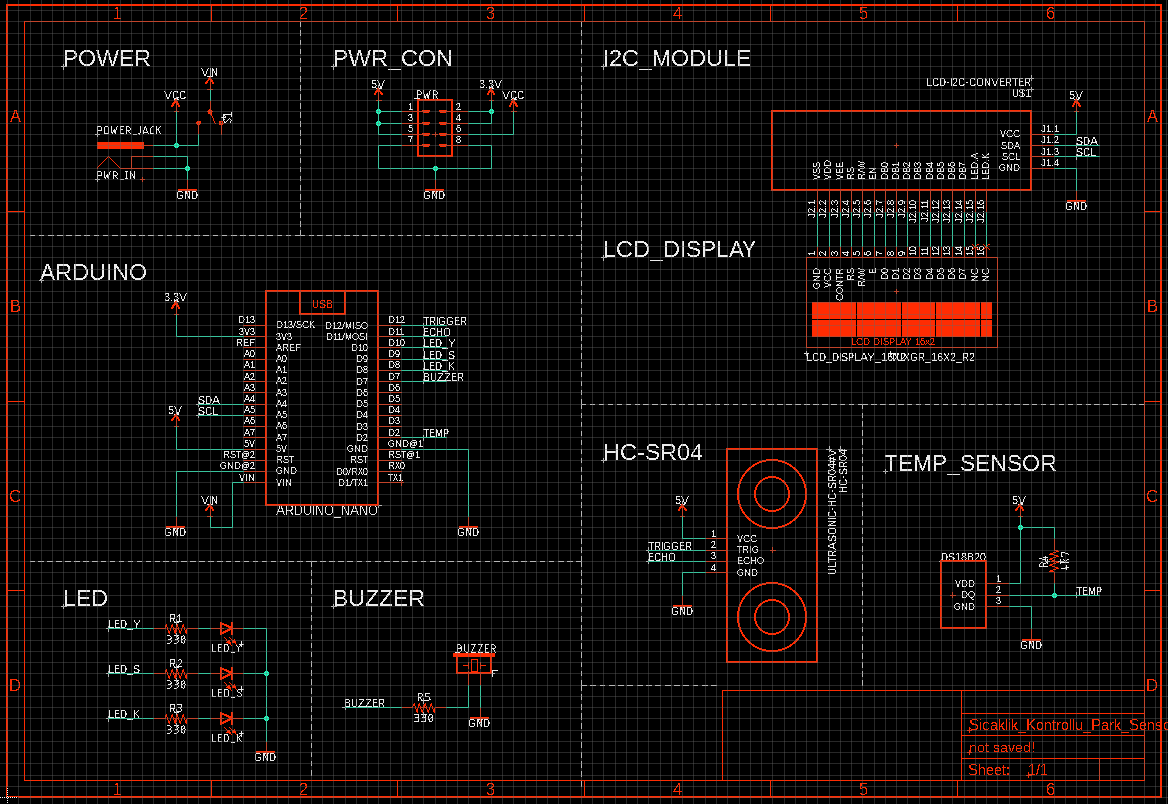
**Proje: Sıcaklık Kontrollü Park Sensörü**

Sıcaklık Kontrollü Park Sensörü Projesi, ultrasonik mesafe sensörlerinin ölçümlerini ses dalgalarının karşıdaki nesneye çarpıp gelme süresi ile yaptıklarından dolayı sesin hızının havanın sıcaklığındaki değişimden etkilenerek sensörden ölçülen mesafe değerindeki hatayı minimize etmek amacıyla bir sıcaklık sensörü yardımıyla havanın sıcaklığının ölçülmesi ve sesin o anki havadaki hızının bu sıcaklık değerine bağlı olarak hesaplanması ve mesafe sensöründe kullanılan ses dalgasının gidip gelme süresine bağlı olan mesafenin hesaplanması olarak açıklanabilir.

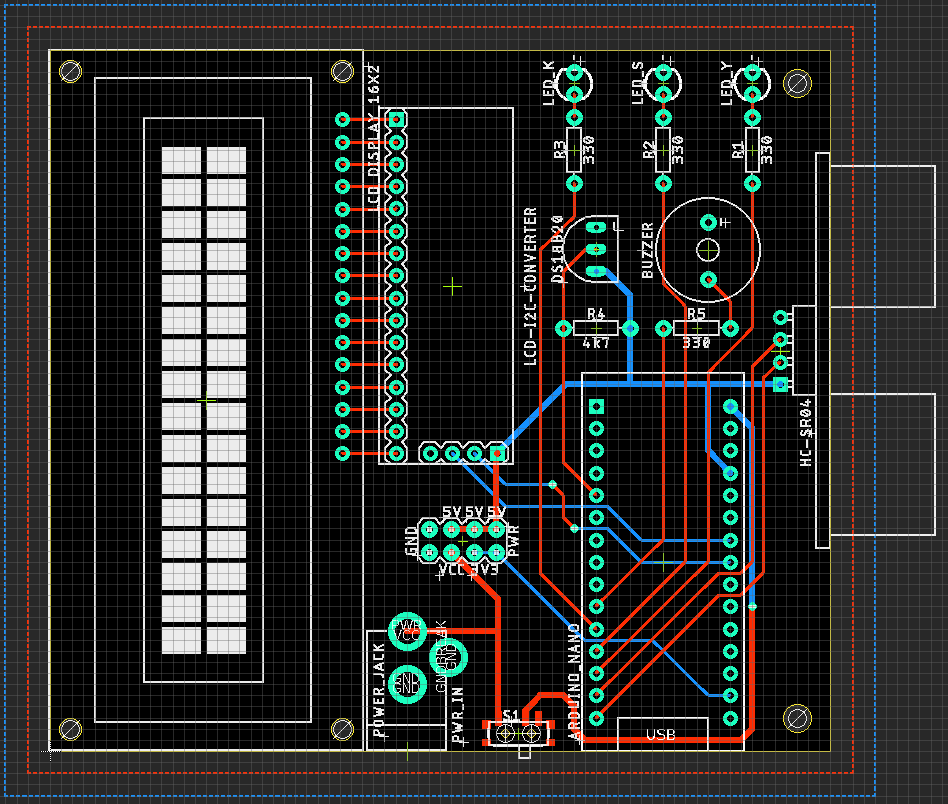
Projede bir sıcaklık sensörü ile sıcaklık ölçülür, ölçülen değer yardımıyla mesafe sensörü ile mesafe ölçümü yapılır. Bu değerler bir LCD ekrana yazdırılır ve kırmızı sarı yeşil ledler ile mesafenin güvenli olup olmadığı teyit edilir. Buzzer ise güvenli olmayan bölgeye girildiği zaman uyarı vermek amacıyla öter. Bütün bu işlemler ise bir Arduino Nano ile kontrol edilir.

**Projede kullanılan malzemeler:**

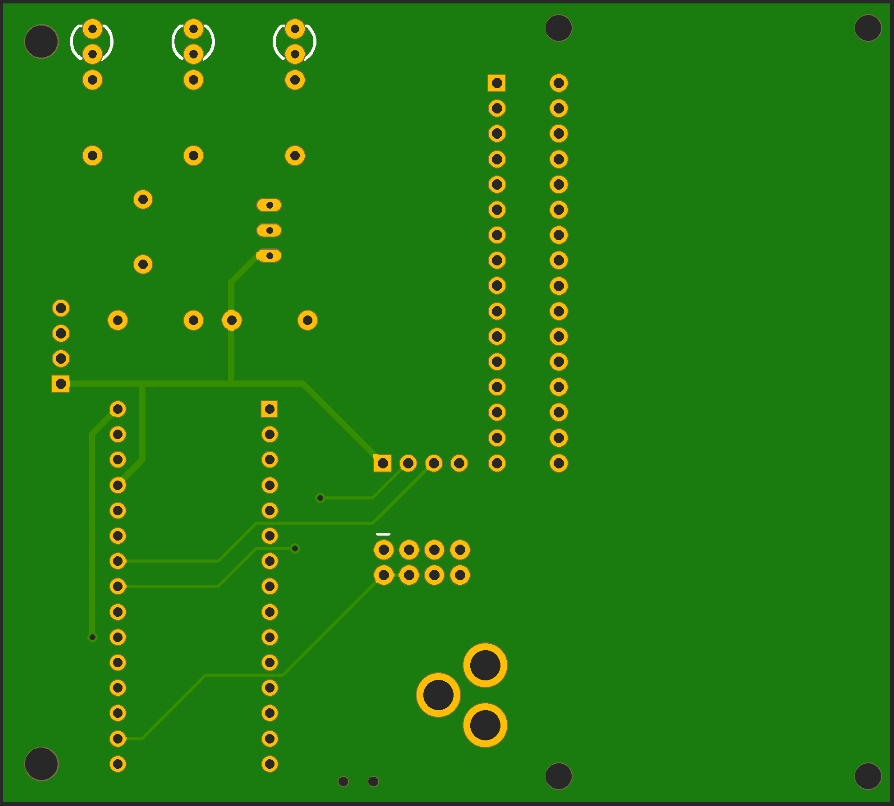
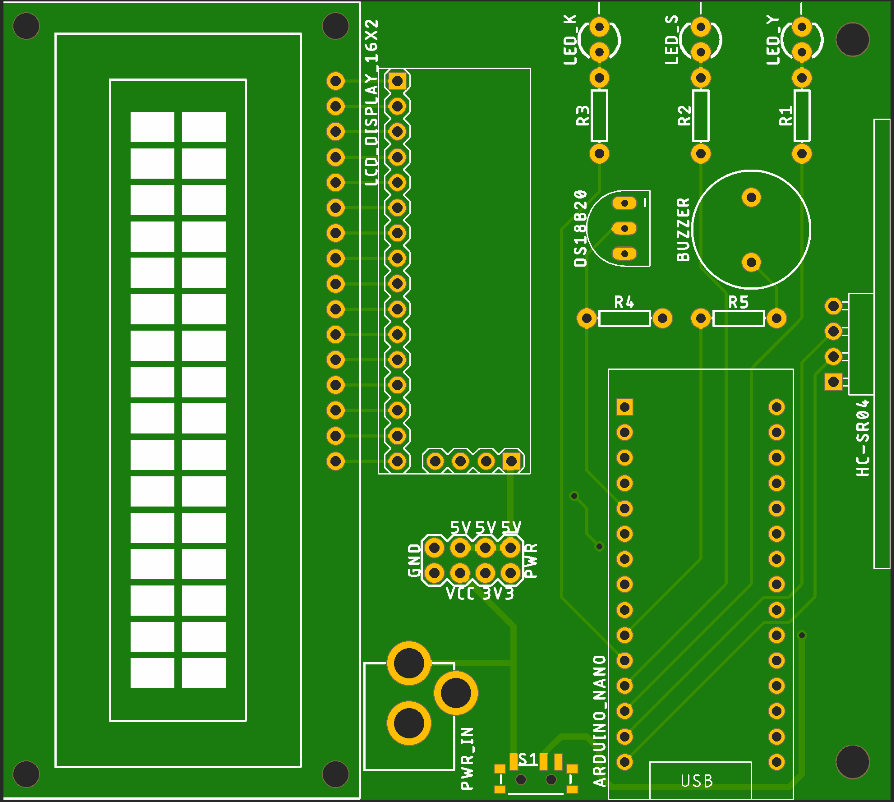
1. 1xArduino Nano
2. 1xHC-SR04 Ultrasonik Mesafe Sensörü
3. 1xDS18B20 Sıcaklık Sensörü
4. 1xLCD Display (16x2)
5. 1xI2C Habeleşme Modülü
6. 1xBuzzer
7. 3xLed (Kırmızı, Sarı, Yeşil)
8. 4x330Ohm ve 1x4.7KOhm Direnç
9. 1xPower Jack
10. 1xConnector (4x2)
11. 1xSwich



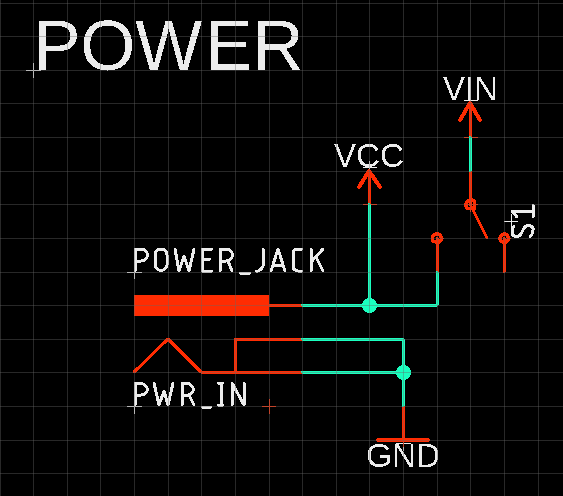
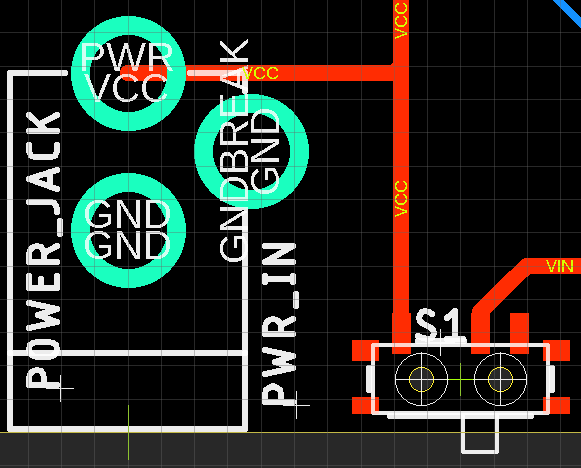
Projenin Şematik Çizimi



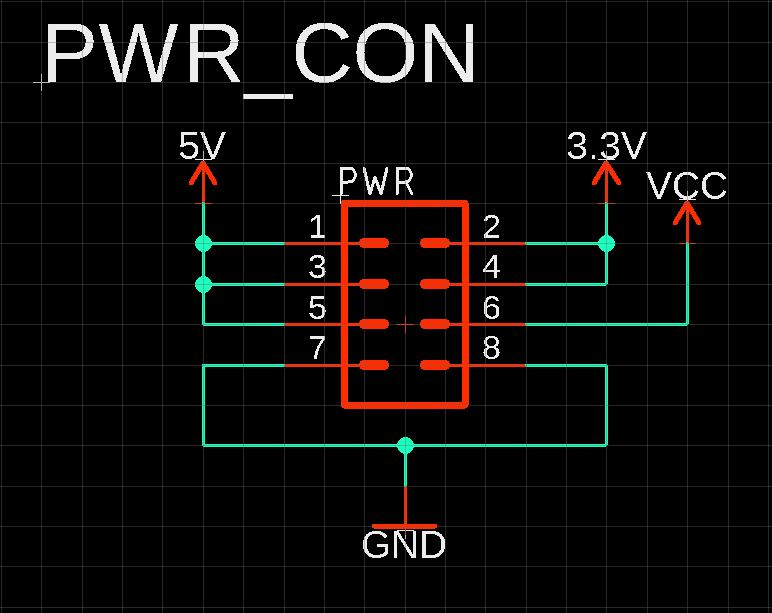
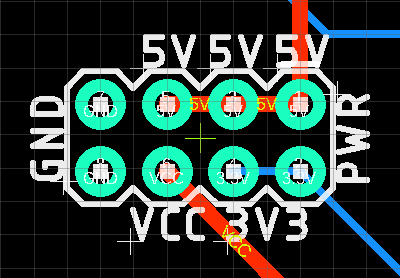
Projenin PCB Çizimi



PCB’nin Ön ve Arkadan Görünümü

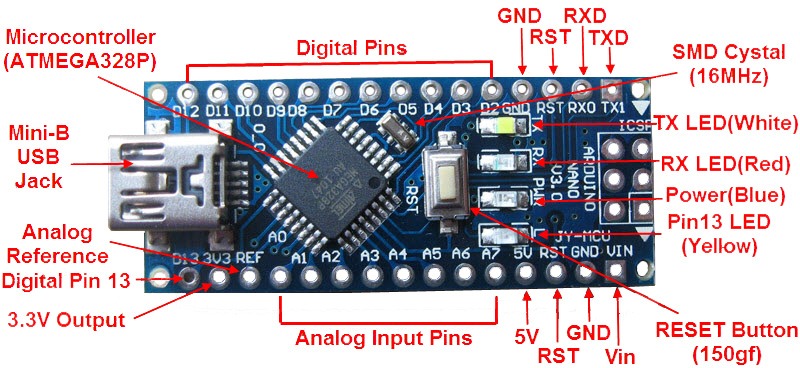
Devrenin giriş kısmı, kart üzerindeki Arduino Nanonun bir power jack yardımıyla 12V’luk bir gerilim ile beslendiği ve açma kapama için bir switch bulunan kısımdır.

Devrenin konnektör kısmı, kart üzerinden 5V gerilim alabileceğimiz 3 pin, 3.3V gerilim alabileceğimiz 2 pin, Arduino Nanonun beslenebileceği VCC pinini ve 2 adet Ground pini içeren kısımdır.

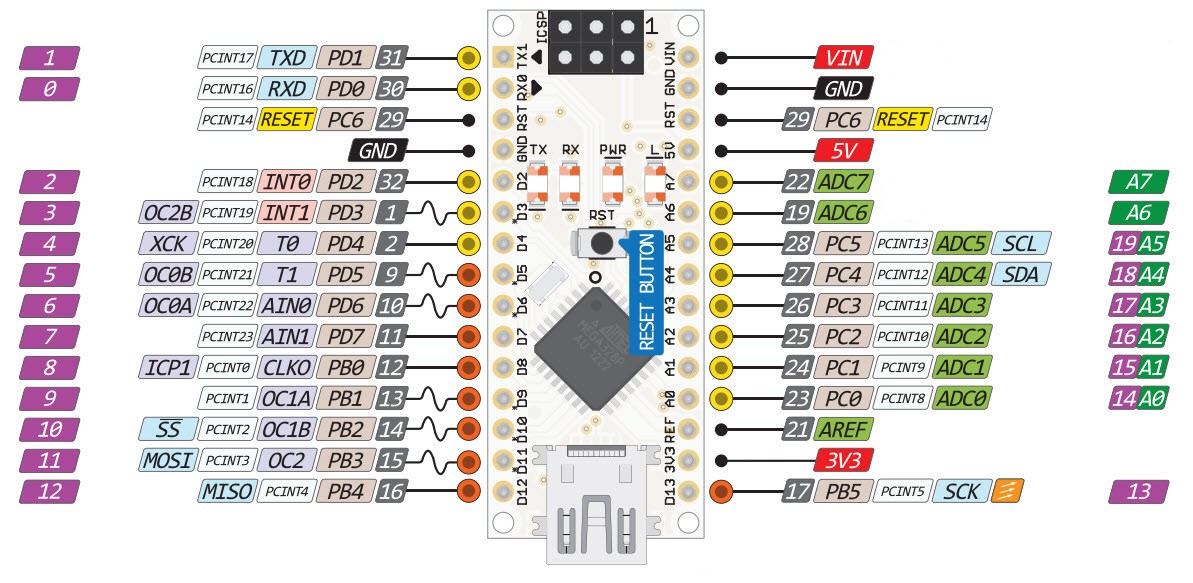
1. **Arduino Nano**

Arduino Nano devre üzerindeki sensörleri, ledleri, buzzerı, ve LCD ekranı control ettiğimiz modüldür.



**Teknik Özellikler:**

* Mikrodenetleyici ATmega328
* Çalışma Gerilimi 5V
* Giriş Gerilimi (önerilen) 7-12V
* Giriş Gerilimi (limit) 6-20V
* Dijital I/O Pinleri 14 (6 tanesi PWM çıkışı)
* Analog Giriş Pinleri 8
* Her I/O için Akım 40 mA
* 3.3V Çıkış için Akım 50 mA
* Flash Hafıza 32 KB (ATmega328) 2 KB kadarı bootloader tarafından kullanılmaktadır
* SRAM 2 KB (ATmega328)
* EEPROM 1 KB (ATmega328)
* Saat Hızı 16 MHz
* Uzunluk 45 mm
* Genişlik 18 mm
* Ağırlık 5g



**Güç:**

Arduino Nano gücünü usb üzerinden veya harici güç kaynağından alabilir. Harici güç kaynağı AC-DC adaptör olabileceği gibi bataryada olabilir. Adaptör ve batarya kart üzerindeki GND ve Vin pinleri üzerinden bağlanabilir.

Kartın çalışması için sürekli olarak usb'nin bağlı olması şart değildir. Kart sadece adaptör veya batarya ile çalıştırılabilir. Bu sayede kart bilgisayardan bağımsız olarak çalıştırılabilir.

Harici güç kaynağı olarak 6-20V arası kullanılabilir. Ancak bu değerler limit değerleridir. Kart için önerilen harici besleme 7-12V arasıdır. Çünkü kart üzerinde bulunan regülatör 7V altındaki değerlerde stabil çalışmayabilir. 12V üstündeki değerlerde de aşırı ısınabilir.

Nano kartının üzerindeki mikrodenetleyicinin çalışma gerilimi 5V'dur. Vin pini veya güç soketi üzerinden verilen 7-12V arası gerilim kart üzerinde bulunan voltaj regülatörü ile 5V'a düşürülerek karta dağılır.

Güç pinleri aşağıdaki gibidir:

VIN: Harici güç kaynağı kullanılırken 7-12V arası gerilim giriş pini.  
 5V: Bu pin regülatörden çıkan 5V çıkışı verir.

Eğer kart sadece usb (5V) üzerinden çalışıyor ise usb üzerinden gelen 5V doğrudan bu pin üzerinden çıkış olarak verilir. Aynı zamanda bu pin üzerinden 5V girişi yapılabilir.  Eğer karta güç Vin (7-12V) üzerinden veriliyorsa regülatörden çıkan 5V doğrudan bu pin üzerinden çıkış olarak verilir.

3V3: Kart üzerinde bulunan 3.3V regülatörü çıkış pinidir. Maks. 50mA çıkış verebilir.

GND: Toprak pinleridir.

**Hafıza:**

Atmega328 32 KB'lık flash belleğe sahiptir (2 KB kadarı bootloader tarafından kullanılmaktadır).  2 KB SRAM ve 1 KB EEPROM'u bulunmaktadır.

**Giriş ve Çıkış:**

Nano üzerindeki 14 adet dijital pinin hepsi giriş veya çıkış olarak kullanılabilir. 8 tane analog giriş pinide bulunmaktadır. Bu analog giriş pinleride aynı şekilde dijital giriş ve çıkış olarak kullanılabilir. Yani kart üzerinde toplam 20 tane dijital giriş çıkış pini vardır. Bu pinlerin tamamının lojik seviyesi 5V'dur. Her pin maks. 40mA giriş ve çıkış akımı ile çalışır. Ek olarak, bazı pinlerin farklı özellikleri bulunmaktadır. Özel pinler aşağıda belirtildiği gibidir.

Seri Haberleşme, D0 (RX) ve D1 (TX): TTL Seri veri alıp (RX), vermek (TX) için kullanılır. Bu pinler doğrudan kart üzerinde bulunan FT232 usb-seri dönüştürücüsüne bağlıdır. Yani bilgisayardan karta kod yüklerken veya bilgisayar-nano arasında karşılıklı haberleşme yapılırkende bu pinler kullanılır. O yüzden karta kod yüklerken veya haberleşme yapılırken hata olmaması için mecbur kalınmadıkça bu pinlerin kullanılmamasında fayda vardır.

Harici Kesme, D2 (interrupt 0) ve D3 (interrupt 1): Bu pinler yükselen kenar, düşen kenar veya değişiklik kesmesi pinleri olarak kullanılabilir.

PWM, D3,D5,D6,D9,D10 ve D11: 8-bit çözünürülükte PWM çıkış pinleri olarak kullanılabilir.

SPI, D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK): Bu pinler SPI haberleşmesi için kullanılır.

LED, D13: Nano üzerinden 13. pine bağlı olan dahili bir led bulunmaktadır. Pin HIGH yapıldığında led yanacak, LOW yapıldığında led sönecektir.

Analog, A0-A7: Nano 8 tane 10-bit çözünürlüğünde analog giriş pinine sahiptir. Bu pinler dijital giriş ve çıkış içinde kullanılabilir. Pinlerin ölçüm aralığı 0-5V'dur. AREF pini ve analogReference() foksiyonu kullanılarak alt limit yükseltilip, üst limit düşürülebilir.

I2C, A4 veya SDA pini ve A5 veya SCL pini: Bu pinler I2C haberleşmesi için kullanılır.

AREF: Analog giriş için referans pini.

Reset: Mikrodenetleyici resetlenmek istendiğinde bu pin LOW yapılır. Reset işlemi kart üzerinde bulunan Reset Butonu ile de yapılabilir.

**Haberleşme:**

Arduino Nano'nun bilgisayarla, başka bir arduino veya mikrodenetleyici ile haberleşmesi için bir kaç farklı seçenek vardır. Atmega328, 0 (RX) ve 1 (TX) pinleri üzerinden UART TTL (5V) seri haberleşme imkanı sunar. Kart üzerinde bulunan FT232 usb-seri dönüştürücüde bilgisayarda sanal bir com port açarak Atmega328 ile bilgisayar arasında bir köprü kurar. Arduino bilgisayar programı içerisinde barındırdığı seri monitör ile arduino ile bilgisayar arasında text temelli bilgilerin gönderilip alınmasını sağlar. Usb-seri dönüştürücü ile bilgisayar arasında usb üzerinden haberleşme olduğu zaman kart üzerinde bulunan RX ve TX ledleri yanacaktır.

Nano üzerinde donanımsal olarak bir adet seri port bulunmaktadır. Ancak [SoftwareSerial kütüphanesi](http://www.arduino.cc/en/Reference/SoftwareSerial" \t "_blank) ile bu sayı yazılımsal olarak arttırılabilir.

Atmega328 aynı şekilde I2C ve SPI portlarıda sağlamaktadır. Arduino bilgisayar programı ile gelen [Wire kütüphanesi](http://www.arduino.cc/en/Reference/Wire" \t "_blank) I2C kullanımını, [SPI kütüphanesi](http://www.arduino.cc/en/Reference/SPI) de SPI haberleşmesini sağlamak için kullanılır.

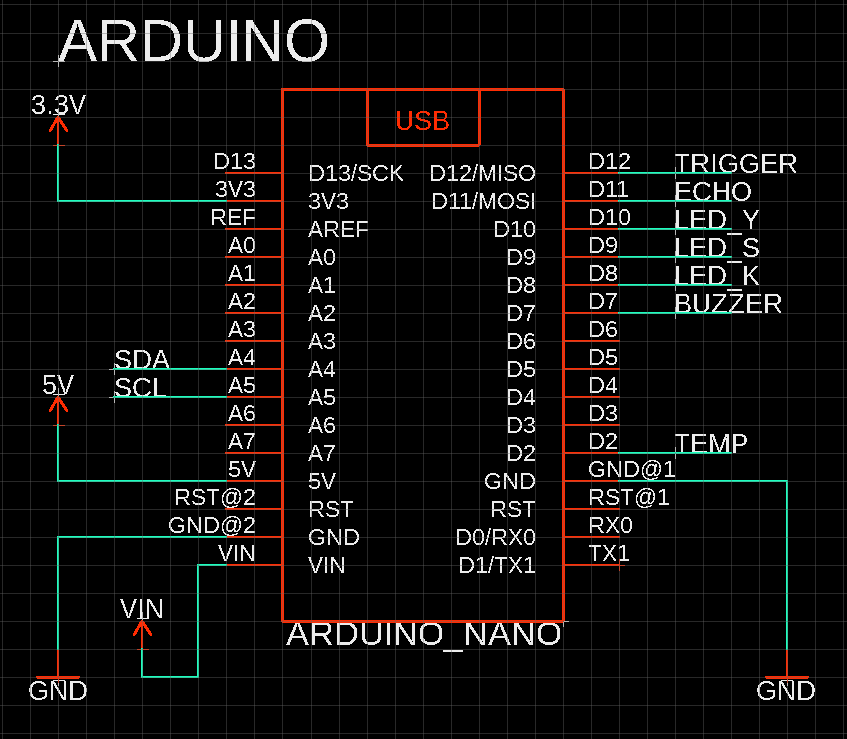
**Programlama:**

Arduino Nano kartı [Arduino bilgisayar programı](http://www.arduino.cc/en/Main/Software) (Arduino IDE) ile programlanır. Programda Tools > Board sekmesi altında Arduino Nano'yu seçip programlamaya başlayabilirsiniz. Ayrıntılı bilgi için [referans](http://www.arduino.cc/en/Reference/HomePage" \t "_blank)ve [temel fonksiyonlar](http://www.arduino.cc/en/Tutorial/HomePage" \t "_blank) sayfasını inceleyebilirsiniz. Arduino Nano üzerindeki Atmega328 üzerine [bootloader](http://www.arduino.cc/en/Hacking/Bootloader?from=Tutorial.Bootloader) denilen özel bir yazılım yüklü gelir. Bu sayede kartı programlarken ekstra bir programlayıcı kullanmanıza gerek yoktur.Haberleşme orjinal [STK500](http://www.atmel.com/dyn/resources/prod_documents/doc2525.pdf) protokolü ile sağlanır.

Bootloader yazılımı bypass edilerek kart doğrudan mikrodenetleyicinin ICSP header'i üzerinden [ISP programlayıc](https://www.robotistan.com/USBtinyISP-AVR-Programlayici-Karti-Arduino-Bootloader-Programlayici,PR-2123.html)ı ile programlanabilir.

**Atmega328 Özellikleri:**

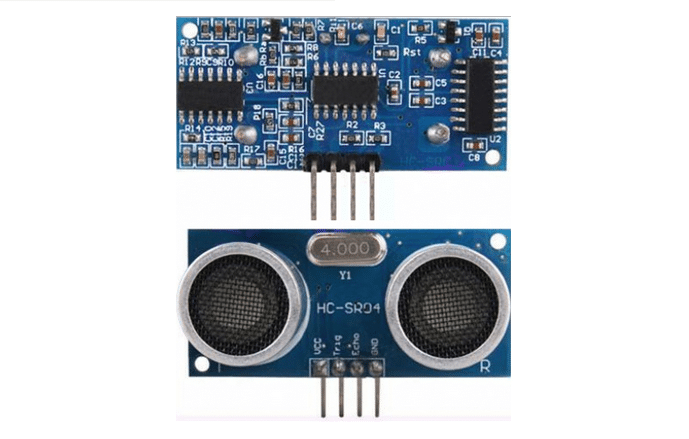
* 131 işlemci komutu
* Çoğu komut tek çevirimde çalışır. (Single cycle)
* 32×8 Genel kullanım yazmacı
* 20MHz’e kadar saat hızı
* Entegre 2 çevirim çarpıcı (multiplier)
* 32KB Programlanabilir FLASH program hafızası
* 1KB EEPROM
* 2KB Dahili SRAM Bellek
* 10.000 Flash / 100.000 EEPROM okuma-yazma kapasitesi
* 85 derecede 20 yıl, 25 derecede 100 yıl veri koruma.
* Kilit Bitleri ve Kod koruma
* İki adet 8-bit zamanlayıcı ve sayıcı ölçeklendirme ve karşılaştırma modlarıyla.
* Bir adet 16-bit zamanlayıcı ve sayıcı ölçeklendirme, yakalama ve karşılaştırma modlarıyla.
* Gerçek zaman sayıcı ayrı osilatör ile
* Altı PWM kanalı
* 6 kanal 10-bit ADC, Sıcaklık ölçümü
* İki adet ana/uydu SPI Seri Arayüz
* Bir adet programlanabilir Seri USART
* Bir I2C arayüzü
* Programlanabilir WDT
* Bir Analog Karşılaştırıcı
* Ayağa bağlı uyandırma ve kesme
* Açılışta reset ve kararma (brown-out) saptaması
* Dahili ayarlı osilatör
* Harici ve Dahili Kesme Kaynakları
* Altı uyku modu
* 23 programlanabilir giriş ve çıkış ayağı
* 1.8 – 5.5 V arası çalışma gerilimi
* -40 ve +105C arası çalışma sıcaklığı
* Aktif modda 0.2mA
* Power-down modunda 0.1uA
* Power-save  modunda 0.75uA. Güç tüketimi. (Buna diğer unsurlar dahil değildir.)



Arduino Nano Şematik Gösterimi

* 3.3V bağlantısı yardımıyla konnektör üzerinden 3.3V gerilim alınabilmektedir.
* SDA ve SCL bağlantıları ile LCD Ekran için kullanılan I2C Modülünün bağlantısı gerçekleşmektedir.
* 5V bağlantısı yardımıyla konnektör üzerinden 5V gerilim alınabilmektedir.
* GND bağlantıları Arduino Nanonun ground pinleridir.
* VIN bağlantısı power jack ve konnektör ile bağlantılıdır ve Arduino Nanoyu beslemek için kullanılır.
* TRIGGER bağlantısı HC-SR04 Ultrasonik Mesafe Sensörü üzerinden ses dalgasının gönderilmesine yarayan Trigger pini bağlantısıdır.
* ECHO bağlantısı HC-SR04 Ultrasonik Mesafe Sensörü üzerinden gönderilen ses dalgasının algılanmasına yarayan Echo pini bağlantısıdır.
* LED\_Y, LED\_S ve LED\_K bağlantıları sırasıyla Yeşil, Sarı ve Kırmızı ledlerin bağlantısıdır.
* TEMP bağlantısı DS18B20 Sıcaklık Sensörünün bağlantısıdır.

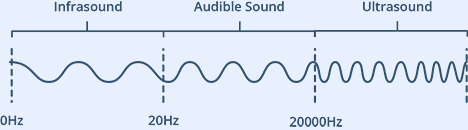
1. **HC-SR04 Ultrasonik Mesafe Sensörü**



HC-SR04 Ultrasonik Mesafe Sensörü, genelde robotik projelerde engel tanımlama amacıyla kullanılan hesaplı bir mesafe sensörüdür. HC-SR04 Ultrasonik Mesafe Sensörü projede çevresel farkındalık kazandırarak engellere çarpmasını önlemek için park sensörü olarak da kullanılmıştır.

HC-SR04 Ultrasonik Mesafe Sensörü karşısındaki nesneleri belirlemek için yarasalar ve yunuslar gibi sonar kullanır. Kolay kullanımıyla yüksek mesafe ve doğrulukta düzenli veri akışı sunar. Performansı kızılötesi sensörlerde olduğu gibi güneş ışığından ve koyu malzemelerden etkilenmez ancak akustik açıdan düşük iletimi olan malzemelerin fark edilmesi zor olabilir.

Ultrasonik mesafe sensörü, insan işitmesinin işitilebilir sınırından daha yüksek frekanslara sahip yüksek perdeli ses dalgaları yayan, geri gelen dalganın dönüş süresine göre cime olan uzaklık mesafesini ölçen cihazdır.



İnsan kulağı saniyede yaklaşık 20 kez (derin bir gürültü) ile saniyede yaklaşık 20.000 kez (yüksek tiz bir ıslık) titreşen ses dalgalarını duyabilir. Bununla birlikte, ultrasonik ses dalga boyları 20.000 Hz’den daha fazla frekanslara sahiptir. Ancak 20.000 den daha yüksek frekansları insan kulağı duyamaz.

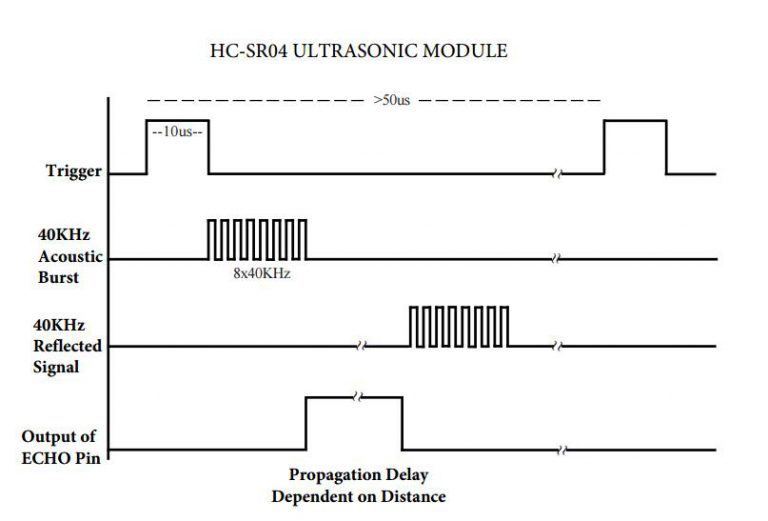
HC-SR04 Ultrasonik mesafe sensörü iki [ultrasonik dönüştürücüden](https://tr.qwe.wiki/wiki/Ultrasonic_transducer" \t "_blank) oluşur. Dönüştürücüler (alıcı-verici), elektrik sinyalini 40 KHz ultrasonik ses darbelerine dönüştüren bir verici görevi görür. Alıcı, iletilen darbeleri dinler. Gelen dalgaların dönüş süresine göre mesafeyi ölçer.

Trigger pinine en az 10 µS (10 mikrosaniye) süreli bir **pulse** uygulandığında başlar. Buna yanıt olarak, sensör 40 KHz’de 8 darbeden oluşan bir ses patlaması iletir. Bu 8 darbe, cihazın benzersiz ses imzası ile çıkar ve alıcının gelen özel ses dalgalarını, ortam gürültüsünden ayırt etmesini sağlar.8 ultrasonik ses darbesi vericiden çıktıktan ve cisme çarpıp geri gelip ECHO pinine ulaştıktan sonra, ECHO pini sinyalin başlangıcını oluşturmaya başlamak için HIGH olur.

Giden ses dalgaları geri gelmezse, ECHO sinyali 38 mS (38 milisaniye) sonra zaman aşımına uğrar ve azalır. Böylece 38 mS’lik bir darbe sensör aralığında herhangi bir engel olmadığını gösterir.

Darbeler bir nesneye çarpıp geri gelirse, sinyal alınır alınmaz ECHOLOW olur. Bu olay, sinyalin alınması için geçen süreye bağlı olarak genişliği 150 uS ile 25 mS arasında değişen bir darbe üretir.

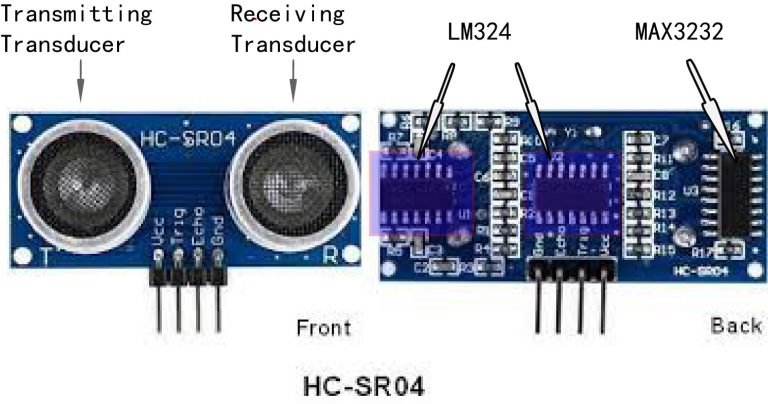
Alınan darbenin gidiş ve geliş süresi arasında geçen zaman, nesneye olan mesafeyi hesaplamak için kullanılır.



formülü ile mesafe hesaplanır.

Zaman değişkeni, ultrasonik darbenin sensörden çıkması, nesneden sekmesi ve sensöre geri dönmesi için geçen süredir. Aslında bu süreyi yarıya bölüyoruz, çünkü nesneye olan mesafeyi değil, sadece sensöre olan mesafeyi ölçmemiz gerekiyor. Hız değişkeni, sesin havadan geçtiği hızdır. Havadaki ses hızı sıcaklık ile değişir. Bu nedenle, mesafeyi doğru bir şekilde hesaplamak için ortam sıcaklığını dikkate almamız gerekir.

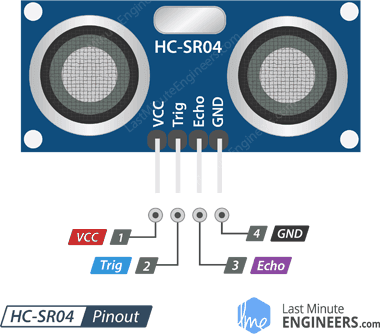
Hesaplanan sıcaklık ile havadaki ses hızı için formül: 331x√(1+T/273)



Ultrasonik mesafe bulucunun önünde iki metal silindir bulunur. Bunlar dönüştürücülerdir. Transdüserler mekanik kuvvetleri elektrik sinyallerine dönüştürür. Ultrasonik mesafe bulucuda, verici bir dönüştürücü ve alıcı dönüştürücü vardır. İleten dönüştürücü bir elektrik sinyalini ultrasonik darbeye dönüştürür ve alıcı dönüştürücü yansıyan ultrasonik darbeyi tekrar elektrik sinyaline dönüştürür. Telemetre arkasına bakarsanız, MAX3232 etiketli verici dönüştürücünün arkasında bir IC göreceksiniz. Bu, verici dönüştürücüyü kontrol eden IC'dir. Alıcı dönüştürücünün arkasında LM324 etiketli bir IC bulunur. Bu, alıcı dönüştürücü tarafından üretilen sinyali Arduino'ya iletecek kadar güçlü bir sinyale yükselten dörtlü bir Op-Amp'dir.

**Arduino HC-SR04 Ultrasonik Mesafe Sensörü Teknik Özellikleri**

* Çalışma Voltajı: DC 5V
* Çektiği Akım: 15 mA
* Çalışma Frekansı: 40 kHz
* Maksimum Görme Menzili: 4m
* Minimum Görme Menzili: 2cm
* Görme Açısı: 15°
* Tetik Bacağı Giriş Sinyali: 10 us TTL Darbesi
* Echo Çıkış Sinyali: Giriş TTL sinyali ve Mesafe Oranı
* Boyutları: 45mm x 20mm x 15mm
* Çözünürlük: 0.3cm
* Ağırlık: 9g

**[](https://hayaletveyap.com/wp-content/uploads/2018/04/Arduino-ile-hcsr04-ultrasonik-mesafe-sensoru-kullanimi-pinout.png)**

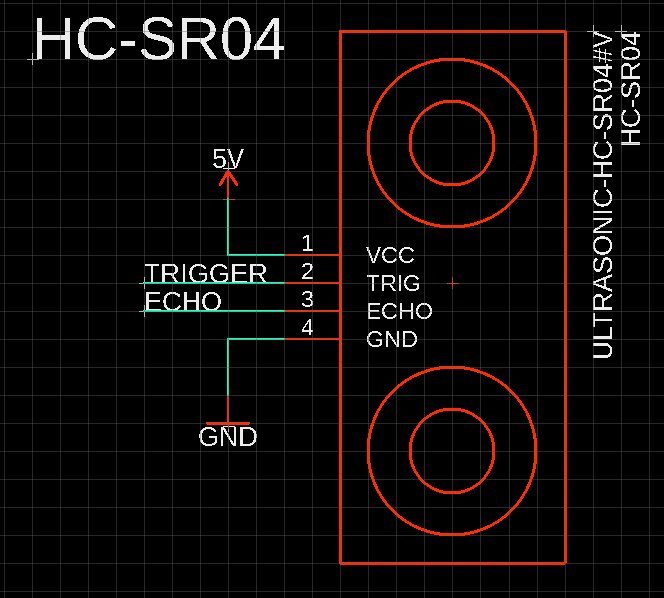
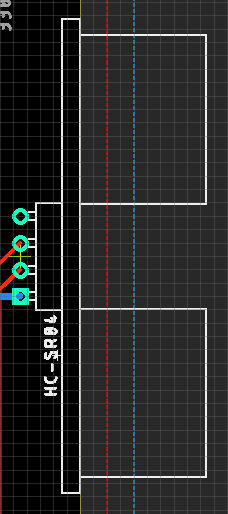
HC-SR04 Ultrasonik Mesafe Sensörü Pinout

VCC Arduino’ya 5V pin bağladığımız HC-SR04 Ultrasonik mesafe sensörü için güç kaynağıdır.

Trig (Trigger) pini, ultrasonik ses sinyallerini tetiklemek için kullanılır.

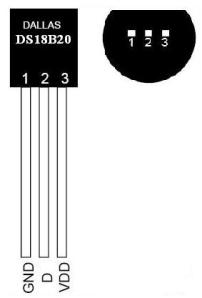
ECHO pini, yansıtılan sinyal alındığında bir BPM üretir. Pulse uzunluğu, iletilen sinyalin algılanması için geçen süre ile orantılıdır.

GND Arduino topraklarına bağlıdır.

HC-SR04 Ultrasonik Mesafe Sensörünün Şematik ve PCB Görünümü

1. **DS18B20 Dijital Sıcaklık Sensörü**



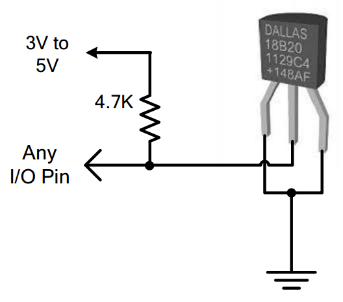
DS18B20 Sıcaklık Sensör Entegresi içinde bulunan sıcaklık sensör devresi sayesinde bulunduğu ortamın sıcaklığını ölçmektedir. Hassas ölçüm gerektiren uygulamalarda, endüstriyel sistemlerde DS18B20 sensörü kullanılabilir.

DS18B20 dijital bir sıcaklık sensörüdür. en önemli üstünlüğü hassasiyetinin yüksek olmasıdır ve analog değil dijital bir sensör olmasıdır. Bu sayede daha kararlı bir sıcaklık ölçümü gerçekleşmektedir. Ancak bu sensörün kütüphanesiz ve datasheet bilgilerinden yola çıkarak kullanılmaya çalışılması uğraş ve emek gerektirmektedir. Sebebi ise kullanılan 1-Wire (OneWire) yani tek hat üzerinden iletişim protokolüdür. I2C protokolüne benzeyen 1-Wire iletişim protokolü hem biraz daha yavaş çalışmaktadır hemde Arduinoda  1-Wire donanımı olmadığından bu iletişim Arduinoda yazılımsal çalışmaktadır. Yazılımsal çalışmasının dezavantajı gönderilen verilerin bazen gecikmesi gecikmeden dolayı bazı bitlerin hatalı yollanabilmesi gibi sorunlarla karşılaşılabilmektedir.

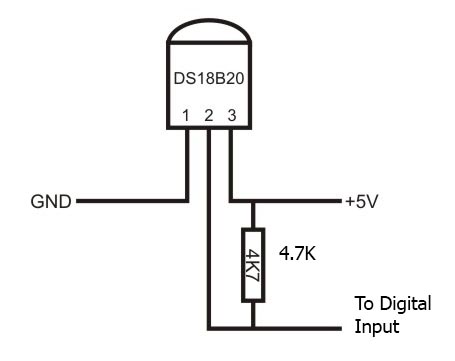
**DS18B20 Dijital Sıcaklık Sensörü Teknik Özellikleri**

* Giriş Gerilimi: 3V - 5.5V
* Sıcaklık Hassasiyeti: ± 0.5
* Sıcaklık Ölçüm Aralığı: -55 ° C - +125 ° C
* Çözünürlük: 9-12 bit
* Arayüz: One-Wire
* 1-Wire arayüzü ile haberleşme için sadece bir port pini gereklidir.
* Her sensör kendine ait benzersiz 64 bit seri numarasına sahiptir.
* Azami çevrim süresi(12 bit): 750 ms
* Paket Tipi: TO-92
* Montaj Tipi: DIP
* Pin Sayısı: 3

Sensör, iki farklı bağlantı şekline sahiptir. Birincisi parasitic power mode ( Bu mod şemada görüldüğü şekilde) bağlandığından, başlangıç enerjisini DATA ucundan alır, içerisindeki kondansatörü şarj eder, sonrasında ise aldığı şarjı kullanarak, hem sıcaklık değerlerini gönderir, hem de içerisindeki elektrik kesildiğinde kaybolabilecek değerleri kendi eeprom’una kayıt etmekte kullanır. Bu şekilde bağlantı tekli kullanımda sorun yaratmasa da çoklu sensör kullanımında sorunlar yaratabileceğinden çoklu sistemlerde harici bağlantı kullanılması gerekir.



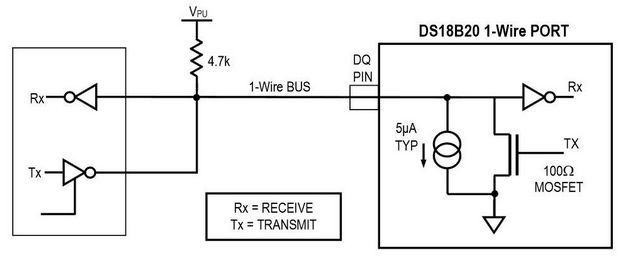
İkincisi ise Harici bağlantıdır. Bu bağlantıda ise Vdd ucuna +3V-5V bağlanır.  
Her iki sistemde de PULL UP direnci ihmal edilmemelidir. Pull up direnci 1,5ma akımdan daha fazla çekilmeyecek şekilde hesaplanabilir ve master ve slave cihaz arasındaki mesafeye göre değişkenlik göstermektedir. Eğer sensör mikrodenetleyiciden uzak bir yerde konumlandırılacaksa direnç değerini düşürmek gerekmektedir.



Bu sensörler çoklu kullanım olarak kullanılır Aşağıdaki devrede olduğu gibi çoklu bağlantı yapılır. DATA ucu ile tek bir haberleşme hattından kontrol edilir.

**One-Wire Haberleşmesi**

1-Wire bir master tarafından kontrol edilen bir veya daha fazla slave cihazın bağlanabildiği haberleşme protokolüdür. DS18B20, DS1822 gibi sensörler slave olarak hatta bağlanırlar. 1-wire haberleşmede master çoğu zaman mikrodenetleyicidir. Hatta tek slave cihaz bağlı ise bu sistem single-drop olarak adlandırılır. Eğer hatta birden çok slave cihaz var ise sistem multi-drop olarak adlandırılır. 1-wire haberleşmede tüm komutlar ve veriler LSB ilk olarak hat üzerinden gönderilir.

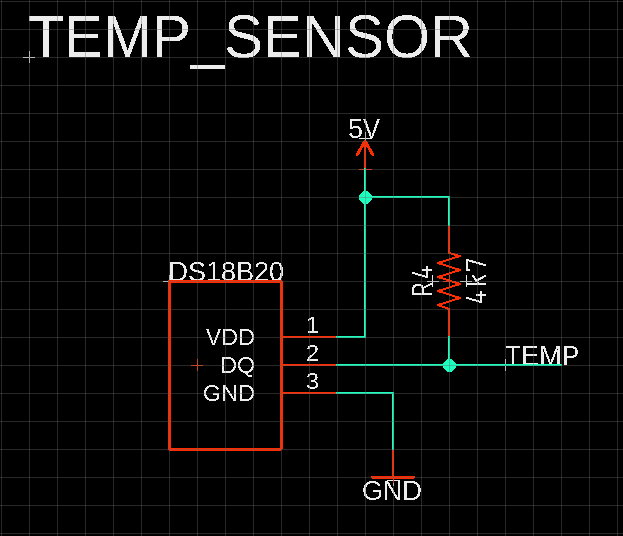
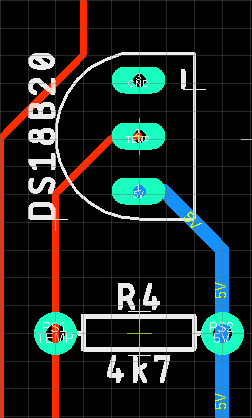


1-Wire İletişim Donanım Yapısı

Şekilde 1-Wire iletişim için gerekli donanım yapısı görülmektedir. 1-Wire iletişim için master ve slave cihazların port yapıları open-drain yada 3-durumlu yapıda olmadır. Böylece hattı kullanmayan cihazlar yüksek empedans konumuna geçerek hattı boş bırakabiliriler. DS18B20 sensörünün DQ haberleşme pini opendrain yapıdadır. 1-Wire haberleşmede hattın boşta olma durumu bir olma(yüksek) durumudur. Hattı bire(yüksek) çekmek için ortalama 5K bir direnç kullanılmalıdır. 1-Wire haberleşmede eğer haberleşme hattı 480us’de fazla düşük seviyede tutulursa hattaki tüm cihazlar kendini resetler. 1-Wire ile DS18B20 haberleşmesi için işlem sırası aşağıda ki gibidir.

**İşlem Sırası**  
 1. Başlatma  
 2. ROM Komutu  
 3. DS18B20 Fonksiyon Komutu

DS18B20 için yukarıdaki işlem sırası çok önemlidir. İşlem sırasında herhangi bir aksama yada eksiklik durumunda DS18B20 ile iletişim kurulamaz.

DS18B20 Dijital Sıcaklık Sensörü Şematik ve PCB Görünümü

* TEMP bağlantısı Arduino Nono ile yapılan One-Wire haberleşmesi için kullanılan bağlantıdır.
* 5V bağlantısı sensörün besleme bağlantısıdır.
* GND bağlantısı ise sensörün toprak bağlantısıdır.

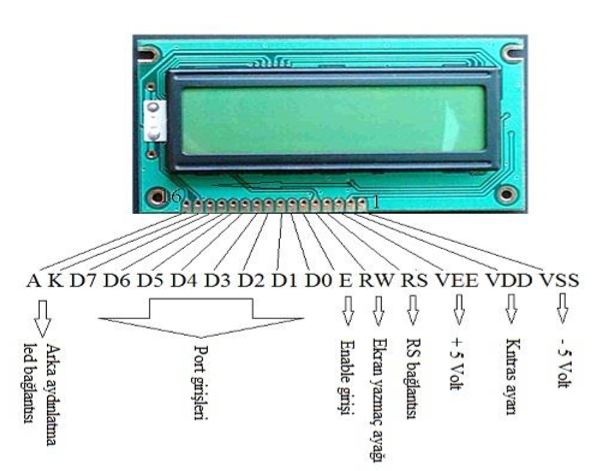
1. **16x2 LCD Ekran**

****

LCD Ekran Sensörler ile ölçtüğümüz sıcaklık ve mesafe verilerini görüntülediğimiz ve uyarıları alabildiğimiz ekrandır.

**16x2 LCD Ekran Teknik Özellikleri**

* Çalışma Voltajı: 5V
* Çalışma Akımı: Arka fon ışığı olmadan 4mA
* Boyutları: 80x36x9.4
* Yapı: COB(Chip on Board)
* Ekran Formatı: 16x2 Karakter
* Ekran Tipi: STN, Transflectif, Pozitif, Y-G
* Kontrolcü: Hitachi HD44780
* Arayüz: 8-bit parallel
* Backlight: Yellow-Green/Arka Işık
* Görüntüleme Yönü: Saat 6 Yönü
* Sürüş Şeması: 1/16 Duty Cycle, 1/5 Bias
* Çalışma Sıcaklığı: -10℃ to +60℃
* Saklama Sıcaklığı: -20℃ to +70℃



|  |  |  |
| --- | --- | --- |
| **PİN NO** | **İŞLEVİ** | **İSMİ** |
| 1 | Topraklama (0v) | Ground |
| 2 | Besleme Gerilimi (+5v) | Vcc |
| 3 | Potansiyometre Giriş ( LCD Konstrast Ayarını Yapmak İçin) | Vee |
| 4 | Komut Register‘ı ve Veri Register‘ı arasında geçiş yapmak için \* | Register Select |
| 5 | LCD’ye Okuma/Yazma işlemleri için kullanılır \*\* | Read/Write |
| 6 | Registera yazma işlemini sağlayan etkinleştirme pini | Enable |
| 7 | 8 Bitlik Veri Pinleri \*\*\* | DB0 |
| 8 | 8 Bitlik Veri Pinleri \*\*\* | DB1 |
| 9 | 8 Bitlik Veri Pinleri \*\*\* | DB2 |
| 10 | 8 Bitlik Veri Pinleri \*\*\* | DB3 |
| 11 | 8 Bitlik Veri Pinleri \*\*\* | DB4 |
| 12 | 8 Bitlik Veri Pinleri \*\*\* | DB5 |
| 13 | 8 Bitlik Veri Pinleri \*\*\* | DB6 |
| 14 | 8 Bitlik Veri Pinleri \*\*\* | DB7 |
| 15 | Arka Plan Işığının Gerilim Ayarı | Led+ |
| 16 | Arka Plan Işığıın Topraklaması | Led- |

LCD Pin Açıklamaları

\* RS Pin LOW (0) olduğunda Komut Register‘ı seçer, RS Pini HIGH (1) olduğunda Veri Register‘ı seçer

\*\* R/W Pinine LOW (0) değerini dönderdiğimizde Yazma işlemi, HIGH (1) gönderdiğimizde Okuma işlemi gerçekleştirir

\*\*\* Bu Pinlerle LCD’ye Veri veya Komut gönderilir.

**LCD Ekranın Arkasındaki Siyah Dairelerin Görevi**



HD44780 Denetleyicisi 16×2 LCD Ekran Arka Yüzü

LCD ekranın arkasındaki bu siyah daireler Microdenetleyicimiz ile LCD arasında köprü görevi görmektedir.

LCD ekranın 16×2 nokta vuruşlu olup toplamda 16×2=32 karakter gösterilmesine yardımcı olur. Her bir karakter 5 sütun 8 satırdan oluşmak üzere 8×5=40 pixel ile ifade edilir.



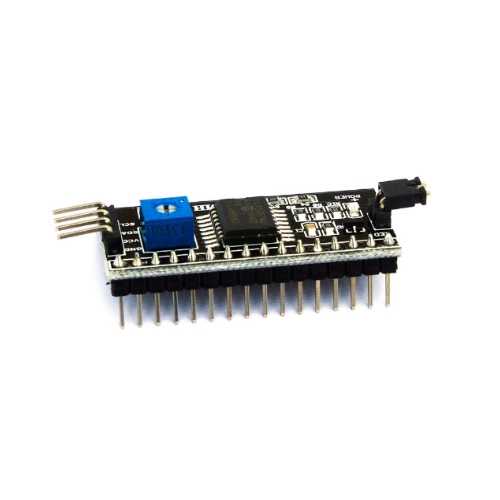
Tüm pixelleri aktif edilmiş 8×5 lik bir karakter görseli

LCD ekranımızda her bir karakter 8×5=40 pixelden oluşmaktadır. LCD ekranda, 16×2=32 karakter gösterebildiğine göre LCD ekran toplamda 40×32=1280 pixelden oluşmaktadır.

Pixellerin konumları hakkında LCD’ye bilgi vermek gerekmektedir. Bu tür işleride kullanılan microdenetleyici aracılığıyla yapılırsa mikro denetleyici üzerine çok yük bineceğinden dolayı bu tür görevleri, anlamlı komutları ve verileri microdenetleyiciden alıp LCD ekranda gösterme görevini yerine getirmek üzere LCD ekran arka yüzüne monte edilmiş HD44780 ara yüzü kullanılır.

|  |  |
| --- | --- |
| **Hex Code** | **LCD Ekran Karşılığı** |
| 0F | LCD açık, İmleç açık |
| 01 | Ekranı Temizle |
| 02 | Satır Başına Dön |
| 04 | İmleci Sola Kaydır |
| 06 | İmleci Sağa Kaydır |
| 05 | Ekranı Sağa Kaydır |
| 07 | Ekranı Sola Kaydır |
| 0E | Ekran Açık, İmleç Yanıp Sönüyor |
| 80 | İmleci İlk Satırın Başında Konumlandır |
| C0 | İmleci İkinci Satırın Başında Konumlandır |
| 3C | İkinci Satırı Etkinleştir |
| 08 | Ekran Kapalı, İmleç Kapalı |
| OC | Ekran Açık, İmleç Kapalı |

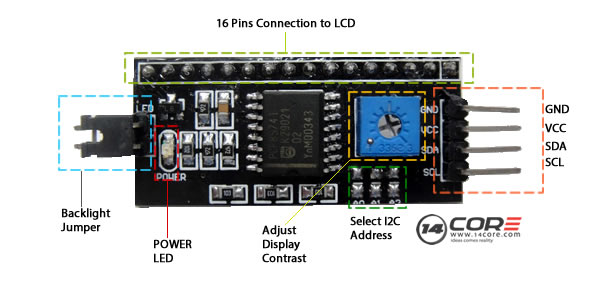
1. **1602 I2C Modülü**



1602 I2C Modülünü LCD Ekran ile Arduino Nano arasındaki haberleşmeyi I2C haberleşme protokolü sayelinde daha az pin kullanarak gerçekleştirebilmek için kullanılmıştır.

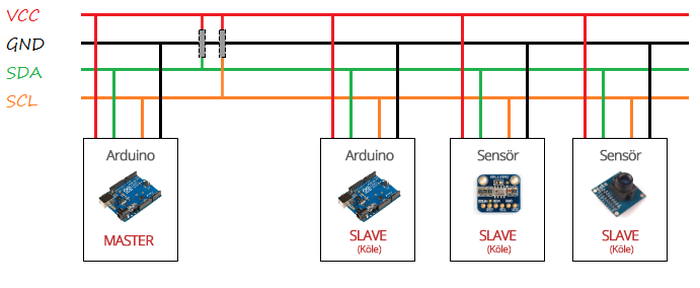
**1602 I2C Modülünün Teknik Özellikleri**

* Boyutlar : 41.5mm (uzunluk) \* 19mm (genişlik) \* 15.3mm (yükseklik)
* Ağırlık: 5g
* PCB rengi: Siyah
* Kontrolcü entegre: PCF8574
* Besleme voltajı :2.5-6V
* I2C protokolünü destekler
* Arka ışık kontrol edebilir. Kart üzerindeki jumper ile güç bağlantısı ayarlanabilir.
* Toplam akım kapasitesi :80mA
* Standby akımı:2.5uA
* Çalışma sıcaklığı -40℃ den +85℃
* Kontrast ayarı kart üzerinde vidalı potansiyometre ile minik bir tornavida yardımıyla ayarlanabilir.
* 8 taneye kadar modül birbirine seri olarak bağlanabilir. A0 A1 A2 pinleri kısa devre yapılarak adres değiştirilebilir. Ana adres ilk başta 0x27 veya 0x3E dir.



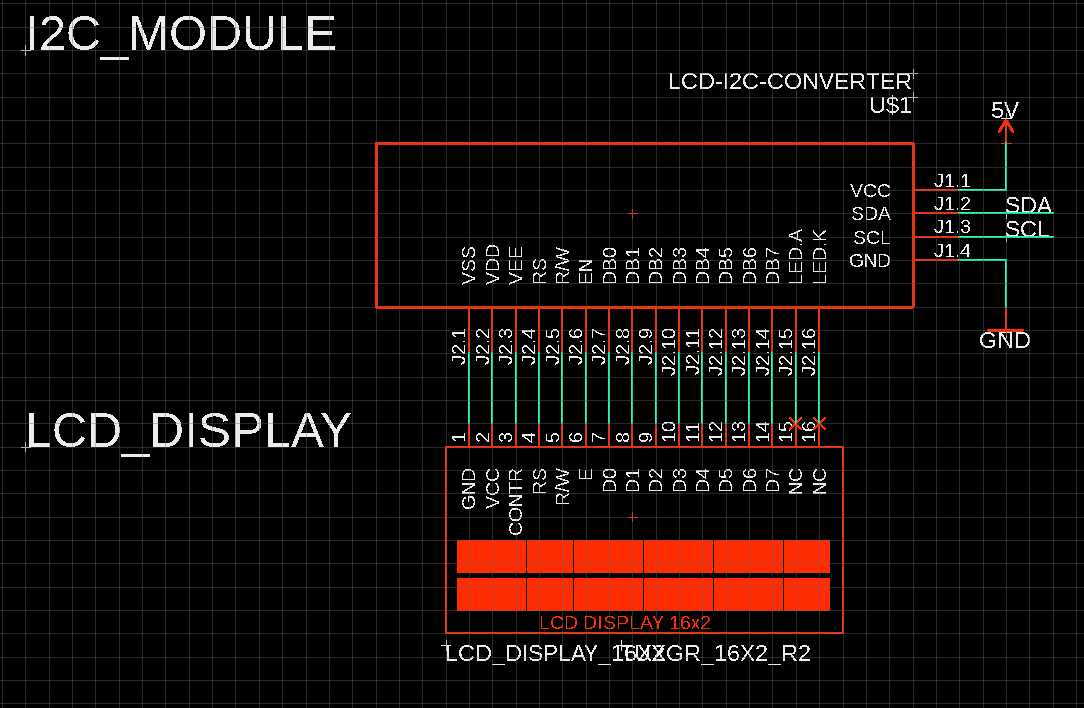
Arduino, diğer Arduino veya sensörlerle haberleşmek için bazı haberleşme protokolleri kullanır. Bu protokollerden birisi de I2C'dir. I2C (Inter-Integrated Circuit), seri haberleşme türlerinden senkron haberleşmeye bir örnektir. Haberleşme için toprak hattı dışında SDA ve SCL olmak üzere iki hatta ihtiyaç duyulmaktadır. Hat sayısının fazla olması nedeniyle, uzun mesafeli haberleşmelerde tercih edilmez. Genellikle kısa mesafeli ve düşük veri aktarım hızının yeterli olduğu yerlerde kullanılır.

I2C haberleşmesinde, haberleşmeyi kontrol eden master cihazı bulunur. Her haberleşmede bir tane master bulunmalıdır. Haberleşmenin sağlanabilmesi için haberleşme hattına en az bir adet slave (köle) cihaz bağlanmalıdır. Hatta bağlanan birden fazla slave cihazlardan hangisinin veri aktaracağına, master cihaz karar verir. Böylece hat sayısında bir değişiklik olmadan birden fazla cihazla haberleşme sağlanır.

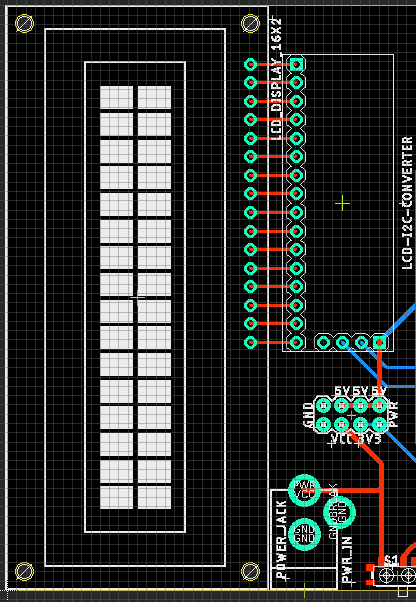


Master ve slave cihazların aynı besleme hattına bağlanmasına gerek yoktur. Fakat iletişimin sağlanması için toprak hatlarının aynı olması gerekir. Bunun yanında veri aktarımı için SDA (Serial Data Line) ve SCL (Serial Clock) olmak üzere iki adet haberleşme hattı bulunur. Bu hatlardan SDA, cihazlar arasındaki veri aktarımının sağlandığı hattır. Bu hatta çift yönlü veri aktarımı olur. Hatta aktarılan verilerin senkronizasyonu, SCL hattı tarafından gerçekleştirilir. SCL hattında master cihaz tarafından üretilen saat sinyali bulunur. SDA hattındaki haberleşme, bu sinyale göre düzenlenir.

Haberleşmenin tüm hat boyunca hatasız bir şekilde sağlanabilmesi için SDA ve SCL hatları, pull-up dirençlerle VCC hattına bağlanmalıdır. SDA ve SCL pinleri, kullanılan Arduino türüne göre değişiklik göstermektedir. Benim kullandığım Arduino Nanoda ise bu pinler A4(SDA) ve A5(SCL) pinleridir.



LCD Ekran ve I2C Modülünün Şematik Gösterimi



LCD Ekran ve I2C Modülünün PCB Gösterimi

1. **Buzzer**



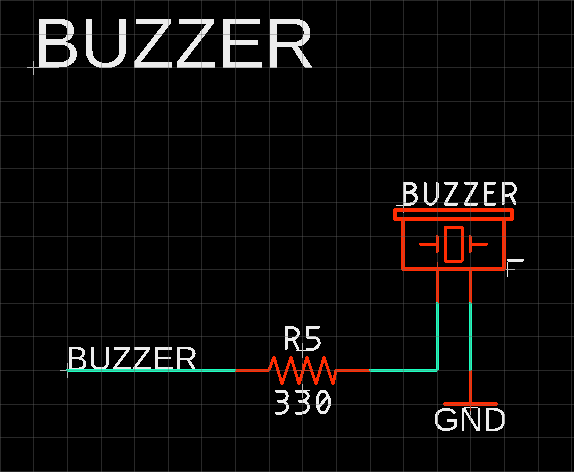
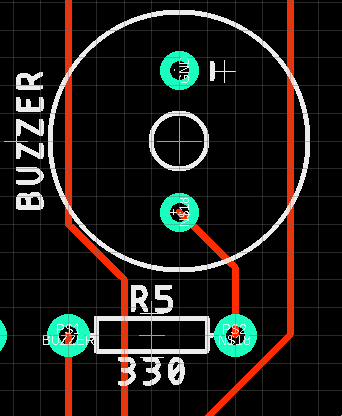
Buzzer projede park sensörünün sesli uyarı aracı olarak kullanılmıştır. Mesafe sensöründen gelen mesafe belli bir değerden aşağı olduğunda çalmaya başlamaktadır ve mesafe azaldıkça hızını artırmaktadır.

Buzzer çeşitli inputlar(sinyaller) alır ve buna cevaben ses yayar.Ses üretmek için çeşitli araçlar kullanılabilir; elektromekanik , elektronik , mekanik.

* Manyetik Buzzer:Manyetik buzzer sesi bobin yardımı ile yayar. Elektrik akısı bobini aktif hale getirir ve ses oluşur.
* Elektromekanik Buzzer:Elektromekanik buzzerın kendi içindeki tasarımı bir çok parçayı birleştirir.Bu çeşidi diğerlerinden ayıran yönü , akım alete doğru giderken temas noktası aktif olur. Eski tip ev zilleri elektromekaniktir fakat günümüzde pek kullanımı kalmamıştır.
* Mekanik Buzzer: Mekanik buzzerlarda elektrik alımı yerine sıkıca sarılmış yaylar kullanılır ki bu sayede mekanik enerji sese çevirilmiş olur.
* Piezo Buzzer: Düşük maliyeti ve yüksek performansı sebebi ile çok popülerdir. Elektrik akımına göre farklı tepkiler verebilen bir çeşit buzzerdir.Bu sayede aynı buzzerdan farklı zamanlarda farklı tonlarda sesler alınabilir.

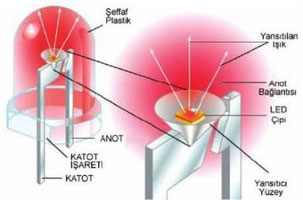
**Buzzer Teknik Özellikleri**

* Teknoloji : Manyetik
* Seri : MCW
* Sürücü Devresi : With Circuit
* Giriş tipi : DC
* Empedans : YOK
* Anma Gerilimi : 5VDC
* Voltaj aralığı : 3VDC ~ 7VDC
* Frekans : 2.3KHz
* Ses Şiddeti : 85dB @ 10cm
* Kaynak Akımı : 30mA
* Montaj tipi : THT
* Eğim : 7.60mm
* Boyutlar (mm) : DxH 12.00x7.50mm

Buzzerın Şematik ve PCB Görünümü

1. **LED(Light Emitting Diode)**



LED(Light Emitting Diode – Işık Yayan Diyot), doğru gerilim uygulandığı zaman ışıyan, diğer bir deyişler elektrik enerjisini ışık enerjisi haline dönüştüren özel katkı maddeli PN diyotlarıdır.

Günümüzde kullanılan ledler görünür ışık, morötesi, kızılötesi gibi çeşitli dalga boylarında, yüksek parlaklıkta ışık verirler. LED düşük enerji tüketimi, daha uzun ömrü, sağlamlığı, küçük boyutu ve hızlı açılıp kapanabilmesi ile diğer ışık kaynaklarına göre daha avantajlıdır.

Farklı boyutlarda (1-1,9-2-2,1-3-5-10 mm vb) LED imalatları yapılır. 2-20mA gibi çok az bir akımla çalıştıkları ve sarsıntılara dayanıklı oldukları için bütün elektronik devrelerinde kullanılır.

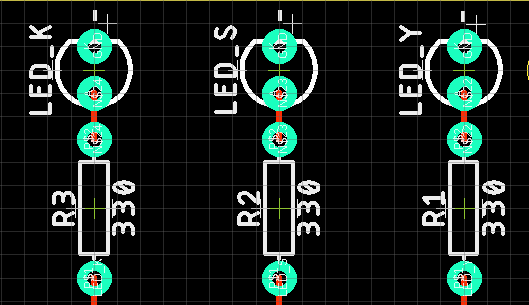
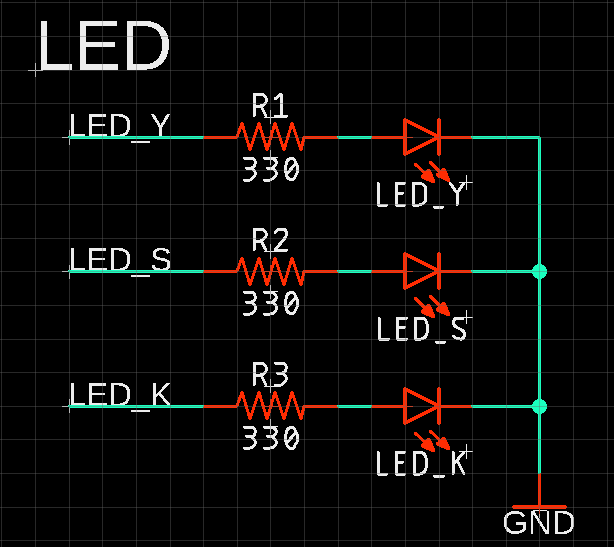
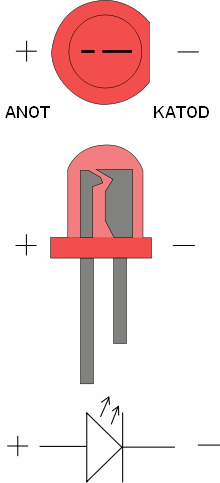
LED’ler doğru polarma altında çalışır. >Farklı renk ve gerilimlerde üretilirler Renklerine göre çalışma gerilimleri Kırmızı için 1.8V-2.2V , Sarı için 1.8V-2.2V, Yeşil için 3V-3.3V’tur

LED’lerin renkleri katkı maddelerinin oranlarına bağlı olarak elde edilir. Yeşil ve Sarı ışık için Galyum - Fosfat, Kırmızı ışık için Galyum - Fosfat içine Çinko - Oksit kullanılır



**LED’lerin Teknik Özellikleri**

* Yarı iletken malzemelerdir.
* Ana maddesi silikondur.
* Üzerinden akım geçtiğinde foton açığa çıkararak ışık verirler.
* Çalışma zamanı nanosaniyeler mertebesindedir.
* Farklı açılarda ışık verecek şekilde üretilirler.
* Çalışma ömrü 100.000-200.000 saat arasındadır.
* LED’lerin gerilim akım grarikleri üsteldir. Uygun çalışma noktasındayken ledin üzerindeki küçük bir gerilim değişimi büyük bir akım değişimine neden olur. Yüksek akım sebebiyle bozulmaması için ledlere seri bir akım sınırlama direnci bağlanır. Böylece hassas olmayan gerilim aralıklarında ledin bozulması engellenir.



LED’lerin Şematik ve PCB Görünümü

Projede LED’ler ölçülen mesafe değerine göre yanmaktadır.

**Yazılım**

**#include <OneWire.h>** // Sıcaklık sensörü için OneWire kütüphanesini ekliyoruz.

**#include<LiquidCrystal\_I2C.h>** // LCD Ekran kütüphanesini ekliyoruz

**LiquidCrystal\_I2C lcd(0x27, 16, 2);** // LCD Ekran için bir nesne tanımlıyoruz ve LCD Ekranın adresini, satır ve sütun bilgilerini giriyoruz

**int DS18B20\_Pin = 2;** // Sıcaklık sensörünü bağladığımız dijital pini 2 olarak belirliyoruz.

**OneWire ds(DS18B20\_Pin);** // Sıcaklık sensörü için bir nesne tanımlıyoruz ve ve pin bilgisini ekliyoruz

**const int trig = 12;** // Mesafe sensörünün Trigger pinini tanımlıyoruz

**const int echo = 11;** // Mesafe sensörünün Echopinini tanımlıyoruz

**int sure = 0,mesafe = 0;**

**float hiz = 0,degisken = 0;**

**const int uyari\_ses = 7;** // Buzzer için kullanacağımız pini tanımlıyoruz

**const int uyari\_led\_kirmizi = 8;** // Kırmızı LED için kullanacağımız pini tanımlıyoruz

**const int uyari\_led\_sari = 9;** // Sarı LED için kullanacağımız pini tanımlıyoruz

**const int uyari\_led\_yesil = 10;** // Kırmızı LED için kullanacağımız pini tanımlıyoruz

**void setup() {**

**pinMode(trig,OUTPUT);** // Trigger pinini OUTPUT olarak ayarlıyoruz

**pinMode(echo,INPUT);** // Echo pinini INPUT olarak ayarlıyoruz

**pinMode(uyari\_led\_kirmizi, OUTPUT);** // Kırmızı LED pinini OUTPUT olarak ayarlıyoruz

**pinMode(uyari\_led\_sari, OUTPUT);** // Sarı LED pinini OUTPUT olarak ayarlıyoruz

**pinMode(uyari\_led\_yesil, OUTPUT);** // Yeşil LED pinini OUTPUT olarak ayarlıyoruz

**pinMode(uyari\_ses, OUTPUT);** // Buzzer pinini OUTPUT olarak ayarlıyoruz

**lcd.begin(16,2);** // LCD ekranı başlatıyoruz

**Serial.begin(9600);** // Seri iletişimi başlatıyoruz.

}

**void loop() {**

**float temperature = getTemp();** // temperature değişkenini sıcaklık değerini alma fonksiyonuna bağlıyoruz ve sıcaklığı santigrat cinsinden elde ediyoruz

**digitalWrite(trig, HIGH);** // Trig pinindenses dalgası üretiyoruz

**delayMicroseconds(1000);** // 1 saniye gecikme veriyoruz

**digitalWrite(trig, LOW);** // Trig pininden ürettiğimiz ses dalgasını kesiyoruz

**hiz = (331.4 \* sqrt(1+(temperature/273)))/10000;** // Ses dalgasının hızını sızaklığa bağlı olarak hesaplıyoruz

**degisken = 1/hiz;** // degisken kullanarak hizi formüle uygun hale getiriyoruz

**sure = pulseIn(echo, HIGH);** // echo pinini trig pininden göderilen ses dalgalarını algılayabilmesi için pulseIn fonksiyonunu kullanıyoruz

**mesafe = (sure/2)/degisken;** // Mesafe hesaplıyoruz

**if(temperature>=10 && temperature<24)** // Sıcaklık 10 ile 24 derece arasında ise uygulanacakları belirliyoruz

**{if(mesafe>=5 && mesafe<15)** // Mesafe 5 ile 15 cm arasındaysa kırmızı led yanacak ve buzzer yüksek derecede ötecek şekilde ayarlıyoruz

**{digitalWrite(uyari\_led\_kirmizi, HIGH);**

**digitalWrite(uyari\_led\_sari, LOW);**

**digitalWrite(uyari\_led\_yesil, LOW);**

**digitalWrite(uyari\_ses, HIGH);**

**delay(50);**

**digitalWrite(uyari\_ses, LOW);**

**delay(50);}**

**if(mesafe>=15 && mesafe<25)** // Mesafe 15 ile 25 cm arasındaysa sarı led yanacak ve buzzer orta derecede ötecek şekilde ayarlıyoruz

**{digitalWrite(uyari\_led\_kirmizi, LOW);**

**digitalWrite(uyari\_led\_sari, HIGH);**

**digitalWrite(uyari\_led\_yesil, LOW);**

**digitalWrite(uyari\_ses, HIGH);**

**delay(100);**

**digitalWrite(uyari\_ses, LOW);**

**delay(100);}**

**else** // Mesafe 25 cm den yukarıdaysa yeşil led yanacak şekilde ayarlıyoruz

**{digitalWrite(uyari\_led\_kirmizi, LOW);**

**digitalWrite(uyari\_led\_sari, LOW);**

**digitalWrite(uyari\_led\_yesil, HIGH);}}**

**else if(temperature>=24 && temperature<30)** // Sıcaklık 24 ile 30 derece arasında ise uygulanacakları belirliyoruz

**{if(mesafe>=10 && mesafe<20)** // Mesafe 10 ile 20 cm arasındaysa kırmızı led yanacak ve buzzer yüksek derecede ötecek şekilde ayarlıyoruz

**{digitalWrite(uyari\_led\_kirmizi, HIGH);**

**digitalWrite(uyari\_led\_sari, LOW);**

**digitalWrite(uyari\_led\_yesil, LOW);**

**digitalWrite(uyari\_ses, HIGH);**

**delay(50);**

**digitalWrite(uyari\_ses, LOW);**

**delay(50);}**

**if(mesafe>=20 && mesafe<30)** // Mesafe 20 ile 30 cm arasındaysa sarı led yanacak ve buzzer orta derecede ötecek şekilde ayarlıyoruz

**{digitalWrite(uyari\_led\_kirmizi, LOW);**

**digitalWrite(uyari\_led\_sari, HIGH);**

**digitalWrite(uyari\_led\_yesil, LOW);**

**digitalWrite(uyari\_ses, HIGH);**

**delay(100);**

**digitalWrite(uyari\_ses, LOW);**

**delay(100);}**

**else** // Mesafe 25 cm den yukarıdaysa yeşil led yanacak şekilde ayarlıyoruz

**{digitalWrite(uyari\_led\_kirmizi, LOW);**

**digitalWrite(uyari\_led\_sari, LOW);**

**digitalWrite(uyari\_led\_yesil, HIGH);}}**

**else if(temperature>=30 && temperature<45)** // Sıcaklık 30 ile 45 derece arasında ise uygulanacakları belirliyoruz

**{if(mesafe>=15 && mesafe<25)** // Mesafe 15 ile 25 cm arasındaysa kırmızı led yanacak ve buzzer yüksek derecede ötecek şekilde ayarlıyoruz

{**digitalWrite(uyari\_led\_kirmizi, HIGH);**

**digitalWrite(uyari\_led\_sari, LOW);**

**digitalWrite(uyari\_led\_yesil, LOW);**

**digitalWrite(uyari\_ses, HIGH);**

**delay(50);**

**digitalWrite(uyari\_ses, LOW);**

**delay(50);}**

**if(mesafe>=25 && mesafe<35)** // Mesafe 25 ile 35 cm arasındaysa sarı led yanacak ve buzzer orta derecede ötecek şekilde ayarlıyoruz

**{digitalWrite(uyari\_led\_kirmizi, LOW);**

**digitalWrite(uyari\_led\_sari, HIGH);**

**digitalWrite(uyari\_led\_yesil, LOW);**

**digitalWrite(uyari\_ses, HIGH);**

**delay(100);**

**digitalWrite(uyari\_ses, LOW);**

**delay(100);}**

**else** // Mesafe 25 cm den yukarıdaysa yeşil led yanacak şekilde ayarlıyoruz

**{digitalWrite(uyari\_led\_kirmizi, LOW);**

**digitalWrite(uyari\_led\_sari, LOW);**

**digitalWrite(uyari\_led\_yesil, HIGH);}**

**Serial.print("Sicaklik: ");** // Serial port ekranında sıcaklık bilgisini yazdırıyoruz

**Serial.println(degisken);**

**Serial.print("Mesafe: ");** // Serial port ekranında sıcaklık bilgisini yazdırıyoruz

**Serial.println(mesafe);**

**lcd.clear();**

**lcd.home();**

**lcd.print("Sicaklik=");** // LCD Ekranda sıcaklık bilgisini yazdırıyoruz

**lcd.setCursor(10,0);**

**lcd.print(temperature);**

**lcd.setCursor(0,1);**

**lcd.print("Mesafe=");** // LCD Ekranda mesafe bilgisini yazdırıyoruz

**lcd.setCursor(10,1);**

**lcd.print(mesafe);**

**delay(500);**

}

// Aşağıdaki fonksiyon DS18B20 sıcaklık sensörümüzden gelen verileri

// Celcius cinsinden sıcaklık değerlerine çevirmek için kullanılıyor.

**float getTemp(){**

**byte data[12];**

**byte addr[8];**

**if ( !ds.search(addr)) {** // Sensörleri arıyoruz

**ds.reset\_search();**

**return -1000; }**

**if ( OneWire::crc8( addr, 7) != addr[7]) {** // Byte iletiminde hata olup olmadığı kontrol ediyoruz

**Serial.println("CRC is not valid!");**

**return -1000;}**

**if ( addr[0] != 0x10 && addr[0] != 0x28) {** // Sensörün tanınıp tanınmadığını control ediyoruz

**Serial.print("Device is not recognized");**

**return -1000 }**

**ds.reset();**

**ds.select(addr);**

**ds.write(0x44,1);** // İletişimi başlatıyoruz

**byte present = ds.reset();**

**ds.select(addr);**

**ds.write(0xBE);** // Sensöre okuma yapacağımızı iletiyoruz

**for (int i = 0; i < 9; i++) {** // 9Byte okuyoruz

**data[i] = ds.read();}**

**ds.reset\_search();**

**byte MSB = data[1];**

**byte LSB = data[0];**

**float tempRead = ((MSB << 8) | LSB);**

**float TemperatureSum = tempRead / 16;**

**return TemperatureSum;}** // Sıcaklık değerini geri döndürüyoruz