

```

1  #include "stm32f10x.h"
2
3
4  int main() {
5
6      //-----BASLANGIC-----
7
8      RCC->APB2ENR |= (1<<4) | (1<<3); //PortB ve PortC Enable durumuna getirildi
9
10     GPIOB->CRH &=~(0xf<<16); //16,19 bitleri 0 oldu
11     GPIOB->CRH |= (1<<19); //19.biti 1 yaparak 12 numarali pin giris oldu
12     GPIOB->ODR &=~(1<<12); //PortB 12 numarali pin pulldown oldu
13     GPIOC->CRH &=~(0xf<<20); //20,23 bitleri 0 oldu
14     GPIOC->CRH |= (1<<21); //21.biti 1 yaparak 13 numarali pin cikis oldu
15
16     while(1){
17         //IDR(Input Data Register): Ilgili pinden gelen veriyi okur
18         if(GPIOB->IDR & (1<<12)) //PortB 12 numarali pin 1 ise kod calisacaktır.
19         {
20             GPIOC->ODR &=~(1<<13); //led yanar
21         }
22         else //PortB 12 numarali pin 0 ise kod calisacaktır.
23         {
24             GPIOC->ODR |= (1<<13); //led soner
25         }
26     }
27
28     //-----BITIS-----
29
30     //-----BASLANGIC-----
31
32     // Yukaridaki kodda input degeri sürekli kontrol edilerek led yakilir veya söndürülür.
33     // Sürekli kontrol etmek yerine 12 numarali pin 1 oldugunda INTERRUPT olusturarak ledi yakalim.
34
35     // Interrupt biriminin devreye girmesi için AFIO mutlaka Enable olmalıdır.
36     // ->AFIO(Alternate-Function I/Os)
37     // ->ETICR[x](External Interrupt Configuration Register);
38     // Hangi portun hangi pinini ayarlamak için kullanilir. (x, 0'dan 3'e kadar deger alabilir.)
39     // Her x degeri, 4 pin için ayarlama yapabiliyor.
40     // ->NVIC(Nested Vectored Interrupt Controller)
41     // ->ISER(Interrupt Set Enable Register)
42     // ->IMR(Interrupt Mask Register)
43     // ->PR(Pending Register)
44
45     RCC->APB2ENR |= (1<<4) | (1<<3) | 1; //PortB, PortC ve AFIO Enable durumuna getirildi
46     GPIOB->CRH &=~(0xf<<16); //16,19 bitleri 0 oldu
47     GPIOB->CRH |= (1<<19); //19.biti 1 yaparak 12 numarali pin giris oldu
48     GPIOB->ODR &=~(1<<12); //PortB 12 numarali pin pulldown oldu
49     GPIOC->CRH &=~(0xf<<20); //20,23 bitleri 0 oldu
50     GPIOC->CRH |= (1<<21); //21.biti 1 yaparak 13 numarali pin cikis oldu
51     GPIOC->ODR |= (1<<13); //led soner
52
53     AFIO->EXTICR[3] |= 1; //PortB 12 numarali pini interrupt girisi olarak ayarladik.
54     EXTI->FTSR |= (1<<12); //Butondan elimizi çektigimizde interrupt olusacak.
55     EXTI->IMR |= (1<<12); //12 numarali pin ile calistigimiz için 12. biti 1 yaptik
56     NVIC->ISER[1] |= (1<<8); //8.biti yani EXTI15_10 interrupt'ini 1 yaptik
57
58     while(1){
59
60     }
61
62     //-----BITIS-----
63
64 }
65
66 void EXTI15_10_IRQHandler() { //10,15 pinlerinden interrupt geldiginde bu fonksiyon cagrilir.
67     if(EXTI->PR & (1<<12)){ //Interrupt'in 12 numarali pinden geldigini kontrol ediyoruz
68         GPIOC->ODR &=~(1<<13); //led yanar
69         EXTI->PR |= (1<<12); // interrupt'i tekrar kullanabilmek için tekrar 1 degerini atadik.
70     }
71
72
73 }
74
75

```