

donanımsal yöntemleri de mevcuttur. En basit donanımsal önlem butona paralel 100nf değerinde bir kondansatör bağlamaktır.

Bu uygulamada sol menü tuşuna basıldığında ledler sola doğru ilerleyerek yanmakta, sağ menü tuşuna basıldığında ledler sağ tarafa doğru sönerek gerilemekte, yukarı menü tuşuna basıldığında power led yanmakta, aşağı menü tuşuna basıldığında power led sönmekte, ve ok tuşuna bir kez basıldığında buzzer ötmekte, ikinci kez basıldığında susmaktadır.

Uygulama Kodları:

```
1 ////////////////////////////////////////////////////////////////////
2 //*****//
3 //      Menü buton setinin ve çeşitli      //
4 //      elemanların kullanımı.            //
5 //      MikroC v6.2 - STM32F407VG        //
6 //      ARMapp-18 Deney Seti için yazılmıştır //
7 //*****//
8 //      http://elektrovadi.com           //
9 //      http://mikrodunya.wordpress.com   //
10 ////////////////////////////////////////////////////////////////////
11
12 // Menü butonları tanımlanıyor.
13 #define yukari      GPIOC_IDR.B4          // 'yukarı' butonu PC4 pininin değer okuma registerına tanımlanıyor.
14 #define asagi      GPIOB_IDR.B14         // 'asagi' butonu PB14 pininin değer okuma registerına tanımlanıyor.
15 #define sol        GPIOB_IDR.B12         // 'sol' butonu PB12 pininin değer okuma registerına tanımlanıyor.
16 #define sag        GPIOB_IDR.B13         // 'sag' butonu PB13 pininin değer okuma registerına tanımlanıyor.
17 #define orta       GPIOB_IDR.B8          // 'orta' butonu PB8 pininin değer okuma registerına tanımlanıyor.
18
19 #define buzzer      GPIOA_ODR.B8          // 'buzzer' pini PA8 pininin çıkış registerına tanımlanıyor.
20 #define powerLed    GPIOB_ODR.B6         // 'powerLed' pini PB6 pininin çıkış registerına tanımlanıyor.
21
22 char ledler=0,buzzerSayac=0;            // char tipinde 'ledler' ve 'buzzerSayac' isminde iki değişken tanımlanıyor.
23
24 void main()                              // Ana program bloğu.
25 {
26     GPIO_Digital_Input(&GPIOB_BASE,_GPIO_PINMASK_8 | // 'B' Portunun 8,12,13 ve 14. pinleri dijital
27                       _GPIO_PINMASK_12 |           // giriş olarak ayarlanıyor. (Menü butonları)
28                       _GPIO_PINMASK_13 |
29                       _GPIO_PINMASK_14);
30
31     GPIO_Digital_Input(&GPIOC_BASE,_GPIO_PINMASK_4 ); // 'C' Portunun 4. pini dijital giriş olarak ayarlanıyor. (Menü butonu)
32
33     GPIO_Digital_Output(&GPIOA_BASE,_GPIO_PINMASK_8); // 'A' Portunun 8. pini dijital çıkış olarak ayarlanıyor. (Buzzer)
34
35     GPIO_Digital_Output(&GPIOB_BASE,_GPIO_PINMASK_6); // 'B' Portunun 6. pini dijital çıkış olarak ayarlanıyor. (Power Led)
36
37     GPIO_Digital_Output(&GPIOE_BASE,_GPIO_PINMASK_0 | // 'E' Portunun 0-7. pinleri dijital çıkış olarak ayarlanıyor.
38                       _GPIO_PINMASK_1 |
39                       _GPIO_PINMASK_2 |
40                       _GPIO_PINMASK_3 |
41                       _GPIO_PINMASK_4 |
42                       _GPIO_PINMASK_5 |
43                       _GPIO_PINMASK_6 |
44                       _GPIO_PINMASK_7);
45     buzzer=0;                               // Buzzer susturuluyor.
46     powerLed=0;                             // Power Led söndürülüyor.
47     GPIOE_ODR=ledler;                       // 'ledler' değişkeni 0 olduğu için hepsi söndürülüyor.
48
49     while(1)                                // Sonsuz Döngü
50     {
51         if(yukari)                          // Yukarı butonuna basıldıysa ( GPIOC_IDR.B4=1 ise)
52         {
53             powerLed=1;                     // Power Led'i yak. (GPIOB_ODR.B6=1 yap)
54             delay_ms(100);                  // Buton arkını önlemek için 100 ms bekle.
55             while(yukari);                  // Yukarı tuşuna basılı olduğu müddetçe burada bekle.
56             delay_ms(100);                  // Buton arkını önlemek için 100 ms bekle.
57         }
58
59         if(asagi)                            // Aşağı butonuna basıldıysa ( GPIOB_IDR.B14=1 ise)
60         {
61             powerLed=0;                     // Power Led'i söndür. (GPIOB_ODR.B6=0 yap)
62             delay_ms(100);                  // Buton arkını önlemek için 100 ms bekle.
63             while(asagi);                   // Aşağı tuşuna basılı olduğu müddetçe burada bekle.
64             delay_ms(100);                  // Buton arkını önlemek için 100 ms bekle.
65         }
66
67         if(sol)                              // Sağ butonuna basıldıysa ( GPIOB_IDR.B13=1 ise)
68         {
69             ledler=(ledler<<1)+1;          // ledler değişkenini 1 kez sola kaydır ve 1 ekle.
70             GPIOE_ODR=ledler;              // E portuna ledler değişkenini yükleyerek istenilen ledlerin yanmasını sağla.
71             delay_ms(100);                  // Buton arkını önlemek için 100 ms bekle.
72             while(sol);                     // Sağ tuşuna basılı olduğu müddetçe burada bekle.
73             delay_ms(100);                  // Buton arkını önlemek için 100 ms bekle.
74         }
75     }
76 }
```

```

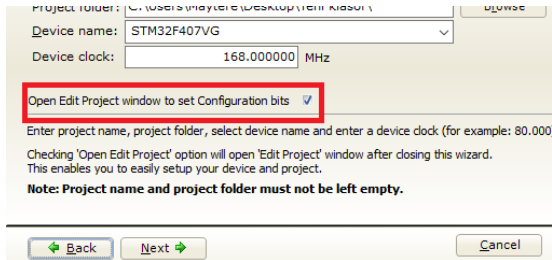
75
76 if(sag) // Sol butonuna basıldıysa ( GPIOB_IDR.B12=1 ise)
77 {
78 if(ledler>0)ledler=(ledler-1)>>1; // Eğer ledler değişkeni 0'dan büyükse 1 eksilt ve 1 kez sağa kaydır.
79 GPIOE_ODR=ledler; // E portuna ledler değişkenini yükleyerek istenilen ledlerin yanmasını sağla.
80 delay_ms(100); // Buton arkını önlemek için 100 ms bekle.
81 while(sag); // Sol tuşuna basılı olduğu müddetçe burada bekle.
82 delay_ms(100); // Buton arkını önlemek için 100 ms bekle.
83 }
84
85 if(orta) // Orta butonuna basıldıysa ( GPIOB_IDR.B8=1 ise)
86 {
87 if(buzzerSayac==0) // Eğer buzzerSayac değişkeni 0 ise
88 {
89 buzzer=1; // Buzzerı öttür. (GPIOA_ODR.B8=1 yap)
90 buzzerSayac++; // buzzerSayac değişkenini 1 arttır.
91 }
92 else if(buzzerSayac==1) // Eğer buzzerSayac=1 ise
93 {
94 buzzer=0; // Buzzerı sustur. (GPIOA_ODR.B8=0 yap)
95 buzzerSayac=0; // buzzerSayac değişkenini 0 yap.
96 }
97 delay_ms(100); // Buton arkını önlemek için 100 ms bekle.
98 while(orta); // orta tuşuna basılı olduğu müddetçe burada bekle.
99 delay_ms(100); // Buton arkını önlemek için 100 ms bekle.
100 }
101 }
102 }



```

Kod kısmında dikkat edilecek olursa butona basıldığında ve bırakıldığında 100 ms'lik beklemler yapılmaktadır. Bu beklemler ark oluşumunu engellemektedir. Diğer önemli bir nokta ise onay butonuna basıldığında buzzerSayac isminde bir değişken arttırılmakta ve bu değişkenin durumuna göre buzzer ötürülüp susturulmaktadır.

İşlem Basamakları:

- 1- MikroC Pro for ARM programını çalıştırınız.
- 2- "Project" sekmesinden "New Project" seçeneğine tıklayınız.
- 3- Açılan ekrandan "Standart Project" seçilerek "Next" butonuna tıklayınız.
- 4- Açılan pencereden proje ismi, projenin kaydedileceği dizin, kullanılacak mikrodenetleyiciyi (STM32F407VG) ve mikrodenetleyici frekansını (168.000000 MHZ) belirleyerek ve "Open Edit Project window to set Configuration bits" seçeneğini işaretleyerek "Next" butonuna tıklayınız.



- 5- Açılan pencerede "Finish" butonuna tıklayarak proje oluşturma aşamasını bitiriniz.
- 6- Açılan pencerede sağ tarafta bulunan Load Scheme butonuna tıklayınız ve açılan pencereden "HSI 16 MHz_PLL 168 MHz Clock Ayarı.cfgsch" dosyasını seçiniz.
- 7- "OK" butonuna tıklayarak proje oluşturma işlemini tamamlayınız.
- 8- Açılan ".c" uzantılı sayfada kodları yazınız ve  ikonuna tıklayarak projeyi kaydediniz.
- 9- ARMapp-18 setini USB kablo kullanarak bilgisayara bağlayınız.
- 10- Pişano anahtarlardan sol tarafta bulunan LED, BUZZER, TUŞ TAKIMI, POWER LED anahtarlarını açarak, bu modüllerin GND bağlantılarını sağlayınız.
- 11-  "Build and Program" ikonuna tıklanarak veya (Ctrl+F11) tuşlarına aynı anda basarak yazılan programı derleyip STM32F407VG mikrodenetleyicisine yükleyiniz.
- 12- Programın çalışmasını menü butonlarına basarak gözlemleyiniz. (Dikkat: Power led'e doğrudan uzun süre bakmak gözlerde kalıcı hasara sebep olabilir.)

Sorular:

- 1- "Sol" butonuna basıldıkça ledleri sağdan sola sırayla söndüren, "sağ" butonuna basıldıkça soldan sağa sırasıyla yakan programı yazınız.
- 2- "Onay" butonuna bastıkça ledlerde 0-255 arası binary sayma işlemini gerçekleştiriniz.