

```

1  #include "stm32f10x.h"
2
3  int main() {
4
5  //-----ACIKLAMA-----
6  /*
7      TIM1
8      Timer'lar genel olarak mikrodenetleyicilerde, main fonksiyonu çalisirken belirli periyotlarda
9      timer alt programina inerek oradaki islemleri yapmasini saglar. Periyodik olarak ana program ve
10     timer alt programi sürekli olarak çalisir.
11
12     Interrupt Ozellikleri
13     >Update Event: Sayici tasma yada sifira gelme durumunda olusur.
14     >Trigger Event: Sayicinin baslamasi veya durmasi durumunda olusur.
15
16     TIM1 Registerlari
17     >Counter Register(TIMx_CNT): Sayma sirasinda o anki degeri tutar
18     >Prescaler Register(TIMx_PSC): Gelen frekansi böler
19     >Auto-Reload Register(TIMx_ARR): Saydirmak istedigimiz deger burada tutulur
20     >Status Register(TIMx_SR): Update Event oldugunda calisiyor
21     >DMA Interrupt Enable Register(TIMx_DIER): 0.bit'i enable yaptigimizda timer update oldugunda
22     interrupt olusturur.
23     >Capture/Compare Mode Register 1(TIMx_CCMR1): PWM çikisinin hangi modda çalisacagi belirlenir
24  */
25  //-----
26
27  //-----BASLANGIC-----
28
29      RCC->APB2ENR |= (1<<11);    //Timer Enable
30      TIM1->ARR = 7200;           //7200e kadar sayacak
31      TIM1->PSC = 0;              //72MHz hizinda sayacak, 7200/72000000=100 mikro saniye
32      TIM1->CR1 |= 1;             //Timer Start
33
34      while(!(TIM1->SR & 1)){ }    //Update olmadigi sürece döngü devam edecek
35      TIM1->SR = 0;               //Update flag sifirlanir
36
37      while(1){ }
38  //-----BITIS-----
39
40  //-----BASLANGIC-----
41
42      //TIM1 kullanarak 1Hz'lik sinyal üretelim.
43      RCC->APB2ENR |= ((1<<11)|(1<<4));    //Timer ve PortC Enable
44      GPIOC->CRH |= ((1<<20)|(1<<21));
45      GPIOC->CRH &= ~(1<<22)|(1<<23));
46
47      TIM1->ARR = 7200;
48      TIM1->PSC = 5000;           //7200/(72000000/5000)=500 milisaniye
49      TIM1->CR1 |= 1;             //Timer Start
50
51      while(1){
52          while(!(TIM1->SR & 1)){ }
53          TIM1->SR &= ~1;
54          GPIOC->ODR ^= (1<<13);
55      }
56  //-----BITIS-----
57
58  //-----BASLANGIC-----
59
60      //TIM1 kullanarak 1Hz'lik sinyal üretelim. Interrupt ile
61      RCC->APB2ENR |= ((1<<11)|(1<<4));
62      GPIOC->CRH |= ((1<<20)|(1<<21));
63      GPIOC->CRH &= ~(1<<22)|(1<<23));
64
65      TIM1->ARR = 7200;
66      TIM1->PSC = 5000;
67      TIM1->CR1 |= 1;
68      TIM1->DIER |= 1;           //Update oldugunda interrupt çalisacak
69      NVIC->ISER[0] |= (1<<25);  //TIM1 için interrupt aktif hale getirdik
70
71      while(1){ }
72  //-----BITIS-----
73
74
75
76
77

```

```
78 //-----BASLANGIC-----
79 //Timer ile pwm sinyali olusturma
80 RCC->APB2ENR |= (1<<11) | (1<<2) | 1;
81 GPIOA->CRH &= ~(1<<2);
82 GPIOA->CRH |= (1<<3)|3;
83
84 TIM1->CCMR1 |= (6<<4); //PortA 8.pin için model ayarlandı
85 TIM1->CCER |= 1; //Kanal-1 çıkis olarak ayarlandı
86 TIM1->BDTR |= (1<<15); //Main output enable
87 TIM1->ARR = 3600; //20KHz
88 TIM1->PSC = 0;
89 TIM1->CCR1 = 360; //Duty Cycle %10
90 TIM1->CR1 |= 1; //Timer Start
91
92 while(1){ }
93 //-----BITIS-----
94
95
96 }
97
98
99 void TIM1_UP_IRQHandler(){ // TIM1 Update Interrupt'i, sayma islemi bittiginde devreye girer
100     if(TIM1->SR &1){
101         TIM1->SR &=~1;
102         GPIOC->ODR ^=(1<<13);
103     }
104 }
105
106
```