EXTI & NVIC | STM32

Manual

CHAPTER 1 — INTRODUCTION TO EXTI

1.1 What is EXTI?

EXTI = **External Interrupt/Event Controller**
It allows **any GPIO pin** to act as an interrupt source.

Functions:

- Detect **Rising Edge / Falling Edge / Both**
- Generate **Interrupt** or **Event**
- Route input from **GPIO pins** to **interrupt lines (0–15)**
- Provide separation between GPIO logic and NVIC

EXTI prevents the CPU from **polling** a pin.
Instead, hardware detects changes and signals the NVIC.

CHAPTER 2 — EXTI INTERNAL BLOCK DIAGRAM

```
GPIO Pin → AFIO → EXTI Line → Edge Detector → Pending Register → NVIC → ISR
```

Detailed flow:

1. **GPIO Input**
   - Configured as floating or pull-up/down
2. **AFIO_EXTICR**
   - Selects which GPIO port (A/B/C/D) is connected to EXTI line
3. **EXTI_RTSR / EXTI_FTSR**
   - Rising or falling edge detectors
4. **EXTI_IMR**
   - Unmask interrupt line (enable)
5. **EXTI_PR**
   - When event occurs → PR flag set → NVIC is notified
6. **NVIC**
   - Looks up ISR vector
   - Pushes CPU state
   - Executes ISR

CHAPTER 3 — REGISTERS

3.1 AFIO -> EXTICR

Maps GPIO pin to EXTI line.

Register	Meaning
EXTICR1	EXTI0–3
EXTICR2	EXTI4–7
EXTICR3	EXTI8–11
EXTICR4	EXTI12–15

Each nibble selects port:

```
0000 = PA
0001 = PB
0010 = PC
...
```

3.2 EXTI Registers

IMR — Interrupt Mask Register

- 1 → enable interrupt
- 0 → block interrupt

EMR — Event Mask Register

- For events (not used much)

RTSR — Rising Edge Trigger

- 1 → interrupt on rising edge of that line

FTSR — Falling Edge Trigger

- 1 → interrupt on falling edge

PR — Pending Register

- Shows which interrupt happened
- You MUST write 1 to clear the flag

3.3 NVIC Registers

Though hidden in CMSIS, important registers:

Register	Purpose
ISER	Enable IRQ number
ICER	Disable IRQ
ISPR	Set pending
ICPR	Clear pending
IPR	Interrupt priority

CHAPTER 4 — LAB SETUP

📌 Goal:

Use **PA6** as input (button)
Use **PB0** as LED output
LED toggles **every time PA6 is pressed**.

Proteus or real hardware both supported.

CHAPTER 5 — HARDWARE CONFIGURATION

PA6 → EXTI6

- Input
- Pull-up enabled
- Falling edge → switch press

PB0

- Output push-pull
- LED connected to PB0

CHAPTER 6 — FULL BARE-METAL CODE

```
#include "stm32f1xx.h"

void EXTI6_Init(void);
void GPIO_Init(void);

int main(void)
{
    GPIO_Init();
    EXTI6_Init();

    while(1)
    {
        // CPU is free, waiting for interrupt
    }
}

void GPIO_Init(void)
{
    RCC->APB2ENR |= RCC_APB2ENR_IOPAEN | RCC_APB2ENR_IOPBEN |
RCC_APB2ENR_AFIOEN;

    // PA6 input with pull-up
    GPIOA->CRL &= ~(0xF << 24);
    GPIOA->CRL |=  (0x8 << 24);   // Input + Pull-up/down
    GPIOA->ODR |=  (1 << 6);      // Enable pull-up

    // PB0 output push-pull 2MHz
    GPIOB->CRL &= ~(0xF << 0);
    GPIOB->CRL |=  (0x2 << 0);
}

void EXTI6_Init(void)
{
    // Map PA6 → EXTI6: EXTICR2 (EXTI4-7)
    AFIO->EXTICR[1] &= ~(0xF << 8);  // PA = 0000

    EXTI->IMR  |=  (1 << 6);  // Unmask EXTI6
    EXTI->FTSR |=  (1 << 6);  // Trigger on falling edge

    NVIC_EnableIRQ(EXTI9_5_IRQn);   // Enable IRQ in NVIC
}

void EXTI9_5_IRQHandler(void)
{
    if (EXTI->PR & (1 << 6))
    {
        EXTI->PR |= (1 << 6); // Clear interrupt

        GPIOB->ODR ^= (1 << 0); // Toggle LED
    }
}
```

CHAPTER 7 — DETAILED CODE EXPLANATION

GPIO Config

- `GPIOA->CRL |= 0x8 << 24`
  - MODE = 00
  - CNF = 10 → Input with pull-up/down

AFIO → EXTI

- Each EXTI line can connect to only ONE PORT
- EXTICR routing required

Trigger Selection

- **FTSR** chosen because falling edge matches button press

NVIC IRQ Selection

PA6 → EXTI6 → belongs to group EXTI5–9
Hence ISR is:

```
void EXTI9_5_IRQHandler()
```

CHAPTER 8 — LAB TEST PROCEDURE

1. Start your STM32 board
2. Set PA6 to HIGH (internally pull-up)
3. Press the button → PA6 goes LOW
4. EXTI detects FTSR → sets PR
5. NVIC calls ISR
6. LED toggles

CHAPTER 9 — TROUBLESHOOTING

Problem	Fix
Interrupt never fires	Check AFIO clock, EXTICR mapping
LED toggles continuously	Switch bouncing → add software debounce
ISR never runs	NVIC IRQ not enabled
PR flag not clearing	Must write **1**, not 0
