

3D Sonar kordinat tespiti ve grafiksel yorumu

Bu yazımda basit bir ultrasonic sonar ile 3D tarama yapıp. Konum ve uzaklık bilgisini hesaplayıp grafik olarak çizdireceğim.

Öncelikle neler kullanacağımıza bakalım.

Ultrasonic Sensör HC-SR04

Cisim ile uzaklığımızı ölçmek için Ultrasonic sensör kullandım. Piyasada çok bilinen HC-SR04 ü tercih ettim. Yaklaşık olarak 2cm ile 4 metre ölçüm alabiliyor.

Çalışma Prensipleri ise Ultrasonik ses dalgalarının belirli atmosferik koşullardaki hızının biliniyor olmasından faydalanarak ses dalgasının gidip gelme süresinin yarısıyla hızının çarpımıyla mesafe bilgisi elde edilmesi.



FPV-PTZ Kit

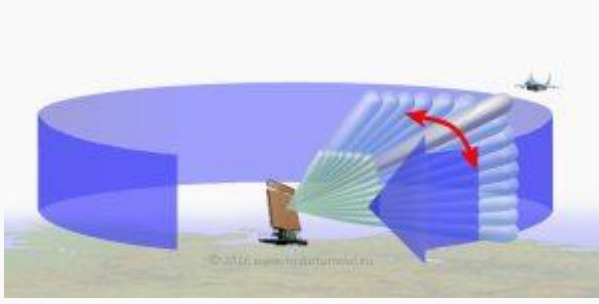
FPV kamera açısı için kullanılan mekanik. 2 tane G9 servo dan oluşuyor. Montajı yaparken servoları 90 dereceye alıp FPV kiti dik açılarla monte edin. Böylece aşağı yukarı, sağ sol için 0-180 derece hareket imkanı tanırız.



Atmel veya Arduino Mikroişlemci

İşlemleri yapıp bilgisayarda yorumlamamız için gerekli mikroişlemci. Bu projede arduino nano yu kullandım. Herhangi bir başka arduino serisi kullanabilirsiniz.

Arduino Nano V 3.0 GRL Pinout				Pin diagram for GRL v0.8 and v0.9	
Atmega 328P					
Pinout Ref					Pinout Ref
D13	Spindle Direction	D13	D12	Spindle Enable	D12
3V3	Not Used	3V3	D11	Limit Z-Axis	D11
VREF	Not Used	VREF	D10	Limit Y-Axis	D10
A0	Reset Abort	A0	D9	Limit X-Axis	D9
A1	Feed Hold	A1	D8	Stopper Enable/Disable	D8
A2	Cycle Start/Resume	A2	D7	Direction Z Axis	D7
A3	Coolant Enable	A3	D6	Direction Y Axis	D6
A4	(Not Used/ Reserve)	A4	D5	Direction X Axis	D5
A5	Probe	A5	D4	Step Pulse Z Axis	D4
A6	Not Used	A6	D3	Step Pulse Y Axis	D3
A7	Not Used	A7	D2	Step Pulse X Axis	D2
			GND		
			RST		
			RX1		
			TX1		



Genel olarak algoritma

Kod algoritma olarak mikroişlemciyle başlıyor. Mikroişlemci servolara teker teker açı sinyalleri göndererek tüm açıları eksiksiz ve tekrarsız olarak tanyor. Bu döngü loop olarak sürekli tekrarlanıyor. Her açı kontrolünde ise ultrasonic sensörden ses dalgasının gidip gelme süresiyle mesafe ölçülüyor. Ölçüldükten sonra ise seri portla bilgisayarımıza servo açıları ve uzaklık bilgisini gönderiyor. Ardından bilgisayarda yapacağımız hesap ise servo y ve servo x açısıyla beraber uzaklık bilgisi. Öncelikle gelen Y açısından 90 derece çıkarıyoruz çünkü kalibrasyonu montajda 90 derece ayarlamıştık. Servo Y açısını sin ve cos olarak mesafe ile çarparak Y koordinatı ve XZ bilgisini bulmuş oluyoruz. Aynı şekilde de Servo X açısıyla XZ bilgisini çarparak X ve Z koordinatlarını da bulmuş oluyoruz. Bundan sonrası ise 3 boyutlu grafik ile koordinatları çizdiriyoruz. Böylece bir cisim tespit edilmişse grafiksel olarak görebiliyoruz. Koordinat, uzaklık gibi bilgilerini de tespit etmiş oluyoruz. Daha güçlü sensörler (lidar, radar vb..) ve alternatif mekanikler (brushless, step motor vb..) ile ölçüm mesafesini metrelerce hatta kilometrelerce arttırabilirsiniz.

Arduino Kodları

```
#include <Servo.h>
#include <Ultrasonic.h>

Ultrasonic ultrasonic(12, 13); //trig_echo
Servo servo_x,servo_y;
int
mesafe,pos_x,pos_y,hiz=1,sonic=1,pos_x_s=0,pos_x_f=180,pos_y_s=0,pos_y_f=180;//başlangıç
ve bitis arasındaki fark çift olmalı

void setup() {
  Serial.begin(9600);
  servo_y.attach(8);servo_x.attach(9);
}

void loop() {
  for (pos_y = pos_y_s; pos_y <= pos_y_f; pos_y += 1) {
    servo_y.write(pos_y);
    delay(hiz);
    for (pos_x = pos_x_s; pos_x <= pos_x_f; pos_x += 1) {
      servo_x.write(pos_x);
      delay(hiz);
      mesafe=ultrasonic.distanceRead();
      if(mesafe>=200){mesafe=200;}
      Serial.print(pos_y);Serial.print(" ");Serial.print(pos_x);Serial.print("
");Serial.print(mesafe);Serial.print("\n");
      delay(sonic);
    }
    pos_y++;
    servo_y.write(pos_y);
    delay(hiz);
    for (pos_x = pos_x_f; pos_x>= pos_x_s; pos_x -= 1) {
      servo_x.write(pos_x);
      delay(hiz);
```

```

mesafe=ultrasonic.distanceRead();
Serial.print(pos_y);Serial.print(" ");Serial.print(pos_x);Serial.print("
");Serial.print(mesafe);Serial.print("\n");
delay(sonic);
}
}
pos_x++;
for (pos_y = pos_y_f; pos_y>= pos_y_s; pos_y -= 1) {
servo_y.write(pos_y);
delay(hiz);
for (pos_x = pos_x_s; pos_x <= pos_x_f; pos_x += 1) {
servo_x.write(pos_x);
delay(hiz);
mesafe=ultrasonic.distanceRead();
if(mesafe>=200){mesafe=200;}
Serial.print(pos_y);Serial.print(" ");Serial.print(pos_x);Serial.print("
");Serial.print(mesafe);Serial.print("\n");
delay(sonic);
}
pos_y--;
servo_y.write(pos_y);
delay(hiz);
for (pos_x = pos_x_f; pos_x>= pos_x_s; pos_x -= 1) {
servo_x.write(pos_x);
delay(hiz);
mesafe=ultrasonic.distanceRead();
Serial.print(pos_y);Serial.print(" ");Serial.print(pos_x);Serial.print("
");Serial.print(mesafe);Serial.print("\n");
delay(sonic);
}
}
}

```

Matlab Kodları

```

clc;
clear all;

delete(instrfindall);
baglanti = serial ('COM4','BaudRate',9600); % buradan COM portunu deęiřtirmelisiniz

fopen(baglanti);

while(1)
data = fscanf(baglanti);
[y, xd] = strtok(data);
[x, d] = strtok(xd);

ya = real(str2num(y));
xa = real(str2num(x));
da = real(str2num(d));

ya=ya-90;

[z,xz]=sincos(ya,da);

```

```
[x,y]=sincos(xa,xz);  
  
yf = round(y);  
xf = round(x);  
zf = round(z);  
  
disp(yf);  
disp(xf);  
disp(zf);  
  
scatter3(xf,yf,zf,'filled')  
hold on  
drawnow;  
  
end  
fclose(baglanti);
```