

Perifera och externa avbrott

Ur innehållet:

- NVIC - Nested Vectored Interrupt Controller
- Avbrott från interna periferienheter, ("Timers")
- EXTI - External Interrupt Configuration
- Avbrottsvippa med programstyrd återställning
- Pulsgenerator för periodiska avbrott

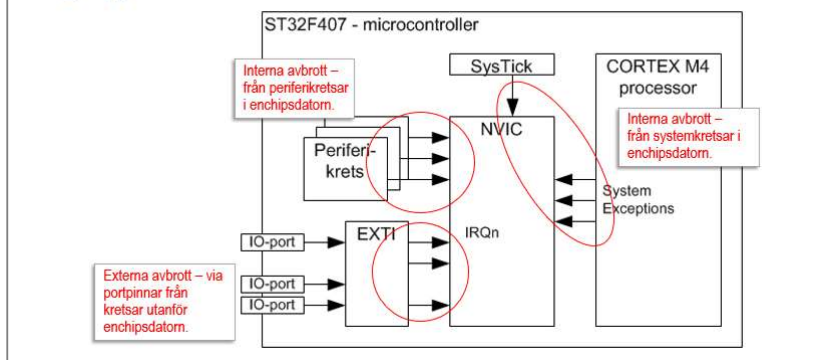
Läsanvisningar:

Arbetsbok kap 6.3, 6.6, 6.7

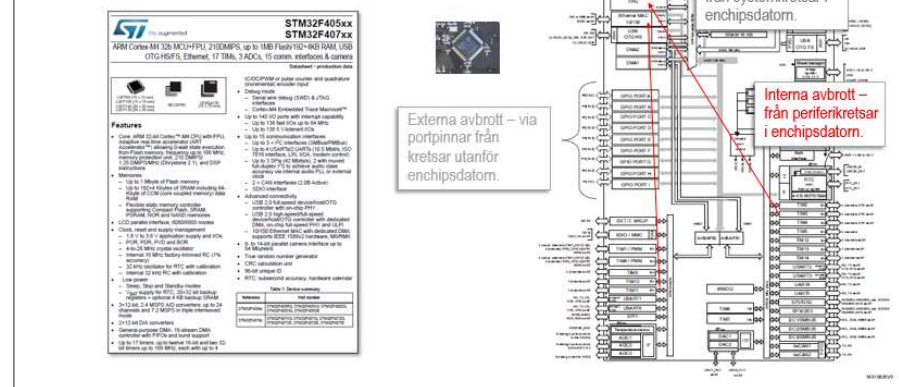
Målsättningar:

- Konfigurera NVIC för interna och externa periferikretsar
- Konfigurera EXTI för portpinne som en avbrottsingång (extern periferikrets)
- Programmera extern avbrottsvippa, och hantera vektoriserade avbrott

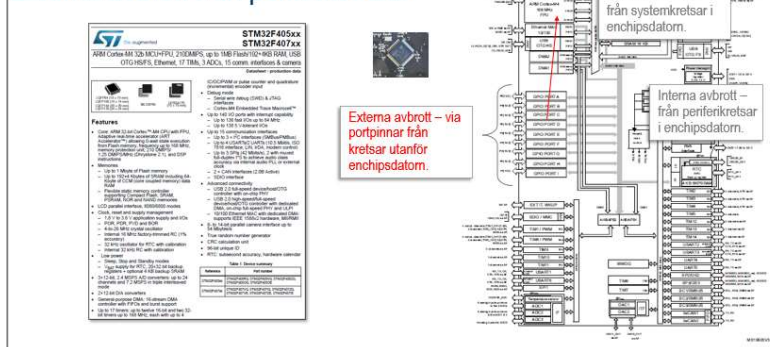
Tre grupper av avbrottskällor



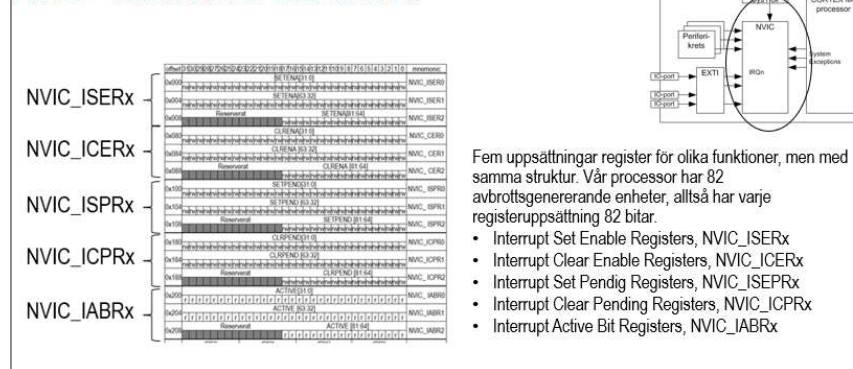
Interna avbrott från periferikretsar



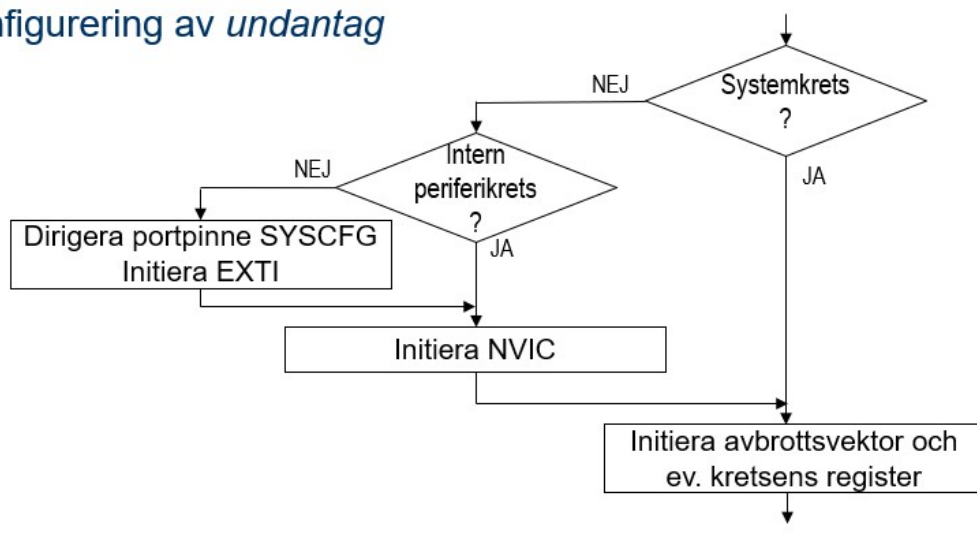
Externa avbrott från periferikretsar



NVIC – Kontrollerar alla avbrott



Konfigurering av *undantag*



SYSCFG EXTICRx används för att koppla IO-pinnar till processorns avbrottssystem

[illegible]

EXTICRx är fyra kontrollregister med samma struktur:

[illegible]

Bitarna i EXTICRx bestämmer hur en IO-pinne dirigeras till någon av de 16 avbrottslinorna
EXTI0..EXTI15 genom att motsvarande fält skrivs med fyra bitar.
Undantag: PKI15[8] används ej.

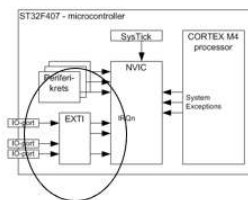
0000: PA[x]	0001: PB[x]	0010: PC[x]	0011: PD[x]	0100: PE[x]	0101: PF[x]
0110: PG[x]	0111: PH[x]	1000: PI[x]	1001: PJ[x]	1010: PK[x]	

EXTI-modul

Totalt finns det 23 olika möjliga avbrottslinor i EXTI-modulen, i STM32F407 används de 16 första för externa avbrott. Några av dessa har gemensam avbrottsvektor.

Irk	Hänteringsrutin	Beskrivning
EXT10_IRQn	EXT10_IRQhandler	för IO-pennar anslutna till anslutnings 0
EXT11_IRQn	EXT11_IRQhandler	för IO-pennar anslutna till anslutnings 1
EXT12_IRQn	EXT12_IRQhandler	för IO-pennar anslutna till anslutnings 2
EXT13_IRQn	EXT13_IRQhandler	för IO-pennar anslutna till anslutnings 3
EXT14_IRQn	EXT14_IRQhandler	för IO-pennar anslutna till anslutnings 4
EXT19_5_IRQn	EXT19_5_IRQhandler	för IO-pennar anslutna till anslutnings 5 till 9
EXT115_10_IRQn	EXT115_10_IRQhandler	för IO-pennar anslutna till anslutnings 10 till 15

Idq	Beskrivning
EXT105	PVD släppte last ansluten till EXT105
EXT117	RTC "Alarm event" last ansluten till EXT107
EXT118	USB OTG FS "Wakeup event" last ansluten till EXT108
EXT119	Ethernet "Wakeup event" last ansluten till EXT109
EXT120	USB OTG HS "Wakeup event" last ansluten till EXT120
EXT121	RTC "Tamper and TimeStamp" last ansluten till EXT1021
EXT122	RTC "Wakeup event" last ansluten till EXT1022

[illegible]

Port pinnar Px 15 .. Px 0 (x=A,B,C,D,E,F)

NVIC – Kontrollerar alla avbrott

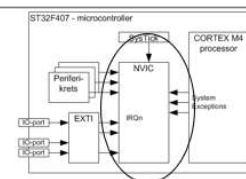


Figure 10: NVIC registers bit fields. The diagram shows the bit fields for the following registers:

- NVIC_USR** (0x000 to 0x00F): USR[0] to USR[31]
- NVIC_ICR** (0x010 to 0x01F): ICR[0] to ICR[31]
- NVIC_IPR** (0x020 to 0x02F): IPR[0] to IPR[31]
- NVIC_ICP** (0x030 to 0x03F): ICP[0] to ICP[31]
- NVIC_IAB** (0x040 to 0x04F): IAB[0] to IAB[31]

The bit fields are labeled with their names and bit positions. The diagram shows the bit fields for each register and their corresponding bit positions.

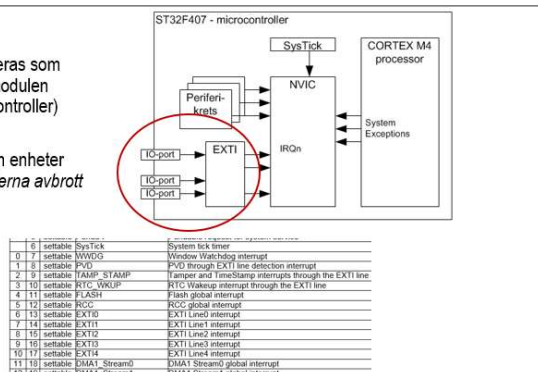
Fem uppsättningar register för olika funktioner, men med samma struktur. Vår processor har 82 avbrottsgenererande enheter, alltså har varje registeruppsättning 82 bitar.

- Interrupt Set Enable Registers, NVIC_ISERx
- Interrupt Clear Enable Registers, NVIC_ICERx
- Interrupt Set Pending Registers, NVIC_ISEPRx
- Interrupt Clear Pending Registers, NVIC_ICPRx
- Interrupt Active Bit Registers, NVIC_IABRx

Externa avbrott

En portpinne kan konfigureras som avbrottsingång via EXTI-modulen (External interrupt/event controller)

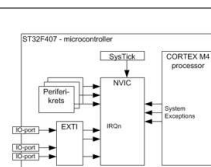
Detta möjliggör avbrott från enheter utanför enchipsdatom, *externa avbrott*



Vektorer för externa avbrottskällor

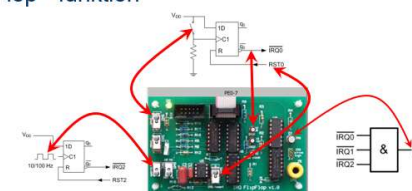
Avbrottsingångar konfigureras i tre steg

- SYSCFG-modulen används för att koppla en GPIO-pinne från någon port till en specifik avbrottslinje (1 av 16).
- EXTI-modulen programmeras.
- NVIC-modulen (Nested vectored interrupt controller) programmeras för att hantera avbrott från denna avbrottsvektor (NVIC_ISERx).

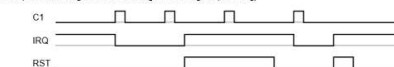


irq	Hanteringarutin	Beskrivning
EXTI0_IRQn	EXTI0_IRQhandler	för IO-pinna analys till avbrottslinje 0
EXTI1_IRQn	EXTI1_IRQhandler	för IO-pinna analys till avbrottslinje 1
EXTI2_IRQn	EXTI2_IRQhandler	för IO-pinna analys till avbrottslinje 2
EXTI3_IRQn	EXTI3_IRQhandler	för IO-pinna analys till avbrottslinje 3
EXTI4_IRQn	EXTI4_IRQhandler	för IO-pinna analys till avbrottslinje 4
EXTI5_9_IRQn	EXTI5_9_IRQhandler	för IO-pinna analys till avbrottslinjer 5 till 9
EXTI10_15_IRQn	EXTI10_15_IRQhandler	för IO-pinna analys till avbrottslinjer 10 till 15

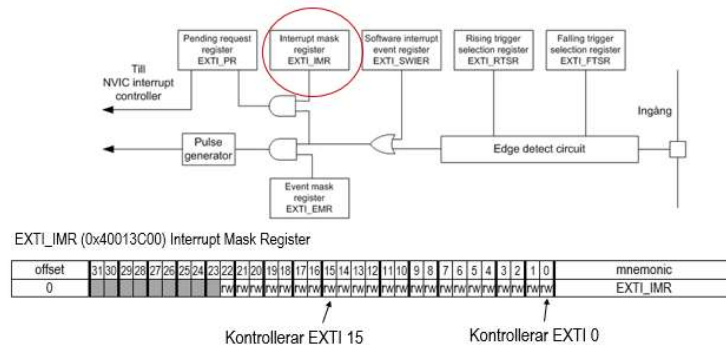
IRQ-FlipFlop - funktion



Exempel: Aktivering och återställning av IRQ-signal (aktiv låg)



EXTI- registren har samma struktur



EXTI_IMR (0x40013C00) Interrupt Mask Register

offset	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	mnemonic
0																																	EXTI_IMR

Kontrollerar EXTI 15

Kontrollerar EXTI 0

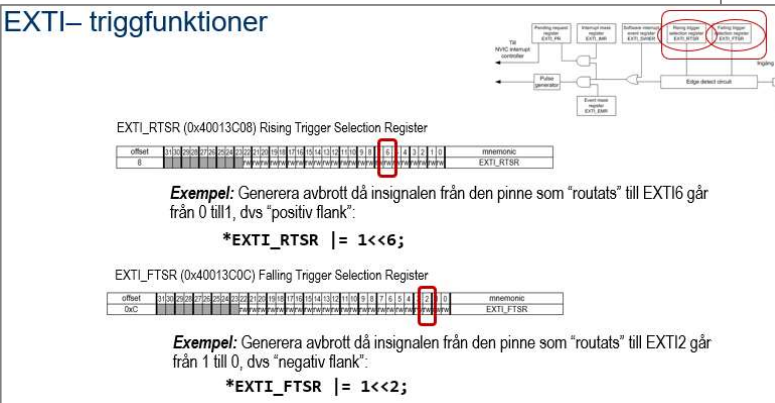
Exempel: Tillåt avbrott från EXTI0

$*EXTI_IMR = 1;$

EXTI- "software interrupt"

Avbrott kan också genereras via EXTI-SWIER. Om avbrott är tillåtet bit i EXTI_SWIER.

EXTI_SWIER (0x40013C04) Software Interrupt Event Register



EXTI_RTSR (0x40013C08) Rising Trigger Selection Register

offset	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	mnemonic
0																																	EXTI_RTSR

Exempel: Generera avbrott då insignalen från den pinne som "routats" till EXTI6 går från 0 till 1, dvs "positiv flank".

$*EXTI_RTSR = 1 << 6;$

EXTI_FTSR (0x40013C0C) Falling Trigger Selection Register

offset	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	mnemonic
0																																	EXTI_FTSR

Exempel: Generera avbrott då insignalen från den pinne som "routats" till EXTI2 går från 1 till 0, dvs "negativ flank".

$*EXTI_FTSR = 1 << 2;$

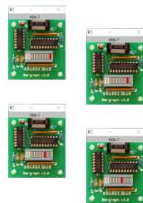
Pseudoparallell exekvering - en enkel "TASK-switch"

En enkel applikation med fyra olika program, (*processerna*) p0, p1, p2 och p3
Växling mellan programmen utförs av ett överordnat program SV(*Supervisor*) vid ett avbrott från EXTIO (bit 0).



```
"Processerna"
void p0( void )
{
  while(1)
  {
    *GPIO_ODR_LOW = 1;
  }
}
void p1( void )
{
  while(1)
  {
    *GPIO_ODR_LOW = 2;
  }
}
void p2( void )
{
  while(1)
  {
    *GPIO_ODR_LOW = 4;
  }
}
void p3( void )
{
  while(1)
  {
    *GPIO_ODR_LOW = 8;
  }
}
```

Processerna tänds var sin diod i diodrampen, vi kan då tydligt se vilken process som körs...



Processtillstånd



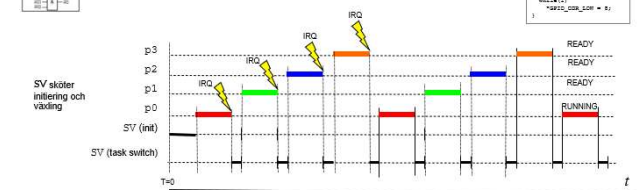
Avbrotten kan också triggas manuellt eller periodiskt från FLIP-FLOP-kortet...



Vi har bara en CPU, som vi växlar mellan sig i olika *tilstånd*.



```
"Processerna"
void p0( void )
{
  while(1)
  {
    *GPIO_ODR_LOW = 1;
  }
}
void p1( void )
{
  while(1)
  {
    *GPIO_ODR_LOW = 2;
  }
}
void p2( void )
{
  while(1)
  {
    *GPIO_ODR_LOW = 4;
  }
}
void p3( void )
{
  while(1)
  {
    *GPIO_ODR_LOW = 8;
  }
}
```



Task-switch

```
"Processerna"
void p0( void )
{
  while(1)
  {
    *GPIO_ODR_LOW = 1;
  }
}
void p1( void )
{
  while(1)
  {
    *GPIO_ODR_LOW = 2;
  }
}
void p2( void )
{
  while(1)
  {
    *GPIO_ODR_LOW = 4;
  }
}
void p3( void )
{
  while(1)
  {
    *GPIO_ODR_LOW = 8;
  }
}
```

