

T.C.  
FIRAT ÜNİVERSİTESİ  
TEKNOLOJİ FAKÜLTESİ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ



## **ARDUİNO İLE DEPREM VE SARSINTI ALGILAYICI**

**BİTİRME PROJESİ**

Hazırlayan

**16503050 KENAN AYBERK ÖZKAN**

Bitirme Yöneticisi

**DOÇ.DR. ÇETİN GENÇER**

**ELÂZİĞ 2020**

**T.C.**  
**FIRAT ÜNİVERSİTESİ**  
**TEKNOLOJİ FAKÜLTESİ**  
**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ**

**ARDUİNO İLE DEPREM VE SARSINTI ALGILAYICI**

**BİTİRME PROJESİ**

**Hazırlayan**

**16503050 KENAN AYBERK ÖZKAN**

**Bitirme Yöneticisi**

**DOÇ.DR. ÇETİN GENÇER**

Bu tez, ..... tarihinde aşağıda belirtilen jüri tarafından oybirliği  
/oyçokluğu ile başarılı / başarısız olarak değerlendirilmiştir.

Danışman:

Üye:

Üye:

**ELÂZİĞ 2020**

## ÖNSÖZ

“Arduino ile deprem ve sarsıntı algılayıcı” projesi Fırat Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü Bitirme Projesi (EET437) dersi için DOÇ.DR. ÇETİN GENÇER yardımıyla hazırlanmıştır.

Bu projede, ADXL335 kullanılarak Arduino üzerinden depremlerin veya sarsıntıların ölçülmesi ve veri haline getirilmesi sağlanmıştır.

Bu proje kapsamında proje seçiminden sonuna kadar bizlere yardımcı olan, sorularımızı öğretici bir şekilde anlatan ve edindiği bilgi ve tecrübeleri bize aktaran DOÇ.DR. ÇETİN GENÇER hocama teşekkür ederim.

## İÇİNDEKİLER

ÖNSÖZ.....	I
İÇİNDEKİLER.....	II
ŞEKİLLER DİZİNİ.....	III
1. GİRİŞ .....	1
2. KULLANILAN MALZEMELER VE PROGRAMLAR .....	2
2.1. Arduino.....	3
2.1.1. Arduino Kullanım Avantajları .....	4
2.1.2. Arduino Nano.....	4
2.1.3. Nano Kartın Pinleri ve Özellikleri .....	5
2.1.4. Nano Kartın Giriş/Çıkışları.....	6
2.2. Altium Designer .....	8
2.3. ADXL335 3 Axis Accelerometer .....	12
2.4. LCD Display.....	14
2.5. Buzzer .....	15
2.6. LED .....	16
2.7. Jumper Kablolar .....	16
2.8.Processing .....	17
2.9. TinkerCad .....	18
3. ARDUİNO NANO’NUN YENİDEN TASARLANMASI .....	19
3.1 Şematik Tasarım .....	19
3.2 PCB Tasarım.....	20
4. TINKERCAD BAĞLANTI ŞEMASI VE İLK TEST AŞAMASI.....	21
5. SONUÇ VE TARTIŞMA .....	22
Kaynakça.....	24

Proje Çalışma Kodları .....	25
Arduino IDE Kodları .....	25
Processing 3 IDE Kodları .....	30

## ŞEKİLLER DİZİNİ

Şekil 2.1 Arduino Nano.....	6
Şekil 2.2 Arduino Nano Giriş/Çıkışlar .....	8
Şekil 2.3 Altium Designer .....	9
Şekil 2.4 ADXL 335 Pin bağlantı şeması .....	13
Şekil 2.5 ADXL335 Acceloremeter Modülü .....	13
Şekil 2.6 2x16 LCD Display .....	14
Şekil 2.7 Buzzer .....	15
Şekil 2.8 Kırmızı ve Yeşil LED .....	16
Şekil 2.9 Jumper Kablo .....	17
Şekil 2.10 Processing .....	17
Şekil 2.11 Sismografin Anlık Görüntüsü .....	18
 Şekil 3.1 Şematik Tasarım .....	19
Şekil 3.2 4 Katmanlı PCB Tasarım ve 3D Görüntüsü.....	20
 Şekil 4.1 TinkerCad Bağlantı Şeması .....	21
Şekil 4.2 Devrenin Breadboard Bağlantısı .....	22

## 1. GİRİŞ

Deprem sensörleri sismik hareketler algıladığında uyarı sağlamak ve binalardaki bazı sistemleri kesmek amacıyla yani acil otomasyon sistemi olarak kullanılır. Deprem dalgaları mekanik dalgalardır ve ses dalgaları ile aynıdır. P (Primary) ve S (Secondary) olmak üzere iki türlü dalgaya sahiptir. P dalgası hızlı ve sesle gelen bir dalga olduğu için deprem sensörlerinin ilk algıladıkları yıkıcı etkisinin az olan dalgalardır. (1) İnsanların tedbir alabileceği zaman bu dalganın gerçekleştiği andır. Deprem sensörleri bu anda devreye girerek sesli ve görsel uyarı vermektedir.

İlk sismoskop M.S. 132’de Zhang Heng tarafından icat edilmiştir. Heng’in sismografi, bakırdan yapılmış büyük bir vazoya benzemektedir. Cihazın çevresinde 8 adet metal ejderha kafası vardı ve her ejderhanın ağzına birer bakır top yerleştirilmişti. Ejderhalar; Kuzey, Güney, Doğu, Batı ve 4 ara yönü temsil ediyordu. Her ejderhanın altında ise ağızları açık durumda 8 metal kurbağa vardı. Deprem olunca; kabın içindeki mekanizma sayesinde, sarsıntının geldiği yöndeki ejderhanın ağzındaki bakır top alttaki kurbağanın ağzına düşmekteydi. Böylece depremin hangi kentte olduğu belirlenebilmekteydi. (2)

Projede rasathanelerde ölçüm için kullanılan sismograf cihazlarındaki akselerometre ile birlikte Arduino kodlanarak, evlerimizde kullanabileceğimiz daha küçük ve hassasiyeti daha iyi bir sensör tasarlanmıştır.

## **2. KULLANILAN MALZEMELER VE PROGRAMLAR**

- Arduino Nano (Kişiselleştirilmiş ve yeniden tasarlanmış)
- ADXL335 3 Axis Accelerometer
- 2x16 LCD Display
- Led (Kırmızı ve yeşil)
- Dirençler
- Buzzer
- Jumper kablolar
- Arduino IDE
- Altium Designer
- Processing 3
- Tinkercad

## 2.1. Arduino

Arduino, üzerinde Atmel mikrokontrolörler bulunan açık kaynak kodlu bir donanımdır. Yazılım bilgilerinin yanı sıra, tasarım bilgileri de kullanıcının hizmetine sunulmuştur. PCB baskı devreleri, devre şemaları, yerleşim planları ve üzerindeki programlanabilir elemanların kodları bütün detayları ile kullanıcının kullanımına rahatça ulaşabileceği bir şekilde açılmıştır. Kullanıcı isterse hazır olarak alabileceği gibi, aynısını kendisi de gerçekleştirebilir. Arduino geliştirme kartı üzerindeki mikrokontrolör (AtmegaXX) Arduino programlama dili ile programlanır ve bu program Processing tabanlı Arduino Yazılım Geliştirme Ortamı (IDE) yardımı ile karta yüklenir.

Yazılım Geliştirme Ortamı (IDE) internet üzerinden ücretsiz olarak bilgisayara indirilebilir. (3) Programlama için temel C dili kullanılır. Arduino'yu ön plana çıkaran en önemli özelliği yazılımının kolay ve sade olmasıdır. Ayrıca kullanıcıya uygulama gerçekleştirmesi için diğer çevre birimleriyle (kristal, güç kaynağı, programlama cihazı) uğraşmama imkânı sunar. Programlama cihazına ihtiyaç duymamasının nedeni; ürün, bootloader programı mikro denetleyicinin içine atılmış şekilde gelmektedir.

Arduino'nun en güçlü özelliklerinden biri de genişletilebilir bir kütüphane sistemine sahip olmasıdır. Bu kütüphaneler sayesinde birçok işlem ve çevrebirimi haberleşmesi kolaylıkla gerçekleştirilebilir. Ayrıca yeni çevrebirimleri için yazılan kütüphaneler kolaylıkla entegre edilebilir. Bütün bunları göz önünde bulundurduğumuzda çok ileri bir yazılımcı olmadan birçok uygulamayı Arduino kullanarak gerçekleştirebilmek mümkün olur. Gerçekleştirilmesi istenen projenin özelliğine göre Arduino'nun birçok çeşidi bulunmaktadır. Ayrıca bu kartlara uygun şekilde tasarlanmış shield (katman) olarak adlandırılmış kullanım kolaylığı sağlayan ek donanımlar da üreticinin hizmetine sunulmuştur. (Bluetooth, Wireless, Ethernet, Motor sürücü katmanları vs.).



### 2.1.1. Arduino Kullanım Avantajları

- Arduino, bir önyükleyiciye (bootloader) sahip olduğundan USB port bağlantısı ile mikroişlemciye program yükleyebilir. Bu sayede mikroişlemcilerdeki çıkarılıp takılmalardan kaynaklanan bacak kırılmaları riski ortadan kalkar.
- Arduino programlaması diğer programlamalara göre kolaydır.
- Diğer platformlarla karşılaştırıldığında daha ucuzdur.
- Basit ve açık programlama ortamı sayesinde kolay bir yazılım ortamı sağlar.

### 2.1.2. Arduino Nano

ATmega328 temelli Arduino Nano, breadboard ve farklı platformlar ile beraber kullanılabilir. Arduino IDE'si üzerinden rahatlıkla programlanabilmektedir. DC güç beslemesi bulunmamaktadır. Buna karşın USB üzerinden besleme yapılabilir. Bununla beraber Vin pini üzerinden de 5-12V arası giriş gerilimi uygulanabilir.

Arduino, Processing/Wiring dilini kullanarak çevre elemanları ile temel giriş çıkış uygulamalarını gerçekleştiren açık kaynaklı fiziksel programlama platformudur. Arduino ile bağımsız olarak interaktif uygulamalar gerçekleştirilebilirsiniz. Aynı zamanda Arduino'yu bilgisayar ile Flash, Processing, MaxMSP, C Sharp gibi birçok yazılım üzerinden ya da kendi yazdığınız yazılımlarla haberleştirerek de kullanabilirsiniz. Açık kaynaklı ara yüz yazılımını internet sitesinden Windows, Mac OS X ve Linux platformları için indirebilirsiniz.

### 2.1.3. Nano Kartın Pinleri ve Özellikleri

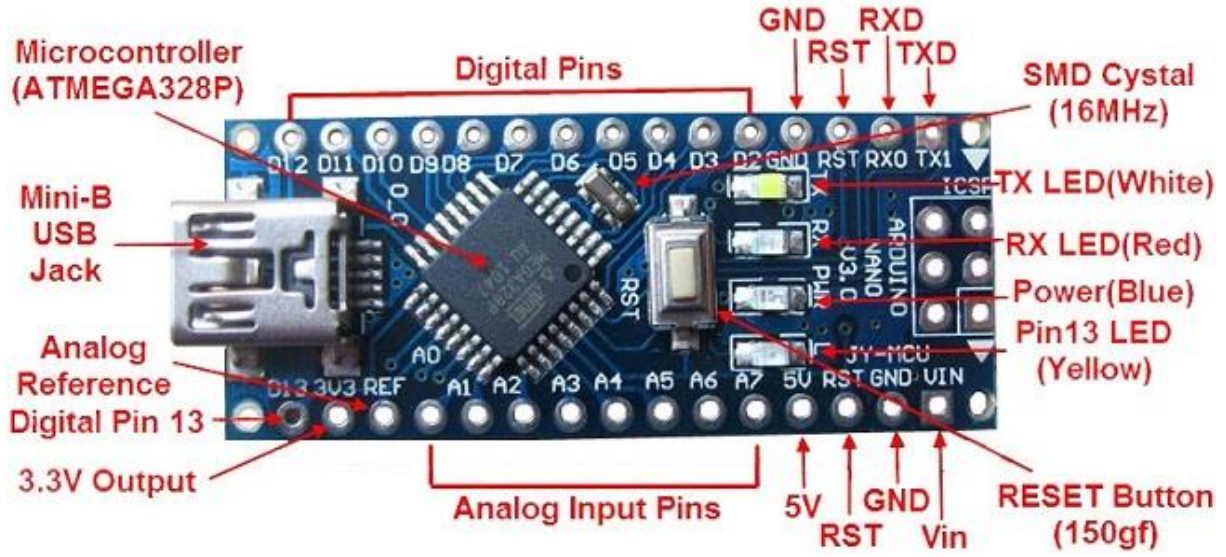
Arduino Nano bir B tipi mini USB kablosu ile bilgisayara bağlanarak çalıştırılabilir ya da harici bir güç kaynağından beslenebilir. 6 – 20 V aralığında bir harici güç kaynağı pin 30'a bağlanabilir ya da 5 V regüle edilmiş bir gerilim ile pin 27'den beslenebilir.

**3V3:** Bu pin daima 3.3V çıkış vermektedir. Bazı hassas elemanlar 3.3V ile çalışmaktadır. Nano'nun Pro Miniye bir üstünlüğü de üzerinde hem 5V hem de 3.3V çıkış bulunmasıdır. Pro Miniler ise iki model olup sadece ya 3.3V ya da 5V verebilmektedir.

**+5V/VCC:** Bazı modellerde +5V bazılarında VCC yazan bu pin arduino çalıştığı sürece 5V çıkış vermektedir. Ayrıca sabit 5V çıkış veren güç kaynağımız varsa bu pine bağlayarak Arduino kartımızı besleyebiliriz. Arduino kartı en fazla 40 mA verebildiği için bu amperden düşük çalışan led gibi elemanları direkt Arduino'ya bağlayabiliriz. Eğer motorlar gibi daha fazla akım çeken eleman bağlamak istersek harici olarak motor sürücü kartlar kullanmak gerekecektir.

**VIN/RAW:** Bazı modellerde VIN bazılarında RAW yazan bu pin Arduino kartımızı beslemek için kullanılmaktadır. 6-20V arasındaki güç kaynağımızı bu pine bağlayarak kartımızı çalıştırabiliriz. Bu pine bağlı olan regülatör sayesinde 6-20V arasındaki voltaj 5 volta dönüştürülerek karta verilir. Genelde 9V pil kullandığımız için bu pini sık kullanacağız.

**GND:** Projelerimizde elektrik akışının tam olabilmesi için neredeyse her eleman ile GND(GROUND) yani toprak hatlarını birleştirmemiz gerekiyor. Bu Arduino'nun (–) kutuplu 4 ve 29 nolu bacağıdır. Sık kullanıldığı için iki tane GND pini konulmuştur.



Şekil 2.1 Arduino Nano

#### 2.1.4. Nano Kartın Giriş/Çıkışları

Arduino Nano'da bulunan 14 tane dijital giriş / çıkış pininin tamamı, pinMode(), digitalWrite() ve digitalRead() fonksiyonları ile giriş ya da çıkış olarak kullanılabilir. Bu pinler 5 V ile çalışır. Her pin maksimum 40 mA çekebilir ya da sağlayabilir ve 20-50 KOhm dahili pull – up dirençleri vardır. Ayrıca bazı pinlerin özel fonksiyonları vardır:

- **Serial 0 (RX) ve 1 (TX) :** Bu ilk iki pin bilgisayarla TTL seri iletişimin yapıldığı pinlerdir. RX (receive) alıcı, TX (Transmit) verici bacaktır. Bu iki bacak ayrıca USB bağlantı noktasına da bağlı olduğu için bu iki bacağı ayrıca kullanmamıza gerek olmayacak.
- **Harici kesmeler (2 ve 3) :** Bu pinler bir kesmeyi tetiklemek için kullanılabilir.
- **PWM: 3, 5, 6, 9, 10, ve 11 :** Bu pinler analogWrite () fonksiyonu ile 8-bit PWM sinyali sağlar.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) :** Bu pinler SPI kütüphanesi ile SPI haberleşmeyi sağlar.

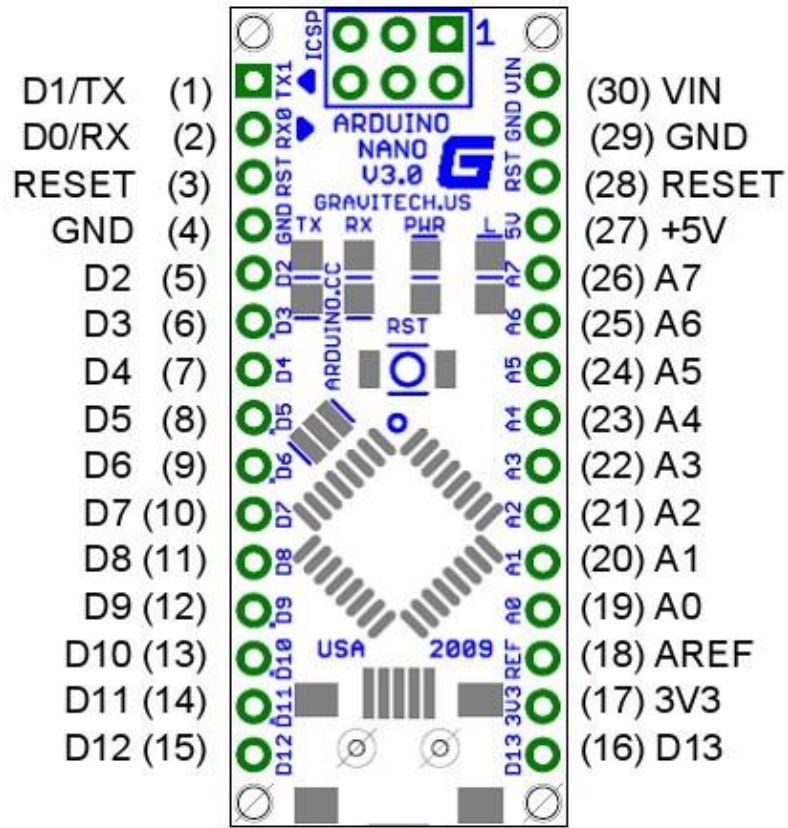
– **LED 13:** Dijital pin 13’e bağılı bir leddir. Pinin değeri High olduğıunda yanar, Low olduğıunda söner.

Arduino Nano’nun 8 adet analog girişı bulnur, her biri 10 bitlik çözünürlük destekler. Varsayılan ayarlarda topraktan 5 V a kadar ölçerler. Ancak, AREF pini ve analogReference() fonksiyonu kullanılarak üst limit ayarlanabilir. Analog pinlerinden 6 ve 7 dijital pin olarak kullanılabilirler. Ayrıca bazı pinlerin özel fonksiyonları vardır.

– **TWI:** A4 ya da SDA pini ve A5 ya da SCL pini Wire kütüphanesini kullanarak TWI haberleşmesini destekler.

– **AREF:** Bu pin analog girişlerden okuyacağımız voltaj değerini ayarlamak için kullanılmaktadır. Normalde nalog giriş pinleri 0-5V arasını okumaktadır. Bu aralığı değıştirmek için AREF pinine okuması istediğı voltajı veriyoruz. Analog girişler için referans voltajıdır. analogReference() fonksiyonu ile kullanılır.

– **RESET:** Arduino içine yazdığımız yazılımın işlem sırasını en başa alarak programı yeniden başlatır. İçindeki programı silmez. Arduino üzerindeki reset tuşu bu pine bağılıdır. Projelerde ayrı bir reset tuşu kullanmak istediğimizde bu pin kullanılır.



Şekil 2.2 Arduino Nano Giriş/Çıkışlar

## 2.2. Altium Designer

Avustralyalı yazılım şirketi Protel tarafından geliştirilen bir PCB çizim ve tasarım programıdır. Elektronik ürün geliştirmek için gereken tüm gereçleri içerisinde barındıran yazılım entegrasyon programıdır. Altium Designer (4) tek bir uygulama ile tasarımın konseptten tamamlanmış haline kadar geliştirilmesini sağlar. Donanım, yazılım ve programlanabilen donanımları birleştirir. Türkiye de piyasa da çok talep gören baskı devre tasarım programıdır.

PCB, elektronik devrelerin bakır yollar ve komponentleri içerecek şekilde, plakalar üzerinde işlenmiş halidir. Dolayısı ile program bir elektronik tasarımın temelini oluşturmaktadır.



Şekil 2.3 Altium Designer

Altium Designer programı çok gelişmiş bir hata analiz motoruna sahip. Ayrıca gelişmiş bir simülasyon motoru sayesinde, devre üzerinde bulunan tüm elemanların zamana göre değişimini gösteren grafiksel analiz yapabiliyorsunuz. HDL, VHDL, C, XML vb. gelişmiş birçok programlama alt yapısına sahip olan Altium Designer ile yazacağınız programları aynı program üzerinde bulunan şematik çizimde deneme şansına sahipsiniz. Programın bir diğer özelliği de gelişmiş kütüphanesi ve program ile birlikte gelen çok geniş şablon dosyalar bulundurmasıdır. Üzerinde gelen birçok şablonu inceleyerek çizimlerinize profesyonellik katabilirsiniz. Bazı şablonları ise hiç değiştirmeden olduğu gibi alıp çizimlerinize ekleme şansına da sahipsiniz. PCB sayfasına baktığımızda, üreticilerinin söylediğine göre Altium Designer, dünyadaki en iyi otomatik çizim motoruna sahip. Ayrıca PCB çiziminin en başında koyduğunuz kuralları, çizim sırasında asla ihmal edemiyorsunuz. Örneğin; iki hat arasındaki mesafenin en az 8mil olmasını tanımlamışsanız, çizim sırasında bir hattın 8mil'den daha yakınına bir hat çizemezsiniz. Bu özelliği sayesinde çizimleri hatasız bir şekilde sürdürebilirsiniz. Daha öncede söylediğim gibi, karmaşık çizimlerde kart üzerinde yapılacak en ufak bir yanlış, onca çizimin ve emeğin boşa gitmesi anlamına gelebilir.

Bu yüzden hataları en sonunda ayıklamak yerine, çizim sırasında hata yapmanız engellenmektedir. Altium Designer'ın bir diğer güzel özelliği de katmanlar arasında çok kolay bir şekilde dolaşabilirsiniz. 3D özelliği ile kartın gerçeğine birebir uyan üç boyutlu görüntüsünü görebilir, 3 boyutluyken de katmanlar arasında dolaşıp analizler yapabilirsiniz.

#### **– Bütünleşik Tasarım Ortamı**

Şematik çizimi, kolay kullanılan PCB tasarım araçları, gelişmiş 3D görüntüleme, üretim dosyaları, tasarım kütüphaneleri ve malzeme listesi oluşturma özelliklerini bütünleşik bir ara yüzde sunmaktadır. Data yönetim sistemi ile kart tasarımını, çevresel birimlere bağlı ve yalnız önemli olana odaklanarak gerçekleştirmeyi sağlamaktadır.

#### **– Kullanım Kolaylığı**

Yeni başlayanlar için tasarımın baştan sona uygulamasını sağlayan dokümantasyonlar içermektedir. İleri düzey özel konulara yönelik eğitim ve videolara erişimin yanı sıra ürünün web sitesinde bulunan forumlar aracılığıyla ve program kullanımı sırasında etkileşimli yardım fonksiyonları ile tasarım boyunca kullanıcıya rehberlik etmektedir.

#### **– Akıllı Tasarım Yönetimi**

Tasarım ekiplerinin aynı proje üzerinde eş zamanlı çalışma imkanını sağlamaktadır. Tek merkezli kütüphane yönetimi sayesinde proje ve komponentlerin yaşam döngüsünü oluşturma ve tekrar kullanabilme; tasarımların varyasyonlarını oluşturma ve üretim dosyalarının da bu değişikliklere göre elde edilmesini sağlamaktadır.

Tüm ekibin kullandığı Altium Designer versiyonun tek merkezden güncelleştirilme ve kurulum işlemlerini gerçekleştirmektedir.

#### **– Gerçek Zamanlı Tedarikçi Erişimi**

Kart tasarımı süreci boyunca gerçek zamanlı maliyet tahmininin yanı sıra, tedarikçilerin verilerine ulaşarak, uygun parça seçimine olanak sağlamaktadır.

### **– Mekanik – Elektronik Tasarım Bütünlüğü**

Üç boyutlu görüntüleme özelliği ile PCB'nin gerçeğe yakın ve 2D/3D görüntü ayarlarına ulaşarak mekanik ver üzerinden “clearance” kontrolü yapmaktadır. Genel olarak kullanılan Dxf, Step, Parasolid vb. mekanik uzantılarda dosya alma ve aktarma imkânı vermektedir.

### **– Geliştirilmiş Dokümantasyon Yetenekleri**

Üretim aşamasında kolaylık sağlayacak PCB dokümantasyon yöntemleri bulunmaktadır. Başka bir programa ihtiyaç duymadan Altium Designer yeteneklerini kullanarak tüm üretim dosyaları elde edilebilmektedir. Üretim öncesinde sunum niteliği taşıyan ve katı model görüntüsü içeren ön izleme dokümantasyonları oluşturabilmekte, güncel ürün maliyeti elde edebilmekte ve üretim prosesinin tek dosya içerisinde özetleyebilmektedir.

### **– Dosya Aktarma ve Alma**

Bilgisayar destekli bilinen birçok elektronik devre tasarım araçlarında oluşturulmuş tasarımları ve kütüphaneleri, desteklenen versiyonlar dahilinde program içerisine alma ve programdan aktarma desteğini sunmaktadır.

### **– Simülasyon Özelliği**

Analog simülasyon imkânı sunarak istenilen noktalardan voltaj, güç, akım vb. değerler okunabilmektedir. Osiloskop verilerine benzer dalga formlarını ve AC, DC sinyal analiz eğrilerini çizdirilebilmektedir. Üçüncü parti yazılımlar ile iş birliği sağlayarak işlemci simülasyonlarını elde etme imkânı sağlamaktadır.

### **– Katı-Esnek PCB Tasarımı**

Katı ve katı-esnek PCB tasarımların oluşturabilmenin yanı sıra bu tasarımların 2D ve 3D görüntülenmesi imkânı sunmaktadır. Esnek yapılar bulunduran tasarımların katlanmış ve katlanmamış hallerini görüntüleme ve çıktısını alma işlemleri gerçekleştirilebilmektedir. Mekanik tasarım programlarına veri aktarımı sağlayarak, oluşturulan tasarımın en iyi şekilde sonlandırılmasında yardımcı olmaktadır. Katı ve katı-esnek PCB tasarım yetenekleri sayesinde aynı kart üzerinde farklı katman yapısı oluşturulabilmekte ve iç katmanlara komponent yerleşimine olanak vermektedir.



### – Hiyerarşik Tasarım Yapısı

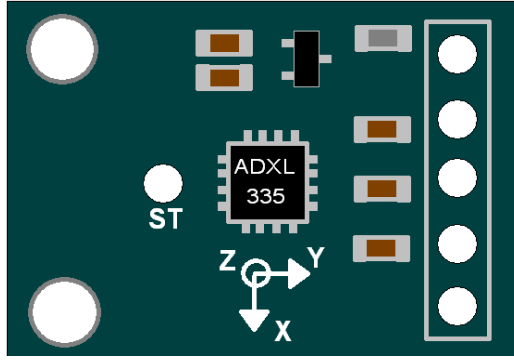
Karmaşık tasarımların okunurluğunu kolaylaştırmak için farklı şematik dosyalarını bağlantıları ile beraber bir üst şematik dosyasına linkleyerek, kolayca takip edilebilir proje taslağına çevirme imkânı vermektedir. Tasarımlarda tekrar eden bölümleri, bloklar haline getirerek ve tek bir blok referans alınarak yerleştirme ve yönlendirme işlemlerinin sürelerini kısaltmaktadır.

### – Güç Dağılım Analizi

Devre üzerinde belirlenen bileşenler veya bileşenler ile besleme kaynağı arasında bulunan bağlantı yolları üzerindeki voltaj düşümü ve akım yoğunluğunu göstermektedir. Belirlenen hat boyunca katmanlar arasındaki kayıpları veya akım yoğunluğunu 2D/3D görüntüleme sistemi sayesinde görselleştirerek, kart üzerindeki sorun oluşturabilecek alanların tespitine olanak sağlamaktadır.

## 2.3. ADXL335 3 Axis Accelerometer

İvmeölçer, ivme kuvvetini ölçen elektromekanik bir cihazdır. Sadece yerçekimi nedeniyle, yani g kuvveti ile ivmeyi gösterir. İvmeyi g birimi cinsinden ölçer. Bu İvmeölçer modülü, X, Y ve Z ivmelenmesini analog girişlerden voltaj değerine dönüştürerek veri üretmektedir. ADXL335 üç eksenli analog ivmeölçer entegre bir devredir. Yerçekimine bağlı ivme miktarını ölçerek bulunduğu yere göre eğildiği açıyı bulabilir. Dinamik ivmelenme miktarını algılayarak, cihazın ne kadar hızlı ve hangi yönde hareket ettiğini bulabilir. İvmeölçer, 3 analog giriş pin kullanarak bir mikro denetleyici ile arabirim oluşturmak çok kolaydır. PIC veya AVR gibi diğer çoğu mikro denetleyici ile kullanılabilir.



Şekil 2.4 ADXL 335 Pin bağlantı şeması

VCC: Giriş pini.

X\_OUT: X eksen analog çıkışı.

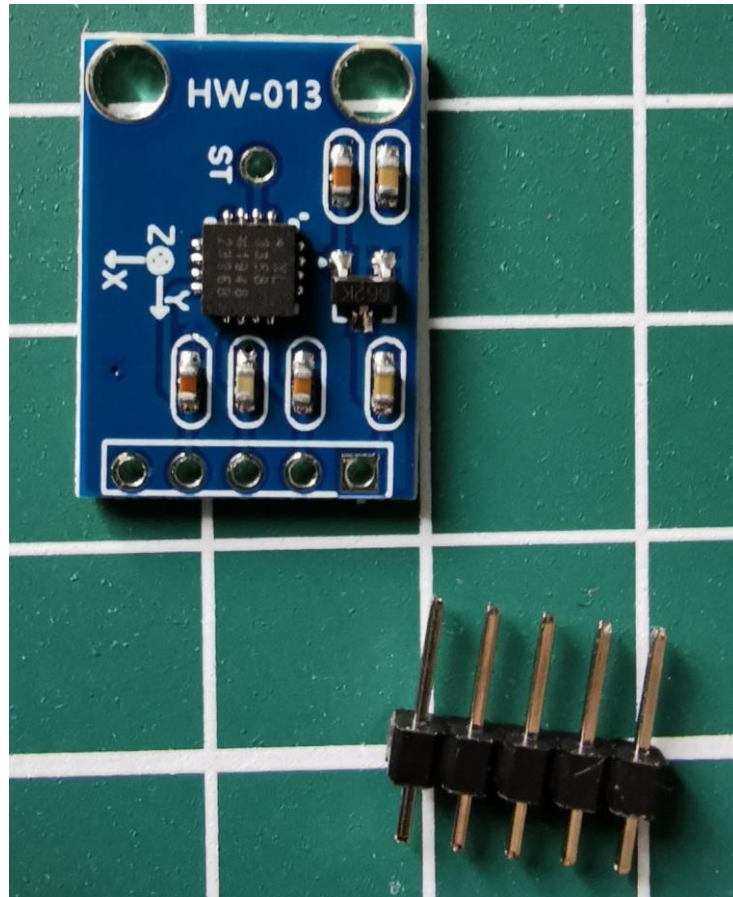
Y\_OUT: Y eksen analog çıkışı.

Z\_OUT: Z eksen analog çıkışı.

GND: Toprak.

ST: Self-Test.

X, Y, Z değerlerini kullanarak eğim hesaplayabiliriz. Ayrıca X, Y ve Z eksenine göre açıları hesaplayabiliriz. Bu yüzden önce 10 bitlik ADC değerlerini g birimine çevirmemiz gerekiyor. ADXL335 veri sayfasına göre 0g'de maksimum gerilim seviyesi 1.65V ve hassasiyet ölçek faktörü 330mV / g'dir. (5)



Şekil 2.5 ADXL335 Acceloremeter Modülü

$$A_{out} = \frac{\frac{ADC \text{ value} * V_{ref}}{1024} - \text{Voltage Level at } 0g}{\text{Sensitivity Scale Factor}}$$

*Denklem 2.1*

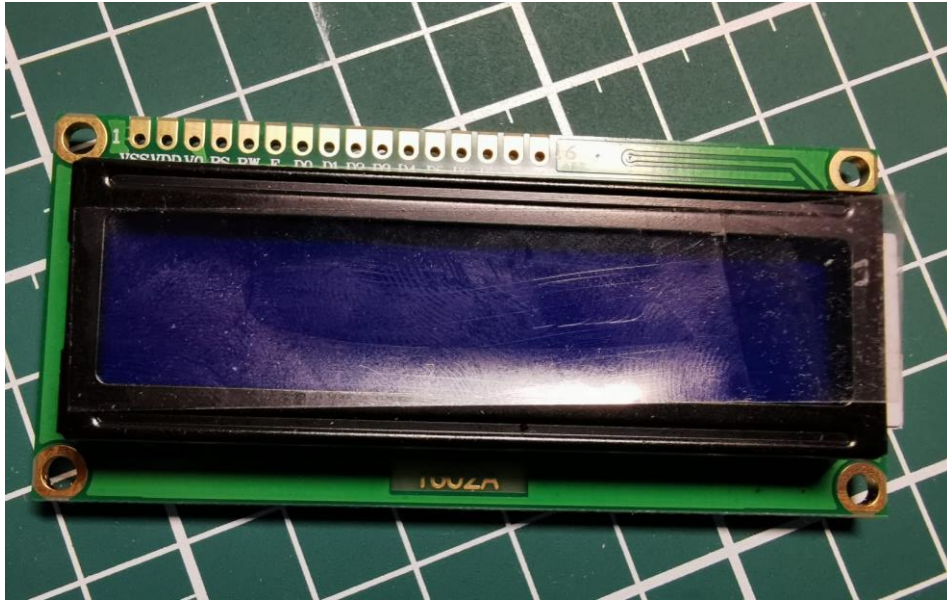
Denklem 2.1 bize X, Y ve Z eksenleri için g birimi cinsinden ivme değerlerini şu şekilde verir:

$$A_{xout} = (((X \text{ eksenini ADC değeri} * V_{ref}) / 1024) - 1.65) / 0.330$$

$$A_{yout} = (((Y \text{ eksenini ADC değeri} * V_{ref}) / 1024) - 1.65) / 0.330$$

$$A_{zout} = (((Z \text{ eksenini ADC değeri} * V_{ref}) / 1024) - 1.65) / 0.330$$

#### 2.4. LCD Display



**Şekil 2.6** 2x16 LCD Display

2x16 LCD Display 2 satır 16 sütun, bağlantı için 16 adet pin bulunmaktadır. Bazı ekranlarda arka aydınlatma ışığı bulunmadığından 14 adet pin yer almaktadır. (6) 15 ve 16 numaralı pinler ekran aydınlatması bulunan ekranlarda ışığı yakmak için kullanılır.

Diğer bağlantılar ise şu şekildedir:

- **VSS Girişi:** GND girişidir. Negatif gerilim bağlanır.
- **VDD (5V) Girişi:** 5 volt pozitif gerilim bağlanır.
- **V0 Girişi:** Ekrandaki yazının netliğini ayarlamak için kullanılan voltaj girişidir.
- **RS (Register Select):** LCD ekranın ne yapacağını, register da bir değer olup olmadığını kontrol eder.
- **R/W (Read/Write):** Pinlerin modlarını okuma veya yazma olarak değiştirmek için kullanılır.
- **E (Enable):** Pinleri okuma modunda aktif hale getirir.
- **D0...D7:** Bit bit aldığı değerleri okur. Buradan ekrana yazdırılacak yazılar bit bit gönderilir. LCD de okur.
- **A(Anot) ve K(Katot):** Sonda bulunan bu iki girişe + ve – olmak üzere voltaj uygulanır. Bu sayede LCD'nin arka plan ışığı yanar.

## 2.5. Buzzer

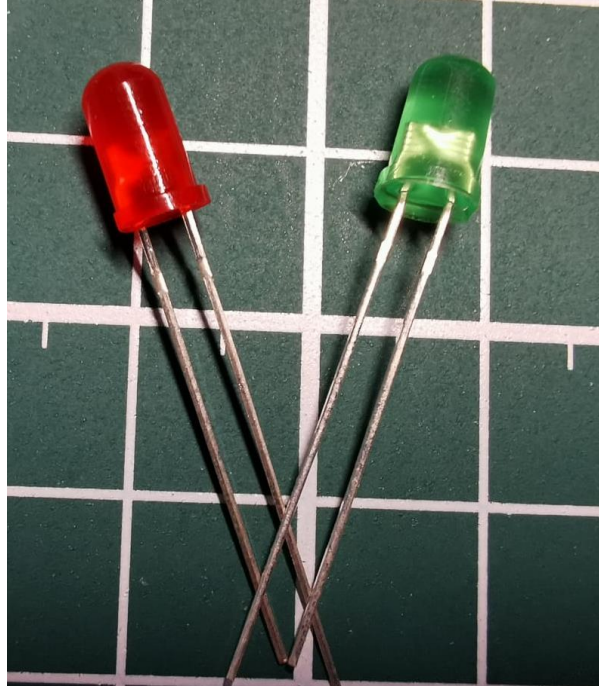
Buzzer; mekanik, elektromekanik ya da piezoelektrik prensiplerine bağlı olarak çalışan işitsel ikaz cihazı çeşididir. Kullanım alanları oldukça fazla olan buzzerlar, genel itibarıyla piezoelektrik prensibiyle çalışmaktadırlar. Buzzerlar, kullanım alanlarına da bağlı olarak alarm, zamanlayıcı, onaylama cevap ikazı gibi işlevlerde kullanılabilirler. (7)



Şekil 2.7 Buzzer

## 2.6. LED

LED ("Light Emitting Diode", Işıık Yayan Diyot), yarı-iletken, diyot temelli, ışıık yayan bir elektronik devre elemanıdır.



Şekil 2.8 Kırmızı ve Yeşil LED

## 2.7. Jumper Kablolar

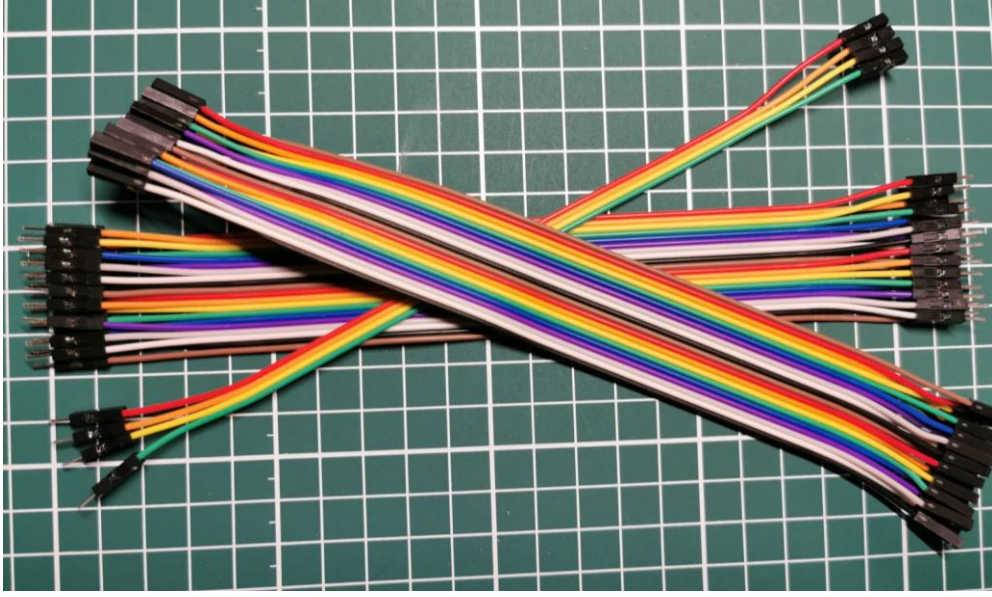
Birçok test projesinde, daha kolay ve daha hızlı işlem yapabilmek için tasarlanmış bir üründür. Kısaca bir çeşit bağlantı kabloları diyebiliriz. Breadbord ve Arduino arasında bağlantı kurmak için oldukça kullanışlıdır. Uçlarında erkek ve dişi girişlerin bulunmasına göre 3 çeşit jumper kablo bulunmaktadır.

-Erkek Erkek

-Erkek Dişi

-Dişi Dişi





Şekil 2.9 Jumper Kablo

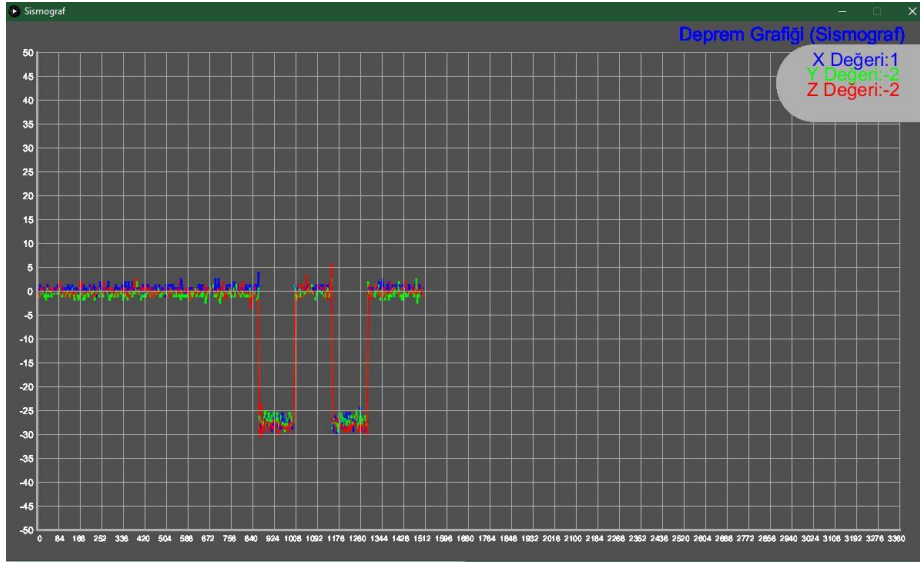
## 2.8.Processing



Şekil 2.10 Processing

Processing, dijital sanat ve görsel tasarım toplulukları tarafından programlamanın temellerini görsel bir yaklaşımla öğretmek için kullanılan veya elektronik bir eskiz defteri görevi de görebilen açık kaynaklı bir programlama dili ve tümleşik geliştirme ortamıdır.

2001 yılında Casey Reas ve Benjamin Fry tarafından başlatılan proje Java üzerine kurulmuş olup daha basit bir grafik programlama modeli ve sözdizimine sahiptir. (8)



Şekil 2.11 Sismografin Anlık Görüntüsü

## 2.9. TinkerCad

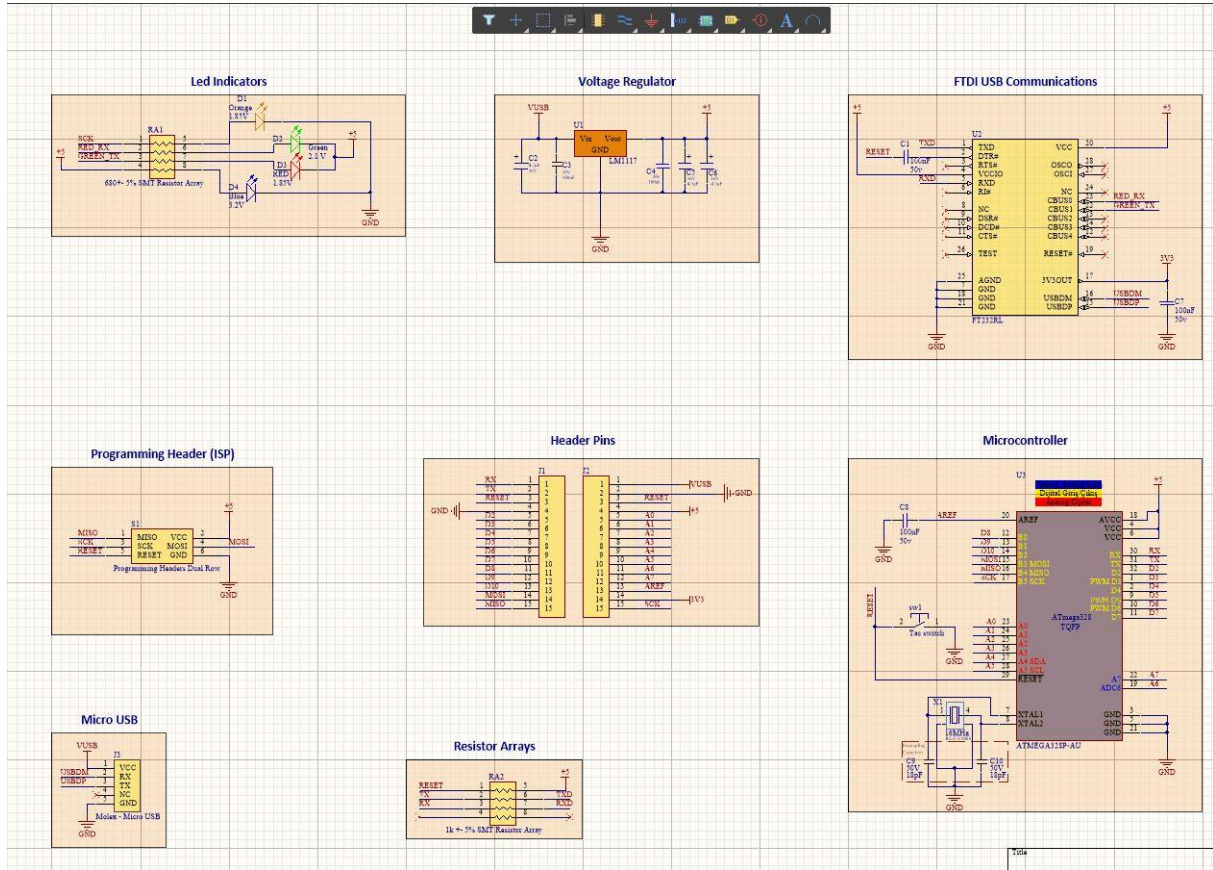
Tinkercad, basitliği ve kullanım kolaylığı ile bilinen bir web tarayıcısında çalışan ücretsiz, çevrimiçi bir 3B modelleme programıdır. Dünyanın her yerinden insanların düşünmesine ve yaratmasına yardımcı olan ücretsiz bir çevrimiçi yazılım araçları koleksiyonudur. 3D tasarım, mühendislik ve eğlence yazılımlarında lider olan Autodesk için ideal bir giriş yapıyor. 2011'de kullanıma sunulduğundan bu yana, okullarda yapıcı katı geometriye giriş düzeyinde bir giriş olmasının yanı sıra 3D baskı için modeller oluşturmak için popüler bir platform haline geldi. (9)

### 3. ARDUİNO NANO’NUN YENİDEN TASARLANMASI

Bu projede kullanılacak olan Arduino Nano’yu boyut olarak daha az yer kaplaması için küçültülmüş, yenileştirilmiş ve projenin ihtiyaçları doğrultusunda tasarlanmıştır. Altium Designer kullanarak şematik ve 4 katmanlı PCB tasarımı yapılmıştır.

#### 3.1 Şematik Tasarım

Sadece Mikrokontrolör değiştirilmeden diğer komponentler değiştirilerek tasarım güncellenmiştir. Komponentlerin seçiminde kalite, ucuz ve kolay bulunabilmesi amaçlanmıştır.

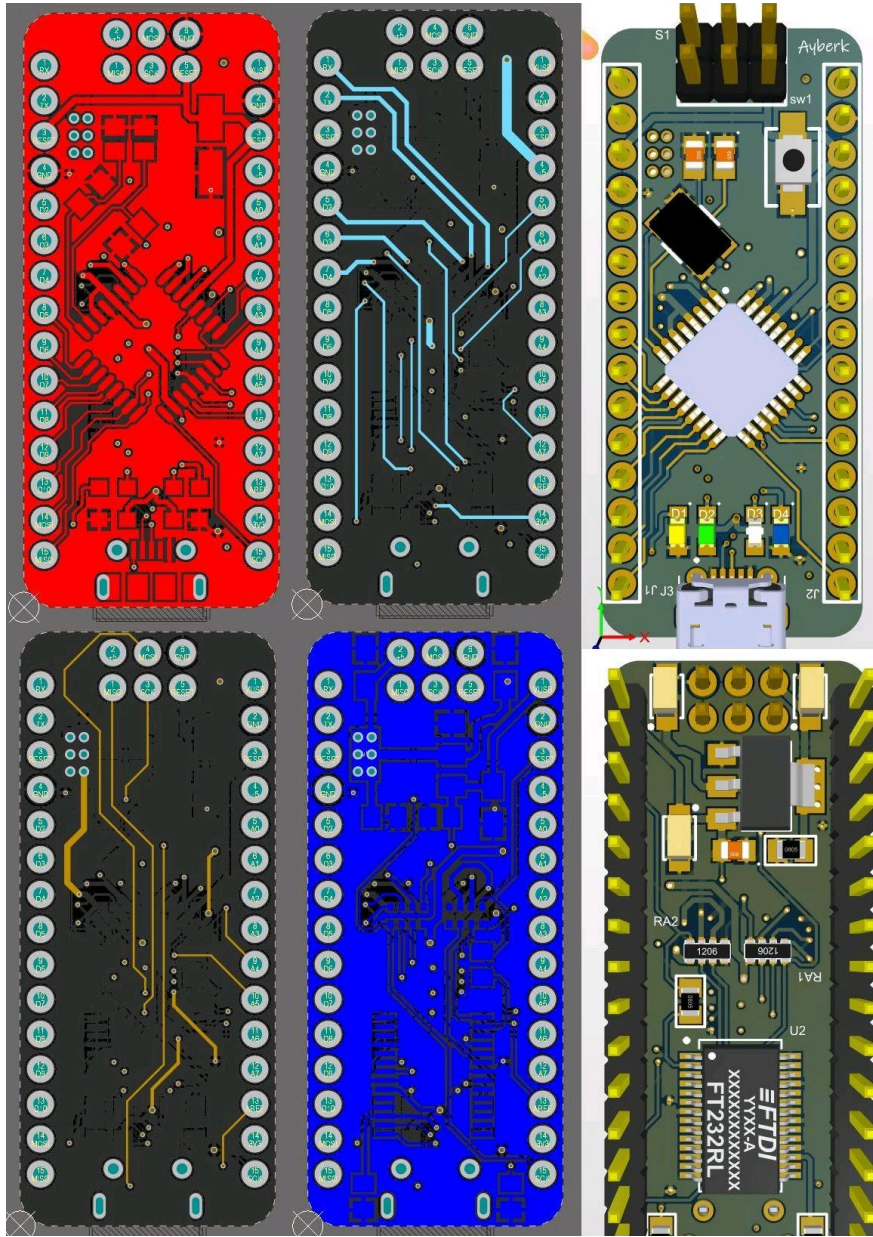


Şekil 3.1 Şematik Tasarım



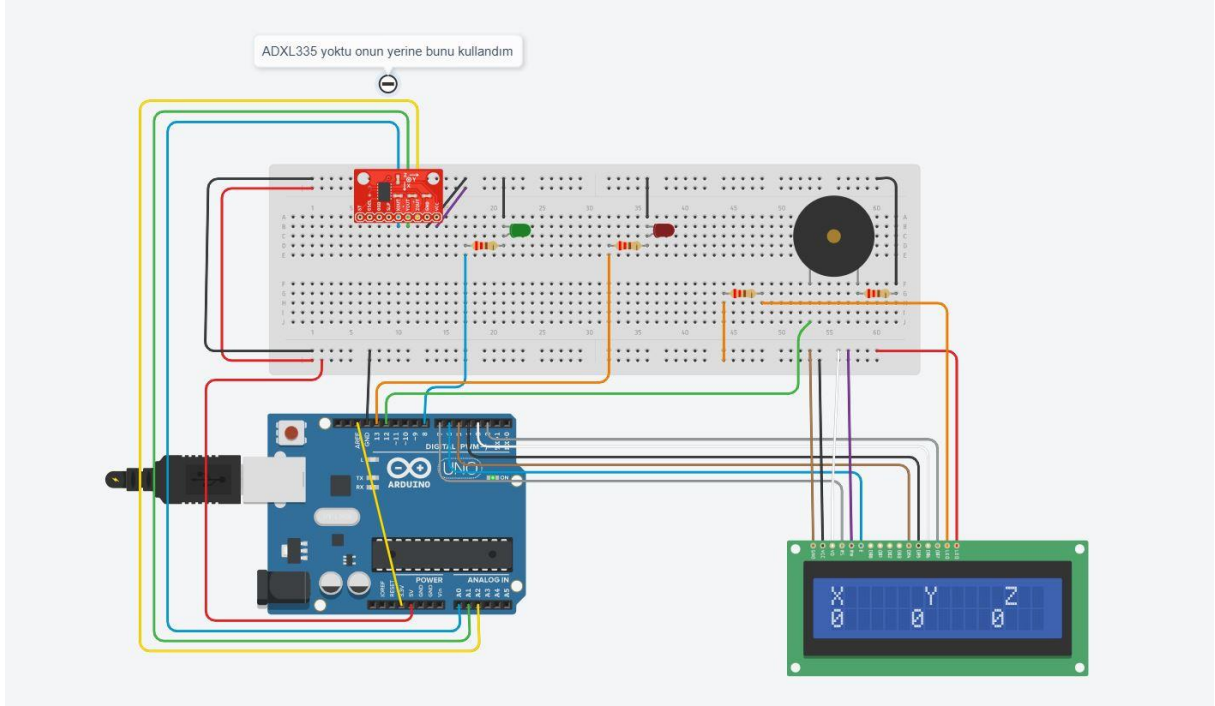
### 3.2 PCB Tasarım

PCB Tasarımında orijinal Arduino Nano'ya nazaran daha az yer kullanılmış ve bakır yollar daha kısaltılmıştır. Ayrıca ısınma sorunu olan yerlere soğutucu Via'lar eklenmiştir. 4 katmanlı olmasına rağmen daha az Via kullanılmıştır. Gürültüyü azaltmak için kartın köşeleri orijinaline göre farklı tasarlanmıştır. İleride aşınması muhtemel olan bakır yollar daha kalın çizilmiştir.



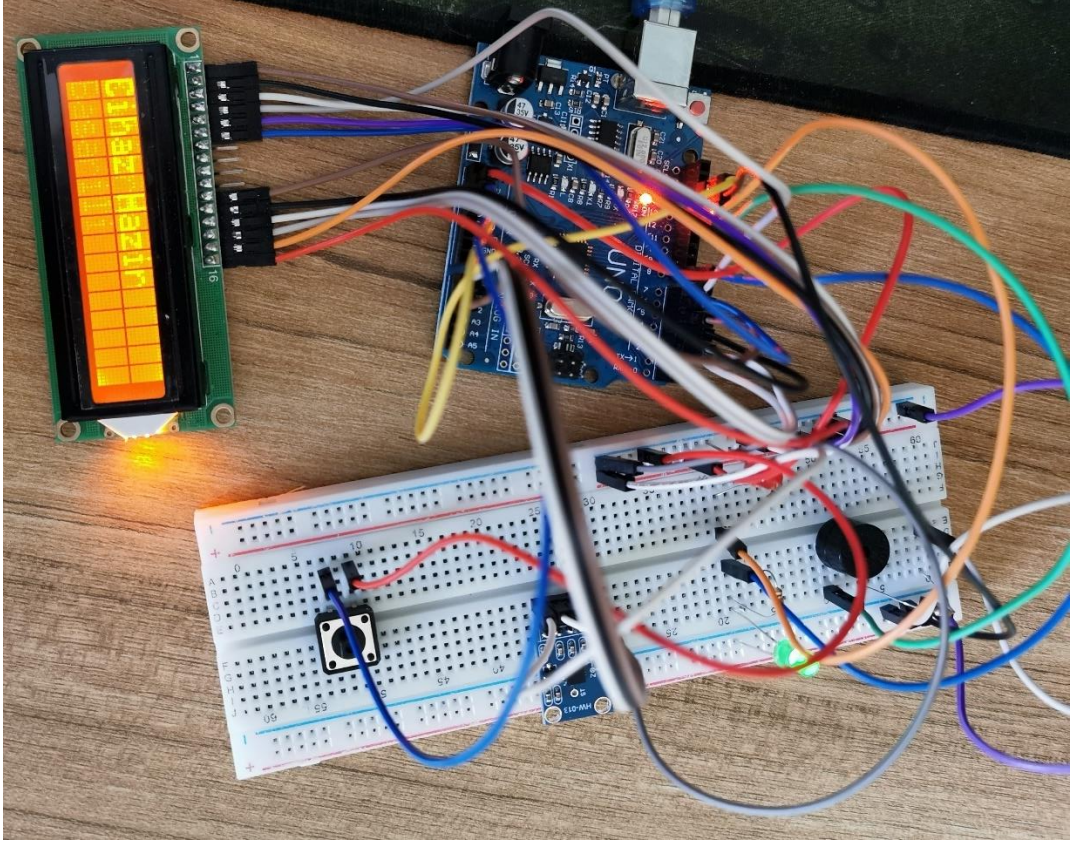
Şekil 3.2 4 Katmanlı PCB Tasarım ve 3D Görüntüsü

#### 4. TINKERCAD BAĞLANTI ŞEMASI VE İLK TEST AŞAMASI



Şekil 4.1 Tinkercad Bağlantı Şeması

İhtiyaçlar doğrultusunda Arduino Nano baştan tasarlandıktan sonra gerber dosyaları üreticiye gönderildi. Fakat üretim aşaması yoğun olduğu için gecikme olacağı söylendi. Bu sırada Arduino Uno kullanılarak program yazılmaya başlandı. Tinkercad kullanılarak hızlı ve kolay bir şekilde bağlantı simülasyonu yapıldı bu sayede gerçek cihazda herhangi bir problemle karşılaşılmadan bağlantıları kolayca yapıldı. İlk çalıştırıldığında hassasiyeti çok düşük düzeydeydi. Hassasiyetini arttırmak için AREF pini ile 3V3 pini arasında bağlantı yapılarak analog girişler 5V yerine 3.3V girişe ayarlandı. Buda ADXL335'in daha düzgün çalışmasını sağlamış oldu.



Şekil 4.2 Devrenin Breadboard Bağlantısı

## 5. SONUÇ VE TARTIŞMA

Projeyi yapay titreşimler ve gerçek deprem ile test etme imkânım oldu ve beklenen sonuçlar elde edildi. Tasarımının küçük olması sayesinde monte edilmek istenilen yere gayet uyum sağlıyor. Bu yerin binanın kolonları olması sensörün daha doğru çalışmasını sağlayacaktır. Buna ek olarak Arduino ile önemli sistemlerin doğalgaz, su veya akıllı diğer ev sistemlerinin deprem anında nasıl davranması gerektiğini ayarlayabiliriz. Bir bilgisayar yardımıyla Processing üzerinden canlı olarak titreşim dalgaları izlenebilir ve bunların kaydı tutulabilir.

Hassasiyetinin daha da artırılması için ek bir devre tasarladım. Bu devrenin ana mantığı şu oldu: Zararlı gürültüyü yükseltmeden sinyalin yalnızca faydalı kısmını yükseltmek. Bunu yapmak için diferansiyel modunda bağlanan enstrümantal işlemsel amplifikatör yani OPAMP, OP07 IC örnek olarak kullanılabilir. Herhangi bir OPAMP IC’de

kullanılabilir. Potansiyometre kullanarak, V2 voltajını V1'den biraz daha düşük olacak şekilde ayarlanmalı ve ikinci bir potansiyometre ile faydalı sinyali ayarlamalıyız. Amplifikasyon katsayısı için formül  $V_{out} = (V1 - V2) * K$ 'dir, burada  $K = 1 + 100 / P2$  (kOhm). Bu güçlendirilmiş sinyal, 1k direnç üzerinden bir Arduino mikrodnetleyicisinin analog girişine bağlanmalı. ADXL335'in üç çıkışını yükseltmek için, her bir eksen (x, y ve z) için ayrı ayrı bu tür üç özdeş amplifikatör kullanmamız gerekecektir.



## Kaynakça

1. **Tech, Michigan.** What is seismology and what are seismic waves?  
<http://www.geo.mtu.edu/UPSeis/waves.html>. [Çevrimiçi]
2. **Boğaziçi Üniversitesi, Kandilli Rasathanesi ve Deprem Araştırma Enstitüsü.**  
[http://www.koeri.boun.edu.tr/sismo/bilgi/sss\\_tr.htm#:~:text=İlk%20sisMoskop%20M.S.,Chang%20Heng%20tarafından%20icat%20edilmiştir](http://www.koeri.boun.edu.tr/sismo/bilgi/sss_tr.htm#:~:text=İlk%20sisMoskop%20M.S.,Chang%20Heng%20tarafından%20icat%20edilmiştir).
3. **Arduino.** <https://www.arduino.cc/en/software>. [Çevrimiçi]
4. **Altium Designer.** <https://www.altium.com>. [Çevrimiçi]
5. **Devices, Analog.** <https://www.analog.com/media/en/technical-documentation/data-sheets/adxl335.pdf>. [Çevrimiçi]
6. **İkizoğlu, Kemal.** <http://blog.ikizoglu.com/2017/10/arduino-2x16-lcd-ekran-kullanimi/>. [Çevrimiçi]  
30 Ekim 2017.
7. **İLX.** <https://medium.com/@ilxmuhendislik1/buzzer-nedir-nasil-calisir-7f902167f595>. *Medium*. [Çevrimiçi]
8. [https://en.wikipedia.org/wiki/Processing\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Processing_(programming_language)). [Çevrimiçi]
9. **AutoDesk.** <https://www.tinkercad.com>. *TinkerCad*. [Çevrimiçi]

## Proje Çalışma Kodları

### Arduino IDE Kodları

```
#include<LiquidCrystal.h>    // LCD kütüphanesi

LiquidCrystal lcd(7,6,5,4,3,2); // LCD bağlantı pinleri


#define buzzer 12            // Buzzer pin'i seçildi.

#define led 13               // Led pin'i seçildi.

#define led2 8               // 2.Led Pini seçildi.(Durum Ledi)


#define x A0                 // Accelerometer'ın X çıkışı Analog olarak tanımlandı.

#define y A1                 // Accelerometer'ın Y çıkışı Analog olarak tanımlandı.

#define z A2                 // Accelerometer'ın Z çıkışı Analog olarak tanımlandı.

    /*Değişkenler*/

int xdegeri=0;

int ydegeri=0;

int zdegeri=0;

long Baslat;

int buz=0;

    /*Değişmeyen Girdiler*/

#define ornekler 50

#define maxDeg 5             // max değişmesini istediğin değer limiti

#define minDeg -500          // min değişmesini istediğin değer limiti

#define buzTime 5000         // buzzer ötme süresi
```

```

void setup()

{

  lcd.begin(16,2);          // LCD Başlatılıyor.

  analogReference(EXTERNAL); // Daha iyi sonuçlar alabilmek için AREF girişini
3.3V'a çevirdim. Arduino Uno'da bağlantı yapmayı unutma. **Bu kodu yazmazsan ve
bağlantı yaparsan Arduino'yu yakabilirsin!

  Serial.begin(9600);       // Serial bağlantı başlatılıyor.

  delay(1000);              // Gecikme

  lcd.print("Deprem ");

  lcd.setCursor(0,1);       // LCD Karakter ayarlaması için.

  lcd.print("Sensoru ");

  delay(2000);

  lcd.clear();              // LCD ekranı sil.

  lcd.print("Kalibre Ediliyor.....");

  lcd.setCursor(0,1);

  lcd.print("Bekleyin...");

  delay(100);

  pinMode(buzzer, OUTPUT);  // Takılı kalmasın diye kapalı olduklarından emin
oluyorum.

  pinMode(led, OUTPUT);

  //pinMode(led2, OUTPUT);

  buz=0;                    // Buzzer'ın pasif konumda olduğundan emin oluyorum

  digitalWrite(buzzer, buz);

```

```

digitalWrite(led, buz);

for(int i=0;i<ornekler;i++)    // Kalibrasyon için örnek alıyor.
{
    xdegeri+=analogRead(x);
    ydegeri+=analogRead(y);
    zdegeri+=analogRead(z);
}

xdegeri/=ornekler;           // X için ortalama değer alınıyor.
ydegeri/=ornekler;           // Y için ortalama değer alınıyor.
zdegeri/=ornekler;           // Z için ortalama değer alınıyor.

delay(5000);

lcd.clear();

lcd.print("Kalibrasyon ");

lcd.setCursor(0,1);

lcd.print("Basarili");

digitalWrite(led2, HIGH);

delay(1000);

lcd.clear();

lcd.print("Cihaz Hazir");

delay(1000);

lcd.clear();

lcd.print(" X Y Z ");

}

```



```

void loop()

{

int deger1=analogRead(x);    // X Çıkışı okunuyor.

int deger2=analogRead(y);    // Y Çıkışı okunuyor.

int deger3=analogRead(z);    // Z Çıkışı okunuyor.

int xDeger=xdegeri-deger1;    // X'teki değişim bulunuyor.

int yDeger=ydegeri-deger2;    // Y'teki değişim bulunuyor.

int zDeger=zdegeri-deger3;    // Z'teki değişim bulunuyor.

    /*LCD'de değerlerin gözükmesi için.*/

    lcd.setCursor(0,1);

    lcd.print(xDeger);

    lcd.setCursor(6,1);

    lcd.print(yDeger);

    lcd.setCursor(12,1);

    lcd.print(zDeger);

    delay(100);

    /* Aralıkların belirlenmesi*/

    if(xDeger < minDeg || xDeger > maxDeg || yDeger < minDeg || yDeger > maxDeg ||
zDeger < minDeg || zDeger > maxDeg) //x,y ve z değerlerinin aralıklarındaki değişim
şartları.

    {

        if(buz == 0)

            Baslat=millis();          // Zamanlayıcı

            digitalWrite(led2, LOW);    // Durum Ledinin sönmesi

```

```

buz=1;                // buzzer ve Led'in aktif olması

}

else if(buz == 1)      // Deprem alarmı verildiğinde buzzer'ın aktif olması 1 aktif
0 pasif

{

lcd.setCursor(0,0);

lcd.print("Deprem Alarmı ");

if(millis() >= Baslat+buzTime)

buz=0;

}

else

{

lcd.clear();

lcd.print("X   Y   Z ");

digitalWrite(led2, HIGH);    // Durum Ledi: stabil ve her şey çalışıyorsa aktif.

}

digitalWrite(buzzer, buz);    // buzzer aktif-pasif komutu

digitalWrite(led, buz);      // led aktif-pasif komutu

/*Değerleri serial port'una gönderiyor.*/

Serial.print("x=");

Serial.println(xDeger);

Serial.print("y=");

Serial.println(yDeger);

Serial.print("z=");

```

```
Serial.println(zDeger);

Serial.println("_");

}
```

### **Processing 3 IDE Kodları**

```
import processing.serial.*;

PFont f6, f8, f12, f10;

PFont f24;

Serial myPort; // Seri bağlantı noktası

int xPos = 0; // Grafiğin yatay konumunu ayarladım.

float y1=0;

float y2=0;

float y3=0;

void setup ()

{

    /**Pencere Boyut Ayarı ve Yazı Tipi Boyutu**/

    f6 = createFont("Arial", 6, true);

    f8 = createFont("Arial", 8, true);

    f10 = createFont("Arial", 10, true);

    f12 = createFont("Arial", 12, true);

    f24 = createFont("Arial", 24, true);

    size(1600, 800); //Pencere Boyutu

    /**Aktif olan tüm seri bağlantı noktaları**/

    println(Serial.list());
```

```

myPort = new Serial(this, "COM8", 9600); // Arduino ile bağlantı port'u COM8

println(myPort);

myPort.bufferUntil('\n');

background(80);

}

void draw ()

{

    serial ();

}

void serial()

{

    String inString = myPort.readStringUntil('_'); // Gelen veriyi porttan okuyor.

    if (inString != null)

    {

        /** 3 eksen için gereken değerlerin elde edilmesi */

        int l1=inString.indexOf("x=")+2;

        String temp1=inString.substring(l1, l1+3);

        l1=inString.indexOf("y=")+2;

        String temp2=inString.substring(l1, l1+3);

        l1=inString.indexOf("z=")+2;

        String temp3=inString.substring(l1, l1+3);

        /** X, Y ve Z değerlerinin grafik ile ölçülendirilmesi */

        float inByte1 = float(temp1+(char)9);

```

```

inByte1 = map(inByte1, -80, 80, 0, height-80);

float inByte2 = float(temp2+(char)9);

inByte2 = map(inByte2, -80, 80, 0, height-80);

float inByte3 = float(temp3+(char)9);

inByte3 = map(inByte3, -80, 80, 0, height-80);

float x=map(xPos, 0, 1120, 40, width-40);

    /*Grafıęı penceresini izdirmek iin*/

strokeWeight(2);

stroke(175);

Line(0, 0, 0, 100);

textFont(f24);

fill(0, 00, 255);

textAlign(RIGHT);

xmargin("Deprem Grafıęı (Sismograf)", 200, 100);


fill(100);

strokeWeight(100);

line(1450, 80, 1600, 80); // Gri Alan

strokeWeight(1);

textAlign(RIGHT);

fill(0, 0, 255);

String temp="X Deęeri:"+temp1;

Text(temp, 100, 95);

```

```

fill(0, 255, 0);

temp="Y Değeri:"+temp2;

Text(temp, 100, 92);

fill(255, 0, 0);

;

temp="Z Değeri:"+temp3;

Text(temp, 100, 89);

    /**X, Y ve Z değerlerini grafik üstünde çizdirme*/

strokeWeight(2);

int shift=40;


stroke(0, 0, 255);

if (y1 == 0)

    y1=height-inByte1-shift;

line(x, y1, x+2, height-inByte1-shift) ;

y1=height-inByte1-shift;

stroke(0, 255, 0);

if (y2 == 0)

    y2=height-inByte2-shift;

line(x, y2, x+2, height-inByte2-shift) ;

y2=height-inByte2-shift;

stroke(255, 0, 0);

if (y2 == 0)

```

```

    y3=height-inByte3-shift;

    line(x, y3, x+2, height-inByte3-shift) ;

    y3=height-inByte3-shift;

    xPos+=1;

    if (x >= width-30)    // Başlangıç noktasına dönüyor...

    {

        xPos = 0;

        background(80);

    }

}

}

void Line(int x1, int y1, int x2, int y2)

{

    float xx1=map(x1, 0, 100, 40, width-40);

    float xx2=map(x2, 0, 100, 40, width-40);

    float yy1=map(y1, 0, 100, height-40, 40);

    float yy2=map(y2, 0, 100, height-40, 40);

    line(xx1, yy1, xx2, yy2);

    xx2=map(100, 0, 100, 40, width-40);

    yy2=map(0, 0, 100, height-40, 40);

    line(xx1, yy1, xx2, yy2);

    strokeWeight(1);

    for (int i=1; i<21; i++)

```

```

{
    yy2=map(i*10, 0, 200, height-40, 40);

    yy1=yy2;

    line(xx1, yy1, xx2, yy2);
}

yy2=map(100, 0, 100, height-40, 40);

yy1=map(0, 0, 100, height-40, 40);

for (int i=1; i<41; i++)
{
    xx1=map(i*5, 0, 200, 40, width-40);

    xx2=map(i*5, 0, 200, 40, width-40);

    line(xx1, yy1, xx2, yy2);
}

textAlign(RIGHT);

// sonuc+=yy1; XXX

fill(255);

strokeWeight(1);

textFont(f12);

for (int i=-10; i<11; i++)
{
    String sonuc="";

    sonuc+=5*i;

    ymargin(sonuc, x1, y1);
}

```



```

        y1+=5;
    }

    x1=0;

    y1=0;

    strokeWeight(1);

    textFont(f10);

    for (int i=0; i<41; i++)
    {

        String sonuc="";

        sonuc+=28*3*i;

        xmargin(sonuc, x1, y1);

        x1+=5;

    }

    textAlign(RIGHT);

    textAlign(RIGHT);

}

void ymargin(String deger, int x1, int y1)
{

    float xx1=map(x1, 0, 100, 40, width-40);

    float yy1=map(y1, 0, 100, height-40, 40);

    text(deger, xx1-5, yy1+5);

}

```

```

void xmargin(String deger, int x1, int y1)
{

    float xx1=map(x1, 0, 200, 40, width-40);
    float yy1=map(y1, 0, 100, height-25, 25);
    text(deger, xx1+7, yy1);
}

```

```

void Text(String deger, int x1, int y1)
{

    float xx1=map(x1, 0, 100, 40, width-40);
    float yy1=map(y1, 0, 100, height-25, 25);
    text(deger, xx1, yy1);
}

```