



GÖRÜNTÜ İŞLEME DERSİ YARDIMCI DERS NOTLARI -2023-



Python Seri Port Haberleşme ve Görüntü İşleme ile Kullanımı

Seri port haberleşme, bir bilgisayarın veya cihazın diğer cihazlarla haberleşebilmesi için kullandığı bir veri iletim yöntemidir. Bu haberleşme yöntemi, verileri sıralı olarak bir bit (1 veya 0) olarak gönderir ve alır.

Python dilinde seri port haberleşme için birkaç farklı kütüphane bulunmaktadır. Bunlar arasında en popüler olanlar pySerial ve PyVISA'dır. pySerial, seri port haberleşme işlemlerini yapmak için kullanılacak bir Python kütüphanesidir. PyVISA ise, seri port haberleşmesi dışında USB, GPIB ve Ethernet gibi birçok veri iletim yöntemini destekleyen bir kütüphanedir.

Seri port haberleşme işlemlerini yapmak için, öncelikle seri port bağlantısını açıp kapatmak, veri göndermek ve almak gibi temel işlemleri yapabilecek bir kod bloğu yazmanız gerekmektedir. Bu kod bloğunu yazdıktan sonra, seri port haberleşme işlemlerini yapmak istediğiniz cihazla bağlantı kurarak haberleşmeyi gerçekleştirebilirsiniz.

Python pySerial kütüphanesinin kurulması

İlk olarak, Python dilinin yüklü olduğundan emin olun.

Python dilinin yüklü olduğundan emin olduktan sonra, Python paket yöneticisi olan “**pip**” komutunu kullanarak pySerial kütüphanesini kurun. “**pip**” komutunu aşağıdaki gibi terminal ekranında veya command prompt penceresinde çalıştırın:

```
>> pip install pyserial
```

Bu komutu çalıştırdıktan sonra, pySerial kütüphanesi kurulmuş olacaktır. Kurulum tamamlandıktan sonra, import serial komutuyla pySerial kütüphanesini içe aktararak seri port haberleşme işlemlerini yapmaya başlayabilirsiniz.

Python ile pySerial kütüphanesi ile seri haberleşme

Öncelikle pySerial kütüphanesini indirin ve projenize dahil edin. Daha sonra aşağıdaki adımları izleyerek seri port haberleşme işlemlerini yapabilirsiniz:

```
import serial
```

komutuyla pySerial kütüphanesini içe aktarın.

Seri port bağlantısını açmak için serial.Serial() fonksiyonunu kullanın. Bu fonksiyonun argümanları olarak seri port ismini (örneğin 'COM2'), baud rate'i ve diğer ayarları verin. Örnek kod şu şekildedir:

```
import serial
```

```
ser = serial.Serial('COM2, 9600, timeout=1)
```

Seri port bağlantısını kontrol etmek için **ser.is_open()** özelliğini kontrol edin. Eğer seri port bağlantısı açıksa, True değerini döndürür.



GÖRÜNTÜ İŞLEME DERSİ YARDIMCI DERS NOTLARI -2023-



Veri göndermek için **ser.write()** fonksiyonunu kullanın. Bu fonksiyonun argümanı olarak gönderilecek veri dizisini verin. Örnek kod şu şekildedir:

```
data = "Hello World!"
```

```
ser.write(data.encode())
```

Veri almak için **ser.read()** fonksiyonunu kullanın. Bu fonksiyon, belirtilen miktar kadar veri okur. Örnek kod şu şekildedir:

```
data = ser.read(32)
```

```
print(data)
```

Seri port bağlantısını kapatmak için **ser.close()** fonksiyonunu kullanın.

Bu adımları izleyerek pySerial kütüphanesi kullanarak seri port haberleşme işlemlerini yapabilirsiniz.

Örnek Kod:

```
import serial
import time

ser = serial.Serial("COM10", 9600)
komut="ON"
x=1;
while x<10:
    ser.write(komut.encode('utf-8'))
    time.sleep(1)
    x=x+1
    print('x= ',x)

komut="OFF"
ser.write(komut.encode('utf-8'))
ser.close()
```

Görüntü İşlemede seri haberleşme kullanımı:

Görüntü işlemede seri port haberleşmesi, görüntü işleme algoritmalarının çalıştırıldığı bilgisayar ve görüntülerin çekildiği cihazlar arasında veri iletimi için kullanılabilir. Örneğin, bir görüntü sensörü veya kamera ile bir bilgisayar arasında veri iletimi için seri port haberleşmesi kullanılabilir. Bu sayede, görüntü sensörü tarafından çekilen görüntüler bilgisayar tarafından işlenebilir ve görüntü işleme algoritmaları tarafından analiz edilebilir.

Görüntü işlemede seri port haberleşmesinin diğer bir önemi ise, görüntü sensörü veya kamera gibi cihazların çalışma parametrelerinin değiştirilebilmesidir. Örneğin, görüntü sensörünün çözünürlük ayarı veya kameranın açıklaması gibi parametreler bilgisayar tarafından değiştirilerek cihazın çalışma şeklinin değiştirilebilir. Bu sayede, görüntü işleme sistemlerinin performansı ve doğruluğu artırılabilir.

Bir başka önemi ise görüntü sinyallerinden elde edilen bilgilerin dış çevreye gönderilebilmesini sağlamasıdır. Bu sayede dış çevredeki aktivatörler çalıştırılabilir. Bir örnek vermek gerekirse:



GÖRÜNTÜ İŞLEME DERSİ YARDIMCI DERS NOTLARI -2023-



Girişin yasak olduğu bir bölgeye yerleştirilen kamera ile piksel hareketleri izlenebilir. Piksel hareketi belli bir eşiği geçtiğinde seri port ile bir alarm kontağı tetiklenebilir, SMS gönderilebilir vs. Piksel hareketini tespit etmek için ardışık video çerçeveleri birbirinden aritmetik olarak çıkarılabilir. Böylece korunan alanda herhangi bir hareket olmazsa fark ekran siyah olacaktır. Eğer bu alanda bir hareket olursa fark ekran tamamen siyah olmayacaktır. Görüntü işleme algoritmaları ile bu durum tespit edilerek seri porttan bilgi gönderilebilir.

Başka bir örnek vermek gerekirse; Bir sabit kanat veya döner kanatlı İHA ile hedef tespit edildiğinde görüntü işleme programının çalıştığı bilgisayarın seri portundan bomba haznesini tutan servo motora sinyal gönderilerek bombanın bırakılması sağlanabilir.

Bu işlem için örnek bir kod oluşturalım:

```
import serial
import cv2
import numpy as np
import time
ser = serial.Serial("COM10", 9600, timeout=1)
x = cv2.VideoCapture(0)
# ret, frame = x.read() #Bir çerçeve oku
sayac=0
# h, w, _ = frame.shape # Çerçevenin en ve boyunu çıkart

while(x.isOpened()):

    sayac=sayac+1
    ret, frame = x.read()
    # Kırmızı renk tespiti yapılıyor.
    xGri=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    xred=frame[:, :,0]
    kirmiziRenk=cv2.subtract(xred, xGri)
    z, xIkili=cv2.threshold(kirmiziRenk, 75, 255, cv2.THRESH_BINARY)
    kernel = np.ones((2,2), np.uint8)
    xIkili=cv2.erode(xIkili, kernel, iterations=10)
    # ÖNEMLİ ÖNEMLİ ÖNEMLİ
    (toplamlBlob, etiket_id, degerler, centroid) = cv2.connectedComponentsWithStats(xIkili, 4, cv2.CV_32S)
    print('sayac: '+str(sayac)+ ' '+str(toplamBlob))

    # Kırmızı renk tespit edildiğinde seri porttan "ON" verisi gönderiliyor.
    if (toplamlBlob>1):
        komut="ON"
        ser.write(komut.encode('utf-8'))
        # time.sleep(1)
        # print('Burada en az bir kırmızı nesne var')
    else:
        # Kırmızı renk yoksa "OFF" seri porttan gönder.
        komut="OFF"
        ser.write(komut.encode('utf-8'))

    cv2.imshow("frame", frame)
    cv2.imshow("Ikili", xIkili)
    time.sleep(1)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

ser.close()
x.release()
cv2.destroyAllWindows()
```

Yukarıdaki kod parçasında:

Videodan gelen canlı görüntüde kırmızı renkli nesne tespit edildiği anda seri porttan "ON" değeri 10. Com porta bağlı olan cihaza gönderilmektedir. Eğer kırmızı nesne yok ise yine aynı cihaza "OFF" değeri gönderiliyor.



GÖRÜNTÜ İŞLEME DERSİ YARDIMCI DERS NOTLARI -2023-



Eğer seri portun ucunda bir Arduino cihazı var ise, bu cihazın gelen bilgiyi alıp bir aksiyonda bulunması için aşağıdaki gibi bir Arduino kodu yeterli olacaktır.

```
int ledPin=46;
int led_error_pin=13;

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(led_error_pin, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  if(Serial.available()>0) {

    String veri=Serial.readString();

    if(veri == "ON"){
      digitalWrite(ledPin,HIGH);}
    else if (veri == "OFF") {
      digitalWrite(ledPin,LOW); }
    else {
      digitalWrite(led_error_pin,HIGH);
      delay(100);
      digitalWrite(led_error_pin,LOW);
    }
  }
}
```

Bu koda göre; gelen veri ON ise 46. Pine bağlı ledi yakacak, OFF ise bu ledi söndürecek, her ikisi de değilse 13. Pine bağlı ledi döngüsel olarak yakıp söndürecektir.

Python Seri porttan veri okuma

Arduino ile Python arasında seri port haberleşmesi gerçekleştirilebilir. Bu haberleşme işlemi için, öncelikle Arduino tarafında gönderilecek verilerin seri port üzerinden gönderilmesi gerekmektedir. Arduino kodlarında, **Serial.println()** fonksiyonu kullanılarak veriler seri porta gönderilebilir.

Örnek olarak, aşağıdaki Arduino kodu çalıştırıldığında, "Merhaba Python" yazısı seri port üzerinden gönderilecektir:



GÖRÜNTÜ İŞLEME DERSİ YARDIMCI DERS NOTLARI -2023-



```
void setup() {  
  // Seri port başlat  
  Serial.begin(9600);  
}  
  
void loop() {  
  // Veriyi seri porta gönder  
  Serial.println("Merhaba Afyon Kocatepe Üniversitesi");  
  delay(1000); // 1 saniye bekle  
}
```

Python tarafında ise, pySerial kütüphanesini kullanarak seri porttan veri alınabilir.

Şimdi kendi kodumuzu yazalım:

Yukarıdaki arduino kodu aynen kalsın. Bu kodu python tarafında okuyarak ekrana yazdıracak bir kod oluşturalım:

```
1 import serial  
2 import time  
3  
4 ser = serial.Serial("COM10", 9600)  
5 x=0  
6 while x<10:  
7     data=ser.readline()  
8     time.sleep(1)  
9     x=x+1  
10    print('x= ',x, data)  
11  
12 ser.close()
```

Yukarıdaki koda Arduino'dan seri olarak gönderilen ifade 10 defa sayaç ile birlikte yan yana yazdırılacaktır.

Sonuç:

```
x= 1 b'\x00\x00Merhaba Afyon Kocatepe Üniversitesi\r\n'  
x= 2 b'Merhaba Afyon Kocatepe Üniversitesi\r\n'  
x= 3 b'Merhaba Afyon Kocatepe Üniversitesi\r\n'  
x= 4 b'Merhaba Afyon Kocatepe Üniversitesi\r\n'  
x= 5 b'Merhaba Afyon Kocatepe Üniversitesi\r\n'  
x= 6 b'Merhaba Afyon Kocatepe Üniversitesi\r\n'  
x= 7 b'Merhaba Afyon Kocatepe Üniversitesi\r\n'  
x= 8 b'Merhaba Afyon Kocatepe Üniversitesi\r\n'  
x= 9 b'Merhaba Afyon Kocatepe Üniversitesi\r\n'  
x= 10 b'Merhaba Afyon Kocatepe Üniversitesi\r\n'
```

```
In [7]:
```