

Gömülü Sistemler

Tek LED Yakıp Söndürme

```
from machine import Pin
from time import sleep

led = Pin(9, Pin.OUT)

while True:
    led.value(1) # LED'i yak
    print("LED yandı")
    sleep(1)

    led.value(0) # LED'i söndür
    print("LED söndü")
    sleep(1)
```

Kullanıcıdan Giriş Alarak LED Yakma

```
from machine import Pin
from time import sleep

led = Pin(9, Pin.OUT)

while True:
    value = int(input("Bir sayı giriniz: "))
    if value == 1:
        led.value(1)
        print("LED Yandı")
        sleep(1)
        led.value(0)
    else:
        print("Yanlış sayı girdiniz!")
```

Klavyeden Harf Alarak LED Kontrolü

```
from machine import Pin
from time import sleep

led_kirmizi = Pin(0, Pin.OUT)
led_yesil = Pin(1, Pin.OUT)
led_mavi = Pin(8, Pin.OUT)
while True:
    value = input("Bir harf giriniz (k, y, m): ")
    if value == "k":
        led_kirmizi.value(1)
        sleep(1)
        led_kirmizi.value(0)
        k = 1
    elif value == "y":
        led_yesil.value(1)
        sleep(1)
        led_yesil.value(0)
        y = 1
    elif value == "m":
        led_mavi.value(1)
        sleep(1)
        led_mavi.value(0)
        m = 1
    else:
        print("Yanlış harf girdiniz!")
```

Buton Kullanımı

```
from machine import Pin
from time import sleep

led_kirmizi = Pin(0, Pin.OUT)
led_yesil = Pin(1, Pin.OUT)
button = Pin(15, Pin.IN, Pin.PULL_UP)
```

```

while True:
    if button.value() == 0: # Butona basınca 0 olur
        led_kirmizi.value(1) # Kırmızı yanar
        led_yesil.value(0) # Yeşil söner
    else:
        led_kirmizi.value(0) # Kırmızı söner
        led_yesil.value(1) # Yeşil yanar

    sleep(0.1)

```

Sayıyı Binary Olarak LED'lerle Gösterme

```

from machine import Pin

# LED pinleri (MSB'den LSB'ye doğru yorumlanmış)
led_sari = Pin(3, Pin.OUT) # MSB (En Yüksek Değerli Bit)
led_mavi = Pin(2, Pin.OUT)
led_kirmizi = Pin(1, Pin.OUT)
led_yesil = Pin(0, Pin.OUT) # LSB (En Düşük Değerli Bit)

# LED'leri sırayla bir listeye ekle (MSB → LSB)
ledler = [led_sari, led_mavi, led_kirmizi, led_yesil]

# Fonksiyon: Onluk sayıyı 4-bit ikili string'e çevirir
def sayiyi_binarye_cevir(sayi):
    binary = bin(sayi)[2:] # 'bin()' fonksiyonu '0b' ile başlar, [2:] ile bunu atarız
    while len(binary) < 4: # Eğer 4 bittten kısaysa başına '0' ekleriz
        binary = '0' + binary
    return binary

# Fonksiyon: İkili string'e göre LED'leri yakar/söndürür
def led_durumunu_yazdir(binary):
    print("LED Durumları (MSB → LSB):")
    for i in range(4): # 4 bit (LED) için döngü
        # binary string'deki '1' ise durum=1 (yanık), '0' ise durum=0 (sönük)
        durum = 1 if binary[i] == '1' else 0
        ledler[i].value(durum) # İlgili LED'in durumunu ayarla
        renk = ["SARI", "MAVİ", "KIRMIZI", "YEŞİL"][i] # LED sırasına göre renk isr

```

```

        print("LED {}: {}".format(renk, "ON" if durum else "OFF")) # Durumu yazdır

# Ana Fonksiyon
def main():
    try: # Hata yakalama bloğu başlar
        sayi = int(input("0 ile 15 arasında bir sayı giriniz: ")) # Sayı al
        if 0 <= sayi <= 15: # Sayı geçerli aralıkta mı?
            binary = sayiyi_binary_cevir(sayi) # İkiliye çevir
            print("Binary karşılığı:", binary)
            led_durumunu_yazdir(binary) # LED'leri ayarla
        else: # Geçersiz aralık
            print("Hata: Lütfen 0 ile 15 arasında bir sayı giriniz.")
    except: # Sayısal olmayan giriş vb. hatalar için
        print("Hata: Lütfen geçerli bir tam sayı giriniz.")

main() #

-----

from machine import Pin

ledler = [
    Pin(3, Pin.OUT), # MSB (En anlamlı bit)
    Pin(2, Pin.OUT),
    Pin(1, Pin.OUT),
    Pin(0, Pin.OUT) # LSB (En az anlamlı bit)
]

# LED renk bilgisi (sadece çıktı için)
renkler = ["LED1 (MSB)", "LED2", "LED3", "LED4 (LSB)"]

def sayiyi_binary_cevir(sayi):
    """Sayıyı 4-bit binary formatına çevirir (0000-1111)"""
    return "{0:04b}".format(sayi)

def ledleri_ayarla(binary_str):
    """Binary stringe göre LED'leri kontrol eder"""

```

```

for i in range(4):
    durum = int(binary_str[i])
    ledler[i].value(durum)
    print(f"{renkler[i]}: {'AÇIK' if durum else 'KAPALI'}")

def main():
    print("0-15 arası sayı giriniz")

    while True:
        giris = input("Sayı: ")

        try:
            sayi = int(giris)
            if 0 <= sayi <= 15:
                binary = sayiyi_binary_cevir(sayi)
                ledleri_ayarla(binary)
            else:
                print("Hata: Lütfen 0-15 arasında bir sayı girin!")
        except ValueError:
            print("Hata: Geçersiz giriş! Lütfen sayı girin veya 'q' ile çıkın")

if __name__ == "__main__":
    main()

```

RGB LED Kullanımı

```

from machine import Pin
import time

# RGB LED pin bağlantıları
RED_PIN = 1
GREEN_PIN = 0
BLUE_PIN = 2

# RGB LED pinlerini çıkış olarak ayarlama
red = Pin(RED_PIN, Pin.OUT)
green = Pin(GREEN_PIN, Pin.OUT)
blue = Pin(BLUE_PIN, Pin.OUT)

```

```

# Kullanıcıdan 0-255 arasında üç renk bileşeni al
def get_color_input():
    try:
        # RGB bileşenlerinin değerlerini kullanıcıdan al
        r = int(input("Kırmızı için 0-255 arasında bir değer girin: "))
        g = int(input("Yeşil için 0-255 arasında bir değer girin: "))
        b = int(input("Mavi için 0-255 arasında bir değer girin: "))

        # Değerlerin geçerli olup olmadığını kontrol et
        if not (0 <= r <= 255) or not (0 <= g <= 255) or not (0 <= b <= 255):
            print("Lütfen 0 ile 255 arasında geçerli bir değer girin!")
            return get_color_input() # Yanlış değer girildiyse yeniden dene

        return r, g, b # Renk bileşenlerini döndür

    except ValueError:
        print("Lütfen geçerli bir sayı girin!")
        return get_color_input() # Yanlış giriş yapıldığında tekrar sor

# RGB LED'i yak
def set_led_color(r, g, b):
    # RGB LED'inin her bir bacağını kontrol et (PWM ile parlaklık ayarlama)
    red.value(1 if r > 0 else 0) # 0'dan büyükse kırmızı yanacak
    green.value(1 if g > 0 else 0) # 0'dan büyükse yeşil yanacak
    blue.value(1 if b > 0 else 0) # 0'dan büyükse mavi yanacak

# Ana program fonksiyonu
def main():
    # RGB LED'in tüm renklerini sırayla yakalım
    print("Kırmızı LED yanıyor...")
    set_led_color(255, 0, 0) # Kırmızı
    time.sleep(1) # 1 saniye bekle

    print("Yeşil LED yanıyor...")
    set_led_color(0, 255, 0) # Yeşil
    time.sleep(1) # 1 saniye bekle

```

```

print("Mavi LED yanıyor...")
set_led_color(0, 0, 255) # Mavi
time.sleep(1) # 1 saniye bekle

# Kullanıcıdan renk seçimi yapması için giriş alalım
while True:
    r, g, b = get_color_input() # Kullanıcıdan renk bileşenlerini al
    set_led_color(r, g, b) # RGB LED'ini bu renk ile yak
    print(f"LED şu renk ile yanıyor: RGB({r}, {g}, {b})")
    time.sleep(1) # 1 saniye bekle

if __name__ == "__main__":
    main()

```

7 Segment Display Kullanımı

```

from machine import Pin
from utime import sleep

# 7 Segment Display pinleri (A, B, C, D, E, F, G, DP)
pins = [
    Pin(0, Pin.OUT), # A
    Pin(1, Pin.OUT), # B
    Pin(2, Pin.OUT), # C
    Pin(3, Pin.OUT), # D
    Pin(4, Pin.OUT), # E
    Pin(5, Pin.OUT), # F
    Pin(6, Pin.OUT), # G
    Pin(0, Pin.OUT), # DP (not connected)
]

# 0-9 arasındaki rakamlar için segment dizileri
digits = [
    [0, 0, 0, 0, 0, 0, 1, 1], # 0
    [1, 0, 0, 1, 1, 1, 1, 1], # 1
    [0, 0, 1, 0, 0, 1, 0, 1], # 2
    [0, 0, 0, 0, 1, 1, 0, 1], # 3
    [1, 0, 0, 1, 1, 0, 0, 1], # 4

```

```

[0, 1, 0, 0, 1, 0, 0, 1], # 5
[0, 1, 0, 0, 0, 0, 0, 1], # 6
[0, 0, 0, 1, 1, 1, 1, 1], # 7
[0, 0, 0, 0, 0, 0, 0, 1], # 8
[0, 0, 0, 1, 1, 0, 0, 1], # 9
]

# Rakamları displayde göstermek için fonksiyon
def display_digit(i):
    if 0 <= i <= 9:
        for j in range(7): # Pinleri set etmek için döngü
            pins[j].value(digits[i][j])
            sleep(1)

# Ana program
def main():
    try:
        # İlk sayı kullanıcıdan alınır
        num1 = int(input("Birinci sayıyı giriniz: "))
        display_digit(num1)

        # İkinci sayı kullanıcıdan alınır
        num2 = int(input("İkinci sayıyı giriniz: "))
        display_digit(num2)

        # Sayılar toplanır
        result = num1 + num2

        # Toplama sonucu yazdırılır
        print(f"Sonuç: {result}")
        display_digit(result) # Sonuç displayde gösterilir

    except ValueError:
        print("Geçersiz giriş! Lütfen geçerli bir tam sayı girin.")

if __name__ == "__main__":
    main()

```


