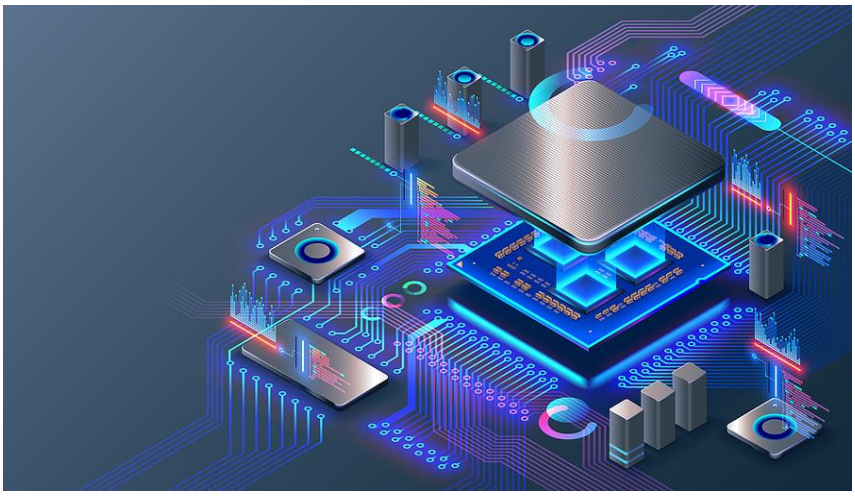


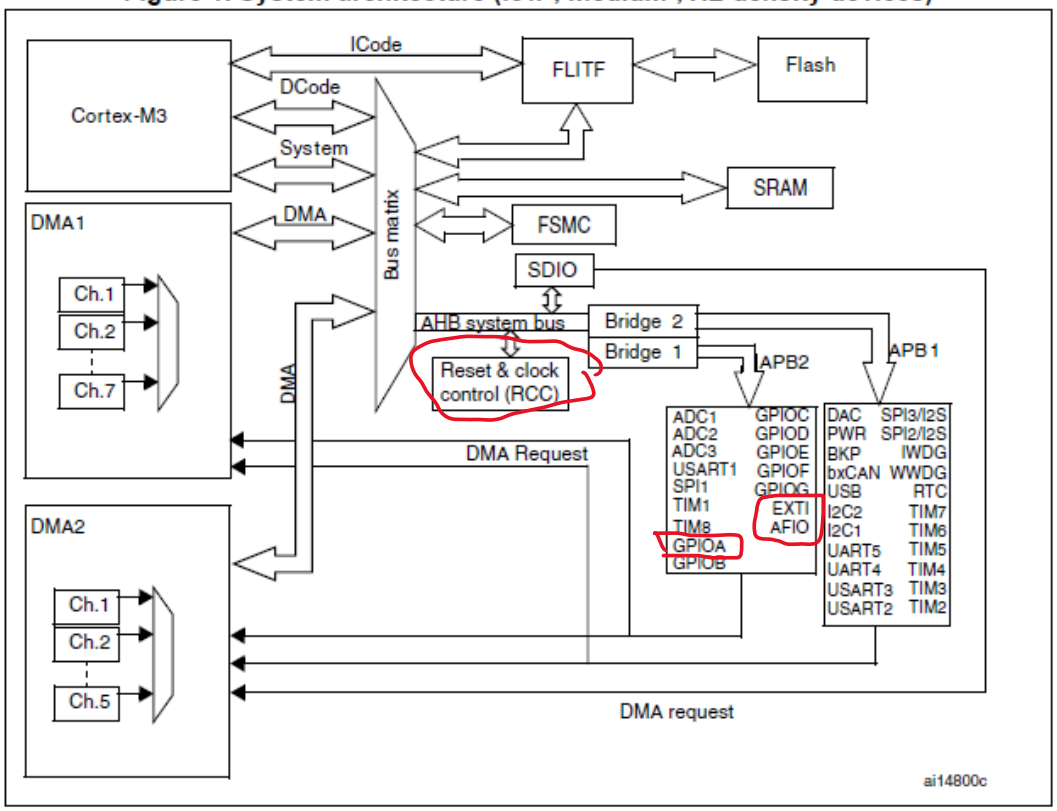
LED CONTROLED BY EXTI PIN

ABSTRACT

This report will explain the logical sequence to enable the EXTI to control an LED.

first of all, we must adjust the system clock of our MCU and then provide this clock to the APB2 Bus to make GPIOA, AFIO, and EXTI peripherals.

Figure 1. System architecture (low-, medium-, XL-density devices)

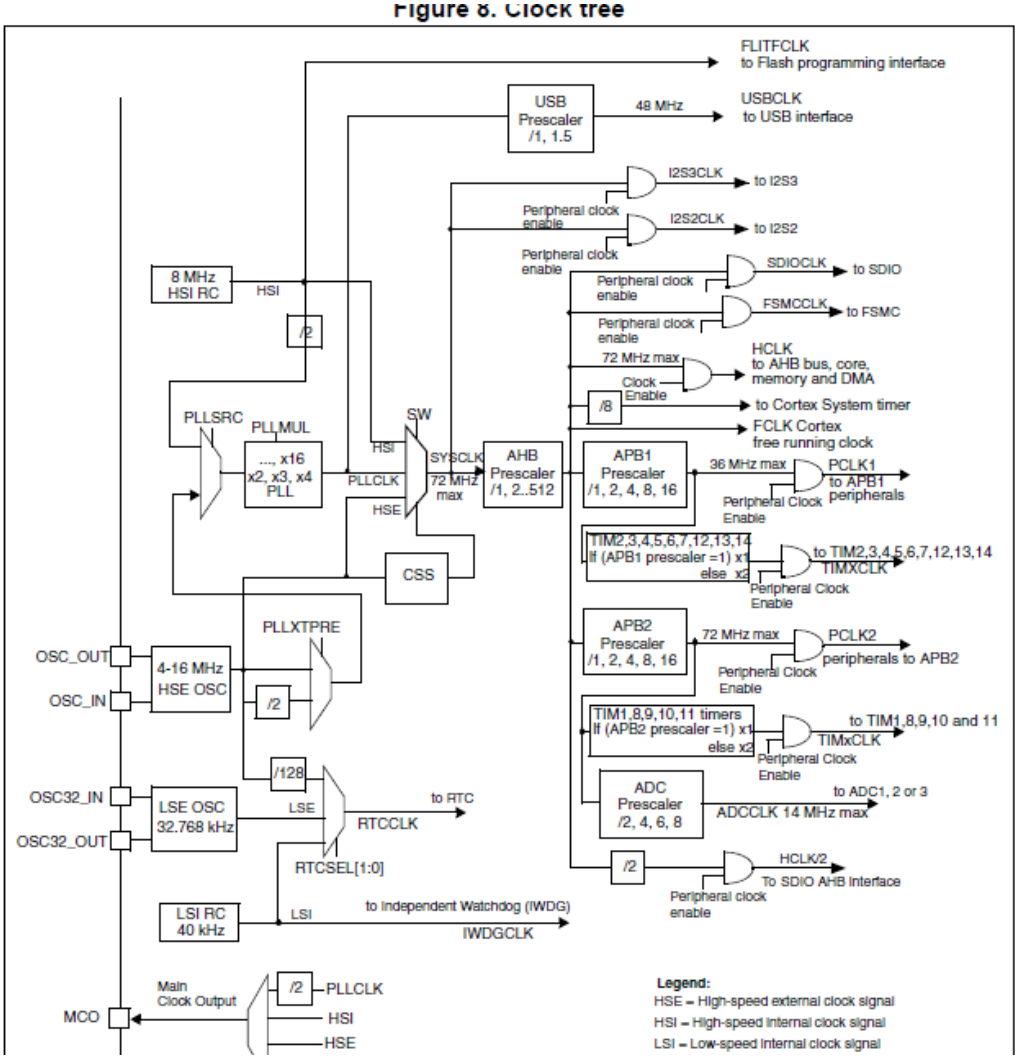


- RCC

I want to provide a 16 MHz to APB2 Bus (GPIOA & EXTI)

Let’s do it in the following diagram.

Figure 8. Clock tree



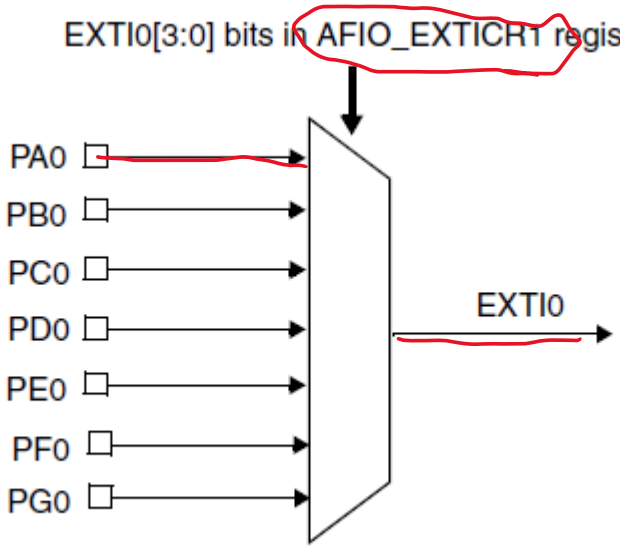
- Adjust pin GPIOA0 to be input pull-down.
- Adjust pin GPIOA13 to be output.

Table 33. Other IOs

Pins	Alternate function	GPIO configuration
TAMPER-RTC pin	RTC output	Forced by hardware when configuring the BKP_CR and BKP_RTCCR registers
	Tamper event input	
MCO	Clock output	Alternate function push-pull
EXTI input lines	External input interrupts	Input floating / input pull-up / input pull-down

AFIO Alternative Function Input output peripheral

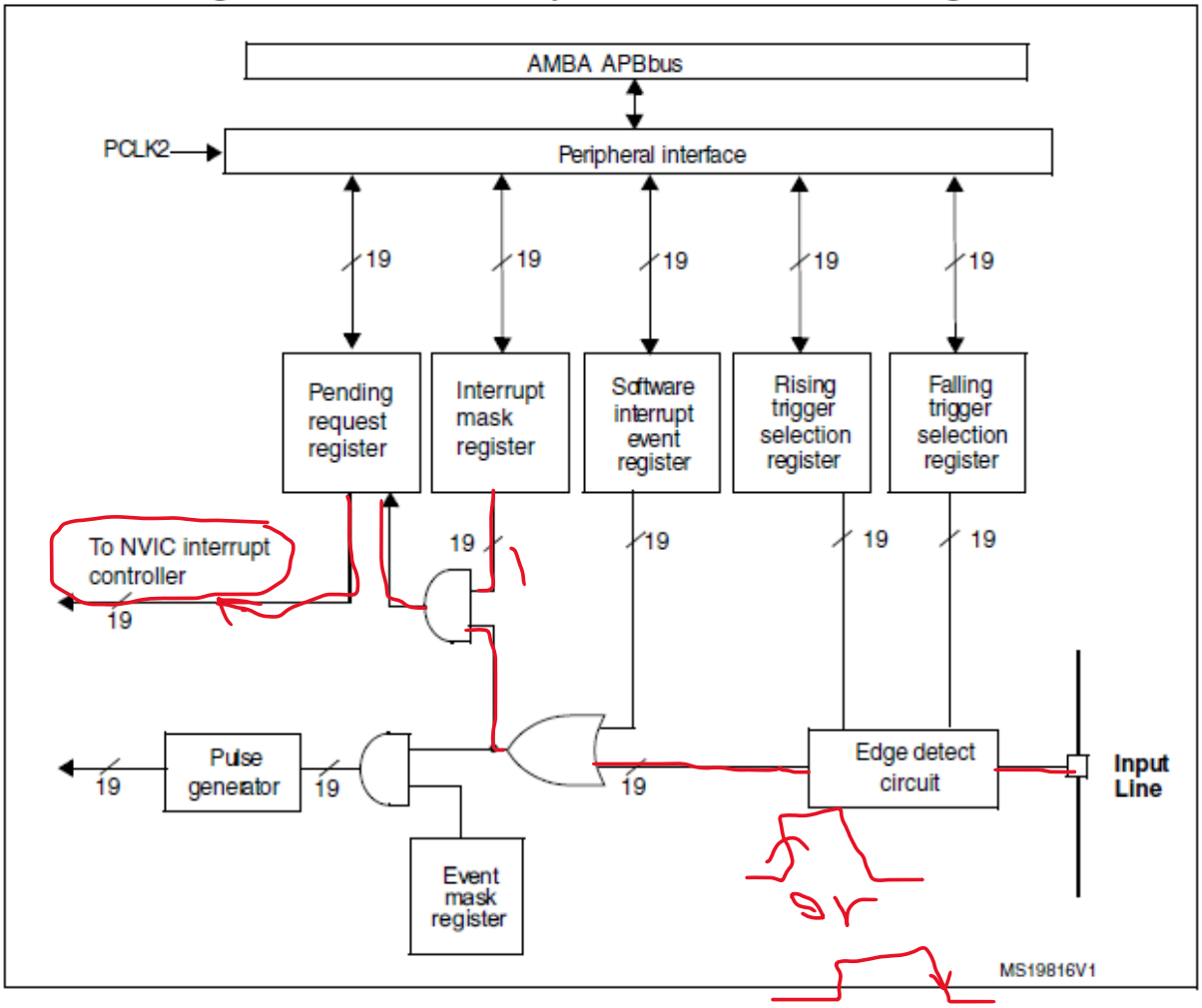
Attach PA0 to EXTI0 from AFIO peripheral



EXTI peripheral

- Adjusting the triggered sensitivity (rising edge or falling edge).
- Enable the interrupt from interrupt mask enable register.

Figure 20. External interrupt/event controller block diagram



NVIC peripheral

Enable interrupt related to EXTI0 in NVIC controller.

Table 8-8 Interrupt Set-Enable Register bit assignments

Bits	Field	Function
[31:0]	SETENA	Interrupt set enable bits. For write operation: 1 = enable interrupt 0 = no effect. For read operation: 1 = enable interrupt 0 = disable interrupt Writing 0 to a SETENA bit has no effect. Reading the bit returns its current enable state. Reset clears the SETENA fields.

EXTI0 is on index 6

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

Finally, after adjusting RCC, GPIO, AFIO, EXTI peripherals now we can write the application.

main.c

```c

#include "STD_TYPES.h"
#include "RCC.h"
#include "GPIO.h"
#include "EXTI.h"
#include "NVIC.h"

void EXTI0_IRQHandler();
volatile int x=0;
int main(void)
{


    RCC_init();
    GPIO_init();
    EXTI_init();
    NVIC_init();


    /* Loop forever */
    while(1);
    return 0;
}
void EXTI0_IRQHandler()
{
            for(x=0;x<100000;x++);
    Toggle_pin_void(GPIO_uint8_PORTC,GPIO_uint8_PIN13);
    SET_BIT(EXTI_PR_uint32_REG,0);

}

```

Simulation

- Toggling pin C13 duing rising edge on A0