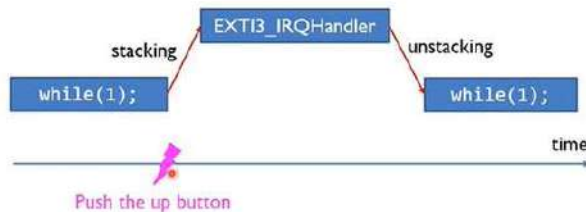
02 Interrupt

Giriş

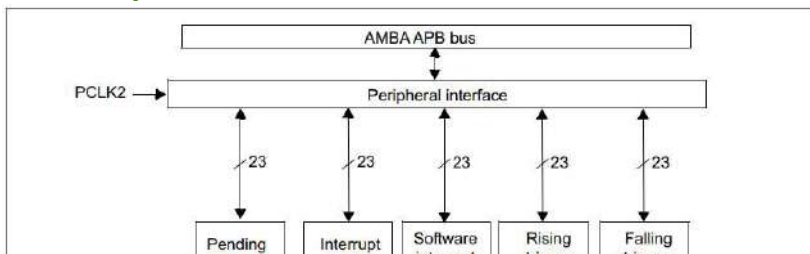
- Polling method sürekli işleciyi meşgul ettiği işlemlerdir. Bu işlemler için **while**, **for** döngüleri kullanılıyor.
- Önceliği yüksek işlerin mikrodenetleyici tarafından ana program akışını keserek yapılmasına interrupt denir.
- Eğer bir kesme kaynağından mikrodenetleyiciye uyarı gelirse mikrodenetleyici yapmakta olduğu işi bekletir, kesme alt programına gider, o programı icra eder, daha sonra ana programda kaldığı yerden devam eder.
- Kesmeleri genellikle **çok hızlı** yapılması gereken işlemlerde, **anlık tepki** verilmesi gereken yerlerde kullanırız.
- Harici bir kaynaktan oluşan olaylardan dolayı meydana gelen kesmelere, **harici kesmeler** denir. Harici kaynak olarak, dış ortamdan pinler vasıtasıyla gelecek kesme ve kendi içindeki donanımlardan gelen kesmeleri anlayabiliriz.
- Karmaşık kesme isteklerinin işlemciye sürekli yük getirmemesi için işlemci içerisinde özel bir donanım bloğu oluşturulmuştur. Bu donanıma **interrupt controller** adı verilir.
- Kesme kontrolörü haklı bir sebeple gelen kesme isteği neticesinde düzgün işleyen programı askıya alarak kesme fonksiyonu (interrupt function) olarak adlandırılan özel kod parçasını işlemeye başlar.
- Kesme fonksiyonunun işletilmesinin bitiminde program kaldığı yerden çalışmaya devam eder.
- **NVIC** kontrolör mikroişlemci içerisindeki önemli donanım kesmelerini (DMA, USART, CAN, I2C ve Timer gibi) ve ayrıca External Interrupt (**EXTI**) adı verilen donanım vasıtasıyla portlardan gelen kesmeleri kontrol eder.



- Interrupt kullanmak için üç farklı yapıyı ayarlamak gerekiyor. **SYSCFG**, **EXTI** ve **NVIC** yapılarını ayarlanarak interrupt kullanabiliriz. İlk ikisi **MCU** ile alakalı iken üçüncüsü **MPU** ile alakalıdır.
- Hatlarımız dışardan multiplexer sayesinde içeriğiye bağlanıyor. Bu bağlanan hatlar aslında içeride EXTI Line olarak tanımlanıyor. Bu bağlantının birinden interrupt bekleniyor ve öncesinden SYSCFG ile söylemem gerekiyor.
- Aynı hatta bağlı yapıda birden fazla interrupt olamaz.
- Daha sonra hatlardan gelen interrupt görmemesi için maskelenmiş durumda olan hattı, EXTI ile önce kaldırmamız gerekiyor sonra gelen sinyalde **yükselen** kenarda mı yoksa **düşen** kenarda mı interrupt girmesini istediğimi belirtmem gerekiyor.
- En son bu gelen Interruptlar NVIC yapısında toplanıyorlar. Bu yapı ile birleşen hatlara **IRQ** olarak adlandırıyoruz.
- NVIC ayarlamasında MPU kısmında Interruptın geleceğini söylemem gerekiyor bunun için işlemcinin kendi datashetini kullanarak ulaşabiliriz.



Birim Yapısı



- STM32 mikrodenetleyicilerindeki interrupt kaynaklarının ve öncelik yapılandırılmaları bir özetini aşağıdaki tabloda göstermektedir. Her bir satır, belirli bir kesme kaynağına dair detayları içerir.
 - **Position**, kesmenin öncelik tablosundaki sıralamasını gösterir. Kesme kaynakları genelde pozisyon numarasıyla tanımlanır. Bu sıralama, mikrodenetleyicinin NVIC (Nested Vector Interrupt Controller) yapılandırmasıyla ilgilidir.
 - **Priority**, kesmenin varsayılan önceliğini belirtir.
 - **Type of Priority**, kesmenin öncelik türünü ifade eder.
 - Bazı kesmeler (örneğin Reset, NMI, HardFault) sabit öncelikli **fixed** olup değiştirilemezdir.
 - Reset: Sistem resetlendiğinde çalışır.
 - HardFault: Ciddi hatalar (örneğin, yanlış bellek erişimi) durumunda çalışır.
 - Çoğu çevresel birim kesmeleri (örneğin, EXTI, CAN, USART v.b) ise yazılım tarafından ayarlanabilir **settable** yapıdadır. Kullanıcı, uygulamanın gereksinimlerine göre bu kesmelerin **Preemption Priority** ve **SubPriority** değerlerini ayarlayabilir.
 - **Acronym**, kesmenin kısa adı verilmiştir. Bu, kesmenin hangi kaynağa ait olduğunu anlamamızı sağlar. Örneğin; NMI: Non-Maskable Interrupt (maskelenemez kesme), EXTI0: EXTI hattı üzerinden tetiklenen dış kesme, CAN1_RX0: CAN1 modülünden gelen RX kesmesi
 - **Address**, kesmenin vektör tablosundaki vektör adresini gösterir.

Position	Priority	Type of priority	Acronym	Description	Address
-	-	-	-	Reserved	0x0000 0000
-	-3	fixed	Reset	Reset	0x0000 0004
-	-2	fixed	NMI	Non maskable interrupt. The RCC Clock Security System (CSS) is linked to the NMI vector.	0x0000 0008
-	-1	fixed	HardFault	All class of fault	0x0000 000C
-	0	settable	MemManage	Memory management	0x0000 0010
-	1	settable	BusFault	Pre-fetch fault, memory access fault	0x0000 0014
-	2	settable	UsageFault	Undefined instruction or illegal state	0x0000 0018
-	-	-	-	Reserved	0x0000 001C - 0x0000 002B
-	3	settable	SVCall	System service call via SWI instruction	0x0000 002C
-	4	settable	Debug Monitor	Debug Monitor	0x0000 0030
-	-	-	-	Reserved	0x0000 0034
-	5	settable	PendSV	Pendable request for system service	0x0000 0038
-	6	settable	SysTick	System tick timer	0x0000 003C
0	7	settable	WWDG	Window Watchdog interrupt	0x0000 0040
1	8	settable	PVD	PVD through EXTI line detection interrupt	0x0000 0044
2	9	settable	TAMP_STAMP	Tamper and TimeStamp interrupts through the EXTI line	0x0000 0048
3	10	settable	RTC_WKUP	RTC Wakeup interrupt through the EXTI line	0x0000 004C
4	11	settable	FLASH	Flash global interrupt	0x0000 0050
5	12	settable	RCC	RCC global interrupt	0x0000 0054
6	13	settable	EXTI0	EXTI Line0 interrupt	0x0000 0058
7	14	settable	EXTI1	EXTI Line1 interrupt	0x0000 005C

5	12	settable	RCC	RCC global interrupt	0x0000 0054
6	13	settable	EXTI0	EXTI Line0 interrupt	0x0000 0058
7	14	settable	EXTI1	EXTI Line1 interrupt	0x0000 005C
8	15	settable	EXTI2	EXTI Line2 interrupt	0x0000 0060
9	16	settable	EXTI3	EXTI Line3 interrupt	0x0000 0064
10	17	settable	EXTI4	EXTI Line4 interrupt	0x0000 0068
11	18	settable	DMA1_Stream0	DMA1 Stream0 global interrupt	0x0000 006C
12	19	settable	DMA1_Stream1	DMA1 Stream1 global interrupt	0x0000 0070
13	20	settable	DMA1_Stream2	DMA1 Stream2 global interrupt	0x0000 0074
14	21	settable	DMA1_Stream3	DMA1 Stream3 global interrupt	0x0000 0078
15	22	settable	DMA1_Stream4	DMA1 Stream4 global interrupt	0x0000 007C
16	23	settable	DMA1_Stream5	DMA1 Stream5 global interrupt	0x0000 0080
17	24	settable	DMA1_Stream6	DMA1 Stream6 global interrupt	0x0000 0084
18	25	settable	ADC	ADC1, ADC2 and ADC3 global interrupts	0x0000 0088
19	26	settable	CAN1_TX	CAN1 TX interrupts	0x0000 008C
20	27	settable	CAN1_RX0	CAN1 RX0 interrupts	0x0000 0090
21	28	settable	CAN1_RX1	CAN1 RX1 interrupt	0x0000 0094
22	29	settable	CAN1_SCE	CAN1 SCE interrupt	0x0000 0098
23	30	settable	EXTI9_5	EXTI Line[9:5] interrupts	0x0000 009C
24	31	settable	TIM1_BRK_TIM9	TIM1 Break interrupt and TIM9 global interrupt	0x0000 00A0
25	32	settable	TIM1_UP_TIM10	TIM1 Update interrupt and TIM10 global interrupt	0x0000 00A4
26	33	settable	TIM1_TRG_COM_TIM11	TIM1 Trigger and Commutation interrupts and TIM11 global interrupt	0x0000 00A8
27	34	settable	TIM1_CC	TIM1 Capture Compare interrupt	0x0000 00AC
28	35	settable	TIM2	TIM2 global interrupt	0x0000 00B0
29	36	settable	TIM3	TIM3 global interrupt	0x0000 00B4
30	37	settable	TIM4	TIM4 global interrupt	0x0000 00B8
31	38	settable	I2C1_EV	I ² C1 event interrupt	0x0000 00BC
32	39	settable	I2C1_ER	I ² C1 error interrupt	0x0000 00C0
33	40	settable	I2C2_EV	I ² C2 event interrupt	0x0000 00C4
34	41	settable	I2C2_ER	I ² C2 error interrupt	0x0000 00C8
35	42	settable	SPI1	SPI1 global interrupt	0x0000 00CC
36	43	settable	SPI2	SPI2 global interrupt	0x0000 00D0
37	44	settable	USART1	USART1 global interrupt	0x0000 00D4
38	45	settable	USART2	USART2 global interrupt	0x0000 00D8
39	46	settable	USART3	USART3 global interrupt	0x0000 00DC
40	47	settable	EXTI15_10	EXTI Line[15:10] interrupts	0x0000 00E0
41	48	settable	RTC_Alarm	RTC Alarms (A and B) through EXTI line interrupt	0x0000 00E4
42	49	settable	OTG_FS_WKUP	USB On-The-Go FS Wakeup through EXTI line interrupt	0x0000 00E8
43	50	settable	TIM8_BRK_TIM12	TIM8 Break interrupt and TIM12 global interrupt	0x0000 00EC
44	51	settable	TIM8_UP_TIM13	TIM8 Update interrupt and TIM13 global interrupt	0x0000 00F0
45	52	settable	TIM8_TRG_COM_TIM14	TIM8 Trigger and Commutation interrupts and TIM14 global interrupt	0x0000 00F4
46	53	settable	TIM8_CC	TIM8 Capture Compare interrupt	0x0000 00F8
47	54	settable	DMA1_Stream7	DMA1 Stream7 global interrupt	0x0000 00FC
48	55	settable	FSMC	FSMC global interrupt	0x0000 0100
49	56	settable	SDIO	SDIO global interrupt	0x0000 0104
50	57	settable	TIM5	TIM5 global interrupt	0x0000 0108
51	58	settable	SPI3	SPI3 global interrupt	0x0000 010C
52	59	settable	UART4	UART4 global interrupt	0x0000 0110
53	60	settable	UART5	UART5 global interrupt	0x0000 0114
54	61	settable	TIM6_DAC	TIM6 global interrupt, DAC1 and DAC2 underrun error interrupts	0x0000 0118
55	62	settable	TIM7	TIM7 global interrupt	0x0000 011C
56	63	settable	DMA2_Stream0	DMA2 Stream0 global interrupt	0x0000 0120
57	64	settable	DMA2_Stream1	DMA2 Stream1 global interrupt	0x0000 0124
58	65	settable	DMA2_Stream2	DMA2 Stream2 global interrupt	0x0000 0128
59	66	settable	DMA2_Stream3	DMA2 Stream3 global interrupt	0x0000 012C
60	67	settable	DMA2_Stream4	DMA2 Stream4 global interrupt	0x0000 0130
61	68	settable	ETH	Ethernet global interrupt	0x0000 0134
62	69	settable	ETH_WKUP	Ethernet Wakeup through EXTI line interrupt	0x0000 0138
63	70	settable	CAN2_TX	CAN2 TX interrupts	0x0000 013C
64	71	settable	CAN2_RX0	CAN2 RX0 interrupts	0x0000 0140

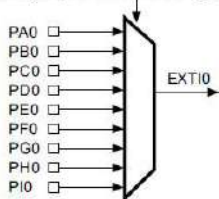
- Kesme öncelikleri iki farklı kavramla ifade edilir.
 - **Preemption Priority**, daha düşük bir Preemption Priority değerine sahip bir kesme, daha yüksek bir Preemption Priority değerine sahip bir kesmenin işlenmesini kesebilir. Kesme sırasındaki en üst seviye

- Kesme öncelikleri iki farklı kavramla ifade edilir.
 - **Preemption Priority**, daha düşük bir Preemption Priority değerine sahip bir kesme, daha yüksek bir Preemption Priority değerine sahip bir kesmenin işlenmesini kesebilir. Kesme sırasındaki en üst seviye kontrolü sağlar. Bu, daha kritik işlemlerin daha az kritik olanları kesintiye uğratmasına izin verir.
 - **SubPriority**, Aynı Preemption Priority seviyesine sahip kesmeler arasında sıralama yapılmasını sağlar. Daha düşük bir SubPriority değerine sahip olan kesme, aynı Preemption Priority seviyesinde daha yüksek önceliğe sahiptir. Yalnızca bir kesme zaten işlenirken birden fazla aynı öncelikteki kesme bekliyorsa kullanılır.
 - Preemption Priority **kritik öncelik sıralamasını** yönetirken, SubPriority **aynı seviyedeki** kesmeler arasında işleme sırasını belirler.
- Bununla ilgili bir örnek senaryo yapalım
 - Diyelim ki bir sistemde üç kesme olsun.
 - Timer (TMR), kritik zamanlama gerektiriyor.
 - USART, veri iletişimi için kullanılıyor.
 - GPIO ise butona basıldığında bir işlem başlatıyor.
 - Şimdi de bunlar için önceliklerin ayarlanması işlemini yapalım.
 - Timer, en kritik olduğu için Preemption Priority: 0, SubPriority: 0
 - USART, orta kritiklikte, Preemption Priority: 1, SubPriority: 0
 - GPIO, en düşük öncelik, Preemption Priority: 2, SubPriority: 1
 - Daha sonrasında çalışma şekli aşağıdaki gibi olur.
 - Timer kesmesi çalışırken başka bir kesme geldiğinde, Timer kesmesi yalnızca daha düşük Preemption Priority değerine sahip kesmeler tarafından kesilebilir (örneğin, hiçbiri bu durumda)
 - USART kesmesi çalışırken Timer kesmesi gelirse, Timer kesmesi (daha yüksek önceliğe sahip olduğu için) USART kesmesini kesebilir.
 - GPIO kesmesi çalışırken USART veya Timer kesmeleri gelirse, GPIO kesmesi kesilir, daha yüksek öncelikli kesmeler işlenir.
- NVIC'nin Preemption Priority ve SubPriority için kaç bit ayıracağı, **Priority Grouping** ile belirlenir.

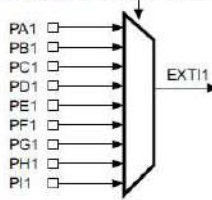
Priority Grouping	Preemption Priority Bits	SubPriority Bits
NVIC_PRIORITYGROUP_0	0	4
NVIC_PRIORITYGROUP_2	2	2
NVIC_PRIORITYGROUP_4	4	0

- Örnek olarak NVIC_PRIORITYGROUP_2 seçildiğinde,
 - 2 bit **Preemption Priority**, 2 bit **SubPriority** kullanılabilir.
 - Bu durumda, **Preemption Priority** değeri [0, 3] arasında olabilir, **SubPriority** ise [0, 3] arasında olabilir.
- STM32F407 mikrodenetleyicisi için porttaki 0.pin EXTI0 kanalına bağlıdır.

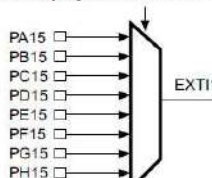
EXTI0[3:0] bits in the SYSCFG_EXTICR1 register



EXTI1[3:0] bits in the SYSCFG_EXTICR1 register



EXTI15[3:0] bits in the SYSCFG_EXTICR4 register



Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	EXTI_IMR	Reserved										MR[22:0]																						
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	EXTI_EMR	Reserved										MR[22:0]																						

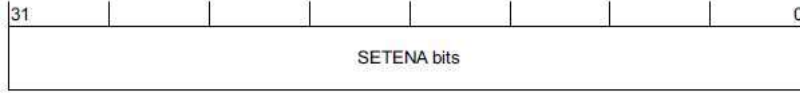
	Reset value		0 0
0x04	EXTI_EMR	Reserved	MR[22:0]
	Reset value		0 0
0x08	EXTI_RTSTR	Reserved	TR[22:0]
	Reset value		0 0
0x0C	EXTI_FTSR	Reserved	TR[22:0]
	Reset value		0 0
0x10	EXTI_SWIER	Reserved	SWIER[22:0]
	Reset value		0 0
0x14	EXTI_PR	Reserved	PR[22:0]
	Reset value		0 0

- **EXTI_IMR** (Interrupt Mask Register), harici kesmelerin genel olarak etkinleştirilip etkinleştirilmeyeceğini kontrol eder. Her bit, belirli bir harici kesme hattını temsil eder ve bu bitlerin set olması, ilgili kesmenin etkinleştirildiği anlamına gelir.
 - **IMx (Interrupt Mask)**: EXTI hattını etkinleştirir veya devre dışı bırakır.
 - **0**: Kesme Devre Dışı (Interrupt Masked)
 - **1**: Kesme Etkin (Interrupt Unmasked)
- **EXTI_EMR** (Event Mask Register), EXTI modülü, hem interrupt hem de event modlarında çalışabilir. Belirli bir harici kesme hattının olay modunda çalışıp çalışmayacağını kontrol eder. Yine, her bit belirli bir kesme hattını temsil eder.
 - **EMx (Event Mask)**: EXTI hattını olay için etkinleştirir veya devre dışı bırakır.
 - **0**: Olay Devre Dışı (Event Masked)
 - **1**: Olay Etkin (Event Unmasked)
- **EXTI_RTSTR** (Rising Trigger Selection Register) ve **EXTI_FTSR** (Falling Trigger Selection Register), bir harici kesmenin hangi kenardan rising ya da falling edge tetikleneceğini belirler. Her bit, bir kesme hattını temsil eder ve bu bitlerin set olması, ilgili kesmenin düşen ya da yükselen kenardan tetikleneceği anlamına gelir.
 - **RTx (Rising Trigger)**: EXTI hattını yükselen kenara duyarlı hale getirir.
 - **0**: Yükselen Kenar Devre Dışı
 - **1**: Yükselen Kenar Etkin
 - **FTx (Falling Trigger)**: EXTI hattını düşen kenara duyarlı hale getirir.
 - **0**: Düşen Kenar Devre Dışı
 - **1**: Düşen Kenar Etkin
- **EXTI_SWIER** (Software Interrupt Event Register), yazılımsal olarak bir harici kesme talebi oluşturmak için kullanılır. Her bit, belirli bir harici kesmeyi temsil eder ve bu bitin set olması, ilgili kesme hattına bir yazılımsal talep gönderileceği anlamına gelir.
 - **SWIx (Software Interrupt)**: EXTI hattını yazılım ile tetikler.
 - **0**: Yazılım Kesmesi Devre Dışı
 - **1**: Yazılım Kesmesi Etkin
- **EXTI_PR** (Pending Register), hangi harici kesmelerin beklediğini gösterir. Her bit, belirli bir kesme hattını temsil eder ve bu bitlerin set olması, ilgili kesmenin beklediği anlamına gelir. Yazılım tarafından temizlenmelidir.
 - **PRx (Pending Flag)**: EXTI hattında kesme bekleyip beklemediğini gösterir.
 - **0**: Bekleyen Kesme Yok
 - **1**: Bekleyen Kesme Var (Bu bit yazılım ile sıfırlanabilir)

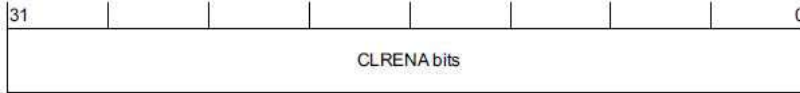
Address	Name	Type	Required privilege	Reset value	Description
0xE000E100-0xE000E11C	NVIC_ISER0-NVIC_ISER7	RW	Privileged	0x00000000	<i>Interrupt Set-enable Registers</i>
0xE000E180-0xE000E19C	NVIC_ICER0-NVIC_ICER7	RW	Privileged	0x00000000	<i>Interrupt Clear-enable Registers</i>
0xE000E200-0xE000E21C	NVIC_ISPR0-NVIC_ISPR7	RW	Privileged	0x00000000	<i>Interrupt Set-pending Registers</i>
0xE000E280-0xE000E29C	NVIC_ICPR0-NVIC_ICPR7	RW	Privileged	0x00000000	<i>Interrupt Clear-pending Registers</i>
0xE000E300-0xE000E31C	NVIC_IABR0-NVIC_IARR7	RW	Privileged	0x00000000	<i>Interrupt Active Bit Registers</i>

0xE000E280-0xE000E29C	NVIC_ICPR0-NVIC_ICPR7	RW	Privileged	0x00000000	Interrupt Clear-pending Registers
0xE000E300-0xE000E31C	NVIC_IABR0-NVIC_IABR7	RW	Privileged	0x00000000	Interrupt Active Bit Registers
0xE000E400-0xE000E4EF	NVIC_IPR0-NVIC_IPR59	RW	Privileged	0x00000000	Interrupt Priority Registers
0xE000EF00	STIR	WO	Configurable ^a	0x00000000	Software Trigger Interrupt Register

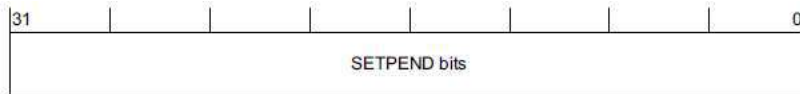
- NVIC_ISERx (Interrupt Set-Enable Registers), NVIC üzerinde belirli bir interrupt hattını etkinleştirmek için kullanılır. Her bit bir interrupt hattını temsil eder. Bit 1 yapılırsa ilgili interrupt etkinleşir.



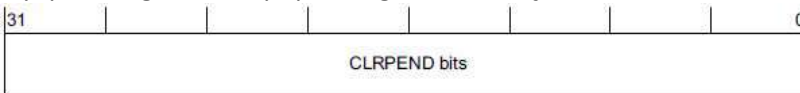
- NVIC_ICERx (Interrupt Clear-Enable Registers), NVIC üzerinde etkinleştirilmiş interrupt'ları devre dışı bırakmak için kullanılır. İlgili bit 1 yapılırsa interrupt devre dışı kalır.



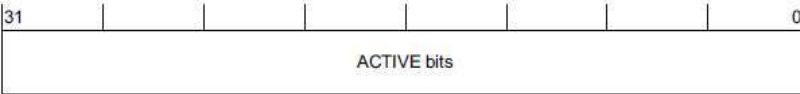
- NVIC_ISPRx (Interrupt Set-Pending Registers), belirli bir interrupt'ı **pending (bekleyen)** duruma almak için kullanılır. Bit 1 yapılırsa interrupt tetiklenmiş gibi görünür ve NVIC tarafından işlenir.



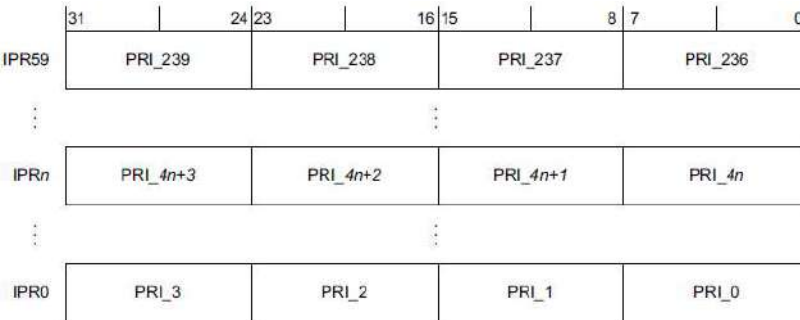
- NVIC_ICPRx (Interrupt Clear-Pending Registers), bekleyen (pending) interrupt'ı temizlemek için kullanılır. Bit 1 yapılırsa ilgili interrupt pending durumdan çıkar.



- NVIC_IABRx (Interrupt Active Bit Registers), her bit, ilgili interrupt hattının **aktif (işlemde)** olup olmadığını gösterir. Okuma ile aktif interrupt'lar görülebilir; yazma etkisi yoktur.



- NVIC_IPRx (Interrupt Priority Registers), Interrupt'ların öncelik seviyesini belirler. Daha düşük değer → daha yüksek öncelik.



- STIR (Software Trigger Interrupt Register), yazılım ile interrupt tetiklemek için kullanılır. İlgili interrupt numarası yazılır, NVIC onu pending duruma alır.



Konfigürasyon Ayarları

1. GPIO Pin Yapılandırması

- Kullanılacak pin **input mode** olarak ayarlanır.
- Gerekirse **pull-up** veya **pull-down** seçilir.

2. SYSCFG (System Configuration) Ayarı

- GPIO pinini uygun **EXTI hattına** yönlendirmek için SYSCFG ayarlanır.
- Örneğin: **SYSCFG_EXTICR** register kullanılır.
 - PA0 → EXTI0
 - PB5 → EXTI5

3. EXTI (External Interrupt/Event Controller) Ayarı

- IMR (Interrupt Mask Register):** İlgili EXTI hattı maskesi açılır → interrupt aktif hale gelir.

3. EXTI (External Interrupt/Event Controller) Ayarı

- **IMR (Interrupt Mask Register):** İlgili EXTI hattı maskesi açılır → interrupt aktif hale gelir.
- **RTSR (Rising Trigger Selection Register):** Yükselen kenarda tetikleme için set edilir.
- **FTSR (Falling Trigger Selection Register):** Düşen kenarda tetikleme için set edilir.
- Gerekirse hem yükselen hem düşen kenar aynı anda kullanılabilir.

4. NVIC (Nested Vectored Interrupt Controller) Ayarı

- **ISER (Interrupt Set-Enable Register):** İlgili interrupt enable edilir.
- **IPR (Interrupt Priority Register):** Öncelik atanır.
- Örneğin: EXTI0_IRQn NVIC'te enable edilir ve önceliği ayarlanır.

5. Interrupt Handler Fonksiyonu

- CMSIS tarafından tanımlı IRQHandler fonksiyonları yazılır.
- EXTI_PR ile pending flag'i temizle