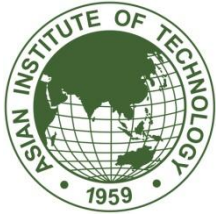


NVI and EXT
Interrupt and Event

Nested Vector Interrupt Controller (NVIC)

- 68 maskable interrupt channels
- 16 programmable priority levels
- Low latency exception and interrupt handling
- Power management control

Vector table

Position	Priority	Type of priority	Acronym	Description	Address
	-	-	-	Reserved	0x0000_0000
	-3	fixed	Reset	Reset	0x0000_0004

Position	Priority	Type of priority	Acronym	Description	Address
	-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	TAMPER	Tamper interrupt	0x0000_0048
3	10	settable	RTC	RTC global interrupt	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
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_Channel1	DMA1 Channel1 global interrupt	0x0000_006C
12	19	settable	DMA1_Channel2	DMA1 Channel2 global interrupt	0x0000_0070
13	20	settable	DMA1_Channel3	DMA1 Channel3 global interrupt	0x0000_0074
14	21	settable	DMA1_Channel4	DMA1 Channel4 global interrupt	0x0000_0078
15	22	settable	DMA1_Channel5	DMA1 Channel5 global interrupt	0x0000_007C
16	23	settable	DMA1_Channel6	DMA1 Channel6 global interrupt	0x0000_0080
17	24	settable	DMA1_Channel7	DMA1 Channel7 global interrupt	0x0000_0084
18	25	settable	ADC1_2	ADC1 and ADC2 global interrupt	0x0000_0088

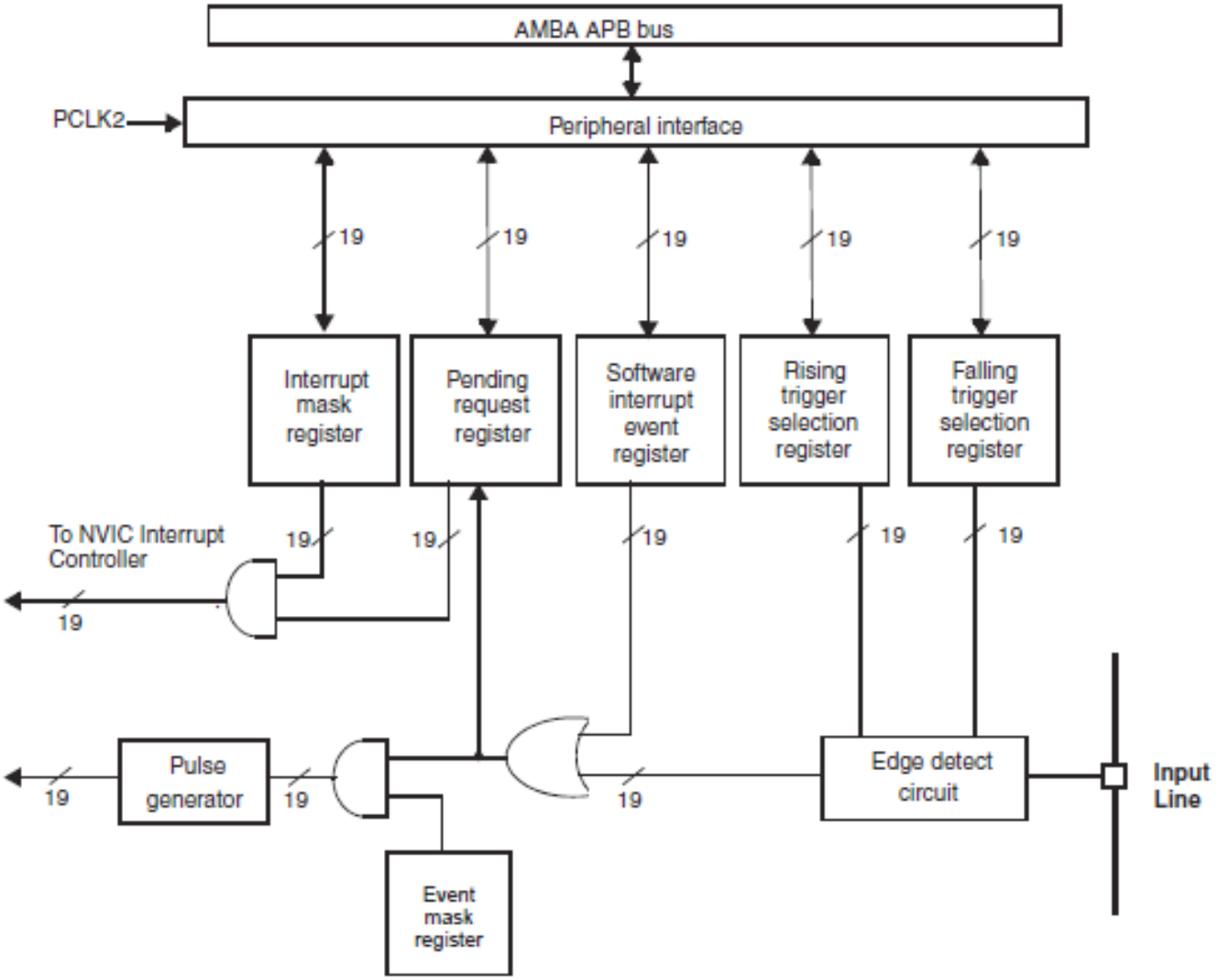
Position	Priority	Type of priority	Acronym	Description	Address
19	26	settable	CAN1_TX	CAN1 TX interrupts	0x0000_008C
20	27	settable	CAN1_RX0	CAN1 RX0 interrupts	0x0000_0090
21	28	settable	CAN1_RX1	CAN1 RX1 interrupt	0x0000_0094
22	29	settable	CAN1_SCE	CAN1 SCE interrupt	0x0000_0098
23	30	settable	EXTI9_5	EXTI Line[9:5] interrupts	0x0000_009C
24	31	settable	TIM1_BRK	TIM1 Break interrupt	0x0000_00A0
25	32	settable	TIM1_UP	TIM1 Update interrupt	0x0000_00A4
26	33	settable	TIM1_TRG_COM	TIM1 Trigger and Commutation interrupts	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^2C1 event interrupt	0x0000_00BC
32	39	settable	I2C1_ER	I^2C1 error interrupt	0x0000_00C0
33	40	settable	I2C2_EV	I^2C2 event interrupt	0x0000_00C4
34	41	settable	I2C2_ER	I^2C2 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	RTCAlarm	RTC alarm through EXTI line interrupt	0x0000_00E4
42	49	settable	OTG_FS_WKUP	USB On-The-Go FS Wakeup through EXTI line interrupt	0x0000_00E8
-	-	-	-	Reserved	0x0000_00EC - 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	TIM6 global interrupt	0x0000_0118
55	62	settable	TIM7	TIM7 global interrupt	0x0000_011C

Position	Priority	Type of priority	Acronym	Description	Address
56	63	settable	DMA2_Channel1	DMA2 Channel1 global interrupt	0x0000_0120
57	64	settable	DMA2_Channel2	DMA2 Channel2 global interrupt	0x0000_0124
58	65	settable	DMA2_Channel3	DMA2 Channel3 global interrupt	0x0000_0128
59	66	settable	DMA2_Channel4	DMA2 Channel4 global interrupt	0x0000_012C
60	67	settable	DMA2_Channel5	DMA2 Channel5 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
65	72	settable	CAN2_RX1	CAN2 RX1 interrupt	0x0000_0144
66	73	settable	CAN2_SCE	CAN2 SCE interrupt	0x0000_0148
67	74	settable	OTG_FS	USB On The Go FS global interrupt	0x0000_014C

External Interrupt controller (EXI)

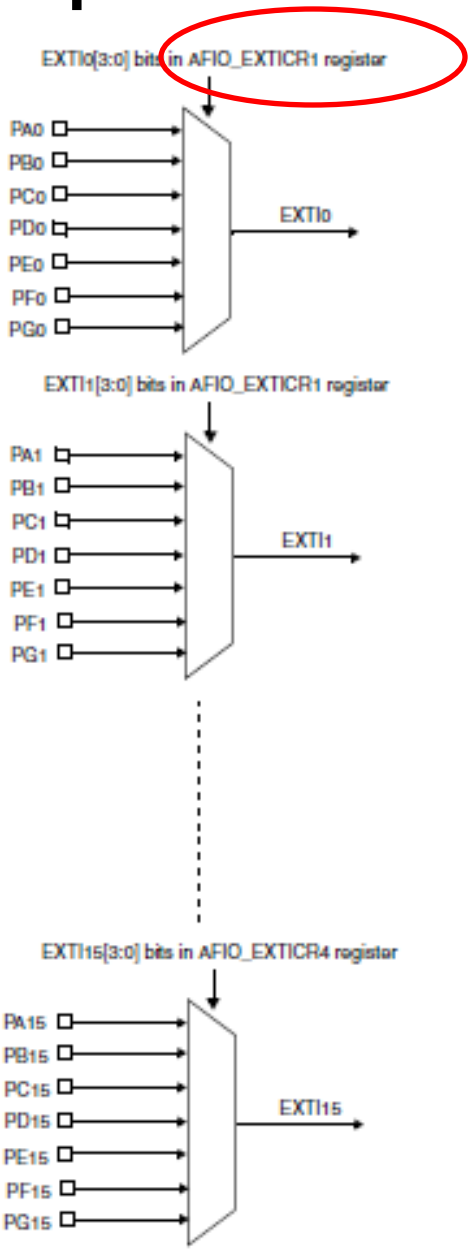
- Consists of up to 20 edge detectors
- Each input line can be independently configured to select the type (pulse or pending) and trigger event (rising or falling edge)
- Each line can masked independently

External interrupt controller block diagram



External interrupt / GPIO mapping

AFIO = Alternate Function I/O

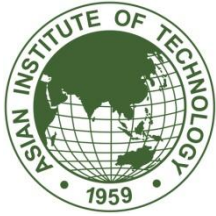


NVIC setting

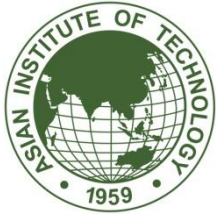
```
NVIC_InitTypeDef NVIC_InitStructure;
 NVIC_PriorityGroupConfig(…);
 NVIC_InitStructure.NVIC_IRQChannel = …;
 NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = ;
 NVIC_InitStructure.NVIC_IRQChannelSubPriority = …;
 NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
 NVIC_Init(&NVIC_InitStructure);…
```

NVIC_IRQChannel

```
#define WWDG_IRQChannel ((u8)0x00) /* Window WatchDog Interrupt */
#define PVD_IRQChannel ((u8)0x01) /* PVD through EXTI Line detection Interrupt */
#define TAMPER_IRQChannel ((u8)0x02) /* Tamper Interrupt */
#define RTC_IRQChannel ((u8)0x03) /* RTC global Interrupt */
#define FLASH_IRQChannel ((u8)0x04) /* FLASH global Interrupt */
#define RCC_IRQChannel ((u8)0x05) /* RCC global Interrupt */
#define EXTI0_IRQChannel ((u8)0x06) /* EXTI Line0 Interrupt */
#define EXTI1_IRQChannel ((u8)0x07) /* EXTI Line1 Interrupt */
#define EXTI2_IRQChannel ((u8)0x08) /* EXTI Line2 Interrupt */
#define EXTI3_IRQChannel ((u8)0x09) /* EXTI Line3 Interrupt */
#define EXTI4_IRQChannel ((u8)0x0A) /* EXTI Line4 Interrupt */
#define DMA1_Channel1_IRQChannel ((u8)0x0B) /* DMA1 Channel 1 global Interrupt */
#define DMA1_Channel2_IRQChannel ((u8)0x0C) /* DMA1 Channel 2 global Interrupt */
#define DMA1_Channel3_IRQChannel ((u8)0x0D) /* DMA1 Channel 3 global Interrupt */
#define DMA1_Channel4_IRQChannel ((u8)0x0E) /* DMA1 Channel 4 global Interrupt */
#define DMA1_Channel5_IRQChannel ((u8)0x0F) /* DMA1 Channel 5 global Interrupt */
#define DMA1_Channel6_IRQChannel ((u8)0x10) /* DMA1 Channel 6 global Interrupt */
#define DMA1_Channel7_IRQChannel ((u8)0x11) /* DMA1 Channel 7 global Interrupt */
```




```
#define ADC1_2_IRQChannel ((u8)0x12) /* ADC1 et ADC2 global Interrupt */
#define USB_HP_CAN_TX_IRQChannel ((u8)0x13) /* USB High Priority or CAN TX
    Interrupts */
#define USB_LP_CAN_RX0_IRQChannel ((u8)0x14) /* USB Low Priority or CAN RX0
    Interrupts */
#define CAN_RX1_IRQChannel ((u8)0x15) /* CAN RX1 Interrupt */
#define CAN_SCE_IRQChannel ((u8)0x16) /* CAN SCE Interrupt */
#define EXTI9_5_IRQChannel ((u8)0x17) /* External Line[9:5] Interrupts */
#define TIM1_BRK_IRQChannel ((u8)0x18) /* TIM1 Break Interrupt */
#define TIM1_UP_IRQChannel ((u8)0x19) /* TIM1 Update Interrupt */
#define TIM1_TRG_COM_IRQChannel ((u8)0x1A) /* TIM1 Trigger and Commutation
    Interrupt */
#define TIM1_CC_IRQChannel ((u8)0x1B) /* TIM1 Capture Compare Interrupt */
#define TIM2_IRQChannel ((u8)0x1C) /* TIM2 global Interrupt */
#define TIM3_IRQChannel ((u8)0x1D) /* TIM3 global Interrupt */
#define TIM4_IRQChannel ((u8)0x1E) /* TIM4 global Interrupt */
#define I2C1_EV_IRQChannel ((u8)0x1F) /* I2C1 Event Interrupt */
#define I2C1_ER_IRQChannel ((u8)0x20) /* I2C1 Error Interrupt */
#define I2C2_EV_IRQChannel ((u8)0x21) /* I2C2 Event Interrupt */
#define I2C2_ER_IRQChannel ((u8)0x22) /* I2C2 Error Interrupt */
#define SPI1_IRQChannel ((u8)0x23) /* SPI1 global Interrupt */
#define SPI2_IRQChannel ((u8)0x24) /* SPI2 global Interrupt */
```




```
#define USART1_IRQChannel ((u8)0x25) /* USART1 global Interrupt */
#define USART2_IRQChannel ((u8)0x26) /* USART2 global Interrupt */
#define USART3_IRQChannel ((u8)0x27) /* USART3 global Interrupt */
#define EXTI15_10_IRQChannel ((u8)0x28) /* External Line[15:10] Interrupts */
#define RTCAlarm_IRQChannel ((u8)0x29) /* RTC Alarm through EXTI Line Interrupt */
#define USBWakeUp_IRQChannel ((u8)0x2A) /* USB WakeUp from suspend through
    EXTI Line Interrupt */
#define TIM8_BRK_IRQChannel ((u8)0x2B) /* TIM8 Break Interrupt */
#define TIM8_UP_IRQChannel ((u8)0x2C) /* TIM8 Update Interrupt */
#define TIM8_TRG_COM_IRQChannel ((u8)0x2D) /* TIM8 Trigger and Commutation
    Interrupt */
#define TIM8_CC_IRQChannel ((u8)0x2E) /* TIM8 Capture Compare Interrupt */
#define ADC3_IRQChannel ((u8)0x2F) /* ADC3 global Interrupt */
#define FSMC_IRQChannel ((u8)0x30) /* FSMC global Interrupt */
#define SDIO_IRQChannel ((u8)0x31) /* SDIO global Interrupt */
#define TIM5_IRQChannel ((u8)0x32) /* TIM5 global Interrupt */
#define SPI3_IRQChannel ((u8)0x33) /* SPI3 global Interrupt */
#define UART4_IRQChannel ((u8)0x34) /* UART4 global Interrupt */
#define UART5_IRQChannel ((u8)0x35) /* UART5 global Interrupt */
#define TIM6_IRQChannel ((u8)0x36) /* TIM6 global Interrupt */
#define TIM7_IRQChannel ((u8)0x37) /* TIM7 global Interrupt */
```

```
#define DMA2_Channel1_IRQChannel ((u8)0x38) /* DMA2 Channel 1 global Interrupt */
#define DMA2_Channel2_IRQChannel ((u8)0x39) /* DMA2 Channel 2 global Interrupt */
#define DMA2_Channel3_IRQChannel ((u8)0x3A) /* DMA2 Channel 3 global Interrupt */
#define DMA2_Channel4_5_IRQChannel ((u8)0x3B) /* DMA2 Channel 4 and DMA2
    Channel 5 global Interrupt */
```

NVIC priority group

- Priority group will be used to specify the splitting between IRQChannelPreemptionPriority and IRQChannelSubPriority
- The IRQChannelPreemptionPriority is referred as group priority
- The sub-priority resolves the priority within the group
- The lower the number will have the higher priority

NVIC Priority Group

NVIC_PriorityGroup	Description
NVIC_PriorityGroup_0	0 bits for pre-emption priority 4 bits for subpriority
NVIC_PriorityGroup_1	1 bits for pre-emption priority 3 bits for subpriority
NVIC_PriorityGroup_2	2 bits for pre-emption priority 2 bits for subpriority
NVIC_PriorityGroup_3	3 bits for pre-emption priority 1 bits for subpriority
NVIC_PriorityGroup_4	4 bits for pre-emption priority 0 bits for subpriority

Setting up Interrupt Vector Table

```
// include file stm32f10x_nvic
#define NVIC_VectTab_RAM             ((u32)0x20000000)
#define NVIC_VectTab_FLASH           ((u32)0x08000000)

// sourcefile stm32f10x_nvic  Sets the vector table location and Offset.
// - NVIC_VectTab: specifies if the vector table is in RAM or
//    Offset: Vector Table base offset field. must be a multiple of 0x100.
void NVIC_SetVectorTable(u32 NVIC_VectTab, u32 Offset){
  /* Check the parameters */
  assert_param(IS_NVIC_VECTTAB(NVIC_VectTab));
  assert_param(IS_NVIC_OFFSET(Offset));
  SCB->VTOR = NVIC_VectTab | (Offset & (u32)0x1FFFFF80);
}

void NVIC_Configuration(void){
     NVIC_SetVectorTable(NVIC_VectTab_FLASH, 0x0);
}
```

Example of code

```
int main(){
  RCC_setup();
  GPIO_setup();
  EXTI_setup();
  NVIC_setup();
}
```

```
void NVIC_setup(){
  NVIC_InitTypeDef NVIC-InitStructure;
  NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);  // configure one bit for
     preemption priority
  // Enable the EXT0 Interrupt
  NVIC_InitStructure.NVIC_IRQChannel = EXTI0_IRQChannel;
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
  NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
  NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
  NVIC_Init(&NVIC_InitStructure);

  // Enable the EXT13 interrupt
  NVIC_InitStructure.NVIC_IRQChannel = EXTI15_10_IRQChannel;
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
  NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
  NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
  NVIC_Init(&NVIC_InitStructure);
}
```

```
 void EXTI_setup()[
  // Enable AFIO for EXT1 module
  EXTI_InitTypeDef EXTI_InitStructure;
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_AFIO,ENABLE);
  // Configure EXT1 line 0 to generate an interrupt on falling edge
    GPIO_EXTLineConfig(GPIO_PortSourceGPIOA, GPIO_PinSource0);
    EXTI_InitStructure.EXTI_Line = EXTI_Line0;
    EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;&
    EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling;
    EXTI_InitStructure.EXTI_LineCmd = ENABLE;
    EXTI_Init(&EXTI_InitStructure);

// Configure EXT1 line 0 to generate an interrupt on falling edge
    GPIO_EXTLineConfig(GPIO_PortSourceGPIOA, GPIO_PinSource13);
    EXTI_InitStructure.EXTI_Line = EXTI_Line13;
    EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;&
    EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling;
    EXTI_InitStructure.EXTI_LineCmd = ENABLE;
    EXTI_Init(&EXTI_InitStructure);
}
```

```
// Function ISR line 0
void EXT0_IRQHandler(void){
  if(EXTI_GetITStatusEXTI_Line0) != RESET){
     GPIO_WriteBit(GPIOC, GPIO_Pin_6,(BitAction)(1-
     GPIO_ReadOutputDataBit(GPIOC,GPIO_Pin_6)));
     EXTI_Clear(PendingBit(EXTI_Line0);
  }
}

// Function ISR line 10 to 15
void EXT15_10_IRQHandler(void){
  if(EXTI_GetITStatusEXTI_Line13) != RESET){
     GPIO_WriteBit(GPIOC, GPIO_Pin_9,(BitAction)(1-
     GPIO_ReadOutputDataBit(GPIOC,GPIO_Pin_9)));
     EXTI_Clear(PendingBit(EXTI_Line13);
  }
}
```

Questions?