

Interrupt on STM32F103 MCU

The external interrupt 0 is enabled and the mask bet is set

The screenshot displays the STM32CubeIDE interface with several windows open to configure an external interrupt on the STM32F103 MCU.

General Purpose I/O A (GPIOA) Configuration:

- Pin:** PA.0
- CNF:** Floating Input
- MODE:** Input
- CNF:** Floating Input
- Configuration & Mode Settings:**
 - GPIOA_CRH: 0x44244444
 - GPIOA_CRL: 0x44444444
 - GPIOA_ODR: 0x00000000
 - GPIOA_LCKR: 0x00000000
 - Pins: 0x00000000
- Settings:** Clock Enabled

Nested Vectored Interrupt Controller (NVIC) Configuration:

- Source:** EXTI Line0 Interrupt (EXTI0)
- Priority:** 0
- Interrupt Control & State:**
 - INT_CTRL_ST: 0x00000000
 - VECTACTIVE: 0x00
 - RETTBASE: 0x00
 - ISRPREEMPT: 0x00
- Application Interrupt & Reset Control:**
 - AIRC: 0xFA050000
 - PRIGROUP: 0: 7.1
 - VECTRESET: 0x00
 - VECTCLRACTIVE: 0x00
 - SYSRESETREQ: 0x00
 - ENDIANNESS: 0x00
- Vector Table Offset:**
 - VTO: 0x00000000
 - TBLOFF: 0x00000000
 - TBLBASE: 0x00
- Software Interrupt Trigger:**
 - SW_TRIG_INT: 0x00000000
 - INTID: 0x00

External Interrupts (EXTI) Configuration:

Line	Source	Port	Mask	Event	Pend	RTsig	FTrig	SwIntr
0	PA.0	0	1	0	0	1	0	0
1	PA.1	0	0	0	0	0	0	0
2	PA.2	0	0	0	0	0	0	0
3	PA.3	0	0	0	0	0	0	0
4	PA.4	0	0	0	0	0	0	0
5	PA.5	0	0	0	0	0	0	0
6	PA.6	0	0	0	0	0	0	0
7	PA.7	0	0	0	0	0	0	0

C Code Snippets:

```
103: while(1);
104:
105:
106: }

97: {
98: clock_init();
99: GPIO_init();
100: EXTI_init();
101:
102: while(1);
103:
104:
105: }

109: // INTERRUPT HANDLER
110: void EXTI0_IRQHandler(void)
111: {
112: //pin0 in GPIOA is pres
113: GPIOA_ODR ^= (1 << 13);
114:
115: //clear pending request
116: EXTI_PR |= (1 << 0);
117: }
```

Registers Window:

- Core:** R0 (0x20), R1 (0x20), R2 (0xE0), R3 (0x00), R4 (0x20), R5 (0x00), R6 (0x00), R7 (0x20), R8 (0x00), R9 (0x00), R10 (0x00), R11 (0x00), R12 (0x00), R13 (SP) (0x20), R14 (LR) (0x08), R15 (PC) (0x08), xPSR (0x61).

Command Window:

```
Load "D:\\embedded diploma\\REPO\\embedded-systems-diploma\\Unit_6_MCU_for_Interrupt\\STM32F103CubeIDE\\STM32F103CubeIDE\\Project\\main.c"
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakPoint
```

When the button is pressed, the pending flag is set and the PC went to the ISR code (Defined in startup)

The screenshot shows the Keil uVision IDE with the following windows open:

- Registers:** Shows the Core registers. R15 (PC) is highlighted at 0x00000000.
- Disassembly:** Shows the assembly code for the interrupt handler. The instruction at address 111 is `GPIO_ODR ^= (1 << 13);`, which is circled in red.
- General Purpose I/O (GPIO):** Shows the configuration for PA0. The `GPIO_ODR` register is 0x00000000, and the `EXTI_PR` register is 0x00000000.
- Nested Vectored Interrupt Controller (NVIC):** Shows the interrupt configuration for EXTI0. The `EXTI0` interrupt is enabled and pending.

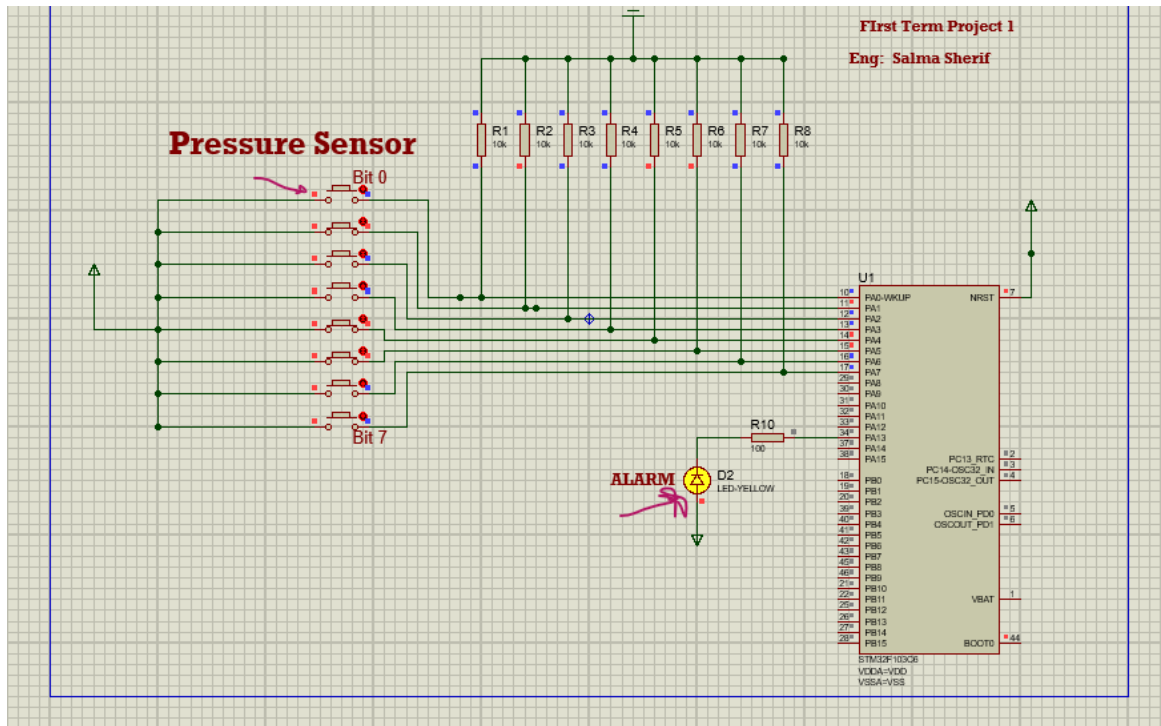
Finally, the LED is toggled, and the pending flag is cleared

The screenshot shows the Keil uVision IDE with the following windows open:

- Registers:** Shows the Core registers. R15 (PC) is highlighted at 0x00000001.
- Disassembly:** Shows the assembly code for the interrupt handler. The instruction at address 111 is `GPIO_ODR ^= (1 << 13);`, which is circled in red.
- General Purpose I/O (GPIO):** Shows the configuration for PA0. The `GPIO_ODR` register is 0x00000001, and the `EXTI_PR` register is 0x00000000.
- Nested Vectored Interrupt Controller (NVIC):** Shows the interrupt configuration for EXTI0. The `EXTI0` interrupt is no longer pending.

ON Proteus:

-before Pressing the button



After pressing the button, the LED is toggled, Interrupt happened

