Erasmus+ KA-202

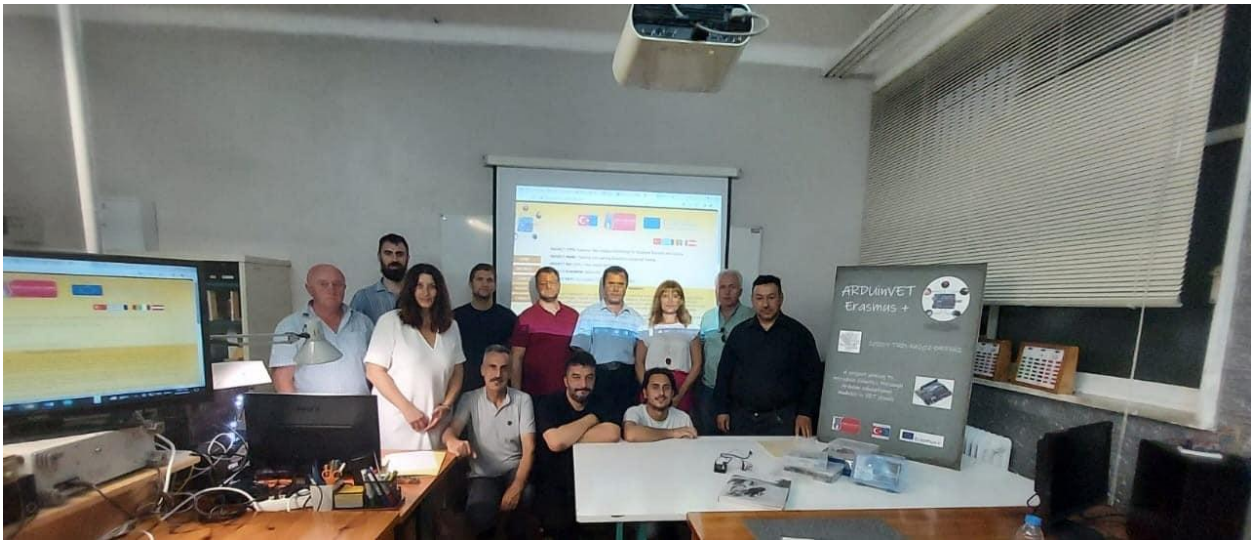
Mesleki Eğitim ve Öğretime Yönelik Stratejik Ortaklık Projesi

Proje Adı: "Mesleki Eğitimde Arduinoları Öğrenme ve Öğretme"

Proje Kısaltması: " ARDUinVET "

Proje Numarası: "2020-1-TR01-KA202-093762"

ARDUinVET KILAVUZU



"Bu proje Erasmus+ Programı kapsamında Avrupa Komisyonu tarafından desteklenmektedir. Ancak burada yer alan görüşlerden Avrupa Komisyonu ve Türkiye Ulusal Ajansı sorumlu tutulamaz."

"This project is Funded by the Erasmus+ Program of the European Union. However, European Commission and Turkish National Agency cannot be held responsi-ble for any use which may be made of the information contained there in"

KOORDİNATÖR:

Gölbaşı Mesleki ve Teknik Anadolu Lisesi (Ankara / TURKEY)

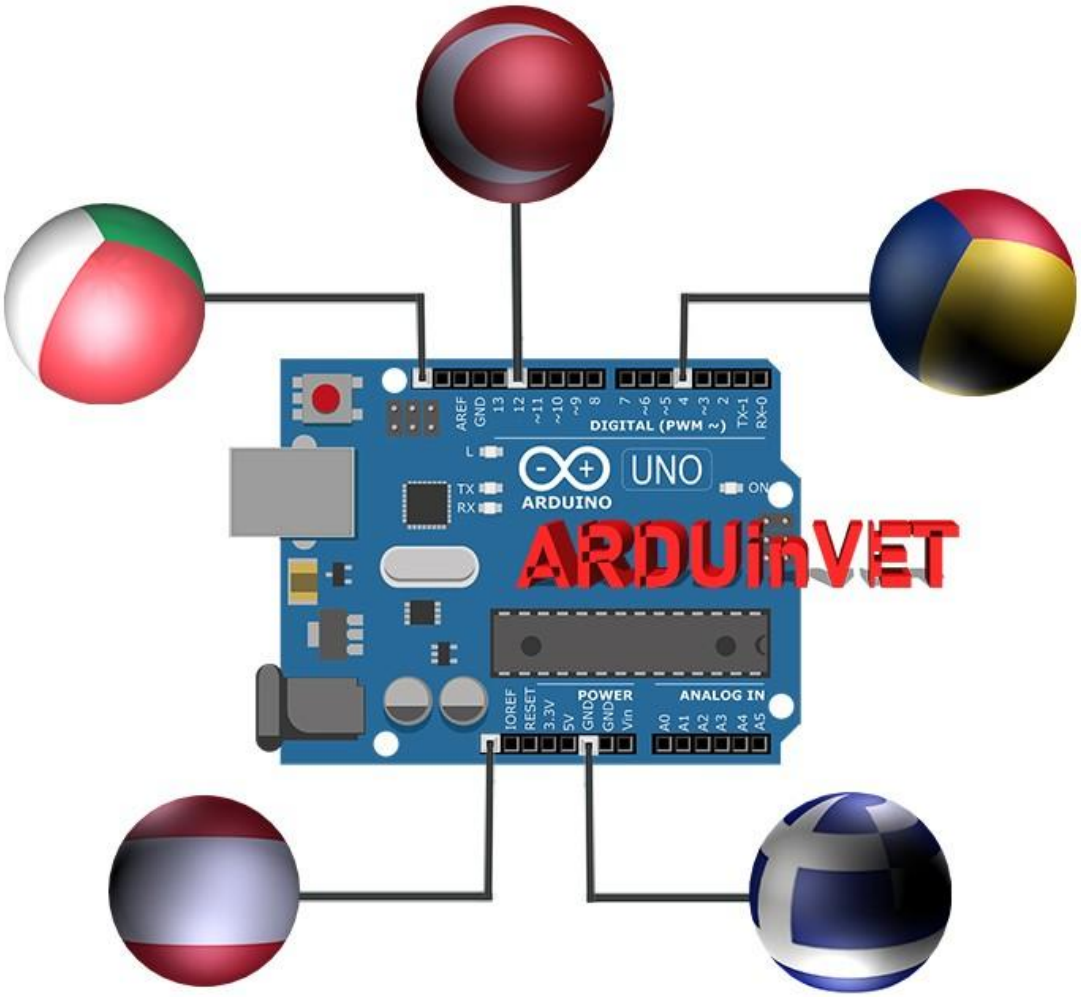
ORTAKLAR:

2 EK Peiraia (Piraeus / Greece)

Liceul Tehnologic Grigore Moisil Braila (Braila / Romania)

Istituto di Istruzione Superiore Einstein De Lorenzo Potenza (Potenza / Italy)

HTBLA Wolfsberg / (College for Engineering Wolfsberg) (Wolfsberg / Austria)



Proje Özeti
Mesleki Eğitimde Arduinoları Öğrenme ve Öğretme

Teknoloji her geçen gün çok hızlı gelişmektedir. Teknolojik gelişmelerin insanı mesleğinde sürekli kendini geliştirmeye zorladığı bir gerçektir. Elektronik, bilgi ve iletişim teknolojisi diğer tüm alanlardan daha fazla hızla gelişmektedir. Bu gelişmeler, daha karmaşık kontrol sistemlerinin üretilmesine yol açmıştır. Arduinolar artık bu karmaşık sistemleri kontrol etmek için kullanılmaktadır.

Teknolojiyi öğreten eğitim kurumları olduğumuz için dünyadaki gelişmeleri takip etmek zorundayız. "Mesleki Eğitimde Arduinoları Öğrenme ve Öğretme" Projesi, Arduino uygulamalarını mesleki eğitime uyarlamayı ve daha verimli bir eğitim seti, mesleki ve teknik eğitim öğrencilerinin atölyeleri ve laboratuarları için bir rehber kitap geliştirmeyi amaçlamaktadır.

Bu projenin temel amacı; bir iyi uygulama rehber kitabı ve Arduino eğitim seti geliştirmek, ev sahibi ülkeye yapılan ziyaretlerde diğer katılımcılara Arduino eğitim modellerini tanıtmak, farklı eğitim sistemlerini ve eğitim yöntemlerini diğer katılımcı okullarla karşılaştırmak ve iyi uygulamaları paylaşmaktır.

Projelerin katılımcıları; elektrik, elektronik, BİT, otomasyon meslek öğretmenleridir. Katılımcılar; mesleki eğitim-öğretim okullarında Arduino öğretmektedir. Projede bir koordinatör ve toplam 4 ortak vardır. Ortaklar; Türkiye, İtalya, Romanya, Avusturya, Yunanistan'da yerleşik meslek okullarıdır. Projedeki toplam hareketlilik sayısı 40'tır. Proje boyunca toplam 5 ulus ötesi toplantı gerçekleştirilecek ve her ortak 2 katılımcı ile her ulus ötesi toplantıya katılacaktır. Her ülkede bir ulus ötesi toplantı düzenlenecektir.

Proje faaliyetleri ile Arduino öğretimi için en iyi uygulamaların paylaşılması ve ortakların bunları eğitim ortamlarına adapte etmeleri sağlanacaktır. Proje ortakları, Arduino deney kitleri ve bir iyi uygulama rehber kitabı üretecektir. Proje faaliyetlerimiz, tüm mesleki eğitim-öğretim öğretmenleri ve öğrencileri için başarılı bir öğrenme ve öğretme ortamı sağlamayı amaçlamaktadır. Başarılı öğrenmenin birinci koşulu olarak; öğretmenlerin öğrenmeyi kolaylaştırmadaki rollerini güçlendirmeleri gerekir. Öğretmenlerin; yenilik, takım çalışması, geri bildirim ve değerlendirme için yeni deney kitlerine ve modüllerine ihtiyaçları vardır. Öğretmenlere sürekli mesleki gelişim için bu fırsat verilmelidir.

Projenin ana çıktıları; mesleki eğitimde Arduino dersleri için bir Arduino Eğitim Seti ve bu set için bir Rehber Kitap, proje web sitesi ve proje DVD'sidir. Projenin yürütülmesinde kullanılacak metodoloji; "Yap-Geliştir-Paylaş" tır. Projemizde önce iyi uygulamalar yapılacak, sonra geliştirilecek ve son olarak paylaşılacaktır. Projemizde her şey açık ve şeffaf olacaktır. Her görev ve sorumluluk projede yazılı olarak kayıt altında olacaktır. Projenin ürünleri sadece yazılı belge ve ürünler değil, aynı zamanda pratik bir Arduino eğitim seti ve kitleridir.

Uzun vadede; projeyi öğretmenlere, öğrencilere ve mesleki eğitim okullarına, yerel eğitim kurumlarına, elektronik ve BİT iş piyasasına yaymayı hedeflemekteyiz. Proje sonuçlarının ve ürünlerinin, yaygınlaştırma faaliyetlerinin mesleki eğitim ve işgücü piyasası üzerinde kısa ve uzun vadeli etkileri olacağına inanıyoruz.

5 ortaklı projenin toplam bütçesi 59.000 Avro'dur.

İÇİNDEKİLER

Co-funded by the
Erasmus+ Programme
of the European Union

ARDUinVET PROJE MODÜLLERİ ve KİTLERİ

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+ KA-202

Mesleki Eğitim ve Öğretime Yönelik Stratejik Ortaklık Projesi

Proje Adı: "Mesleki Eğitimde Arduinoları Öğrenme ve Öğretme"

Proje Kısaltması: " ARDUinVET "

Proje Numarası: "2020-1-TR01-KA202-093762"

ARDUINOLARA GİRİŞ

(ARDUINO'NUN TEMELLERİ)

ARDUNIO'YA GİRİŞ

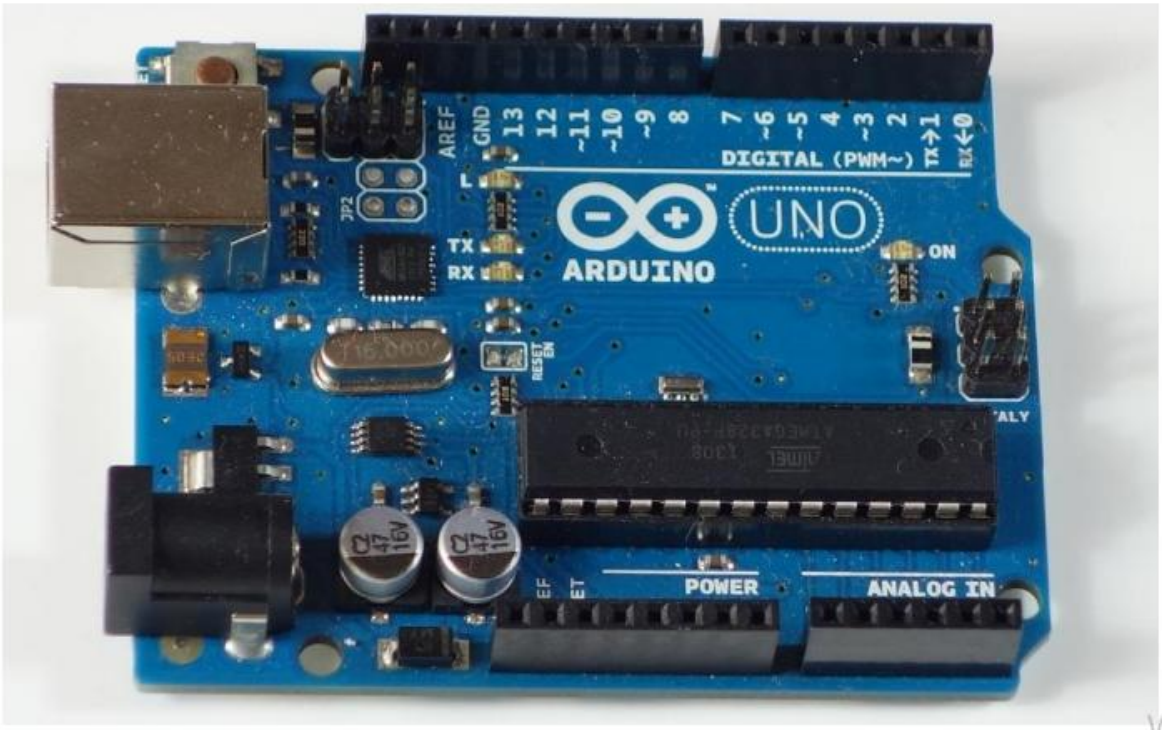
Arduino, donanım ve yazılıma dayalı, kullanımı kolay bir prototip platformudur (açık kaynak). Programlanabilen bir devre kartı (mikrodenetleyici olarak adlandırılır) ve bilgisayar kodunu fiziksel karta yazmak ve yüklemek için kullanılan Arduino IDE (Entegre Geliştirme Ortamı) adlı hazır bir yazılımdan oluşur.

Diğer bir deyişle Arduino, çevrenizdeki dünyadan bilgi okumak ve dış dünyaya komutlar göndermek için programlayabileceğiniz küçük bir bilgisayardır. İstediğinizi yapmak için Arduino'ya çeşitli cihaz ve bileşenler bağlayabilirsiniz. Onunla yapabileceklerinizin sınırı yoktur ve hayal gücünüzü kullanarak harika projeler gerçekleştirebilirsiniz.

Basit bir ifadeyle Arduino, çeşitli girdi ve çıktı biçimleriyle etkileşim kurmak için talimatlarınızla programlanabilen küçük bir bilgisayar sistemidir. Mevcut Arduino Uno modeli ortalama insan eline kıyasla oldukça küçüktür.

Ardunio Nedir?

Arduino kartı aşağıda gösterildiği şekildedir.

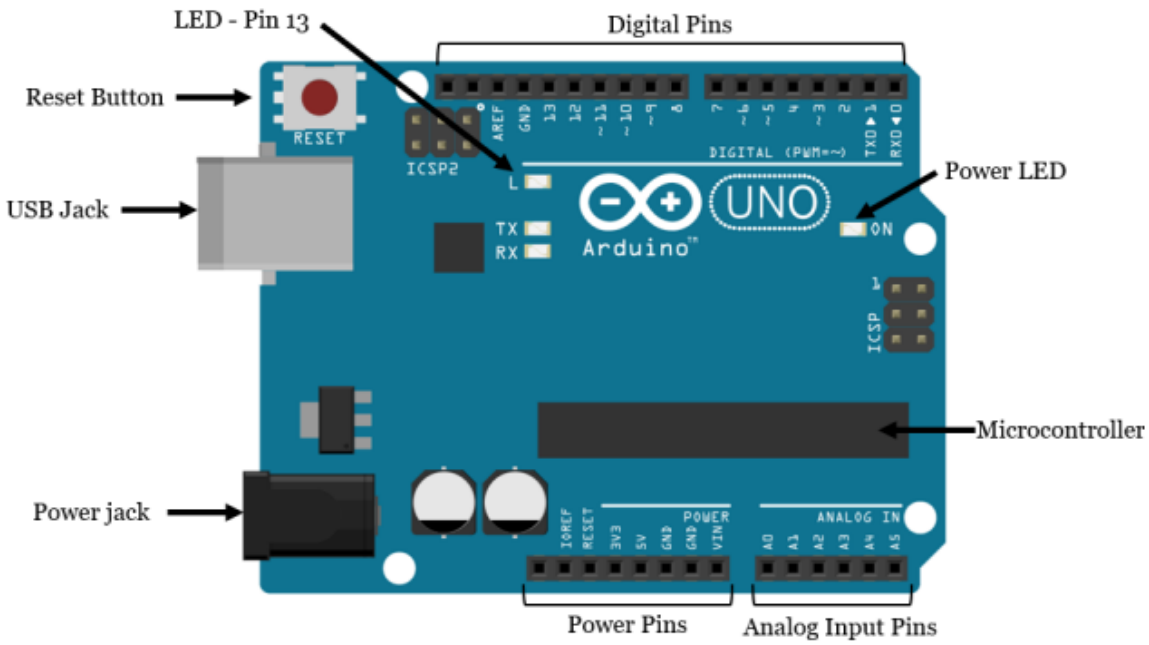


Temel olarak elektrik devrelerine bağlayabileceğiniz, beyni (mikrodenetleyici olarak da bilinir) olan küçük bir geliştirme kartıdır. Bu, girdilerin okunmasını – dışarıdan veri okunmasını – ve çıktıların kontrol edilmesini – dışarıya bir komut gönderilmesini kolaylaştırır. Bu kartın beyni

(Arduino Uno), Arduino'nuza ne yapacağını söyleyecek programlarınızı saklayabileceğiniz bir ATmega328p yongasıdır.

Arduino Uno Kartı Keşfetmek

Aşağıda etiketlenmiş bir Arduino kartı görebilirsiniz. Her parçanın ne yaptığını görelim.



• **Mikrodenetleyici:** ATmega328p, Arduino'nun beynidir. Arduino kartındaki her şey bu mikrodenetleyiciyi desteklemek içindir. Arduino'ya ne yapacağını söylemek için programlarınızı burada saklarsınız.

• **Dijital pinler:** Arduino'nun 0'dan 13'e kadar etiketlenmiş, giriş veya çıkış görevi görebilen 14 dijital pini vardır. Giriş olarak ayarlandığında bu pinler voltaj okuyabilir. Yalnızca iki durumu okuyabilirler: YÜKSEK veya DÜŞÜK. Çıkış olarak ayarlandığında bu pinler voltaj uygulayabilir. Yalnızca 5V (YÜKSEK) veya 0V (DÜŞÜK) uygulayabilirler.

• **PWM pinleri:** Bunlar ~ ile işaretlenmiş dijital pinlerdir (11, 10, 9, 6, 5 ve 3 numaralı pinler). PWM, "darbe genişliği modülasyonu" anlamına gelir ve dijital pinlerin değişen miktarlarda "sahte" voltaj çıkışına izin verir. PWM hakkında daha sonra daha fazlasını öğreneceksiniz.

• **TX ve RX pinleri:** T, "göndermek" ve R ise "almak" anlamına gelir. Arduino, seri bağlantı aracılığıyla diğer elektroniklerle iletişim kurmak için bu pinleri kullanır. Arduino yeni kod yüklerken bilgisayarınızla iletişim kurmak için de bu pinleri kullanır. Gerekmedikçe bu pinleri seri iletişim dışındaki diğer görevler için kullanmaktan kaçının.

• **Dijital pin 13'e bağlı LED:** Arduino senaryolarının kolay hata ayıklaması için kullanışlıdır.

• **TX ve RX LED'leri:** Bilgisayar ve Arduino arasında bilgi alışverişi olduğunda bu ledler yanıp söner.

• **Analog pinler:** Analog pinler A0'dan A5'e kadar etiketlenir ve genellikle analog sensörleri okumak için kullanılır. 0 ile 5V arasında farklı voltaj miktarlarını okuyabilirler. Ayrıca dijital pinler gibi dijital çıkış/giriş pinleri olarak da kullanılabilirler.

• **Güç pinleri:** Arduino bu pinler üzerinden 3.3V veya 5V sağlar. Çoğu bileşenin çalışması için 3.3V veya 5V gerektiğinden bu gerçekten yararlıdır. "GND" olarak etiketlenen pinler topraklama pinleridir.

• **Reset butonu:** Bu butona bastığınızda Arduino'nuzda çalışmakta olan program yeniden başlar. Ayrıca, sıfırlama düğmesi görevi gören güç pinlerinin yanında bir Reset pini vardır. O pine küçük bir voltaj uyguladığınızda Arduino'yu sıfırlayacaktır.

• **Güç AÇIK LED'i:** Arduino'ya güç verildiğinden yanacaktır.

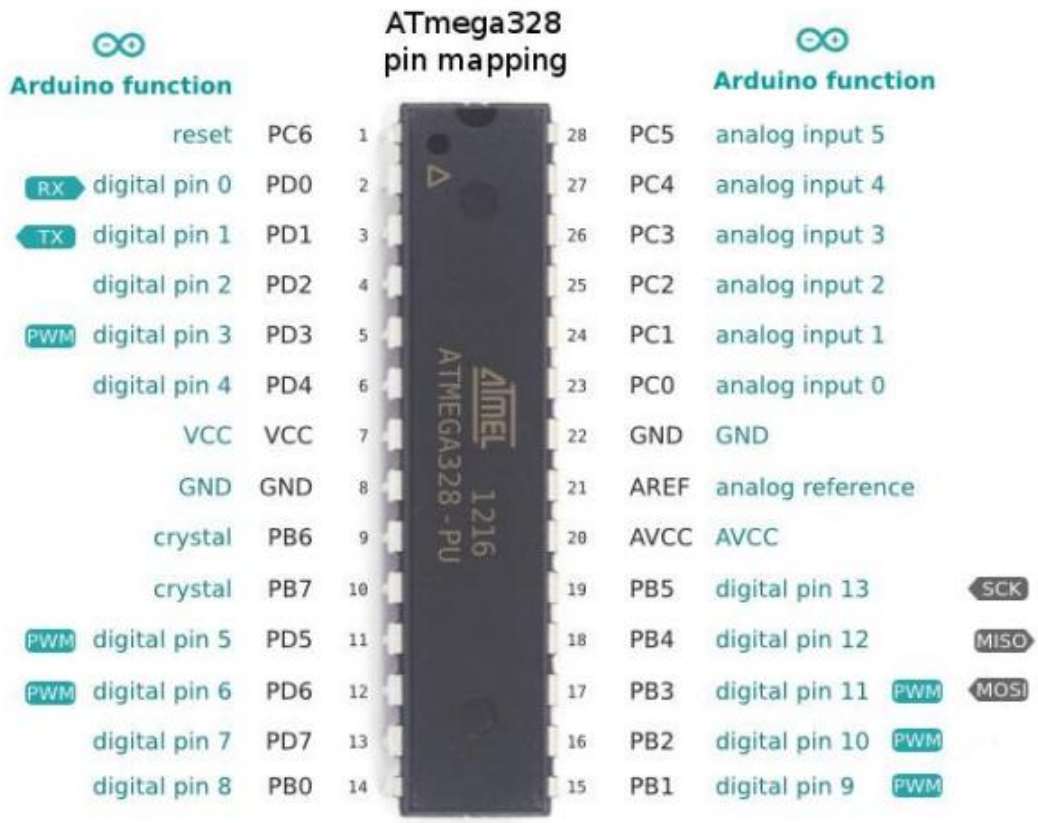
• **USB jakı:** Bilgisayarınızdan Arduino kartınıza program yüklemek için bir erkek USB A - erkek USB B kablosuna (aşağıdaki şekilde gösterilmiştir) ihtiyacınız vardır. Bu kablo aynı zamanda Arduino'nuza da güç sağlar.

• **Güç jakı:** Arduino'yu güç jakı üzerinden çalıştırabilirsiniz. Önerilen giriş voltajı 7V ila 12V'dir. Arduino'nuzu güçlendirmenin birkaç yolu vardır: örneğin; şarj edilebilir piller, tek kullanımlık piller, adaptörler, güneş paneli.

Arduino Özellikleri

Arduino Uno; Atmel Atmega 328P mikrodenetleyiciye sahiptir. Ayrıca USB bağlantı girişi, power jack girişi, reset butonu bulunmaktadır. Arduino bir mikrodenetleyicide olması gereken herşeye sahiptir.

Mikrodenetleyici	Atmega328P
Çalışma voltajı	5V
Giriş voltajı(önerilir)	7-12V
Giriş voltajı(sınır)	6-20V
Dijital giriş / çıkış pini	14
PWM giriş / çıkış pini	6
Analog giriş pini	6
Giriş / çıkış pini başına DC akımı	20mA
3.3V için DC akım	50mA
Flash bellek	32 KB
Sram	2KB
EEPROM	1 KB
Saat hızı	16 MHz
Uzunluk	68.6 mm
Genişlik	53.4 mm
Ağırlık	25 g
Şekil: Arduino Uno Özellikleri	



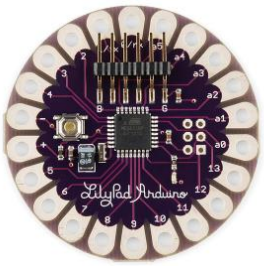
Şekil: Atmega 328P Pinleri

DİĞER ARDUNIO TÜRLERİ

ARDUINO MEGA

Üzerinde Atmega 2560 mikrodenetleyici bulunmaktadır. 54 dijital giriş-çıkış pinine, 16 analog girişe, 4 donanım seri portuna ve 16 mhz kristal osilatöre sahiptir. Hem USB hem de DC adaptör tarafından desteklenmektedir. Genel olarak Arduino UNO ile aynı özelliklere sahip olan kart, daha fazla pine sahip olduğu için daha büyük projelerde tercih edilmektedir.	

ARDUINO LILYPAD

Lilypad elbiselere ve kumaşlara dikilmek üzere tasarlanmıştır. Bu sayede giyilebilir olarak tasarlanabilecek ilginç projelerde kullanılabilir. Üzerinde Atmega 168V mikrodenetleyici bulunmaktadır.	

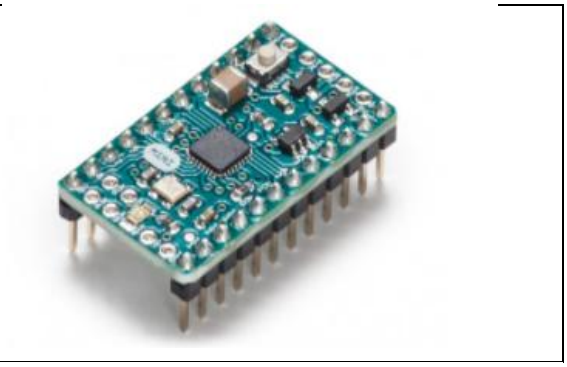
ARDUINO ETHERNET

İnternet bağlantılı projeler yapmak için bir Ethernet yongası ve bir Ethernet bağlantı noktasına sahiptir. Mikrodenetleyici olarak Atmega 328 modelinin bulunduğu kart üzerinde ayrıca bir SD-Kart yuvası bulunuyor.	

ARDUINO BLUETOOTH

Arduino BT üzerinde Bluetooth protokolü ile haberleşen uygulamalar yapmak için ideal bir Bluetooth modülü bulunmaktadır. Bu modül, Arduino'yu Bluetooth üzerinden programlamak için de kullanılabilir.	

ARDUINO MİNİ

Bir breadboard üzerinde çalıştırılmak veya başka bir tasarıma entegre edilmek üzere tasarlanmış bir Arduino modelidir. Üzerinde Atmega 168 veya Atmega 328 model mikrodenetleyici bulunmaktadır. Küçük boyutun özellikle önemli olduğu uygulamalar için idealdir.	

ARDUINO NANO

Atmega 328 veya Atmega 168 mikrodenetleyici, voltaj regülatörü, seriden USB'ye dönüştürücü chip, DC voltaj giriş portu ve mini USB portuna sahiptir.	

ARDUINO LEONARDO

Arduino Leonardo üzerinde bulunan Atmega 32u4 mikrodenetleyici içeren ve USB bağlantısı için ek bir chip gerektirmeyen Arduino kartlarından biridir. 20 adet dijital giriş/çıkış ve 12 adet analog giriş ile kart üzerindeki mikrodenetleyicinin yüzeye montaj kapağı bulunmaktadır. Leonardo, USB bağlantı yetenekleri sayesinde bilgisayara fare veya klavye olarak bağlanabilir.	

ARDUINO ESPLORA

Esplora, diğerlerinden farklı olarak çeşitli sensörler içeren bir Arduino kartıdır. Kart üzerinde bulunan sensörler sayesinde başka ilavelere ve aşırı elektronik bilgisine ihtiyaç duymadan birçok uygulamayı gerçekleştirmek mümkündür. Esplora, sürgülü potansiyometre, ışık ve ses sensörü, sıcaklık sensörü, ses üreteci, 2 eksenli mini analog joystick, 3 renkli LED ve ivme ölçer ile donatılmıştır. Esplora ayrıca Leonarda gibi Atmega 32U4 AVR mikrodenetleyici ile donatılmıştır. Mikro USB bağlantısı ile bilgisayara bağlanıldığında fare veya klavye görevi görebilen uygulamalar geliştirilebilir.	

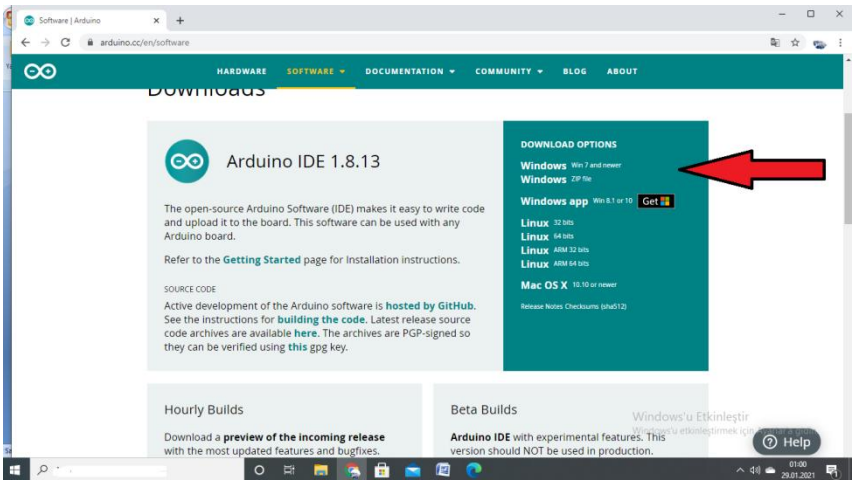
Arduino IDE'yi İndirmek

Arduino IDE (Entegre Geliştirme Ortamı), Arduino'ya ne yapacağını söyleyen programlarınızı geliştirdiğiniz yerdir.

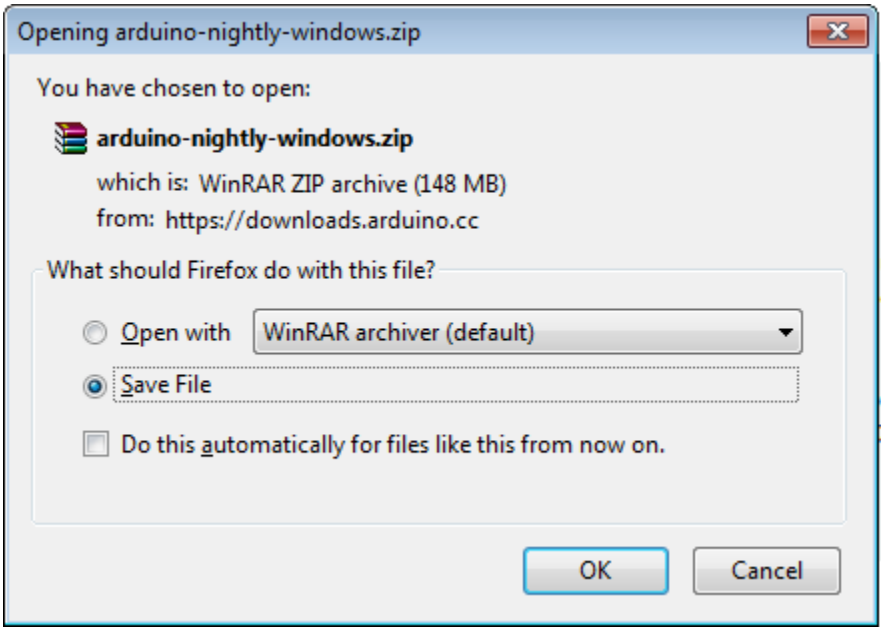
Windows için Arduino IDE'yi kurmak için talimatları izlemeliyiz.

Arduino IDE'yi kullanarak USB üzerinden ana çip olan ATmega328p'ye yeni programlar yükleyebilirsiniz.

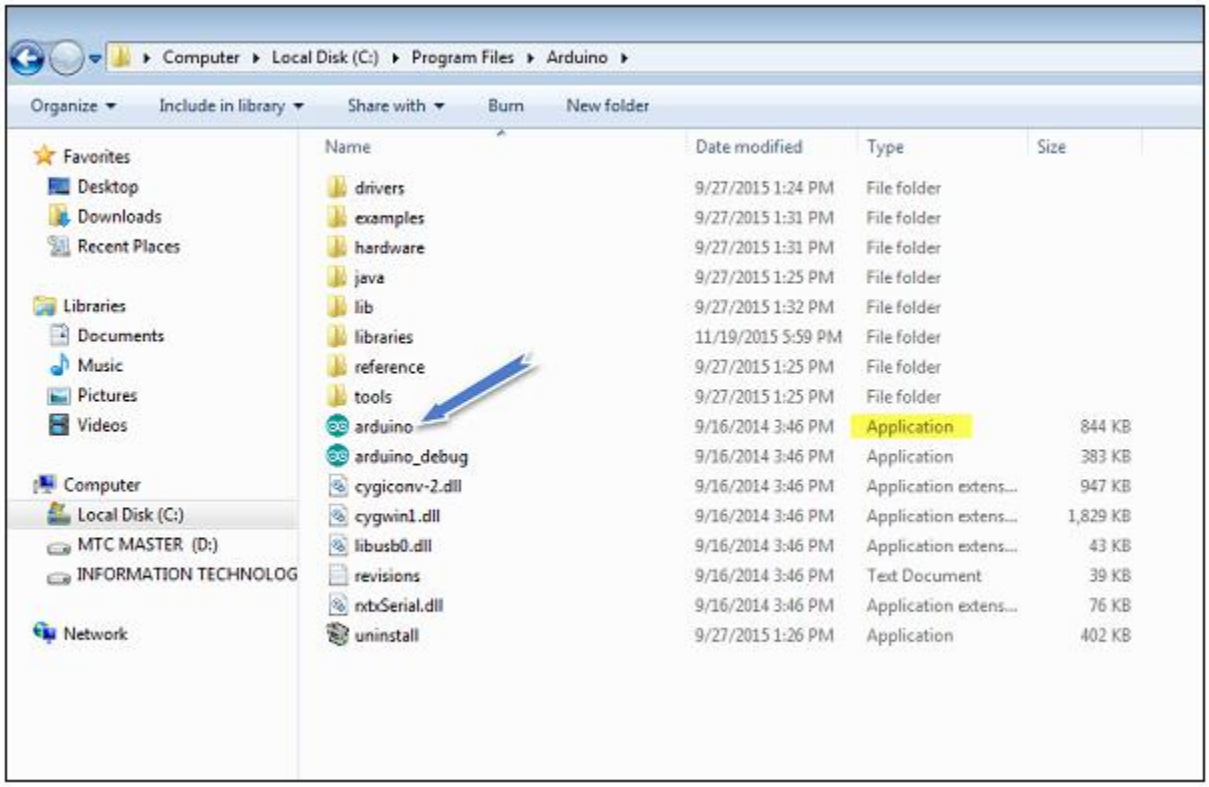
Arduino IDE'yi indirmek için lütfen aşağıdaki bağlantıya tıklayın:
https://www.arduino.cc/en/Main/Software



Kullanmakta olduğunuz İşletim Sistemini seçin ve indirin. Arduino IDE yazılımımız indirildikten sonra klasörü açmamız gerekiyor.



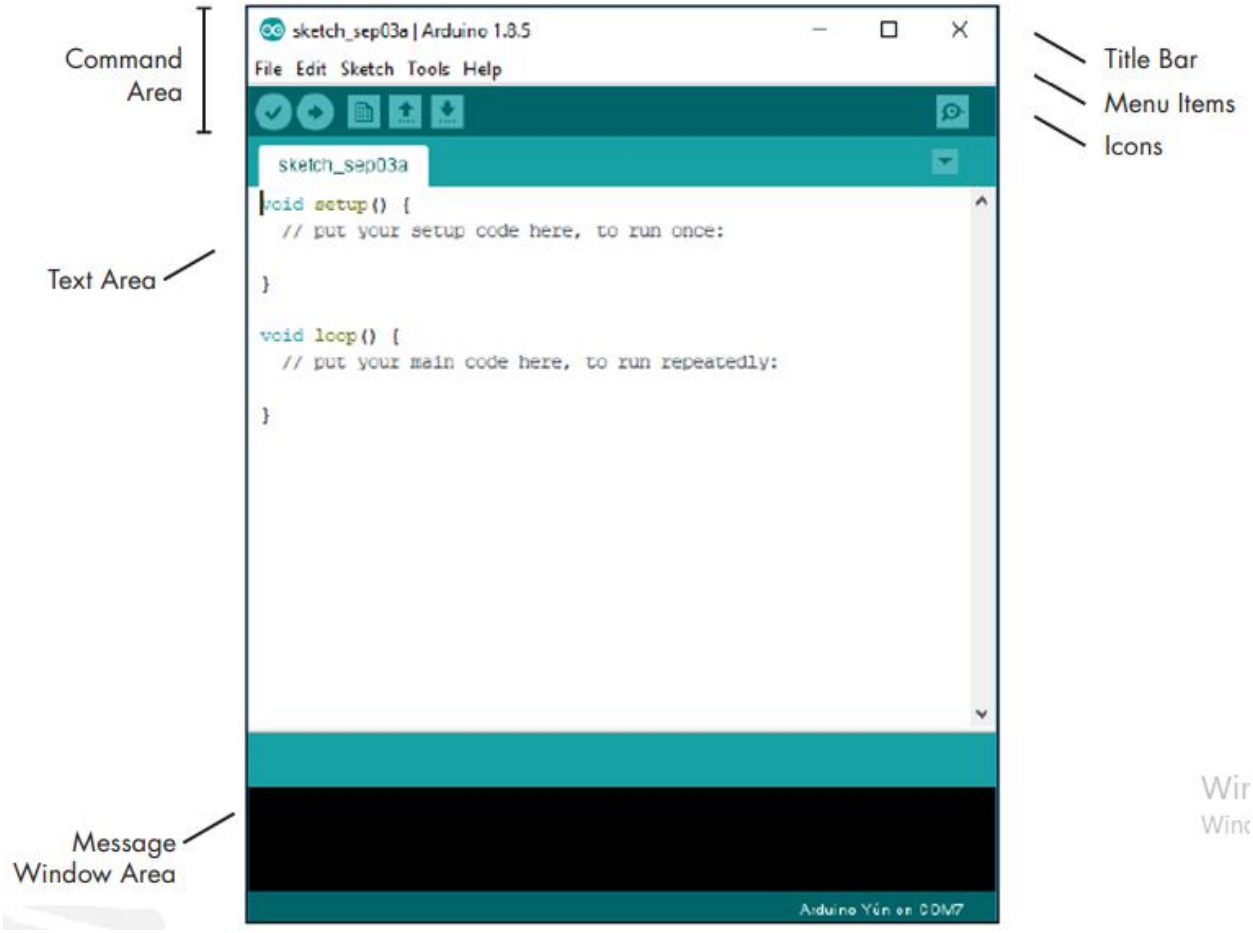
Klasörün içinde, sonsuzluk etiketli (application.exe) uygulama simgesini bulabiliriz. IDE'yi başlatmak için simgeye çift tıklayın. Ardından, Arduino IDE'yi kurmak için kurulum sihirbazını takip etmeniz yeterlidir.



Program Yazmak için Arduino IDE Penceresi

Arduino IDE'yi ilk açtığınızda, aşağıdaki şekle benzer bir şey görmelisiniz.

Aşağıdaki şekilde gösterildiği gibi, Arduino IDE basit bir kelime işlemciye benzer. IDE üç ana alana bölünmüştür: komut alanı, metin alanı ve mesaj penceresi alanı.



Menu Öğeleri

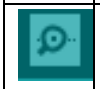
Herhangi bir kelime işlemci veya metin düzenleyicide olduğu gibi, seçeneklerini görüntülemek için menü öğelerinden birine tıklayabilirsiniz.
File(dosya): Senaryoları kaydetme, yükleme ve yazdırma seçeneklerini içerir;
Edit(düzen): Herhangi bir kelime işlemcide ortak olan normal kopyalama, yapıştırma ve arama işlevlerini içerir
Sketch(çizim): Bir panoya yüklemeden önce çiziminizi doğrulama işlevini ve bazı çizim klasörünü ve içe aktarma seçeneklerini içerir.
Tools(araçlar): Arduino kart tipini ve USB bağlantı noktasını seçme komutlarının yanı sıra çeşitli işlevleri içerir.
Help(yardım): Çeşitli ilgi alanlarına ve IDE sürümüne bağlantılar içerir.

Sketch Nedir?

Arduino projesi, belirli bir görevi gerçekleştirmek için oluşturduğunuz bir dizi talimattır; başka bir deyişle, eskiz bir programdır.
Sketch, Arduino'nun gerçekleştirmesi için bir dizi talimattan başka bir şey değildir. Arduino IDE kullanılarak oluşturulan eskizler .pde dosyaları olarak kaydedilir. Bir program oluşturmak için üç ana bölümün olması gerekir: Değişken tanımlamaları, Kurulum işlevi ve ana Döngü işlevi.

Arduino IDE Araç Çubuğu Butonları

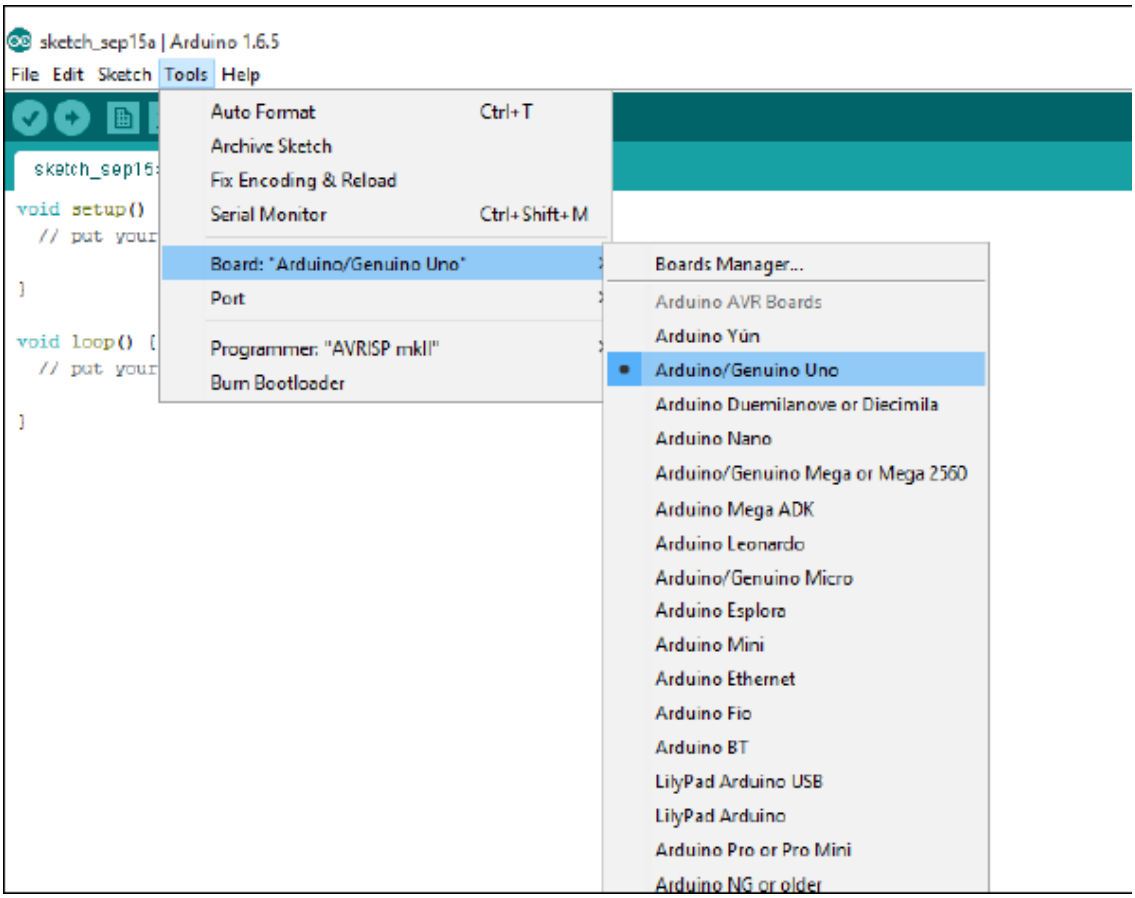
Menü araç çubuğunun altında altı simge bulunur. Adını görüntülemek için fareyi her simgenin üzerine getirin. Simgeler, soldan sağa aşağıdaki gibidir:

	**Doğrula (Derle) :** Arduino taslağının geçerli olduğunu ve herhangi bir programlama hatası içermediğini kontrol etmek için buna tıklayın.
	**Yeni(New):** Yeni bir pencerede yeni bir boş Sketch açmak için buna tıklayın.
	**Aç(Open):** Kaydedilmiş bir programı açmak için buna tıklayın.
	**Kaydet(Save):** Açık programı kaydetmek için buna tıklayın. Sketch daha önce kaydedilmemişse bir isim vermeni istenir.
	**Yükleme(Upload):** Onaylamak ve ardından programınızı Arduino panosuna yüklemek için buna tıklayın.
	**Seri Monitör:** Arduino'nuz ve IDE arasında veri göndermek ve almak için yeni bir pencere açmak için buna tıklayın.

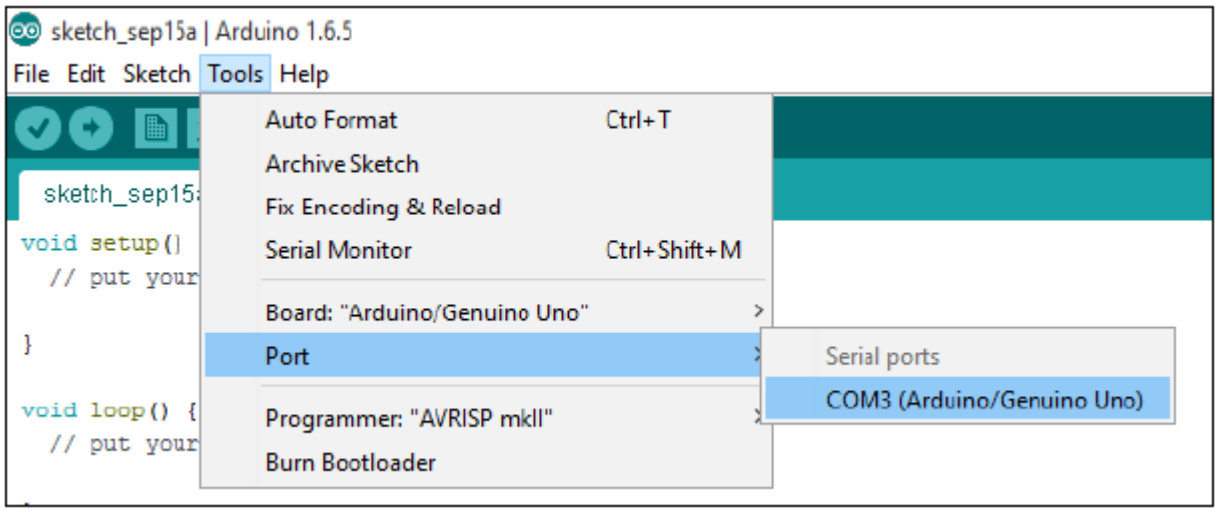
Ardunio'yu Bağlama

Arduino UNO'nuzu USB üzerinden bilgisayarınıza bağlayın.
Arduino'nuzu bir USB kablosu ile bağladıktan sonra Arduino IDE'nin doğru kartı seçtiğinden emin olmanız gerekir.

Arduino Uno kullanıyoruz, bu yüzden **Tools → Board→Arduino/Genuino Uno** seçili olmalıdır.



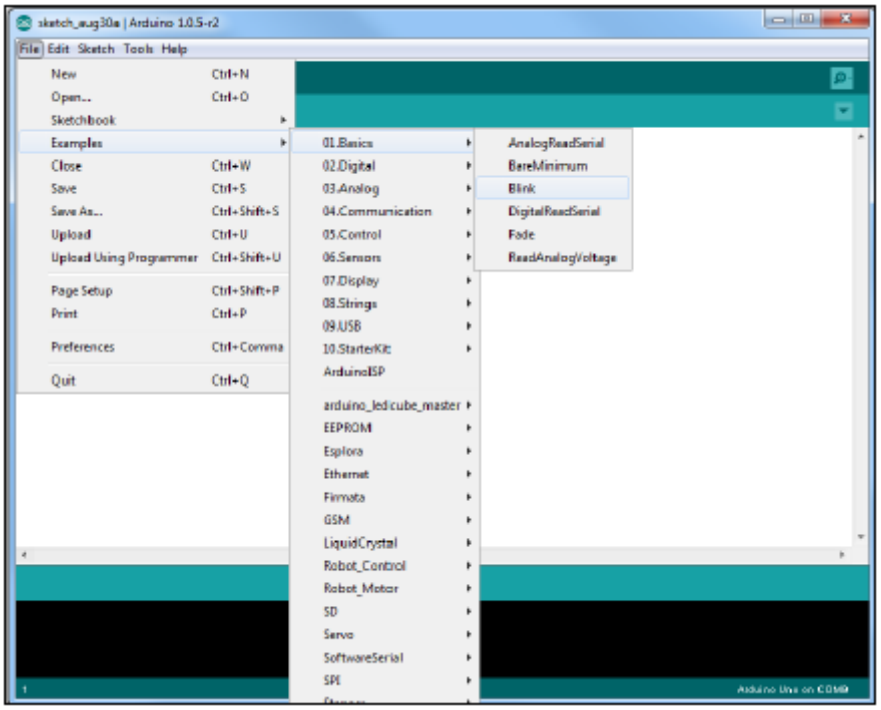
Port'a gidin ve doğru portu seçin. Ardından Arduino'nuzun bağlı olduğu seri portu seçmelisiniz. **Tools→Port**'a gidin ve doğru portu seçin.



Arduino Sketch Yükleme

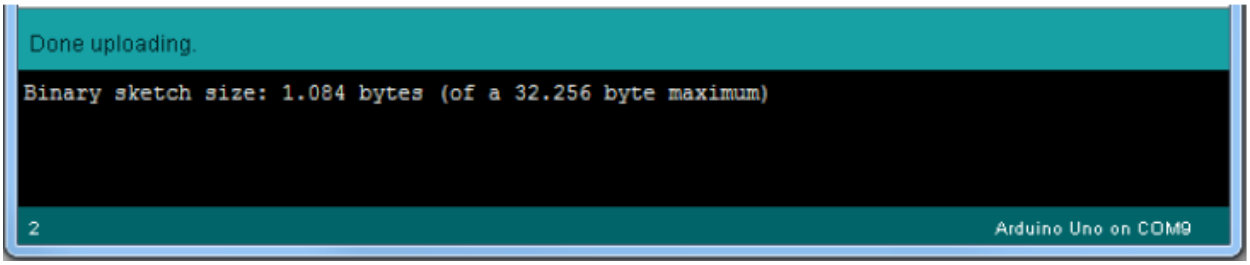
Arduino kartınıza nasıl kod yükleyeceğinizi göstermek için size basit bir örnek. Bu en temel örneklerden biridir. LED yakıp söndürür.

1. Arduino IDE'nizi açın.
2. **File→Examples→ 01.Basics→Blink e gidin**



Varsayılan olarak, Arduino IDE, Arduino UNO için önceden yapılandırılmış olarak gelir. Yükle düğmesini tıklayın ve birkaç saniye bekleyin.

Birkaç saniye sonra, **Yükleme tamamlandı (Done uploading)** mesajını görmelisiniz.



Bu kod, Arduino UNO'nuzdaki yerleşik LED'i yakıp söndürür (kırmızı renkle vurgulanır). Küçük LED'in bir saniye boyunca yandığını ve bir saniye söndüğünü görmelisiniz.

Çıkışı Kontrol Edin ve Girişi Okuyun

Bir Arduino kartı, dijital pinler, analog pinler ve PWM pinleri içerir.

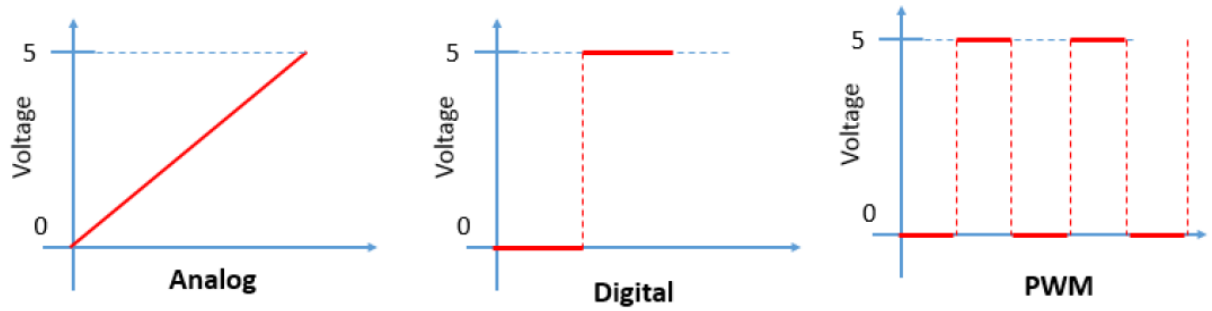
Dijital, analog ve PWM arasındaki fark

Dijital pinlerde, açık veya kapalı olmak üzere sadece iki olası durumunuz vardır. Bunlar ayrıca Yüksek veya Düşük, 1 veya 0 ve 5V veya 0V olarak da adlandırılabilir.

Örneğin, bir LED yanıyorsa durumu Yüksek veya 1 veya 5V'dir. Kapalıysa, Düşük veya 0 veya 0V değerine sahip olursunuz.

Analog pinlerde, 0 ile 1023 arasında sınırsız olası durumunuz vardır. Bu, sensör değerlerini okumanızı sağlar. Örneğin, ışık sensörü ile çok karanlıksa 1023, çok parlaksa 0 okursunuz Karanlık ile çok parlak arasında bir parlaklık varsa 0 ile 1023 arasında bir değer okursunuz.

PWM pinleri dijital pinlerdir, yani 0 veya 5V çıkış verirler. Ancak bu pinler “Pulse Width Modulation” (PWM) yapabildikleri için 0 ile 5V arasında “sahte” ara gerilim değerleri verebilirler. PWM, Arduino'nun çıkış voltajını titreştirerek değişen güç seviyelerini "simüle etmeye" izin verir.



Bir çıkışı kontrol etme

Dijital bir çıkışı kontrol etmek için digitalWrite() işlevini ve yazdığınız parantezler arasında, kontrol etmek istediğiniz pini ve ardından YÜKSEK veya DÜŞÜK kullanın.

Bir PWM pinini kontrol etmek için analogWrite() işlevini kullanırsınız ve parantezler arasına kontrol etmek istediğiniz pini ve 0 ile 255 arasında bir sayı yazarsınız.

Giriş okuma

Bir analog girişi okumak için analogRead() işlevini ve bir dijital giriş için digitalRead() işlevini kullanırsınız.

Not: Arduino öğrenmenin en iyi yolu pratik yapmaktır. Bu yüzden birçok proje yapın ve bir şeyler geliştirmeye başlayın.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+ KA-202

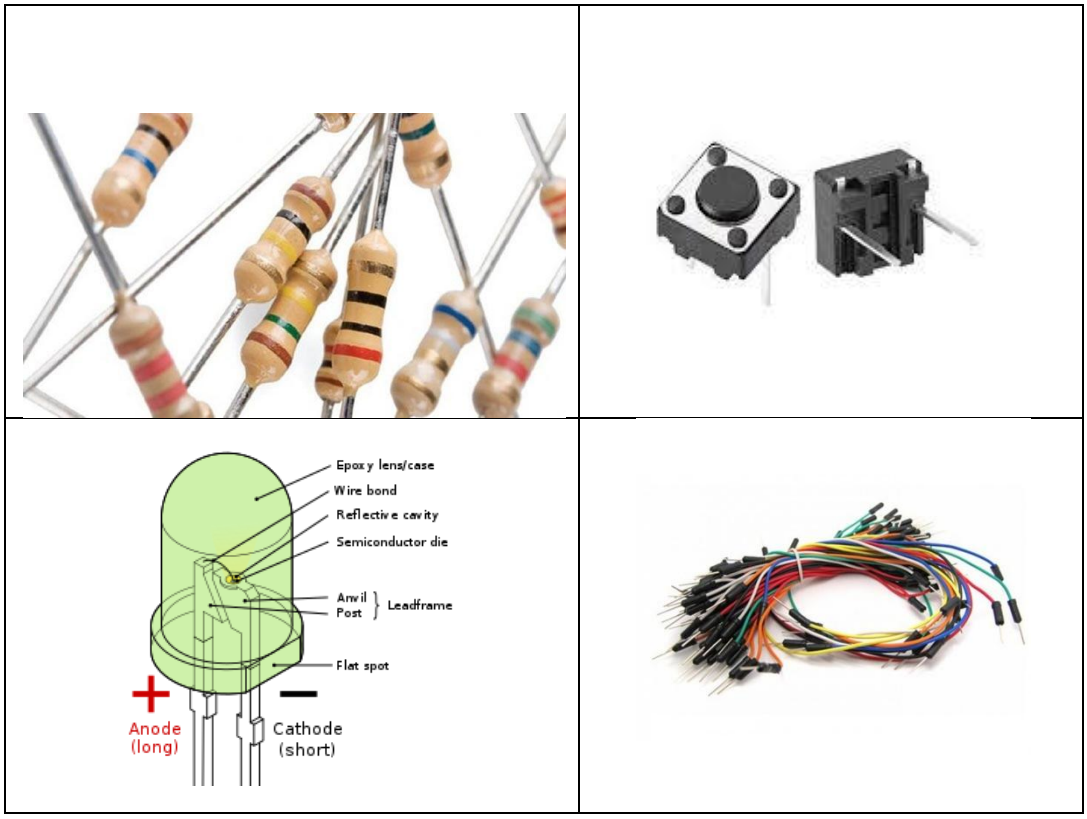
Mesleki Eğitim ve Öğretime Yönelik Stratejik Ortaklıklar Projesi

Proje Başlığı : "Mesleki Eğitimde Arduino'ları Öğretme ve Öğrenme"

Proje Kısaltması: " ARDUinVET "

Proje No: "2020-1-TR01-KA202-093762"

Arduino Giriş/Çıkış Modülü ve Kiti



Proje Planı

İlk projelerimize başlarken, yeni bir fikir bulduktan hemen sonra taslağınızı yazmak isteyebilirsiniz. Ancak yazmaya başlamadan önce birkaç temel hazırlık adımı sıralanmıştır. Sonuçta, Arduino kartınız bir zihin okuyucu değildir; kesin talimatlara ihtiyaç duyar ve bu talimatlar Arduino tarafından yürütülsede, küçük bir ayrıntıyı bile gözden kaçırırsanız sonuçlar beklediğiniz gibi olmayabilir.

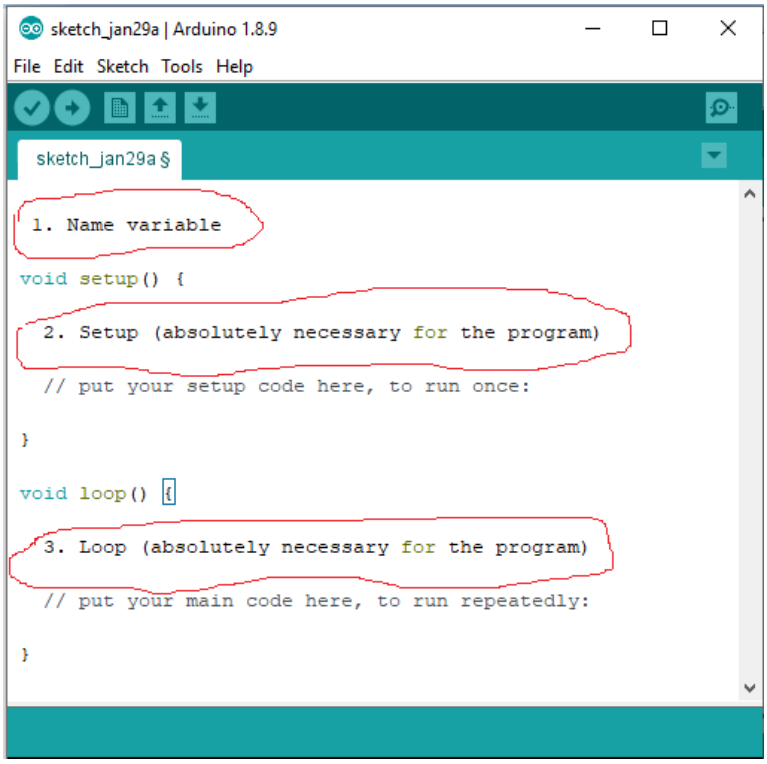
İster basitçe yanıp sönen bir proje, ister otomatikleştirilmiş bir demiryolu sinyali modeli oluşturuyor olun, başarının temeli ayrıntılı bir plandır. Arduino projelerinizi tasarlarken şu temel adımları izleyin:

1. Hedefinizi tanımlayın. Neye ulaşmak istediğinizi belirleyin.
2. Algoritmanızı yazın. Algoritma, projenizi nasıl gerçekleştireceğinizi açıklayan bir dizi talimattır. Algoritmanız, projenizin amacına ulaşmanız için gerekli adımları listeleyecektir.
3. Donanımınızı seçin. Arduino'ya nasıl bağlanacağını belirleyin.
4. Eskizinizi yazın. Arduino'ya ne yapacağını söyleyen ilk programınızı oluşturun.
5. Kabloyu bağlayın. Donanımınızı, devrenizi ve diğer öğelerinizi Arduino kartına bağlayın.
6. Test edin ve hata ayıklayın. Çalışıyor mu? Bu aşamada, ister çizimde, ister donanımda veya algoritmada olsun, hataları tanımlar ve nedenlerini bulursunuz.

Projenizi planlamak için ne kadar çok zaman harcarsanız, test ve hata ayıklama aşaması daha kolay olur.

Bir projenin temel yapısı

Arduino programı "sketch" olarak adlandırılır. Bir Sketch üç bölüme ayrılabilir.



1. Name variable(Değişken tanımlama-zorunlu değildir):
İlk bölümde programın elemanları isimlendirilmiştir. Bu kısım zorunlu değildir.

2. Setup(Kurulum-program için kesinlikle gereklidir):
Setup'daki kodlar sadece bir kez çalıştırılır.
Giriş ve çıkış pinleri belirlenmelidir.
Çıkış pinleri: Pin bir voltaj vermelidir. Örneğin: LED 'in yanması
Giriş pinleri: Kart bir voltaj okumalıdır. Örneğin: Bir anahtarın harekete geçirilmesi

3. Loop(Döngü-program için kesinlikle gereklidir):
Bu döngü kısmı kart tarafından sürekli olarak tekrarlanacaktır. Proje çalışır, sona gelince başa döner ve böyle devam eder.

Diğer Sözdizimi Kuralları

Arduino programları yazarken bu kurallara dikkat etmek gerekir. Aksi takdirde programımız başarısız olacaktır.

; (Noktalı virgül):

; (Noktalı virgül) bir ifadeyi bitirmek için kullanılır. Bir satırı noktalı virgülle bitirmeyi unutmak derleyici hatasına neden olur.

Örnek: int a=13;

{ } (Küme Parantezi):

Küme parantezleri Arduino programlama dilinin önemli bir parçasıdır. Birkaç farklı yapıda kullanılırlar ve bu bazen yeni başlayanlar için kafa karıştırıcı olabilir. Bir açılış küme parantezi "{" her zaman bir kapanış küme ayracı "}" izlemelidir.

Küme parantezinin ana kullanımları: Fonksiyonlar, Döngüler, Koşullu ifadeler

Örnek:

```
 void myfunction(datatype argument)
{           statements(s)

}
```

// (Tek satırlı yorum) ve /* */ (Çok satırlı yorum):

Bunlar, programın çalışma şekli hakkında kendinizi veya başkalarını bilgilendirmek için kullanılan programdaki satırlardır. Derleyici tarafından yok sayılırlar ve işlemciye aktarılmazlar, bu nedenle Atmega çipinde herhangi bir yer kaplamazlar. Yorumların tek amacı, programınızın nasıl çalıştığını anlamanıza (veya hatırlamanıza) yardımcı olmak veya programınızın nasıl çalıştığını başkalarına bildirmektir. Bir satırı yorum olarak işaretlemenin iki farklı yolu vardır:

Örnek:

```
x = 5;      // Bu tek satırlık bir yorumdur. Eğik çizgiden sonraki her şey bir yorumdur

x = 5;    /* Bu çok satırlı bir yorumdur ……

Bu, çok satırlı bir yorumun sonu. */
```

#define:

#define program derlenmeden önce programcının sabit bir değere bir isim vermesini sağlar. Arduino'da tanımlı sabitler çip üzerinde herhangi bir program hafızasında yer kaplamaz. Genel olarak, sabitleri tanımlamak için const anahtar sözcüğü tercih edilir ve bunun yerine #define kullanılmalıdır.

Örnek:

#define ledPin 3 // Derleyici, derleme zamanında ledPin'in değerini 3 değeriyle değiştirecektir.

#include:

#include projenize dış kitaplıkları dâhil etmek için kullanılır. Bu, programcıya geniş bir standart C kitaplığı grubuna (önceden oluşturulmuş işlev grupları) ve ayrıca özellikle Arduino için yazılmış kitaplıklara erişim sağlar.

Örnek:

#include <servo.h>

Ardunio – Veri Türleri

Veri türleri, değişkenleri veya farklı türlerdeki işlevleri bildirmek için kullanılır. Bir değişkenin türü, depolamada ne kadar yer kapladığını ve depolanan bit modelinin nasıl yorumlanacağını belirler.

Herhangi bir bilgiyi bellekte saklamak için kullanılan ve program akışı sırasında değerini değiştirebilen ifadelere değişkenler denir. Değişkenler sayılar, karakterler veya mantıksal ifadeler olabilir. Değişken tipine göre uygun veri tipi seçilmelidir. Değişkeni bellekte tanımlamada kullanılan veri tipine göre belirli bir alan tahsis edilir.

Aşağıdaki tablo Arduino programlama sırasında kullanacağınız tüm veri tiplerini sunmaktadır.

Veri tipi	**Değeri**
boolean	Doğru(True) veya yanlış(False)
char	-128 to 127
byte	0 to 255
unsigned char	0 to 255
int	-32,768 to 32,767
unsigned int	0 to 65,535
word	İşaretsiz int
long (or long int)	-2,147,483,648 to 2,147,483,647
unsigned long	0 to 4,294,967,295
float	-3.4028235E+38 to 3.4028235E+38
double	Float benzeri

Arduino - Değişkenler ve Sabitler

Değişken tanımlanırken değişkenin adı, değişkenin değeri ve değişkene uygun veri türü belirlenmelidir.

Tanım aşağıdaki örnekte görüldüğü gibi yapılmıştır:

int LED =12;

int=veri tipi **LED**= Değişken adı **12**=Değişken değeri

Arduino'nun kullandığı değişkenler, kapsam adı verilen bir özelliğe sahiptir. Kapsam, programın bir bölgesidir ve değişkenlerin bildirilebileceği üç yer vardır. Bunlar:

- Yerel değişkenler olarak adlandırılan bir fonksiyon veya blok içinde.
- Biçimsel parametreler olarak adlandırılan fonksiyon parametrelerinin tanımında.
- Global değişkenler olarak adlandırılan tüm fonksiyonların dışında.

Sabit, değişkenin davranışını değiştiren ve bir değişkeni *"salt okunur"* yapan bir değişken niteleyicidir. Bu, değişkenin kendi türündeki herhangi bir değişken gibi kullanılabileceği, ancak değerinin değiştirilemeyeceği anlamına gelir. Bir const değişkenine değer atamaya çalışırsanız derleyici hatası alırsınız.

const anahtar sözcüğüyle tanımlanan sabitler, diğer değişkenleri yöneten değişken kapsamı kurallarına uyar. Bu ve #define kullanmanın tuzakları, const anahtar sözcüğünü sabitleri tanımlamak için üstün bir yöntem yapar ve #define kullanımına göre tercih edilir.

Örnek:
const float pi = 3.14;

Not: #define veya const: Sayısal veya dize sabitleri oluşturmak için const veya #define kullanabilirsiniz. Diziler için const kullanmanız gerekecektir. Genel olarak sabitleri tanımlamak için #define yerine const tercih edilir.

Arduino – Operatörler

Operatör, derleyiciye belirli matematiksel veya mantıksal işlevleri gerçekleştirmesini söyleyen bir semboldür. Arduino'da aşağıdaki operatör türleri kullanılabilir.

Matematiksel Operatörler:

A değişkeninin 10'u ve B değişkeninin 20'yi tuttuğunu varsayalım, o zaman

Operatör adı	Operatör sembolü	Örnek
atama operatörü	=	A = B
toplama	+	A + B sonuç 30
çıkartma	-	A - B sonuç -10
çarpma	*	A * B sonuç 200
bölme	/	B / A sonuç 2
mod	%	B % A sonuç 0

Karşılaştırma Operatörleri:

A değişkeninin 10'u ve B değişkeninin 20'yi tuttuğunu varsayalım, o zaman

Operatör adı	Operatör sembolü	Örnek
eşit eşit	==	(A == B) yanlış (false)
eşit değil	!=	(A != B) doğru(true)
küçüktür	<	(A < B) doğru(true)
büyüktür	>	(A > B) yanlış (false)

küçük eşit	<=	(A <= B) doğru(true)
Büyük eşit	>=	(A >= B) yanlış (false)

Mantıksal Operatörler

A değişkeninin 10'u ve B değişkeninin 20'yi tuttuğunu varsayalım, o zaman

Operatör adı	Operatör sembolü	Örnek
and	&&	(A && B) doğru(true)
or	||	(A || B) doğru(true)
not	!	!(A && B) yanlış (false)

Bitsel Operatörler

A değişkeninin 60'ı ve B değişkeninin 13'yi tuttuğunu varsayalım, o zaman

Operatör adı	Operatör sembolü	Örnek
and	&	(A & B) sonuç 12 (0000 1100)
or	|	(A | B) sonuç 61 (0011 1101)
xor	^	(A ^ B) sonuç 49 (0011 0001)
not	~	(~A) sonuç -60 (1100 0011)
shift left	<<	A << 2 sonuç 240 (1111 0000)
shift right	>>	A >> 2 sonuç15 (0000 1111)

Arduino - I/O Fonksiyonları (Komutlar)

Fonksiyonlar, programların bireysel görevleri gerçekleştirecek şekilde yapılandırılmasına izin verir. Bir programda aynı eylemi birden çok kez gerçekleştirmeniz gerektiğinde fonksiyonlar kullanılır. Aşağıdaki devrelerde ve Arduino programlarında gerçek program örneklerini gösterdiğimizde daha anlaşılır olacaktır. Çeşitli komut işlevlerini açıklamaya yardımcı olmak için bunları ayrı komutlara ayırdık.

Arduino kartındaki pinler giriş veya çıkış olarak yapılandırılabilir. Bu modlarda pinlerin işleyişini anlatacağız. Arduino analog pinlerinin çoğunluğunun dijital pinlerle tamamen aynı şekilde yapılandırılabileceğini ve kullanılabileceğini belirtmek önemlidir.

pinMode() Fonksiyonu:

pinMode() fonksiyonu, belirli bir pini giriş veya çıkış olarak davranacak şekilde yapılandırmak için kullanılır. INPUT_PULLUP modu ile dahili pull-up dirençlerini etkinleştirmek mümkündür.

Kullanımı: pinMode(pin, mod)

```
Void setup () {
pinMode (pin , mod);
                }
```

Parametreler:
pin: Arduino pin numarası.
mod: INPUT, OUTPUT veya INPUT_PULLUP.

Örnekler:

```
pinMode(13, OUTPUT);      // dijital pin 13'ü çıkış olarak ayarlar
pinMode(5, INPUT);        // dijital pin 5'i giriş olarak ayarlar
```

digitalWrite() Fonksiyonu:

digitalWrite() fonksiyonu, bir dijital pine YÜKSEK veya DÜŞÜK bir değer yazmak için kullanılır. Pin, pinMode() ile bir ÇIKIŞ olarak yapılandırılmışsa, voltajı ilgili değere ayarlanacaktır. YÜKSEK için 5V, DÜŞÜK için 0V (toprak). Pin bir GİRİŞ olarak yapılandırılmışsa, digitalWrite() giriş pinindeki dahili çekmeyi etkinleştirir (YÜKSEK) veya devre dışı bırakır (DÜŞÜK). Dahili pull-up direncini etkinleştirmek için pinMode() öğesini INPUT_PULLUP olarak ayarlamanız önerilir.

pinMode() öğesini OUTPUT olarak ayarlamazsanız ve bir LED'i bir pine bağlamazsanız, digitalWrite(HIGH) çağrılırken LED loş görünebilir. pinMode() açıkça ayarlanmadan, digitalWrite() büyük bir akım sınırlayıcı direnç gibi davranan dahili pull-up direncini etkinleştirmiş olacaktır.

Kullanımı:

digitalWrite (pin ,değer);

pin: Modunu ayarlamak istediğiniz Ardunio pin numarası.

değer: HIGH (1) veya LOW(0).

Örnek:

digitalWrite(LED, HIGH); // Led yanar

digitalWrite(LED, LOW); // Led söner

digitalRead()Fonksiyonu:

digitalRead() fonksiyonu, değeri YÜKSEK veya DÜŞÜK olarak belirtilen bir dijital pinden okur.

Kullanımı:

digitalRead(pin);

pin: Okumak istediğiniz Arduino pin numarası.

Örnekler:

değer = digitalRead(inPin); // giriş pinini oku

Not: Analog giriş olarak adlandırılan A0, A1, vb., pinleri dijital pinler olarak kullanılabilir.

delay() Fonksiyonu:

delay() fonksiyonu, programı parametre olarak belirtilen süre boyunca (milisaniye cinsinden) duraklatır. (Bir saniyede 1000 milisaniye vardır.)

Kullanımı: delay(ms);

ms: duraklatılacak milisaniye sayısı. İzin verilen veri türleri: işaretsiz long

Örnekler:

delay(1000); // 1 saniye bekletir

delay(2000); // 2 saniye bekletir

analogWrite() Fonksiyonu:

analogWrite() fonksiyonu, bir pine bir analog değer (PWM dalgası) yazar. Bir LED'i değişen parlaklıklarda yakmak veya bir motoru çeşitli hızlarda sürmek için kullanılabilir. analogWrite() çağrısından sonra pin, aynı pin üzerinde bir sonraki analogWrite() çağrısına (veya digitalRead() veya digitalWrite() çağrısına) kadar belirtilen görev döngüsünün sabit bir dikdörtgen dalgasını üretecektir.

Örnekler:
Potansiyometreden okunan değerle orantılı olarak çıkışı LED'e ayarlar.

değer = analogRead(analogPin); // giriş pinini oku

analogWrite(ledPin, değer /4); // analogRead değerleri 0'dan 1023'e, analogWrite değerleri 0'dan 255'e

analogRead( ) Fonksiyonu:

Arduino, pinlerinden birine voltaj uygulanıp uygulanmadığını tespit edip digitalRead() fonksiyonu ile raporlayabilmektedir. (Bir nesnenin varlığını algılayan) bir açma/kapama sensörü ile değeri sürekli değişen bir analog sensör arasında fark vardır. Bu tip sensörü okumak için farklı bir pin tipine ihtiyacımız var.

Arduino kartının sağ alt kısmında "Analog In" olarak işaretlenmiş altı pin göreceksiniz. Bu özel pinler sadece kendilerine uygulanan bir voltajın olup olmadığını değil, değerini de söyler. AnalogRead() fonksiyonunu kullanarak pinlerden birine uygulanan voltajı okuyabiliriz.

Bu işlev, 0 ile 5 volt arasındaki voltajları temsil eden 0 ile 1023 arasında bir sayı döndürür. Örneğin, 0 numaralı pine uygulanan 2,5 V'luk bir voltaj varsa, analogRead(0) 512 değerini döndürür.

Kullanımı: analogRead(pin);

pin: 0'dan 5'e kadar okunacak analog giriş pininin numarası.

Örnek:

```
değer= analogRead(analogPin);       // giriş pinini oku

Serial.println(değer);              // değeri Seri monitöre yaz
```

If Komutu:

if deyimi bir koşulu kontrol eder ve koşul "doğru" ise aşağıdaki deyimi veya deyim kümesini yürütür.

Kullanımı:

```
if (koşul) {
        // koşul doğru olduğunda yürütülecek işlemler
}
```

koşul: bir boolean ifadesi (yani, doğru veya yanlış olabilir).

Örnekler: Bir if ifadesinden sonra parantezler atlanabilir. Bu yapılırsa, sonraki satır (noktalı virgülle tanımlanır) tek koşullu ifade olur.

```
if (x > 120) digitalWrite(LEDpin, HIGH);

if (x > 120)
digitalWrite(LEDpin, HIGH);

if (x > 120) {digitalWrite(LEDpin, HIGH);}

if (x > 120) {
  digitalWrite(LEDpin1, HIGH);
  digitalWrite(LEDpin2, HIGH);
}
          // hepsi doğru
```

if-else Komutu:

if…else, birden çok koşulun gruplandırılmasına izin vererek, kod akışı üzerinde temel if ifadesinden daha fazla kontrol sağlar. If deyimindeki koşul yanlış(false) ile sonuçlanırsa, else yan tümcesi (eğer varsa) yürütülür. Else, aynı anda birden fazla, birbirini dışlayan testin çalıştırılabilmesi için bir başka if koşuluna geçebilir.

Doğru koşulla karşılaşılıncaya kadar her koşul bir sonrakine geçecektir. Bir koşul doğru olduğunda ilgili kod bloğu çalıştırılır ve program daha sonra tüm if/else yapısını izleyen satıra atlar. Hiçbir koşul doğru değilse, varsa varsayılan else bloğu yürütülür.

Kullanımı:

```
if (koşul DOĞRU ise) {
  // A işini yap
}
else
  //eğer YANLIŞ ise,  B işini yap
}
```

```
if (koşul1) {
  // A işini yap
}
else if (koşul2) {
  // B işini yap
}
else {
  // C işini yap
}
```

Örnekler: Aşağıda, sıcaklık sensörü sistemi için yazılmış koddan bir alıntı bulunmaktadır.

```
if (sıcaklık >= 70) {
  // Tehlike! Sistemi kapatın.

}
else if (sıcaklık >= 60) { // 60 <= sıcaklık < 70
  // Uyarı! Kullanıcının dikkati gereklidir.

}
else { // sıcaklık< 60
  // Güvenli! Varsayılan işlemleri yap
}
```

for Komutu:

For ifadesi, küme parantezleri içine alınmış bir ifade bloğunu tekrarlamak için kullanılır. Bir artış sayacı genellikle döngüyü artırmak ve sonlandırmak için kullanılır. for ifadesi, herhangi bir tekrarlayan işlem için kullanışlıdır ve genellikle veri/pin koleksiyonları üzerinde çalışmak için dizilerle birlikte kullanılır.

Kullanımı:

```
for (başlangıç değeri; koşul; artış miktarı) {
        // yapılacaklar;
}
```

Parametreler:

başlangıç: önce ve tam olarak bir kez olur.

koşul: Döngü boyunca her seferinde koşul test edilir; doğruysa, deyim bloğu ve artış yürütülürse, koşul yeniden test edilir. Koşul yanlış olduğunda döngü sona erer.

artış: koşul doğru olduğunda döngü boyunca her seferinde yürütülür.

Örnekler:

```
for (int i = 0; i <= 255; i++)          {
   analogWrite(PWMpin, i);              }

for (int x = 2; x < 100; x = x * 1.5) {
        println(x);                     }

for (int i = 0; i > -1; i = i + x)   {
  analogWrite(PWMpin, i);       }
```

switch...case Komutu:

If ifadelerinde olduğu gibi, switch/case, programcıların çeşitli koşullarda yürütülmesi gereken farklı kodlar belirlemesine izin vererek programların akışını kontrol eder. Özellikle, bir switch ifadesi, bir değişkenin değerini case ifadelerinde belirtilen değerlerle karşılaştırır. Değeri değişkeninkiyle eşleşen bir case ifadesi bulunduğunda, o case ifadesindeki kod çalıştırılır.

break anahtar sözcüğü, switch deyiminden çıkar ve genellikle her durumun sonunda kullanılır. Bir break ifadesi olmadan, switch ifadesi bir break veya switch ifadesinin sonuna ulaşılana kadar aşağıdaki ifadeleri yürütmeye devam eder.

Kullanımı:

```
switch (var) {
  case sabit1:
    // yapılacaklar
    break;
  case sabit2:
    // yapılacaklar
    break;
  default:
    // yapılacaklar
    break;
}
```

Örnek Kod:

```
switch (değişken) {
  case 1:
    // değişken 1'e eşit olduğunda
yapılacaklar
    break;
  case 2:
    // değişken 2'e eşit olduğunda yapılacaklar
    break;
  default:
    // başka hiçbir şey eşleşmiyorsa,
varsayılanı yapın
    // varsayılan isteğe bağlıdır
    break;
}
```

var: değeri çeşitli durumlarla karşılaştırılacak bir değişken. İzin verilen veri türleri: int, char.

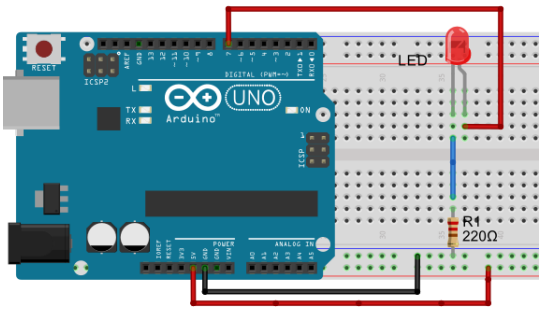
sabit1,sabit2: sabitler. İzin verilen veri türleri: int, char.

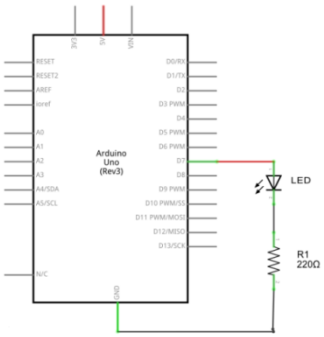
Not:

Arduino devreleri iki şekilde gösterilmiştir;

1-) Breadboard görünümü

2-) Şematik görünüm





1-) Breadboard görünümü 2-) Şematik görünüm

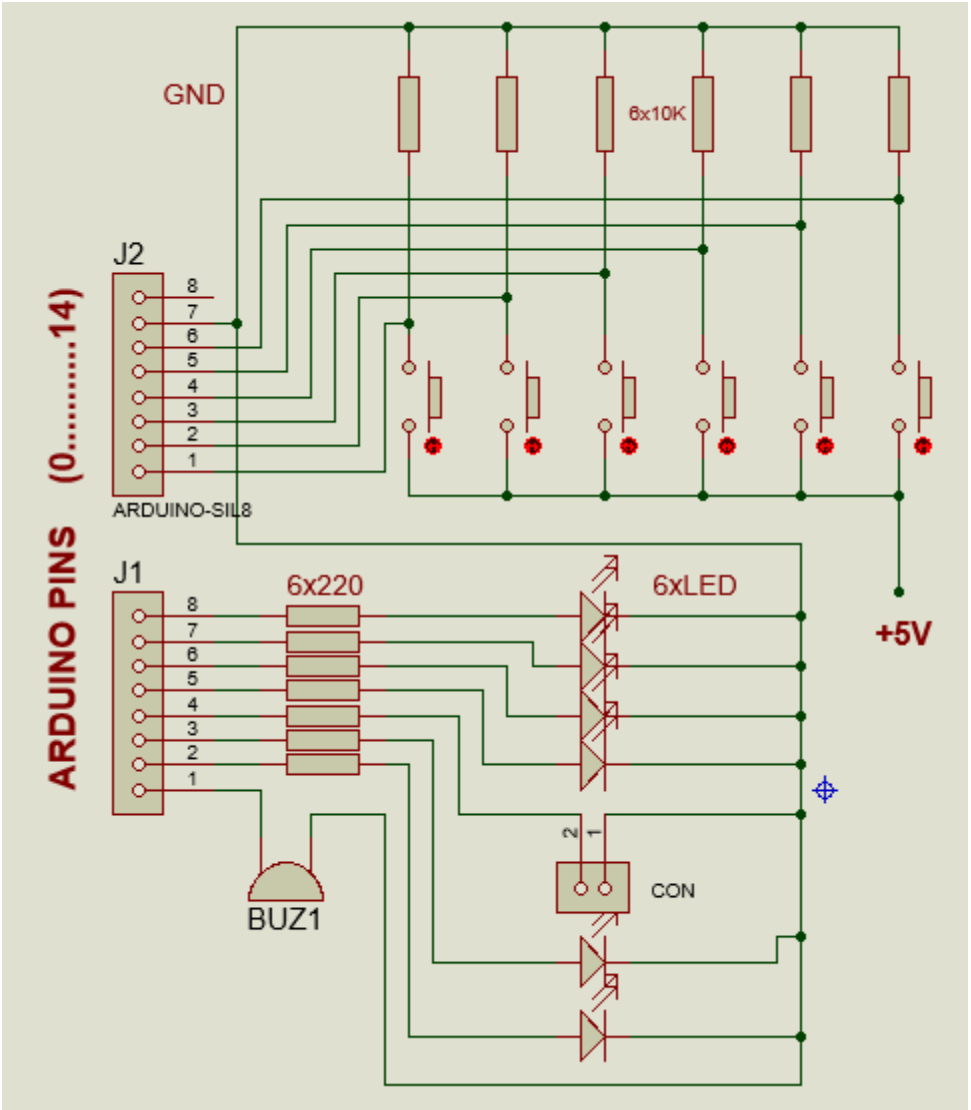
Daha çok kullanılan Breadboard görünümünü kullanacağız.

Yeter konuşma! Hadi bir şeyler yapalım!

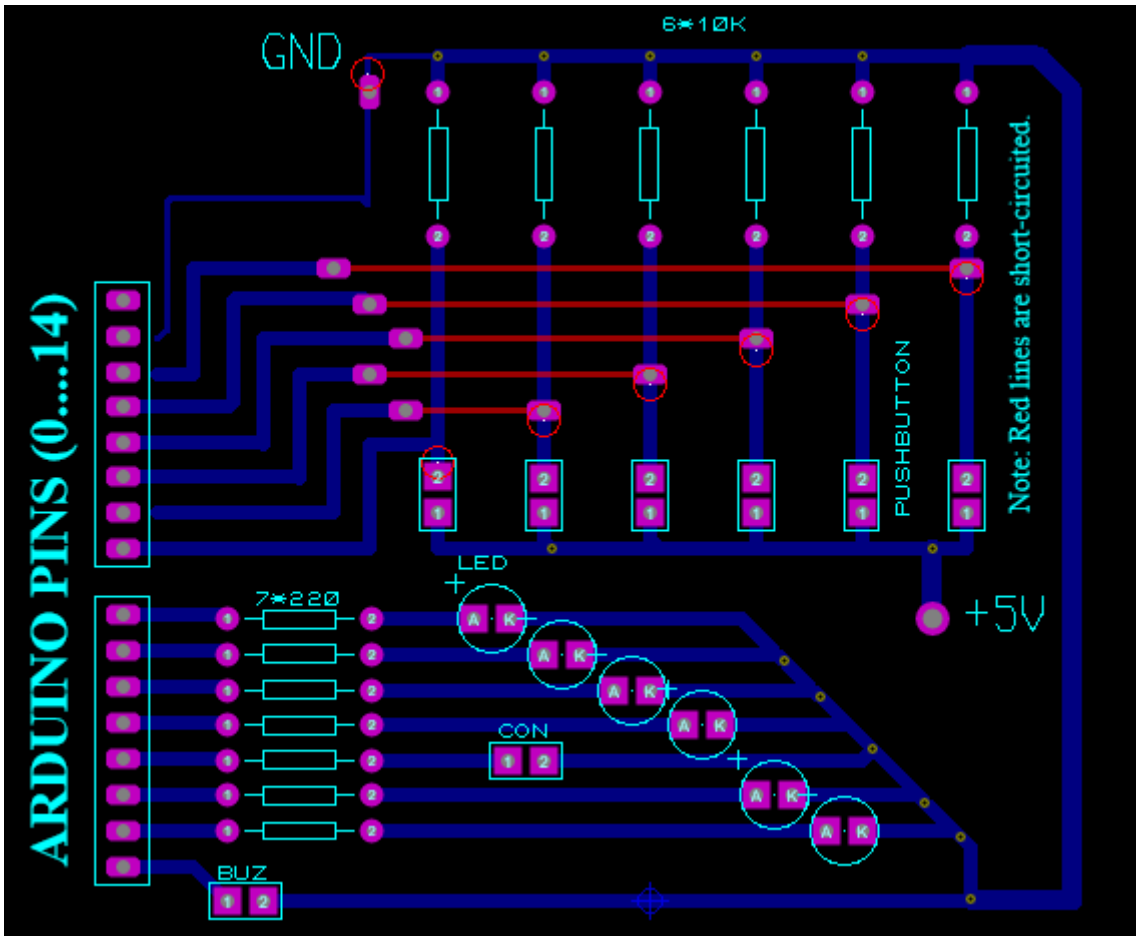
Arduino Buton Devreleri ve LED Modül Kiti

Arduino üzerindeki pinlerin çoğu giriş veya çıkış olarak konfigüre edilebilir. Buton ve LED Modül Kiti, öğrencilerin Arduino'ların I/O sistemlerini öğrenmelerini sağlamak için ARDUinVET'in ilk eğitim Kit'idir. Bu nedenle I/O Arduino eğitim KIT olarak adlandırılabilir. Bu eğitim kitinde aşağıda görüldüğü gibi 6 adet buton giriş olarak Arduino'ya bağlanmıştır. Ve bir buzzer, 2 pinli konnektör ve 6 LED çıkış olarak bağlanır.

Öğrenciler bu eğitim setini Arduino'nun I/O sistemlerini öğrenmek için kullanabilecekleri için bu eğitim seti Arduino devresini daha kolay test etmelerini sağlayabilir. Ayrıca Arduino devrelerini ve deneylerini test etmek için bir devre tahtası kullanabilirler.



Şekil: Buton ve LED Modül Kitinin Açık Devresi



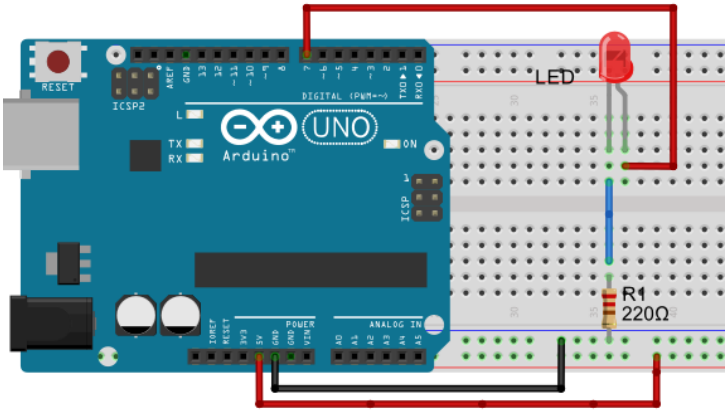
Şekil: Buton ve LED Modül Kitinin PCB Şeması

Örnek Arduio Devreleri ve Programları

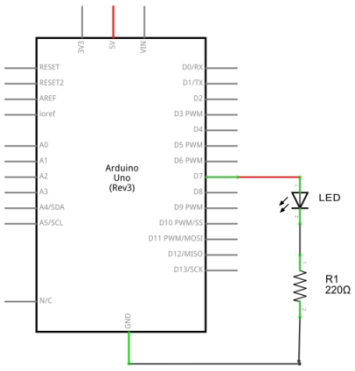
Devre 1:

Devre Adı: Led yakma söndürme .

Devre Açıklaması: Arduino'nun 13. pinine bir LED bağlanmıştır. LED 1 saniye aralıklarla sürekli olarak yanıp söner.



1-) Breadboard görünümü



2-) Şematik görünümü

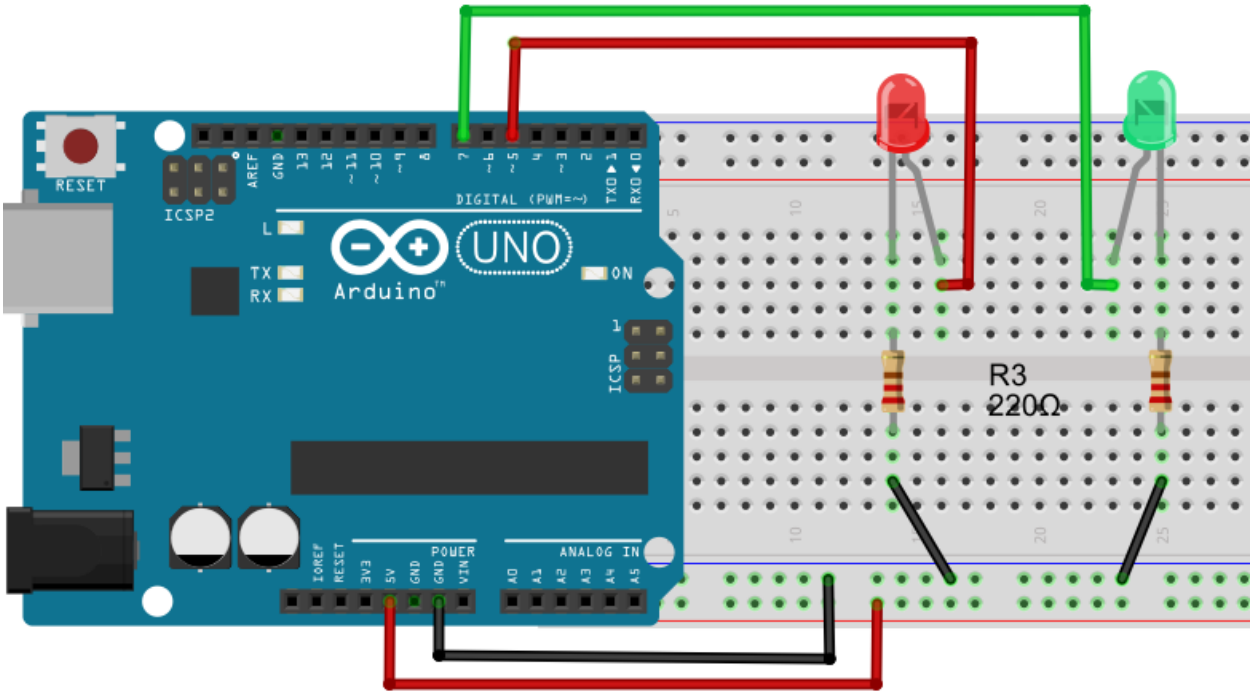
/* Yanıp sönen LED Programı, LED'i açma ve kapatma */

```
int led = 7;                  // led isimli tamsayı değişken tanımlandı, değer verildi
void setup() {                // setup() fonksiyonu sadece bir defa yürütülür
pinMode(led, OUTPUT);         // led PIN'i dijital çıkış olarak ayarlandı
}

void loop() {                 // loop() methodu tekrarlanır
digitalWrite(led, HIGH);      // led'i yakar
delay(1000);                  // programı 1000 milisaniye için durdur
digitalWrite(led, LOW);       // led'i söndürür
delay(1000);                  // programı 1000 milisaniye için durdur
}
```

Devre 2:

Devre Adı: 2 LED'i sırayla yakıp söndürme.

Devre Açıklaması: 2 LED 2 saniye aralıklarla sırayla yanıp söner.

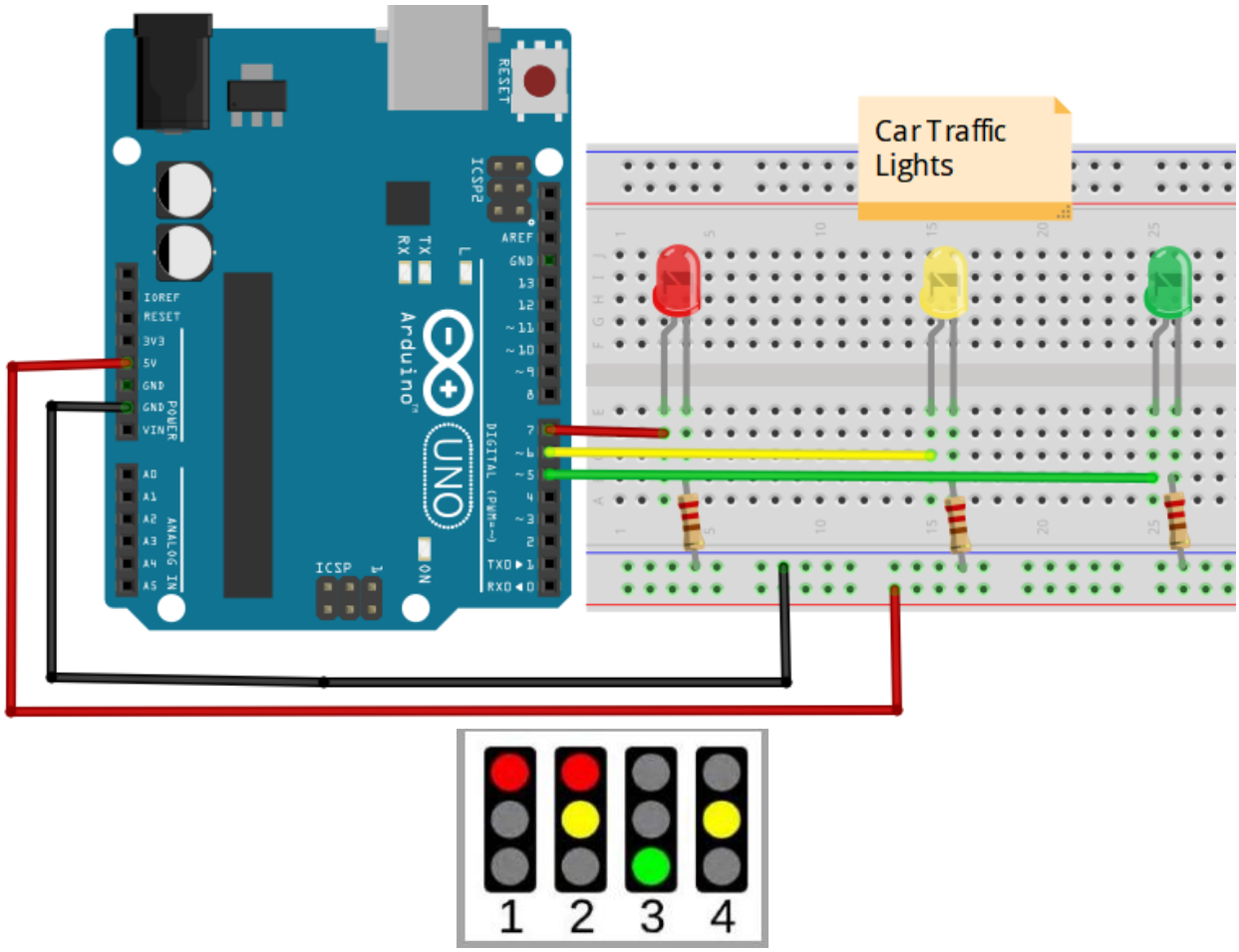



```
/*    Flip Flop     */

int greenLED=5;                    // greenLED isimli değişken tanımlandı, değer verildi
int redLED=7;                      // redLED isimli değişken tanımlandı, değer verildi

void setup()  {
pinMode(greenLED, OUTPUT);         // green LED pin dijital çıkış olarak ayarlandı
pinMode(redLED, OUTPUT);           // red LED pin dijital çıkış olarak ayarlandı
                }

void loop(){
digitalWrite(greenLED,  HIGH);     // green LED'i yak
digitalWrite(redLED,  LOW);        // red LED'i söndür
delay(2000);                       // programı 1000 milisaniye için durdur

digitalWrite(greenLED,  LOW);      // green LED'i söndür
digitalWrite(redLED,  HIGH);       // red LED'i yak
delay(2000);
                }
```

Devre 3:

Devre Adı: Trafik Lambası.

Devre Açıklaması: Bu projede, bir trafik ışığı sistemi kuracağız. Farklı renklerde (yeşil, sarı ve kırmızı) 3 adet led bulunmaktadır.



/* Trafik Lambası */

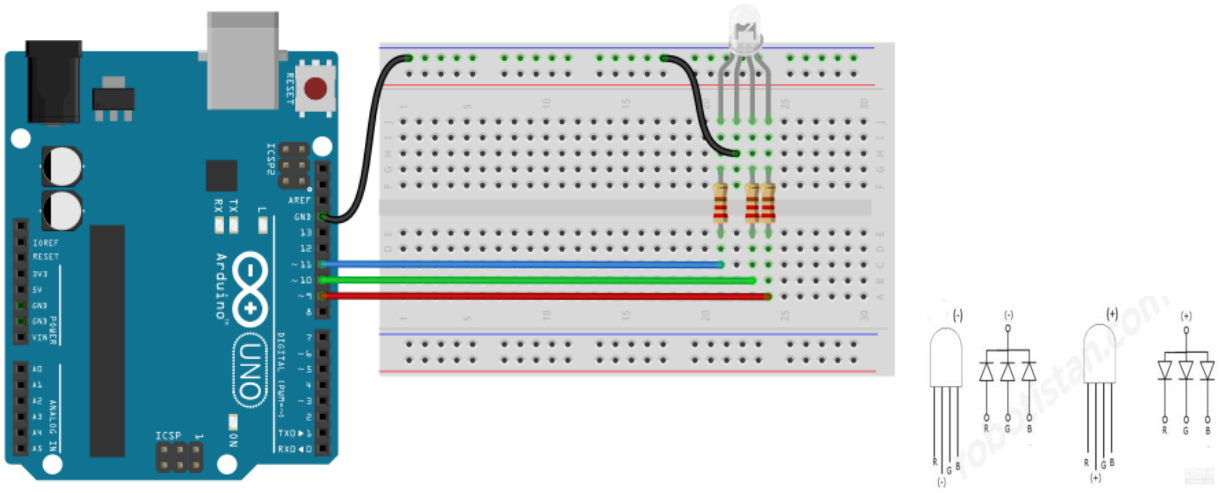
```
int redLED = 7;
int yellowLED =6;
int greenLED = 5;


void setup() {                          // burada pinlerimizi çıkış olarak ayarlıyoruz

pinMode(redLED, OUTPUT);
pinMode(yellowLED, OUTPUT);
pinMode(greenLED, OUTPUT);
}
```

```
void loop() {

digitalWrite(redLED, HIGH);          // redLED'i 9 saniye yak
digitalWrite(yellowLED, LOW);
digitalWrite(greenLED, LOW);
delay(9000);

digitalWrite(redLED, HIGH);          // redLED ve yellowLED'i 2 saniye yak
digitalWrite(yellowLED, HIGH);
digitalWrite(greenLED, LOW);
delay(2000);

digitalWrite(redLED, LOW);           // greenLED'i 9 saniye yak
digitalWrite(yellowLED, LOW);
digitalWrite(greenLED, HIGH);
delay(9000);

digitalWrite(redLED, LOW);           // tekrar, yellowLED'i 2 saniye yak
digitalWrite(yellowLED, HIGH);
digitalWrite(greenLED, LOW);
delay(2000);


/* Döngü yeniden başlıyor */

 }
```

Devre 4:

Devre Adı: RGB LED Uygulaması.

Devre Açıklaması: RGB LED, tek bir kasaya yerleştirilmiş ortak olarak bağlı 3 LED'dir. Üç rengin ışık yoğunluğunu dijital olarak kontrol etmek mümkündür. Ayrıca PWM tekniği kullanılarak istenilen renkler elde edilebilir.



/* RGB LED'in her rengini 1 saniye aralıklarla yakıp söndüreceğiz. Beyaz ışık göstermek istiyorsak tüm ledleri açmamız gerekiyor. */

```
const int BlueLed=11;          // BlueLed'e 11 değerini sabit olarak ata

 const int GreenLed=10;        // GreenLed'e 10 değerini sabit olarak ata

const int RedLed=9;            // RedLed'e 9 değerini sabit olarak ata

        // Ledlerin bağlı olduğu pinleri çıkış olarak atadık.

void setup()  {
pinMode(BlueLed,OUTPUT);
pinMode(GreenLed,OUTPUT);
pinMode(RedLed,OUTPUT);      }

        //  Döngü burada başlıyor

void loop() {
digitalWrite(BlueLed, LOW);         // RedLed açık.
digitalWrite(GreenLed, LOW);
digitalWrite(RedLed, HIGH);
delay(1000);

 digitalWrite(BlueLed,  LOW );     // GreenLed açık.
digitalWrite(GreenLed, HIGH);
digitalWrite(RedLed, LOW );
delay(1000);

 digitalWrite(BlueLed,  HIGH);          // BlueLed açık.
digitalWrite(GreenLed,  LOW);
digitalWrite(RedLed, LOW);
delay(1000);
```

// Tüm ledleri aktif hale getirerek beyaz rengi gösteriyoruz.

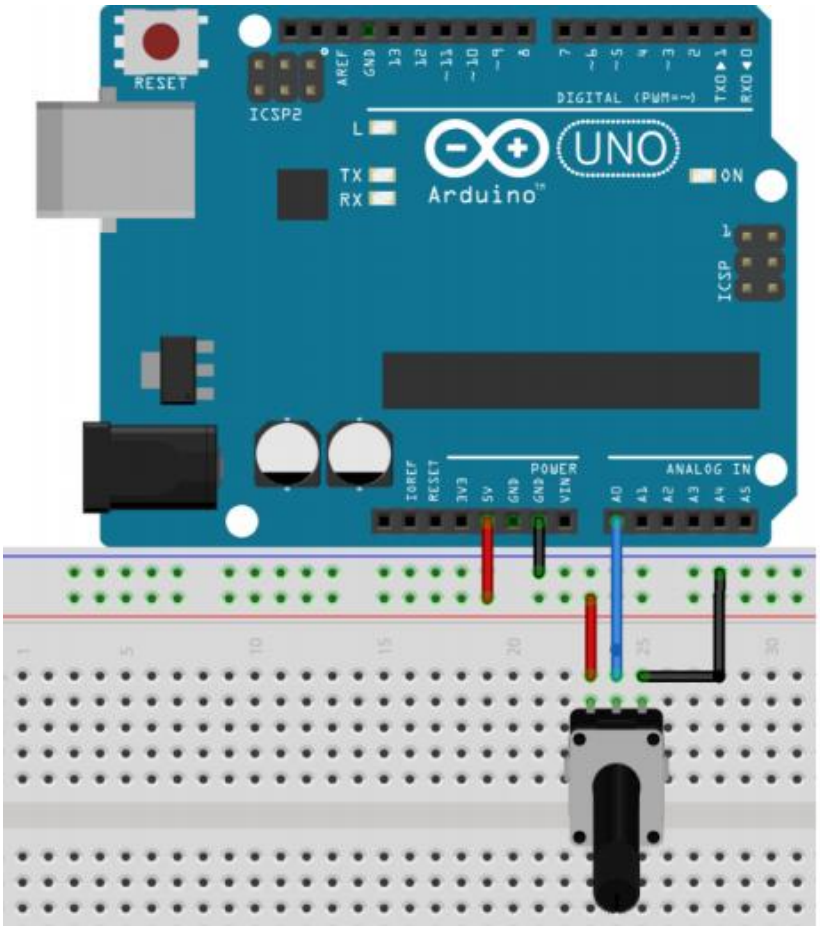
```
digitalWrite(BlueLed,  HIGH);
digitalWrite(GreenLed,  HIGH);
digitalWrite(RedLed,  HIGH);
delay(1000);
}
```

Devre 5:

Devre Adı: Analog Voltaj Okuma, Potansiyometreden Değer Okuma Uygulaması.

Devre Açıklaması: Burada, analog pin-0'da bir analog girişin nasıl okunacağını öğreniyoruz. Giriş, analogRead()'den voltaja dönüştürülür ve Arduino IDE'nin seri monitörüne yazdırılır.

Potansiyometre: Bir potansiyometre (veya pot), basit bir elektro-mekanik dönüştürücüdür. Giriş operatöründen gelen döner veya doğrusal hareketi bir direnç değişikliğine dönüştürür. Bu değişiklik, bir hi-fi sisteminin hacminden, büyük bir konteyner gemisinin yönüne kadar her şeyi kontrol etmek için kullanılabilir (veya kullanılabilir).	



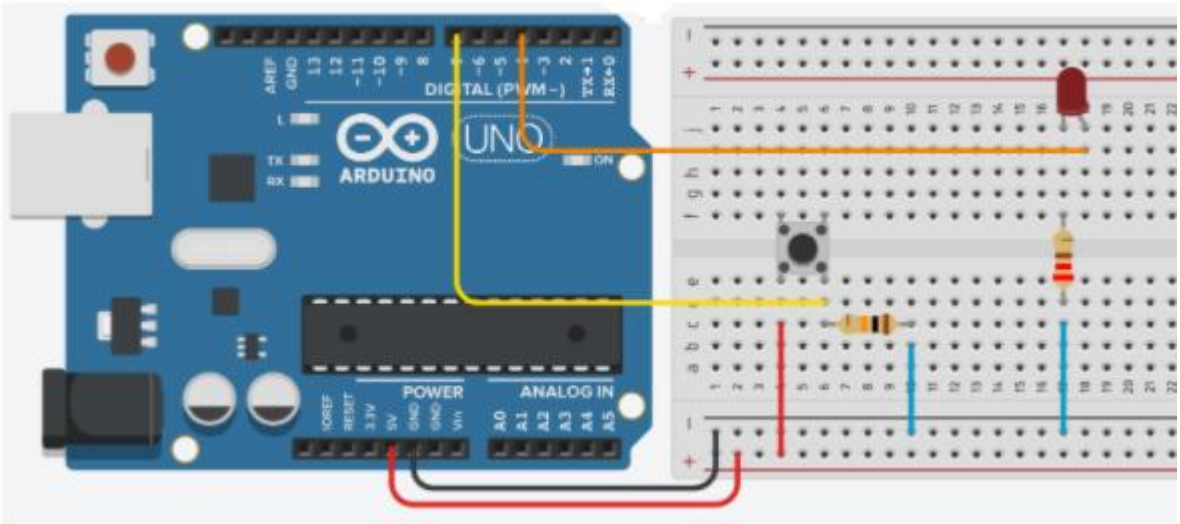
/* ReadAnalogVoltage: Pin 0'daki bir analog girişi okur, onu voltaja dönüştürür ve sonucu seri monitöre yazdırır. Seri çizici kullanılarak grafik gösterimi mevcuttur (Araçlar→Seri Çizici menüsü). Bir potansiyometrenin merkez pinini A0 pinine ve dış pinleri +5V'a ve toprağa takın. */

```
// Reset'e bastığınızda kurulum rutini bir kez çalışır:

void setup()        {

  Serial.begin(9600);          // saniyede 9600 bitte seri iletişimi başlat

}

        void loop() {          // döngü rutini sonsuza kadar tekrar tekrar çalışır:

          int sensorValue = analogRead(A0);       //analog pin 0'daki girişi okuyun

          Serial.println(sensorValue);        // okuduğunuz değeri Seri monitöre yazdırın

          delay(1000);       // stabilite için okumalar arasındaki gecikme

}
```

Devre 6:

Devre Adı: Ardunio ile buton kullanımı.

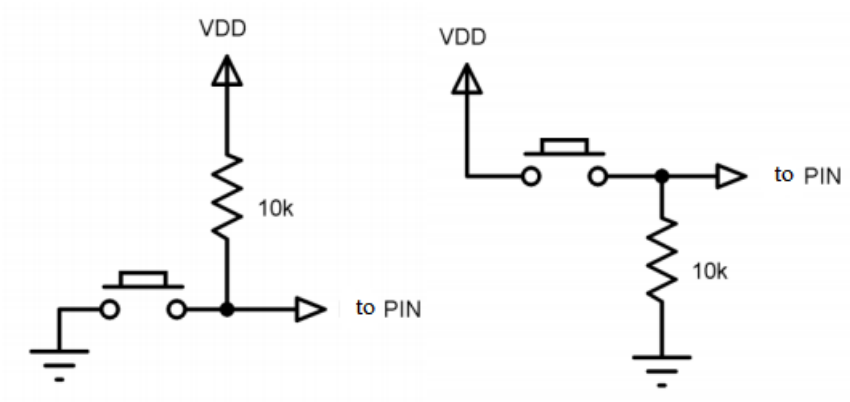
Devre Açıklaması: Pin-7'ye bağlı bir butona basıldığında, dijital pin-4'e bağlı bir (LED) yanar ve söner.



/* Pin 7'ye bağlı bir butona basıldığında, dijital pin 4'e bağlı bir ışık yayan diyotu (LED) açar ve kapatır. */

```
void setup()
{
  pinMode(4, OUTPUT);  // pin-4 çıkış yap
  pinMode(7, INPUT);     //  pin-7 giriş yap
}
void loop()
{
   if (digitalRead(7) == HIGH)              //  Eğer pin-7 açık ise,
    digitalWrite(4, HIGH);                  // Led'i yak,
    if (digitalRead(7) == LOW)              // Eğer pin-7 kapalı ise
     digitalWrite(4, LOW);                  // Led'i söndür
}
```

NOT: IF-THEN komutu, Arduino'un giriş pinlerine butonları bağlamak için de kullanılabilir. Bu bağlantı iki farklı şekilde yapılabilir. Pull_Up bağlantısı ve Pull-Down bağlantısı.

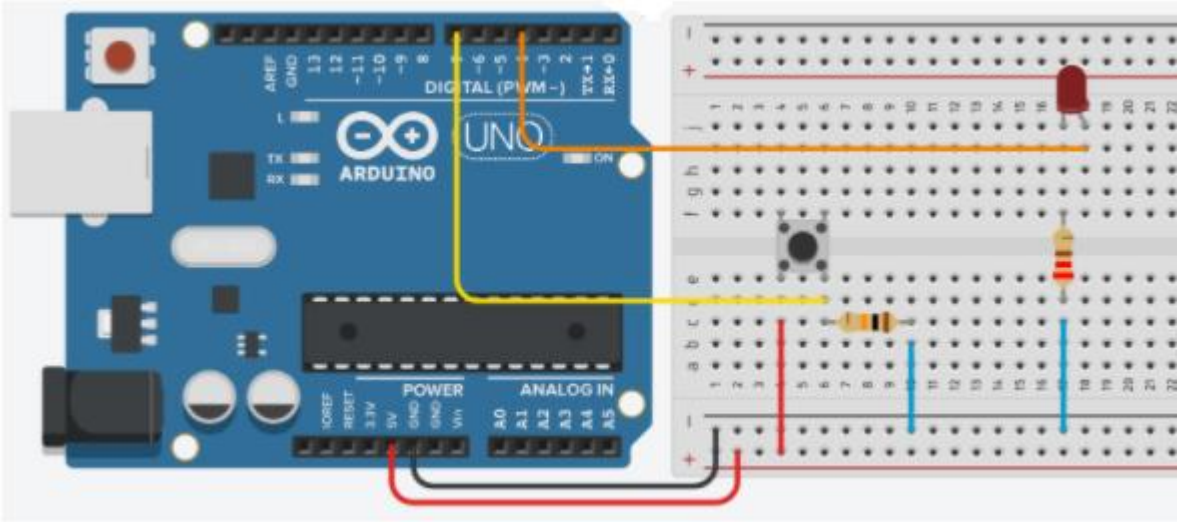


Şekil: IF…THEN komutu için kullanılabilen Anahtarlar veya Düğmeler

Devre 7:

Devre Adı: Arduino'da ELSE komutunu kullanma.

Devre Açıklaması: Pin-7'ye bağlı bir butona basıldığında dijital pin-4'e bağlı bir (LED) buton yanar ve söner. Ayrıca pinleri "int" değişkeni ile belirlemeyi öğreneceğiz.

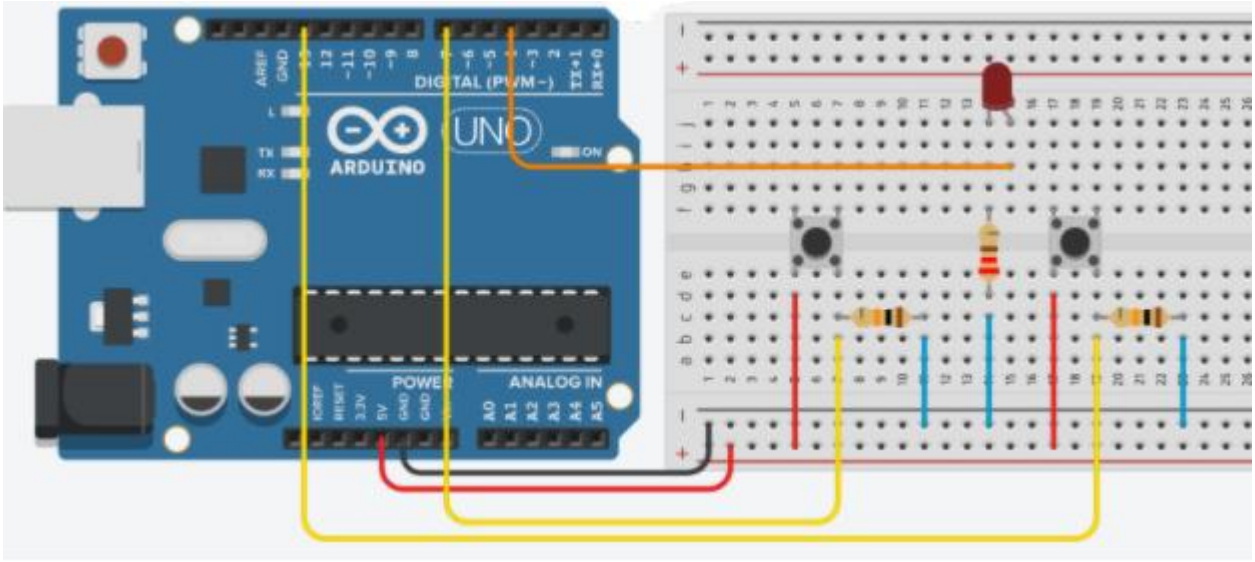



```
 int led = 4;
int buton =7;
void setup()
{
 pinMode(led, OUTPUT);
 pinMode(buton, INPUT);
}
void loop()
{
    if (digitalRead(buton) == HIGH)        // Buton açıksa
       digitalWrite(led, HIGH);            // Ledi yak
     else                                  // Değilse
      digitalWrite(led, LOW);              // Ledi söndür
}
```

Devre 8:

Devre Adı: LED'i iki butonla kontrol etme.

Devre Açıklaması: Bir buton Led'i açar, diğer buton Led'i kapatır.



/* LED'i iki butonla kontrol etme

Düğmeler seri olarak 10K dirençlerle bağlanır. ***/**

```
int led = 4;                    //  Pin-4 "led" olarak tanımlandı
int button1 = 7;                // Pin-7 "button1"  olarak tanımlandı
int button2 = 13;               // Pin-13 "button2"  olarak tanımlandı
void setup()
{
  pinMode(led, OUTPUT);         //  led=Çıkış
  pinMode(button1, INPUT);      //  button1=Giriş
  pinMode(button2, INPUT);      // button2=Giriş
}
void loop()
{
  if (digitalRead(button1) == HIGH)     // Eğer button1 açıksa
     digitalWrite(led, HIGH);           // Led 'i yak
   if (digitalRead(button2) == HIGH)    // Eğer button2 açıksa
     digitalWrite(led, LOW);            // Led' söndür
}
```

ARDUINO'DA SERİ MONİTÖR NASIL KULLANILIR

Arduino IDE, çizimlerde hata ayıklamada veya Arduino'yu bilgisayarınızın klavyesinden kontrol etmede çok yardımcı olabilecek bir özelliğe sahiptir.

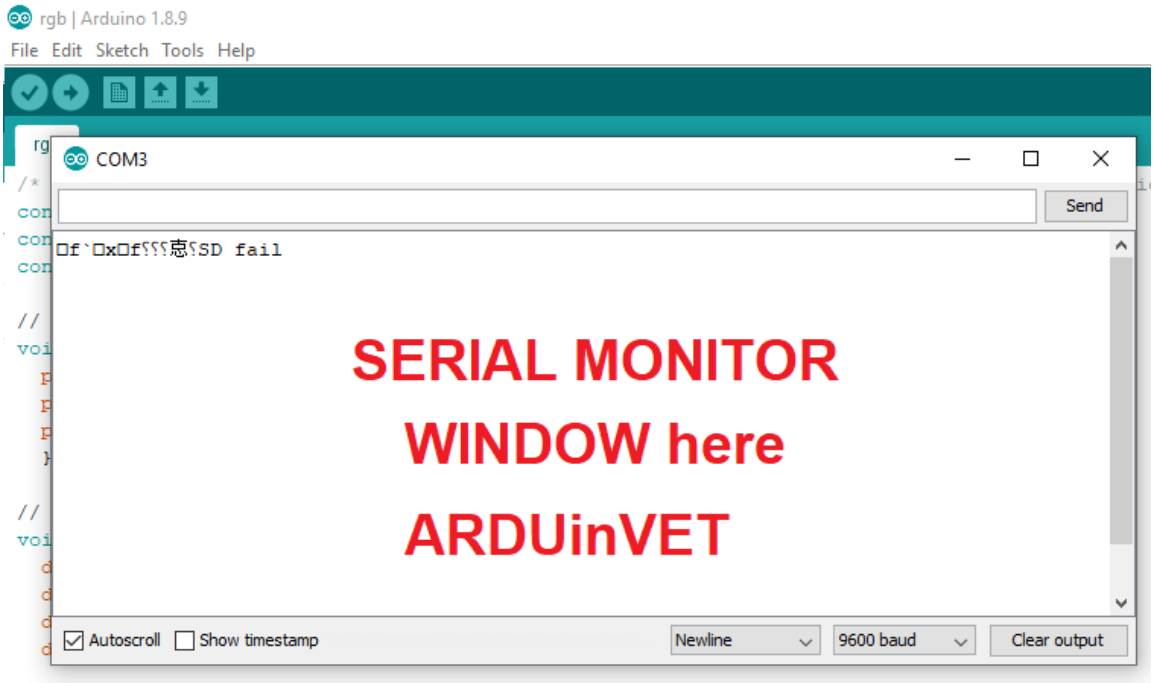
Seri Monitör, Seri verileri alıp göndererek iletişim kuran ayrı bir terminal görevi gören ayrı bir açılır penceredir. 

simgesi ile Seri Monitör gösterilir.

Seri veriler tek bir kablo üzerinden gönderilir (ancak bizim durumumuzda genellikle USB üzerinden geçer) ve kablo üzerinden gönderilen bir dizi 1 ve 0'dan oluşur. Veriler her iki yönde de gönderilebilir (Bizim durumumuzda iki kablo üzerinde).

Arduino uygulamalarında hata ayıklamak veya çalışan bir uygulama tarafından gönderilen verileri görüntülemek için Seri Monitörü kullanacaksınız. Seri Monitörü aktif hale getirebilmek için bilgisayarınıza USB ile bağlı bir Arduino'nuz olmalıdır.

IDE'deki Seri Monitör kutusuna tıklayarak Seri Monitörü açın. Aşağıdaki ekran görüntüsü gibi görünmelidir. Baud'un (hız) 9600'e ayarlandığından EMİN OLUN(sağ alt köşede bulunur). Önemli olan bizim programımızda ve burada aynı ayarlanmış olması. Buradaki default 9600 olduğu için biz kendi ayarlarımızı yapıyoruz.



Seri Monitörü Kullanmak için Ana Komutlar

Serial.begin(): Seri veri iletimi için veri hızını saniyede bit (baud) olarak ayarlar. Seri Monitör ile iletişim kurmak için ekranın sağ alt köşesindeki menüde listelenen baud hızlarından birini

kullandığınızdan emin olun. Bununla birlikte, örneğin baud hızı gerektiren bir bileşenle 0 ve 1 pinleri üzerinden iletişim kurmak için başka hızlar belirleyebilirsiniz.

Kullanımı:

Serial.begin(speed);

Örnek: Serial.begin(9600);

Serial.print(): Verileri seri bağlantı noktasına insan tarafından okunabilen ASCII metni olarak yazdırır. Bu komut birçok şekilde olabilir. Sayılar, her basamak için bir ASCII karakteri kullanılarak yazdırılır. Kayan noktalı sayılar benzer şekilde ASCII basamakları olarak yazdırılır ve varsayılan olarak iki ondalık basamaktır. Baytlar tek bir karakter olarak gönderilir. Karakterler ve dizeler olduğu gibi gönderilir. Örneğin:

Serial.print(78); "78" gösterir

Serial.print('N'); "N" gösterir

Serial.print("Hello ArduinVet."); "Hello ArduinVet." gösterir

Serial.println(): Verileri seri bağlantı noktasına insan tarafından okunabilir ASCII metni ve ardından bir satır başı karakteri (ASCII 13 veya '\r') ve yeni satır karakteri (ASCII 10 veya '\n') olarak yazdırır. Bu komut Serial.print() ile aynı formları alır.

Serial.println(val);

Serial.println(val, format);

Serial.println(13, BIN); Seri Monitörde "1101" ikili değeri 15'i verir.

NOT: Serial.print ve Serial.println arasındaki tek fark, Serial.println'nin, bundan sonra seri bağlantı noktasından gönderilen şeyin bir sonraki satırda başlayacağı anlamına gelmesidir. Fark etmiş olabileceğiniz üçüncü bir yeni şey daha var. Tırnak içinde bir şey var (“). Buna dize denir.

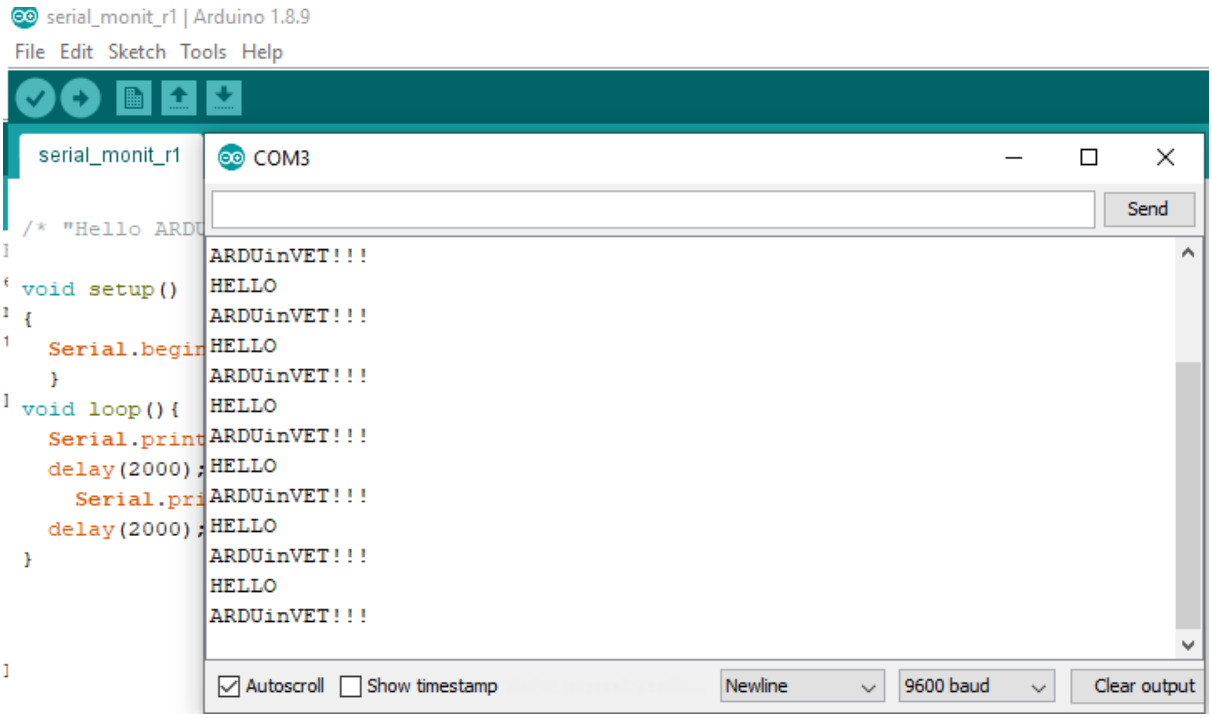
Devre 9:

Devre Adı: Seri Monitörde “Merhaba ARDUinVET” Yazdırma Uygulaması.

Devre Açıklaması: Merhaba ARDUinVET seri Monitörde görülecektir.

/* " Seri Monitörde Hello ARDUinVET" Uygulaması */

```
void setup()
{
  Serial.begin(9600);     // Seri bağlantı hızı
}
void loop()
{
  Serial.println("  HELLO ");
  delay(2000);
  Serial.println("ARDUinVET!!!");
  delay(2000);
}
```

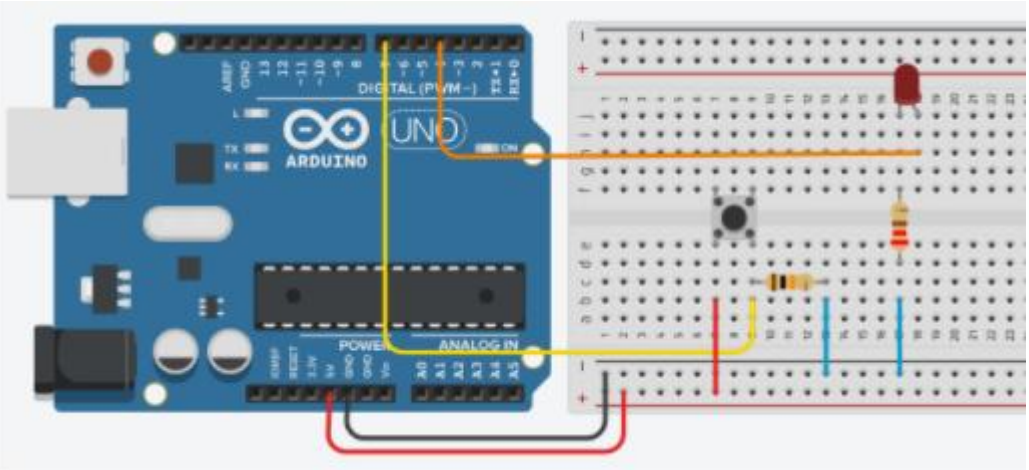


Devre 10:

Devre Adı: Seri Monitörde Düğme Durum Analizi

Devre Açıklaması: Butona basıldığında seri monitörde "Led yanıyor, LED=AÇIK" mesajı çıkıyor ve Led yanıyor.

Butona basılmazsa, seri monitörde ("Buton basılı değil") mesajı görülür ve Led yanar.




```
 /*       Seri Monitörde Buton Durum Analizi      */

void setup()                  {
  pinMode(4, OUTPUT);
  pinMode(7, INPUT);
  Serial.begin(9600);         }
void loop()                     {
   if (digitalRead(7) == HIGH)        // 7 nolu pin'de sinyal varsa
    {
    digitalWrite(4, HIGH);
    Serial.println("Led is ligthing, LED=ON");
    delay(250);
    }
   else                         // Önceki şart doğru değilse
    {
    digitalWrite(4, LOW);
    Serial.println("Button is not pressed");
    delay(1000);
    }

}                               // Seri monitörün görünümü aşağıda görülmektedir.
```

```
Button is not pressed
Button is not pressed
Led is ligthing, LED=ON
Led is ligthing, LED=ON
Led is ligthing, LED=ON
Led is ligthing, LED=ON
Led is ligthing, LED=ON
Button is not pressed
```

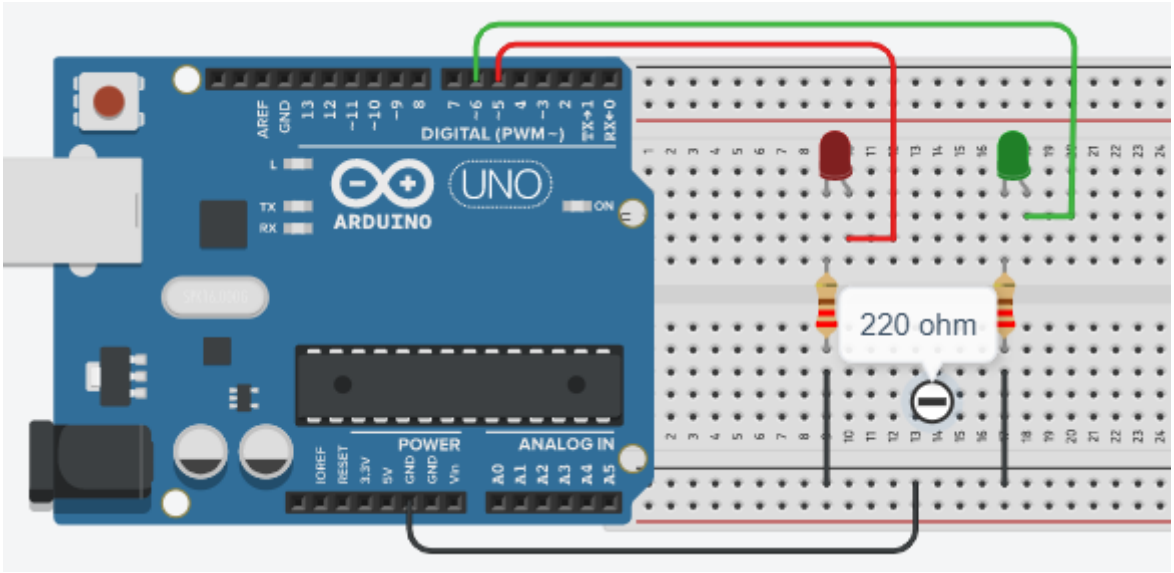
Devre 11:

Devre Adı: Seri ekrandan veri gönderme.

Devre Açıklaması: Klavyede "A" karakterine basılırsa led1 yanar.

Klavyede "3" karakterine basılırsa led2 yanar.

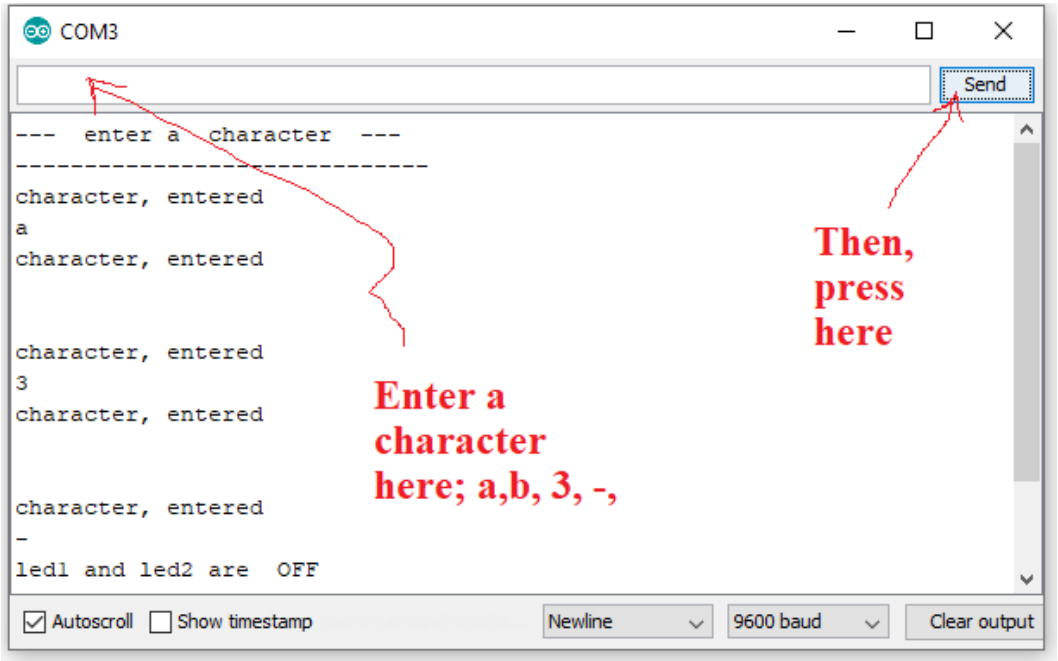
Klavyede "-" karakterine basılırsa led1 ve led2 OFF olur.



/* Seri monitörden veri gönderme */

char character;
#define led1 5
#define led2 6

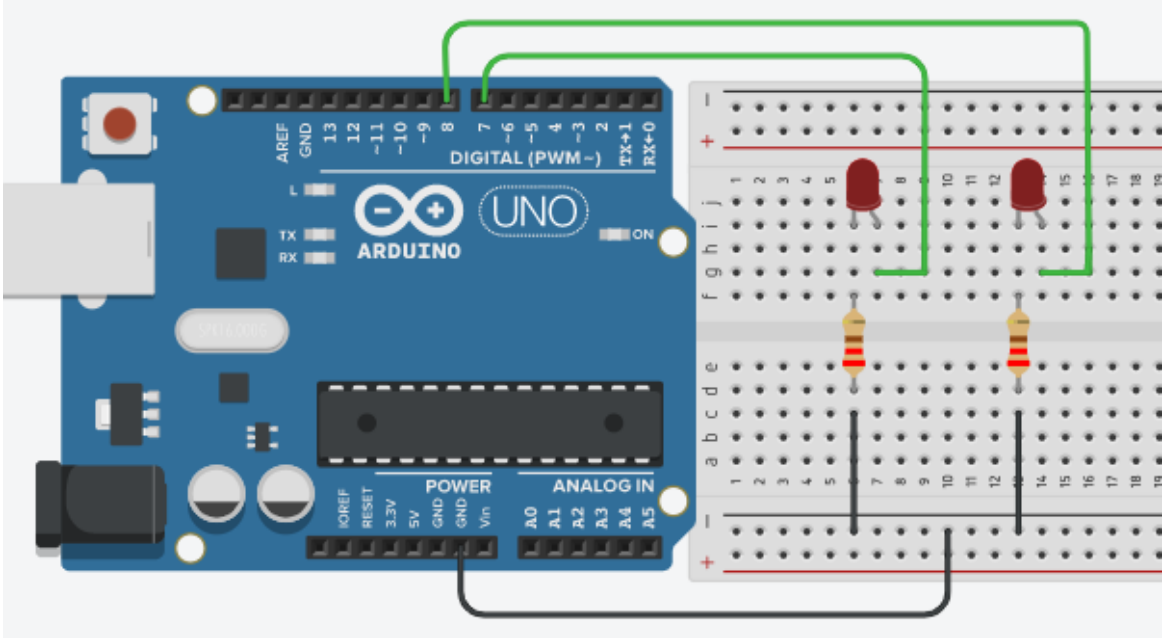
```
 void setup()  {
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
Serial.begin(9600);
Serial.println ("---  enter a  character  --- ");
Serial.println ("-----------------------------");
}

void loop()
{
if (Serial.available()>0)
{
character=Serial.read();
Serial.println ("character, entered ");
Serial.println (character);

 if (character=='A') digitalWrite (led1, 1);

if (character=='a') digitalWrite (led1, 1);

if (character=='3') digitalWrite (led2, 1);

if (character=='-')
{
digitalWrite(led1,0);
digitalWrite(led2,0);
Serial.println ("led1 and led2 are  OFF ");
}
}
}
```



Devre 12:

Devre Adı: FOR komutunu öğrenme uygulaması.

Devre Açıklaması: Bu uygulamada LED'ler 6 kez yanıp sönecektir.




```
/*     FOR komutunu öğrenme   */

int   led1 = 7;
int   led2 = 8;

void setup() {
Serial.begin(9600);
pinMode(7,OUTPUT);
pinMode(8,OUTPUT);
}

void loop()
{
for(int i=1; i<6; i++)
{
Serial.println(i);
digitalWrite(led1, 1);
digitalWrite(led2, 1);
delay(1000);

digitalWrite(led1, 0);
digitalWrite(led2, 0);
delay(1000);
```

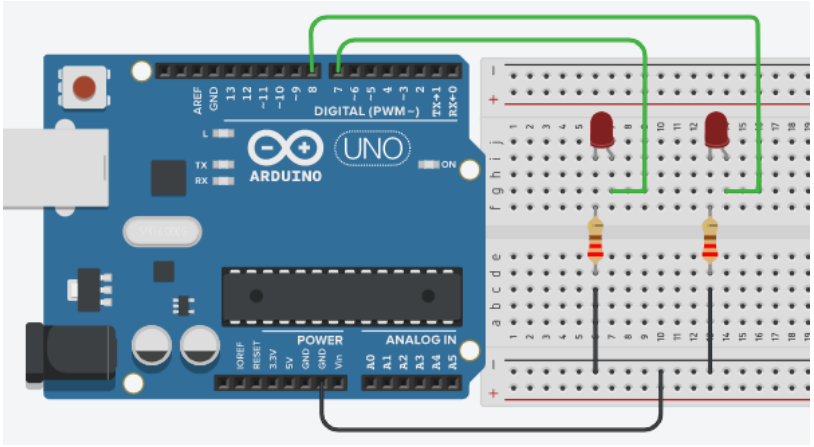
}
}

Devre 13:

Devre Adı: FOR komutu ve seri monitör uygulaması.

Devre Açıklaması: Bu uygulamada LED'ler 6 kez yanıp sönecektir. Seri monitörde 1, 2, 3, 4, 5,6 sayıları görünecektir. Daha sonra for döngüsünden çıkılarak while döngüsüne girilecek ve program duracaktır. Yeniden başlatmak için Reset düğmesine basılmalıdır.

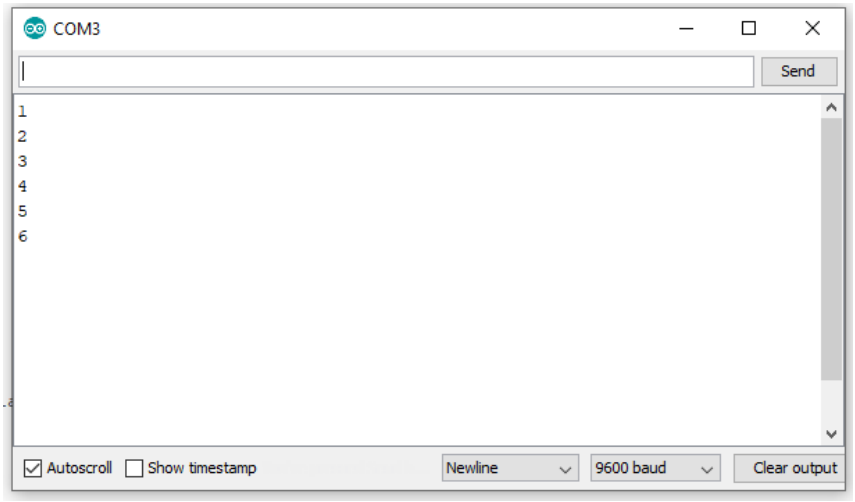
Burada önceki devre ile aynı devreyi kullanıyoruz.




```
/*     FOR komutu ve seri monitor uygulaması   */

int   led1 = 7;
int   led2 = 8;

void setup()                    {
Serial.begin(9600);
pinMode(7,OUTPUT);
pinMode(8,OUTPUT);       }

void loop() {

  for(int i=1; i<=6; i++)                // i = 1, 2,3, 4,5, 6

  {

 Serial.println(i);                      // Seri monitöre i'nin değerlerini yaz

digitalWrite(led1, 1);
digitalWrite(led2, 1);
```

```
delay(1000);
digitalWrite(led1, 0);
digitalWrite(led2, 0);
delay(1000);
}
while(1);                    // for döngüsü sonsuz bir döngüye girmez.

}
```

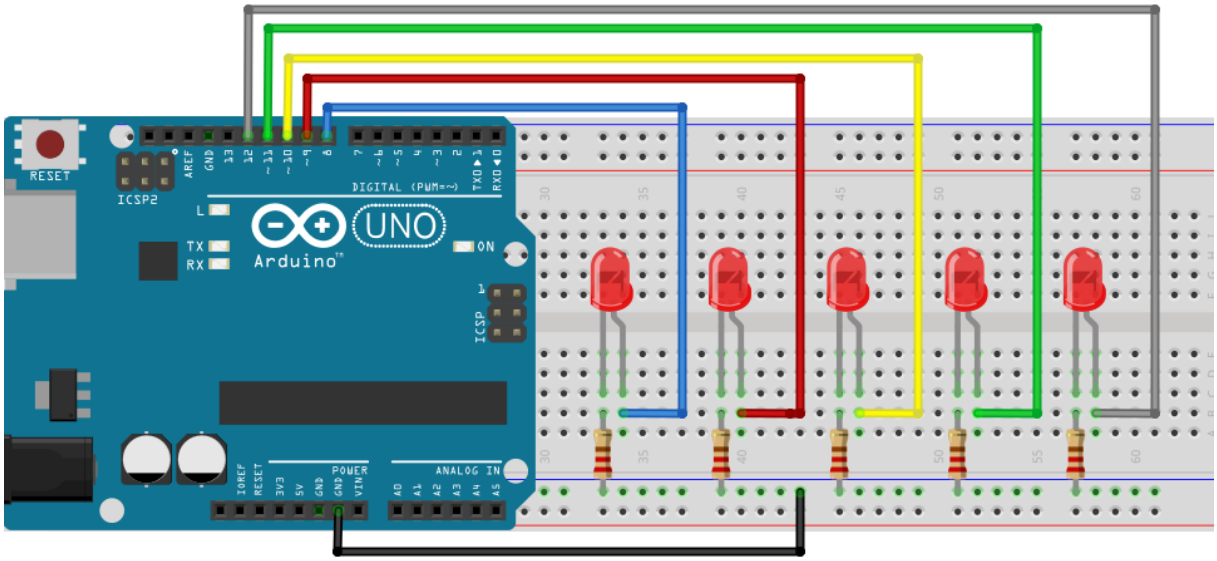
NOT: Programı yeniden başlatmak içinReset düğmesine basılmalıdır.



Devre 14:

Devre Adı: FOR komutu ile karaşimşek uygulaması.

Devre Açıklaması: Devreye bağlı 5 LED sırayla bir yöne doğru yanma/sönme işlemi yapacak, son LED'e geldiğinde ise bu işlemi ters yönde gerçekleştirecektir.




```
/*     FOR komutu kullanarak karaşimşek uygulaması yapmak  */

void setup() {
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
pinMode(12, OUTPUT);     }

void loop() {

 for (int b=8; b<=12; b++)                    // Döngü başlangıcı

  {

 digitalWrite(b, HIGH);
delay(150);
digitalWrite (b, LOW);
delay(150);

  }

  for (int b=12; b>=8; b--)                    // Döngü başlangıcı

  {

 digitalWrite (b, HIGH);
delay(150);
digitalWrite (b, LOW);
delay(150);
```

```
  }
}
```

Devre 15:

Devre Adı: Switch/case komutunu kullanarak RGB led ile rastgele renk üretimi

Devre Açıklaması: Bu uygulamada Random komutu ve Switch/Case komutu kullanılmaktadır. Bu devrede kırmızı, yeşil, mavi renkler rastgele elde edilecektir.

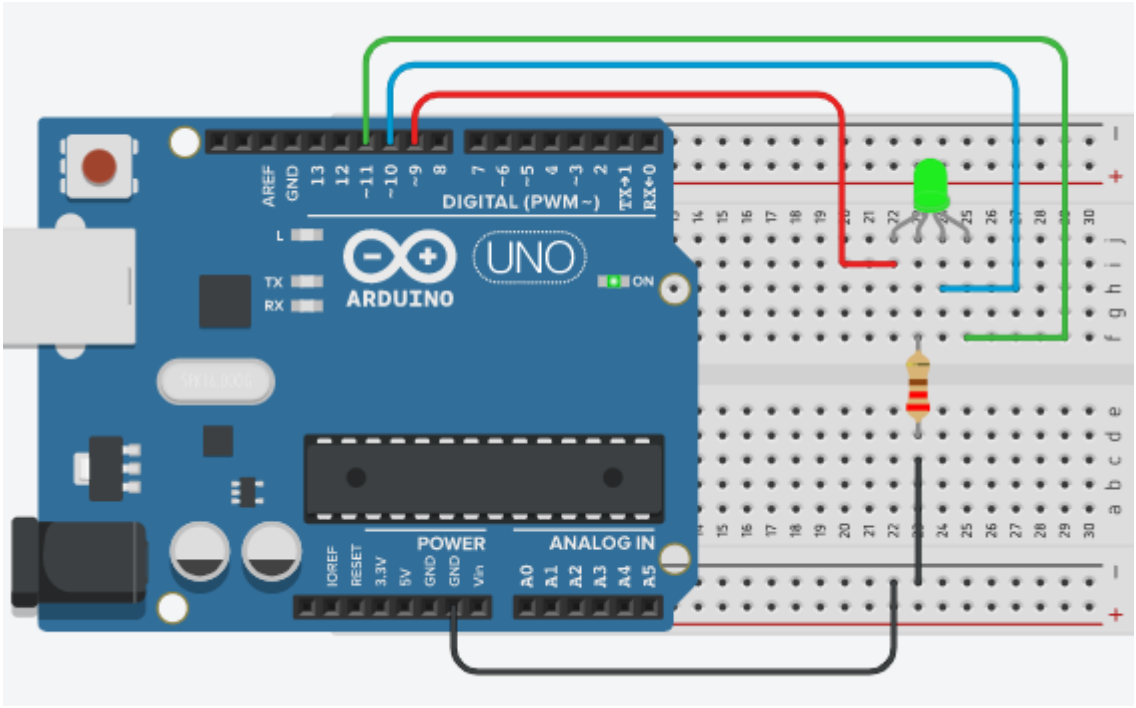
NOT:

random() : Random işlevi rasgele sayılar üretir.

Kullanımı: random(max), random(min, max)

min: rastgele değerin alt sınırı, (isteğe bağlı).

max: rastgele değerin üst sınırı.




```
/*    Switch/case komutunu kullanarak RGB led ile rastgele renk üretimi   */

#define R  9
#define G 10
```

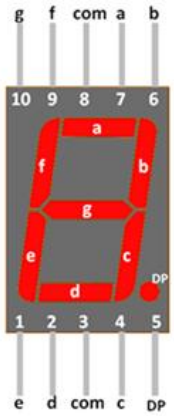
```
#define B 11
int colour;
int dly=3000;

void setup() {
pinMode(R, OUTPUT);
pinMode(G, OUTPUT);
pinMode(B, OUTPUT);
Serial.begin(9600);
}

void loop()
{
colour=random(4);
Serial.println(colour);

switch(colour)  {

 case 0:
digitalWrite (R, 1);              //RedLED=ON
digitalWrite (G, 0);
digitalWrite (B, 0);
delay(dly);
break;

 case 1:
digitalWrite (R, 0);            //GreenLED=ON
digitalWrite (G, 1);
digitalWrite (B, 0);
delay(dly);
break;

 case 2:                          //BlueLED=ON
digitalWrite (R, 0);
digitalWrite (G, 0);
digitalWrite (B, 1);
delay(dly);
break;

 case 3:                          //Renk yok
digitalWrite (R, 0);
digitalWrite (G, 0);
digitalWrite (B, 0);
delay(dly);
break;
}
}
```

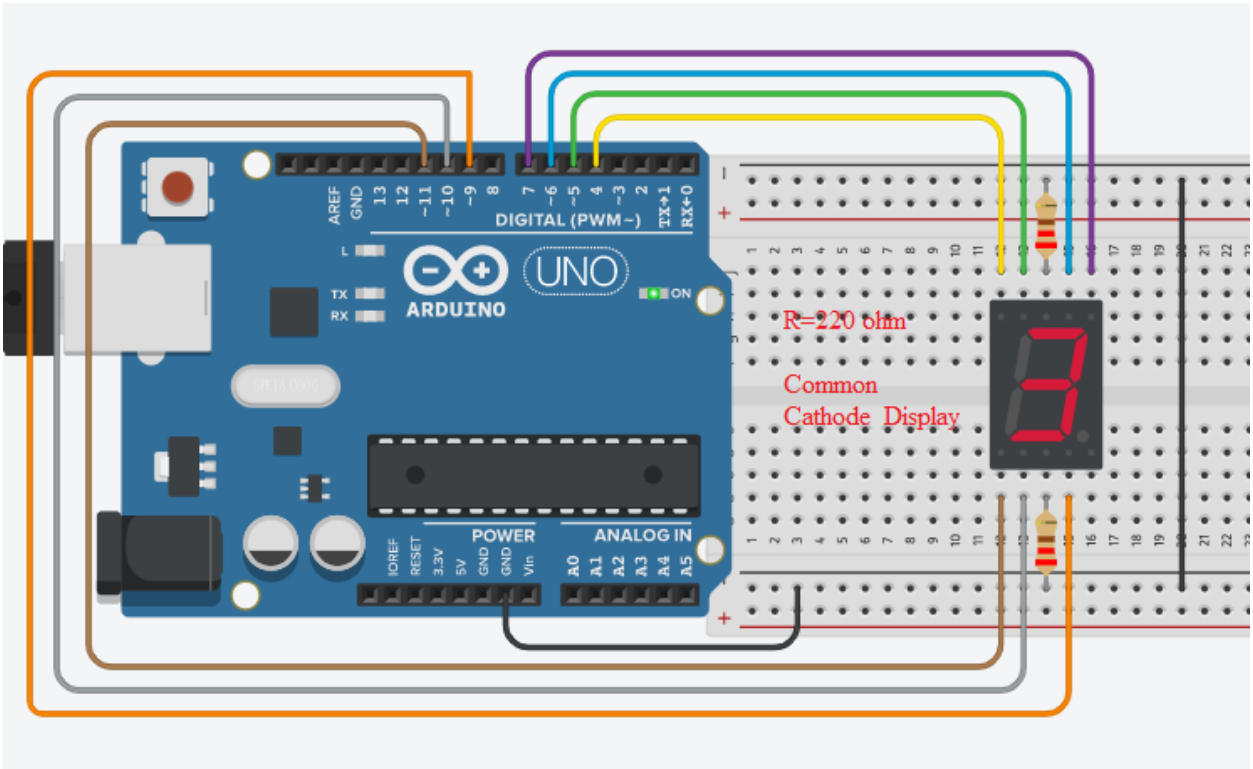
Devre 16:

Devre Adı: Segment Display 0-9 Sayıcı Devresi (Counter)

Devre Açıklaması: Ekranda bir sayı görmek için a, b, c, d, e, f, g harfleriyle temsil edilen 7 LED'den ilgili LED yanar.

NOT: Segment display(Yedi bölmeli ekran), ondalık sayıları görüntülemek için bir elektronik görüntüleme cihazıdır.





/* Segment Display 0-9 Sayıcı Devresi (Counter) */

```
int       a=6,  b=7,  c=9,   d=10,    e=11,   f=5,    g=4;
int number;

void setup() {
pinMode(a,  OUTPUT);
pinMode(b,  OUTPUT);
pinMode(c,  OUTPUT);
pinMode(d,  OUTPUT);
pinMode(e,  OUTPUT);
pinMode(f,  OUTPUT);
pinMode(g,  OUTPUT);
}

void loop()  {
for(number=0;  number< =9;  number++)    {
delay(1000);
switch(number)  {

 case 0:
digitalWrite (a, HIGH);
digitalWrite (b, HIGH);
digitalWrite (c, HIGH);
digitalWrite (d, HIGH);
digitalWrite (e, HIGH);
digitalWrite (f, HIGH);
digitalWrite (g, LOW);
break;

case 1:
digitalWrite (a, LOW);
digitalWrite (b, HIGH);
digitalWrite (c, HIGH);
digitalWrite (d, LOW);
digitalWrite (e, LOW);
digitalWrite (f, LOW);
digitalWrite (g, LOW);
break;

case 2:
digitalWrite (a, HIGH);
digitalWrite (b, HIGH);
digitalWrite (c, LOW);
digitalWrite (d, HIGH);
digitalWrite (e, HIGH);
digitalWrite (f, LOW);
digitalWrite (g, HIGH);
break;

case 3:
digitalWrite (a, HIGH);
digitalWrite (b, HIGH);
digitalWrite (c, HIGH);
digitalWrite (d, HIGH);
digitalWrite (e, LOW);
digitalWrite (f, LOW);
digitalWrite (g, HIGH);
break;

case 4:
digitalWrite (a,LOW );
digitalWrite (b, HIGH);
digitalWrite (c, HIGH);
digitalWrite (d, LOW);
digitalWrite (e, LOW);
digitalWrite (f, HIGH);
digitalWrite (g, HIGH);
break;

case 5:
digitalWrite (a, HIGH);
digitalWrite (b, LOW);
digitalWrite (c, HIGH);
digitalWrite (d, HIGH);
digitalWrite (e, LOW);
digitalWrite (f, HIGH);
digitalWrite (g, HIGH);
break;
```

```
case 6:
digitalWrite (a, HIGH);
digitalWrite (b, LOW);
digitalWrite (c, HIGH);
digitalWrite (d, HIGH);
digitalWrite (e, HIGH);
digitalWrite (f, HIGH);
digitalWrite (g, HIGH);
break;

case 7:
digitalWrite (a, HIGH);
digitalWrite (b, HIGH);
digitalWrite (c, HIGH);
digitalWrite (d, LOW);
digitalWrite (e, LOW);
digitalWrite (f, LOW);
digitalWrite (g, LOW);
break;

case 8:
digitalWrite (a, HIGH);
digitalWrite (b, HIGH);
digitalWrite (c, HIGH);
digitalWrite (d, HIGH);
digitalWrite (e, HIGH);
digitalWrite (f, HIGH);
digitalWrite (g, HIGH);
break;

case 9:
digitalWrite (a, HIGH);
digitalWrite (b, HIGH);
digitalWrite (c, HIGH);
digitalWrite (d, HIGH);
digitalWrite (e, LOW);
digitalWrite (f, HIGH);
digitalWrite (g, HIGH);
break;
}
}
}
```

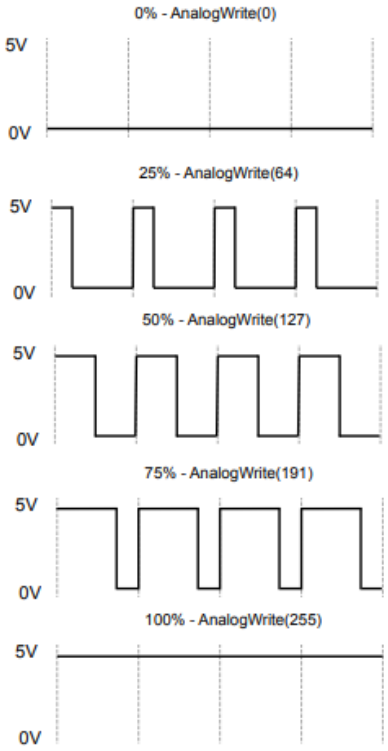
Devre 17:

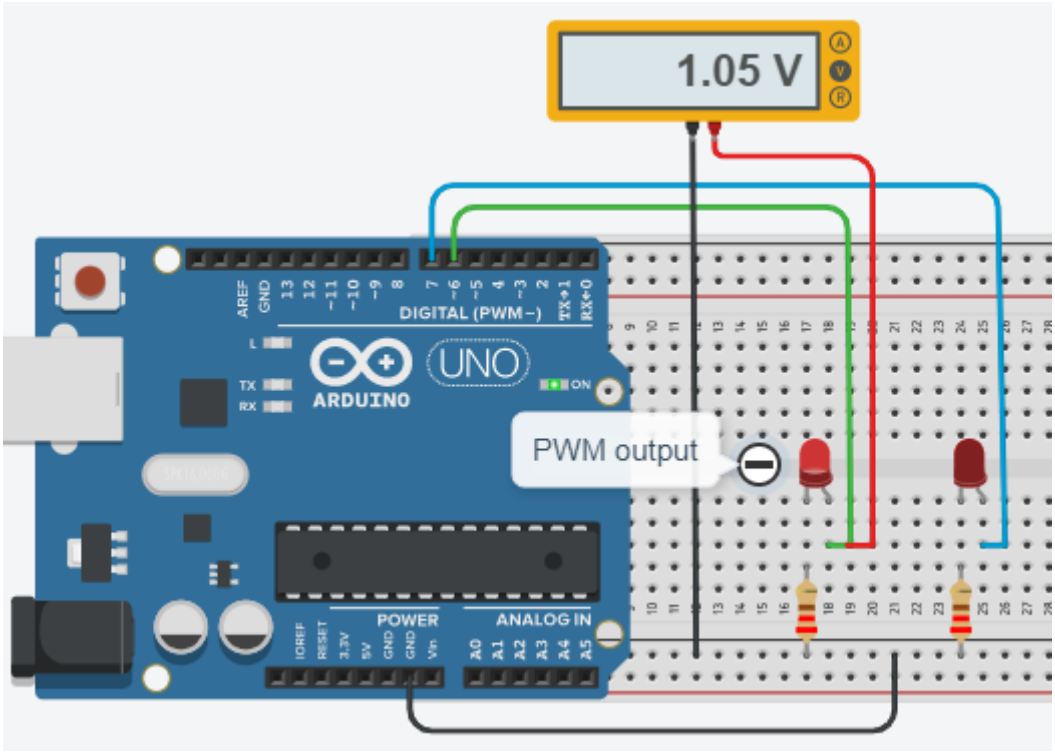
Devre Adı: PWM Tekniğinin Kullanılması

Devre Açıklaması: Pratikte PWM çıkışı olarak pin-6 ve dijital çıkış olarak pin-7 kullanılır. Böylece aralarındaki farkın gözlemlenmesi amaçlanmaktadır. PWM yöntemi ile led parlaklık ayarı, motor hız kontrolü gibi uygulamalar yapılabilmektedir.

NOT: Arduino, 500Hz'de PWM'yi (Arduino kartınızda tilde (~) ile işaretlenmiş belirli pinlerde - pin 3, 4,5,9,10 ve 11) destekler. (Saniyede 500 kez.) 0 ile 255 arasında bir değer verebilirsiniz. 0, asla 5V olmadığı anlamına gelir. 255 her zaman 5V olduğu anlamına gelir. Bunu yapmak için değerle analogWrite()'a bir çağrı yaparsınız. "AÇIK" zamanın toplam süreye oranına "görev döngüsü" denir. Zamanın yarısında AÇIK olan bir PWM çıkışının %50'lik bir görev döngüsüne sahip olduğu söylenir.

PWM'yi x/255 için açık olarak düşünebilirsiniz; burada x, analogWrite() ile gönderdiğiniz değerdir. Aşağıda, darbelerin nasıl göründüğünü gösteren bir örnek verilmiştir:





/* PWM Tekniğini analogWrite() kullanarak öğrenme */

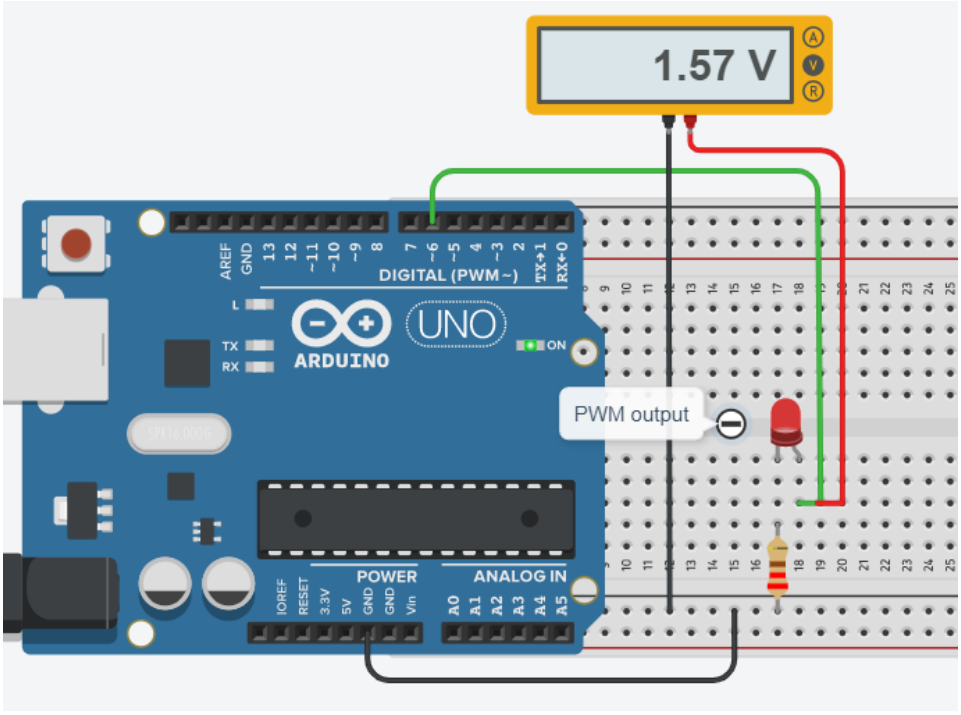
```
#define   led1  6
#define   led2  7

void setup()  {
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);      }

void loop()               {

 analogWrite(led1, 60);       //  60 değeri Led1 pinine gönderilir (1.05 V)


 analogWrite(led2,60);        //  60 değeri Led2 pinine gönderilir (1.05 V)


delay (100);                  // sadece Led1 yanar

}
```

Devre 18:

Devre Adı: Bir LED'in parlaklığını minimumdan maksimuma değiştirmek.

Devre Açıklaması: PWM tekniği kullanılarak bir LED'in parlaklığı minimumdan maksimuma değiştirilir. Burada analogWrite() ve for komutları birlikte kullanılacaktır.

.




```
/*        Bir LED'in parlaklığını minimumdan maksimuma değiştirmek          */

#define led1    6
int a;

void setup()          {
Serial.begin(9600);
pinMode(led1, OUTPUT);    }

void loop() {

 for (a=0; a<=255; a++) {

 Serial.println(a);

analogWrite(led1, a);

 delay (100);
}                }
```

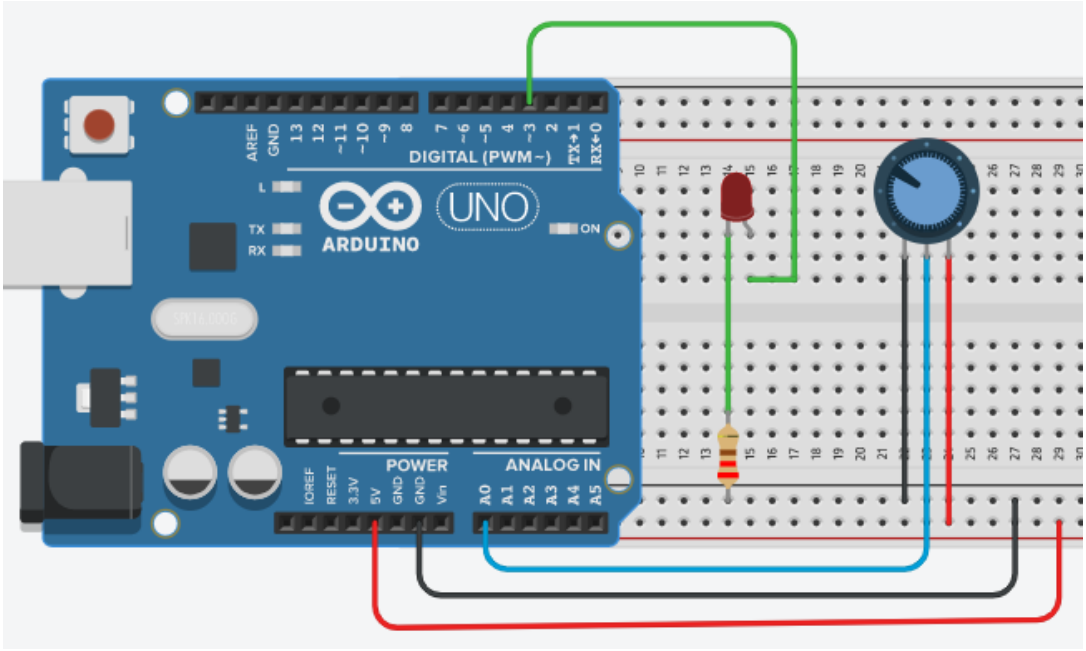
NOT: Voltmetrede 0 ile 5 Volt arasında değişen değerler görüntülenir. 0'dan 255'e kadar sayma sayıları seri monitörde görüntülenir.

.

Devre 19:

Devre Adı: Potansiyometre ile LED yakma.

Devre Açıklaması: Potansiyometre (10K) kullanılarak bir LED'in parlaklığı minimumdan maksimuma değiştirilir. Burada map() fonksiyonu kullanılır.



map() Fonksiyonu:

Bir sayıyı bir aralıktan diğerine yeniden eşler. Yani, fromLow değeri toLow ile, fromHigh değeri toHigh ile, aradaki değerler ile aradaki değerler ile eşlenir.

Aralık dışı değerler bazen amaçlandığından ve kullanışlı olduğundan, değerleri aralık dahilinde sınırlamaz. Aralıklarda sınırlamalar isteniyorsa, bu işlevden önce veya sonra Constrain() işlevi kullanılabilir.

map() işlevi tamsayı matematiği kullanır, bu nedenle matematiğin yapması gerektiğini belirttiğinde kesirler oluşturmaz. Kesirli kalanlar kesilir ve yuvarlanmaz veya ortalaması alınmaz.

Kullanımı: map(value, fromLow, fromHigh, toLow, toHigh);

Örnek: val = map(val, 0, 1023, 0, 255);

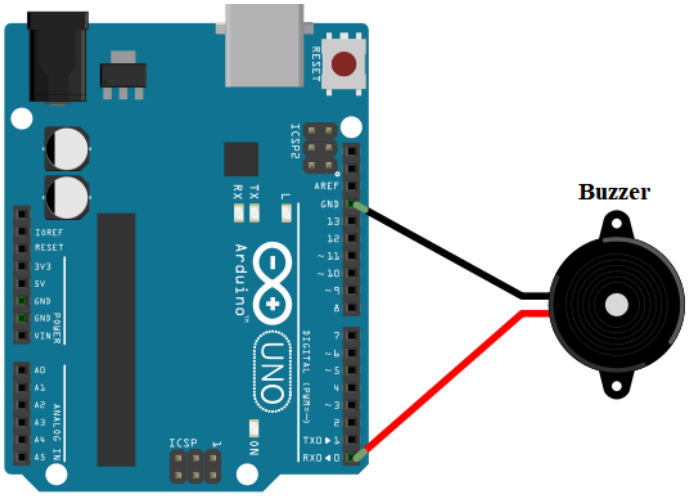
```
/*        Potansiyometre ile LED yakma        */

int LED_PIN = 3;

void setup() {
Serial.begin(9600);
pinMode(LED_PIN, OUTPUT);      }

void loop() {

        int analogValue = analogRead(A0); // A0 pini üzerindeki girişi okur (0 ile 1023
                                                arasındaki değer)

        int brightness = map(analogValue, 0, 1023, 0, 255);       // parlaklığı ölçeklendirir (0 ile
        255 arasındaki değer)

        analogWrite(LED_PIN, brightness);  // pin-3'e bağlanan LED'in parlaklığını ayarlar

         Serial.print("Analog: ");              // değeri Seri monitöre yazdır

        Serial.print(analogValue);

         Serial.print(" Brightness: ");

          Serial.println(brightness);

         delay(100);

}                Seri monitör ekranı aşağıdadır
```

Devre 20:

Devre Adı: Buzzer kullanımı.

Devre Açıklaması: Buzzer, mekanik, elektromekanik veya piezoelektrik (kısaca piezo) olabilen bir ses sinyali cihazıdır. Buzzer'ların endüstrideki tipik kullanımları, uğultu gerektiğinde uğultu veya bip sesi çıkaran bir alarm cihazıdır.

.



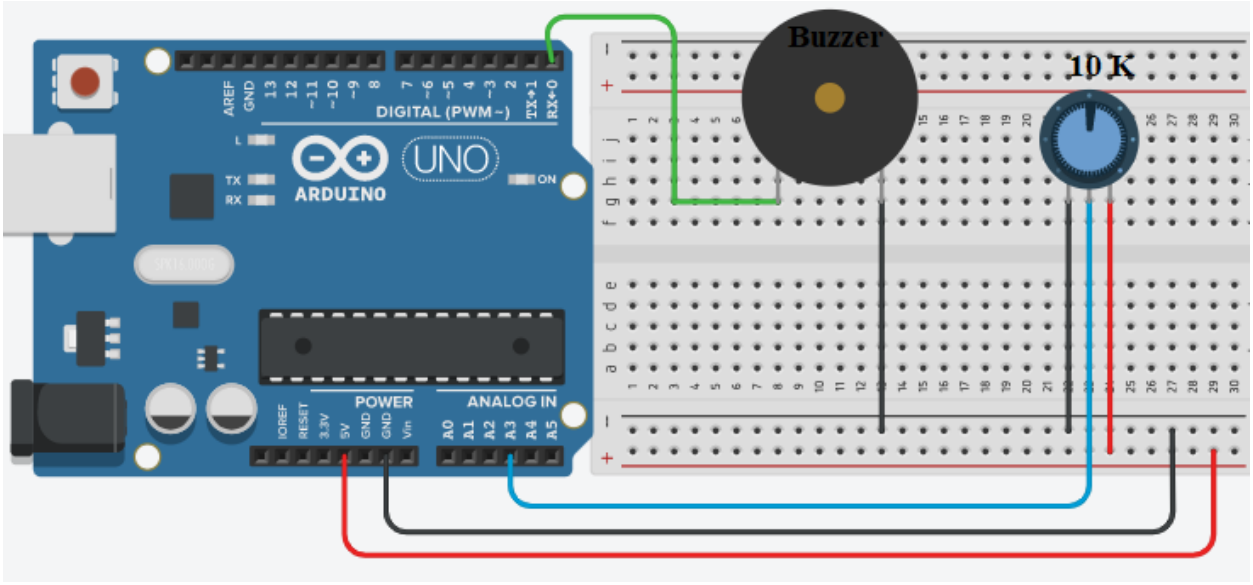
/* Arduino devrelerinde buzzer kullanımı */

```
#define buzzer_pin  0

 void setup()  {

 pinMode(buzzer_pin, OUTPUT);        }

void loop()       {

 digitalWrite(buzzer_pin, HIGH);

 delay(500);

 digitalWrite(buzzer_pin, LOW);

 delay(500);     }
```

Devre 21:

Devre Adı: Potansiyometre ile buzzer'ı kontrol etme.

Devre Açıklaması: Potansiyometre değeri 500'den yüksek olduğunda sesli uyarı çalar.



/* Potansiyometre ile buzzer'ı kontrol etme */

```
const int POT_PIN = A3;              // Pot'a bağlı Ardunio pini
const int BUZZER_PIN  = 0;           // Buzzer'a bağlı Ardunio pini

const int ANALOG_THRESHOLD  = 500;
int analogValue;

void setup() {

  pinMode(BUZZER_PIN, OUTPUT);       // Pin çıkış olarak ayarlandı

}

void loop() {

 analogValue = analogRead(POT_PIN);            //  analog pindeki giriş değerini oku

if(analogValue > ANALOG_THRESHOLD)
digitalWrite(BUZZER_PIN, HIGH);                 // Buzzer'ı aç

else
digitalWrite(BUZZER_PIN, LOW);                  // Buzzer'ı kapat

}
```

Erasmus+ KA-202

Mesleki Eğitim ve Öğretime Yönelik Stratejik Ortaklık Projesi

Proje Adı: “Mesleki Eğitimde Arduinoları Öğrenme ve Öğretme”

Proje Kısaltması: “ ARDUinVET ”

Proje Numarası: “2020-1-TR01-KA202-093762”

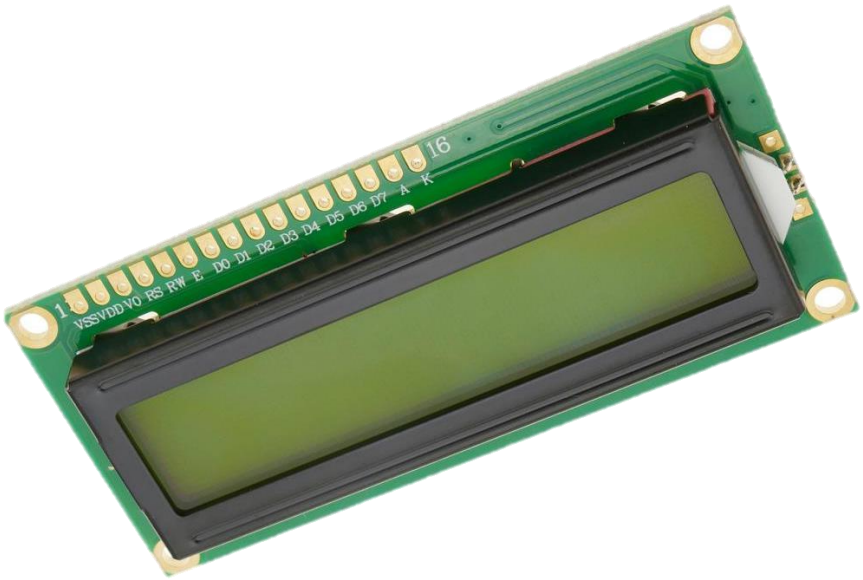
LCD Modül ve Eğitim Kiti

LCD Modül ve Eğitim Kiti

Bu modülde Arduino'nun durum mesajlarının veya sensör okumalarının LCD ekranlarda nasıl görüntüleneceğini açıklamak istiyoruz. Bunlar son derece yaygındır ve projenize okunabilir bir arayüz eklemenin hızlı bir yoludur.
LCD, Liquid Crystal Display'in kısaltmasıdır. Görünür bir görüntü oluşturmak için sıvı kristalleri kullanan bir görüntüleme birimidir. Bu özel tür kristale akım uygulandığında, opaklaşır ve ekranın arkasında kalan arka ışığı engeller. Sonuç olarak, belirli bir alan diğerine kıyasla karanlık hale gelecektir. Ve karakterler ekranda bu şekilde görüntülenir.

Arduino ile 16×2 Karakterli LCD Modülü

Arduino'nuza bağlayabileceğiniz farklı türde LCD ekranlar vardır. En yaygın olanı, Hitachi'nin HD44780 adlı paralel arabirim LCD denetleyici yongasına dayanmaktadır.
Bu LCD'ler yalnızca metin/karakterleri görüntülemek için idealdir, bu nedenle 'Karakter LCD' adı verilir. Ekranın LED arka aydınlatması vardır ve her satırda 16 karakter olmak üzere iki satırda 32 ASCII karakteri görüntüleyebilir.
Ekranda her karakter için küçük bir dikdörtgen vardır, bu dikdörtgenlerin her biri 5×8 piksellik bir ızgaradır.
Yalnızca metin görüntülemelerine rağmen, birçok boyut ve renkte gelirler: örneğin, 16×1, 16×4, 20×4, mavi arka plan üzerinde beyaz metin, yeşil üzerinde siyah metin ve çok daha fazlası.
Arduino topluluğu, HD44780 LCD'leri (LiquidCrystal Kitaplığı) işlemek için zaten bir kitaplık geliştirdi; bu yüzden onları birkaç zaman içinde arayüz haline getireceğiz.



LCD pin çıkışı aşağıdadır:

- VSS: bir topraklama pin'idir ve Arduino'nun toprağına bağlanmalıdır;
- VDD: 5 V'a bağlı;
- VD: kontrast ayarı için (potansiyometrenin merkez pin'ine bağlı);
- RS: gönderilen verilerin hangi lcd bellek bölgesinde saklandığını kontrol eder;
- R/W: okuma/yazma modunu seçmek için pin;
- E: etkinleştirilirse, LCD modülünün özel talimatları gerçekleştirmesini sağlar;
- D0 - D7: veri iletimi;
- A ve K: LCD modülüne arka ışık sağlamak için anot ve katot.

Arduino - LCD Fonksiyonları (Komutlar)

LiquidCrystal lcd() - LiquidCrystal türünde bir değişken oluşturur. Ekran 4 veya 8 veri hattı kullanılarak kontrol edilebilir. İlki ise, d0 ila d3 için pin numaralarını atlayın ve bu satırları bağlantısız bırakın. RW pini Arduino'daki bir pin yerine toprağa bağlanabilir; eğer öyleyse, bu fonksiyonun parametrelerinden çıkarın. Komut Yapısı:

LiquidCrystal(rs, enable, d4, d5, d6, d7)
LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)
LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)
LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)

Parametreler:

rs: LCD üzerindeki RS pinine bağlı olan Arduino pininin numarası
rw: LCD üzerindeki RW pinine bağlı olan Arduino pininin numarası (opsiyonel)
enable: LCD üzerindeki etkinleştirme pinine bağlı olan Arduino pininin numarası
d0, d1, d2, d3, d4, d5, d6, d7: LCD'deki ilgili veri pinlerine bağlı Arduino pinlerinin numaraları. d0, d1, d2 ve d3 isteğe bağlıdır; atlanırsa, LCD sadece dört veri hattı (d4, d5, d6, d7) kullanılarak kontrol edilecektir.
lcd.begin() - LCD ekran arayüzünü başlatır ve ekranın boyutlarını (genişlik ve yükseklik) belirtir. start() diğer LCD kitaplığı komutlarından önce çağrılmalıdır. Sözdizimi:
lcd.begin(cols, rows)

Parametreler:

lcd: LiquidCrystal türünde bir değişken
cols: ekranın sahip olduğu sütun sayısı

rows: ekranın sahip olduğu satır sayısı

lcd.print() - Metni LCD'ye yazdırır. Sözdizimi:

lcd.print(data)

lcd.print(data, BASE)

Parametreler:

lcd: LiquidCrystal türünde bir değişken

data: Yazdırılacak veri (char, byte, int, long veya string)

BASE (isteğe bağlı): sayıların yazdırılacağı taban: ikili için BIN (taban 2), ondalık için DEC (taban 10), sekizlik için OCT (taban 8), onaltılık için HEX (taban 16).

İfadeler:

byte print() yazılan bayt sayısını döndürür, ancak bu sayıyı okumak isteğe bağlıdır

lcd.setCursor() - LCD imlecini konumlandır; yani, LCD'ye yazılan sonraki metnin görüntüleneceği konumu ayarlayın. Sözdizimi:

lcd.setCursor(col, row)

Parametreler:

lcd: LiquidCrystal türünde bir değişken

col: imlecin konumlandırılacağı sütun (0 ilk sütun olmak üzere)

row: imlecin konumlandırılacağı satır (0 ilk satır olmak üzere)

lcd.clear(); - LCD ekranı temizler ve imleci sol üst köşeye konumlandırır.

Sözdizimi: lcd.clear()

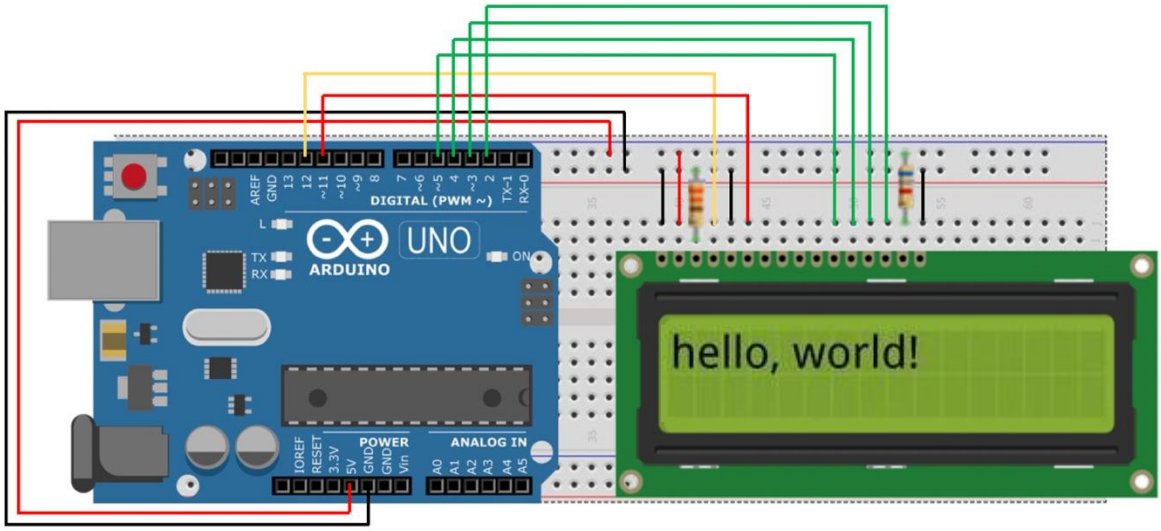
Parametreler: lcd: LiquidCrystal türünde bir değişken

Devre 1:

Devre başlığı: "LCD'ye bir mesaj yazdırın"

Devre Açıklaması: Mikrodenetleyiciye bir LCD ekran bağlanarak ekrana istenilen mesajların yazdırılması mümkündür.

Not: Aşağıda açıklanan LCD işlevlerini kullanmak için LiquidCrystal.h kitaplığını eklemeniz gerekir.




```
/*  Mesaj yazdır  */
#include <LiquidCrystal.h> // kütüphane kodunu dahil et
// LCD'deki satır ve sütun sayısı için sabitler
const int numRows = 2;
const int numCols = 16;
// kütüphaneyi arayüz pinlerinin numaralarıyla başlatın
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup()
{
lcd.begin(numCols, numRows);
lcd.print("hello, world!"); // LCD'ye bir mesaj yazdırın.
}
```

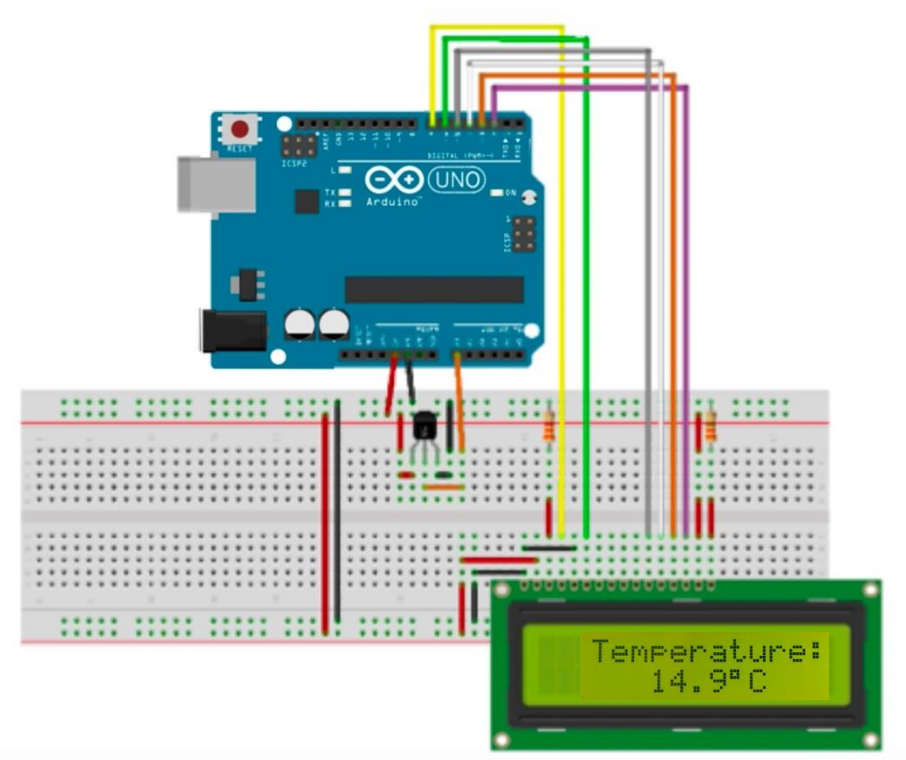
Devre 2:

Devre başlığı: "Dijital Termometre"

Devre Açıklaması: Arduino ile dijital bir termometre oluşturun. Sıcaklık, LM35 sıcaklık sensörü ile alınacak ve bir LCD ekranda görüntülenmeli ve her saniye güncellenmelidir.

LCD ekranın Vo pini, ekranda görüntülenen karakterlerin kontrastını ayarlar. Yukarıda bahsedildiği gibi GND'ye bağlı 3,3 kΩ dirence bağlanmalıdır. Ancak bazı ekranlarda Vo pinini doğrudan GND'ye bağlamanız gerekebilir.

Not: LM35 sensörünün 3 terminali vardır: biri güç kaynağı için, biri kütle için ve diğeri algılanan sıcaklığa orantılı voltaj çıkışı için, her bir santigrat derece için 10 mV'ye eşittir ve Santigrat derece olarak kalibre edilmiştir.

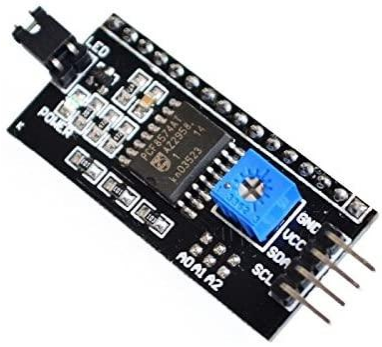



```
/*  Dijital termometre */
#include <LiquidCrystal.h> //Library to drive LCD display
#define pin_temp A0 // Sıcaklık sensörü Vout ayak bağlantı pin’i
```

```
float temp = 0; // Tespit edilen sıcaklığın saklanacağı değişken
LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // Kütüphaneyi LCD ekran pin'leriyle başlatma
void setup()
{
  lcd.begin(16, 2); // Ekrandaki sütun ve satır sayısını ayarlama
  LCD  lcd.setCursor(0, 0); // İmleci ilk satırın (0. satır) ve ilk sütunun üzerine getirin
  lcd.print ("Temperature:"); İlk satıra 'Sıcaklık:' mesajını yazdırın
 //* 1.1V'de Uygulanan ADC Vref
 (sıcaklık hesaplamasında daha fazla doğruluk için)
  NEMLİ: Arduino Mega kullanıyorsanız INTERNAL'ı INTERNAL1V1 ile değiştirin */
  analogReference(INTERNAL);
}

void loop()
{
  /* sıcaklığı hesapla ==============================================*/
  temp = 0;
  for (int i = 0; i < 5; i++) { // Sonraki ifadeyi 5 kez yürütür
    temp += (analogRead(pin_temp) / 9.31); // Sıcaklık ve toplamı 'temp' değişkeninde hesaplar
  }
  temp /= 5; // Sıcaklık değerlerinin matematiksel ortalamasını hesaplar
  /*================================================================*/

  /* LCD ekranda sıcaklığı görüyorum
  ==============================================*/
  lcd.setCursor(0, 1); // lcd.print(temp); İmleci ilk sütunun ve ikinci satırın
  üzerine getirin lcd.print(temp);
  // LCD ekran sıcaklığında kalıp
  lcd.print(" C"); // Ekranda bir boşluk ve 'C' yazı tipi oluşturun
 /*=============================================*/
  delay(1000); // Bir saniye gecikme (değiştirilebilir)
  }
```

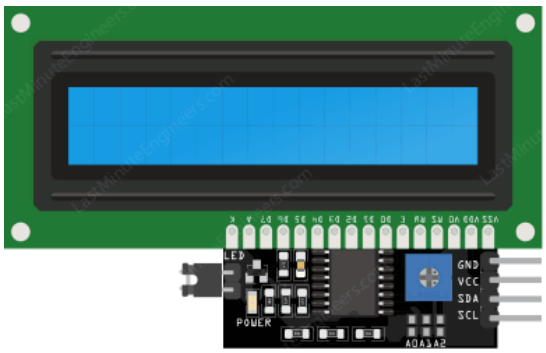
Arduino ile I2C LCD'yi Arayüzleme

Arduino ile LCD ekran bağlamak istiyorsanız Arduino üzerinde çok fazla pin tüketmeniz gerekiyor. 4 bit modunda bile, Arduino hala mevcut dijital G/Ç pinlerinin yarısı olan toplam yedi bağlantı gerektirir.
Çözüm, I2C protokolü ile arayüz oluşturan bir LCD ekran kullanmaktır. Yalnızca bir dijital I/O pin setinin parçası olamayacak ve diğer I2C cihazlarıyla paylaşılabilen iki I/O pini tüketir.
En yaygın I2C LCD ekranı, HD44780 tabanlı bir karakter LCD ekranı ve bir I2C LCD adaptöründen oluşur.



Adaptörün en önemli parçası 8-Bit G/Ç Genişletici yongası – PCF8574. Bu çip, bir Arduino'dan gelen I2C verilerini LCD ekranın gerektirdiği paralel verilere dönüştürür. Kart ayrıca ekranın kontrastında ince ayarlar yapmak için küçük bir potansiyometre ile birlikte gelir. Aynı I2C veriyolu üzerinde birden fazla cihaz kullanıyorsanız, başka bir I2C cihazı ile çakışmaması için kart için farklı bir I2C adresi ayarlamanız gerekebilir. Bu nedenle kartta üç adet lehim jumper'ı (A0, A1 ve A2) bulunur.

Bir I2C LCD, onu Arduino'ya bağlayan sadece 4 pin'e sahiptir.



Pin çıkışı aşağıdaki gibidir:

- GND: bir topraklama pin'idir ve Arduino'nun toprağına bağlanmalıdır;

- VCC: modüle ve LCD'ye güç sağlar. Arduino'nun 5V çıkışına veya ayrı bir güç kaynağına bağlayın;

- SDA: Seri Veri pinidir. Bu hat hem gönderme hem de alma için kullanılır. Arduino üzerindeki SDA pinine bağlanın;

- SCL: Seri Saat pinidir. Bu, Bus Master cihazı tarafından sağlanan bir zamanlama sinyalidir. Arduino üzerindeki SCL pinine bağlayın.

R3 düzenine sahip Arduino kartlarında, SDA (veri hattı) ve SCL (saat hattı), AREF pinine yakın pin başlıklarında bulunur. A5 (SCL) ve A4 (SDA) olarak da bilinirler.
Arduino modeline bağlı olarak doğru pinleri belirlemek için aşağıdaki tabloya bakın.

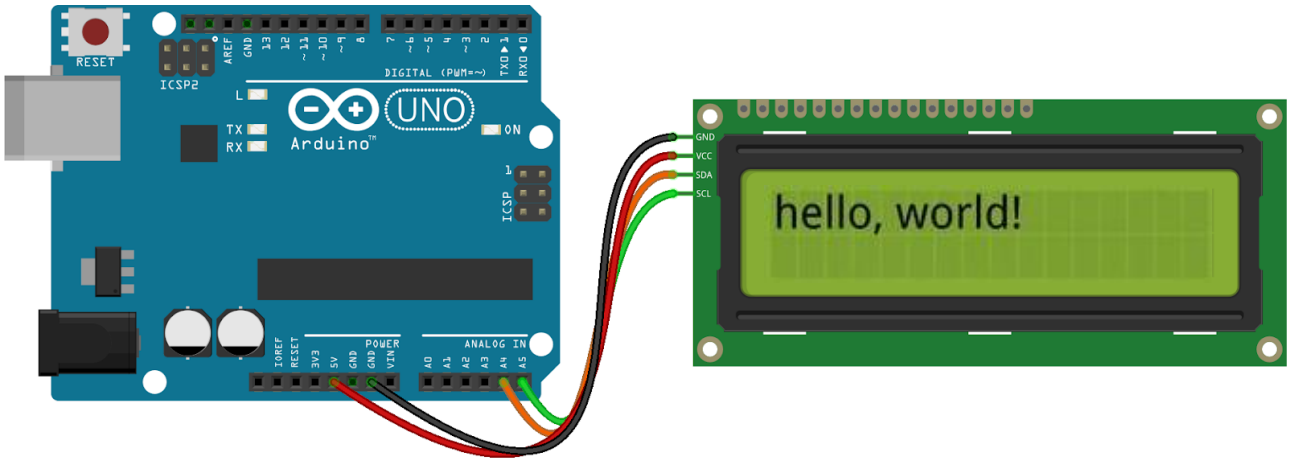
	SCL	SDA
Arduino Uno	A5	A4
Arduino Nano	A5	A4
Arduino Mega	21	20
Leonardo/Micro	3	2

Devre 3:

Devre başlığı: "I2C LCD'ye bir mesaj yazdırın"

Devre Açıklaması: Mikrodenetleyiciye sadece 4 pin ile bir LCD ekran bağlayıp ekrana istenilen mesajları yazdırmak mümkündür.

Not: LiquidCrystal_I2C adlı bir kitaplık kurmanız gerekir. Bu kitaplık, Arduino IDE'nizle birlikte gelen LiquidCrystal kitaplığının geliştirilmiş bir sürümüdür.




```
/*  I2C Ekranında bir mesaj yazdırın */
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F,16,2);  // 16 karakter ve 2 satır ekran için LCD adresini 0x3F olarak ayarlayın
void setup() {
  lcd.init();
  lcd.clear();
  lcd.backlight();      // Arka ışığın açık olduğundan emin olun
  // LCD'nin her iki satırına da bir mesaj yazdırın.
  lcd.setCursor(2,0);   // İmleci 0 satırındaki karakter 2'ye ayarla
  lcd.print("Hello world!");
  lcd.setCursor(2,1);   // İmleci satır 1'deki karakter 2'ye taşıyın
  lcd.print("with I2C protocol!");
}
void loop() {
}
```

Arduino ile OLED Grafik Ekran Modülü

I2C protokolünü kullanmanın bir başka olasılığı da bir OLED (Organik Işık Yayan Diyot) ekranı seçmektir. Süper hafif ve incedirler ve daha parlak ve net bir resim üretirler.

Bir OLED ekran, arka ışık olmadan çalışır. Ekranın bu kadar yüksek kontrasta, son derece geniş görüş açısına sahip olmasının ve derin siyah seviyelerini gösterebilmesinin nedeni budur. Arka ışığın olmaması, OLED'i çalıştırmak için gereken gücü önemli ölçüde azaltır.

Şekilden de görebileceğimiz gibi, pin çıkışı yukarıda zaten görülen klasik dört pinli I2C arayüzüdür.

Arduino topluluğu, bu OLED ekranları işlemek için Adafruit'in SSD1306 kitaplığı gibi birkaç kitaplık geliştirdi. Kitaplığı kurmak için Çizim > Kitaplığı Dahil Et > Kitaplıkları Yönet'e gidin... Kitaplık Yöneticisinin kitaplıklar dizinini indirmesini ve kurulu kitaplıkların listesini güncellemesini bekleyin.

Arduino - OLED Grafik Ekran Modülü Fonksiyonları (Komutları)

pinMode();

display.begin()

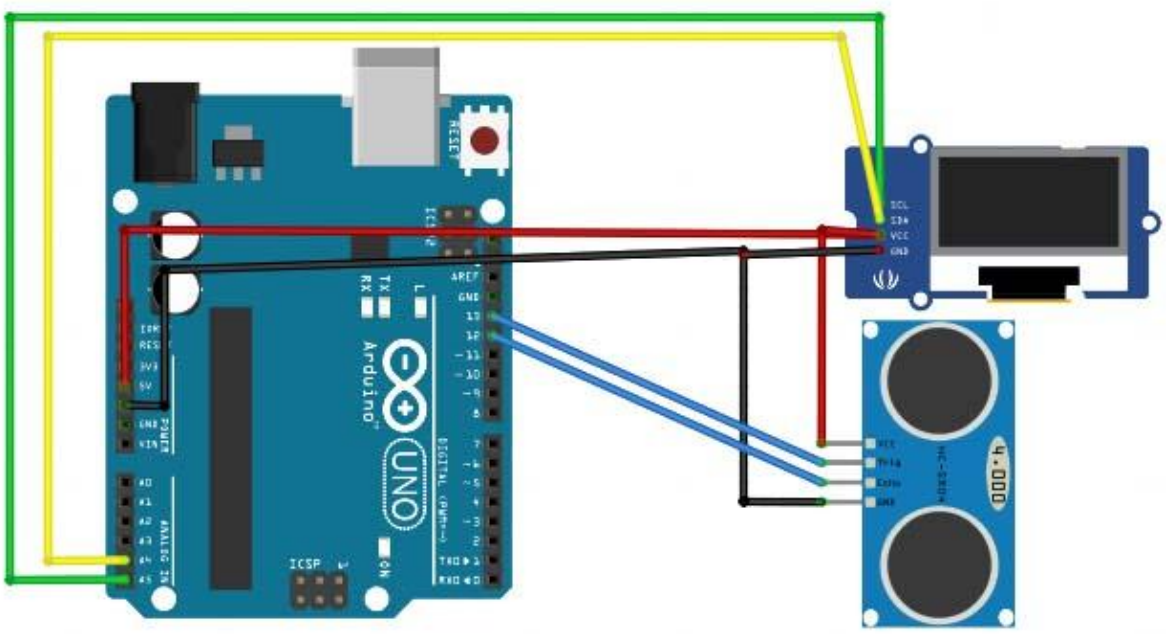
display.clearDisplay();

Devre 4:

Devre başlığı: "Mesafe sensörü"

Devre Açıklaması: Mesafe HC-SR04 ultrasonik sensör ile alınacak ve OLED ekranda görüntülenecektir.

Not: HC-SR04'te endişelenmeniz gereken yalnızca dört pin vardır: VCC (Güç), Trig (Trigger), Echo (Alma) ve GND (Ground).



/* Mesafe sensörü */

```
#include <SPI.h>        // Bu kütüphane, ana cihaz olarak Arduino ile SPI cihazları ile iletişim
                           kurmanıza izin verir.
#include <Wire.h>      // bu kitaplık I2C/TWI cihazlarıyla iletişim kurmanızı sağlar
#include <Adafruit_GFX.h>            // OLED ekranın kitaplıkları
#include <Adafruit_SSD1306.h>

#define CommonSenseMetricSystem
#define trigPin 13      // sensörün pin'lerini tanımlayın
#define echoPin 12
```

```
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // I2C adresi 0x3C (128x64) ile başlat
  display.clearDisplay();

}

void loop() {
  long duration, distance;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  display.setCursor(22,20);    // OLED ekran ayar imleci
  display.setTextSize(3);        // metnin boyutu
  display.setTextColor(WHITE);       // siyah yazarsan her şeyi siler
  display.println(distance);     // değişkenimizi yazdır
  display.setCursor(85,20);
  display.setTextSize(3);

    display.println("cm");

  display.display();

  delay(500);
  display.clearDisplay();

  Serial.println(distance);//debug
}
```

Arduino ile Nokia 5110 Grafik LCD Ekranı Arayüzü

Arduino'yu Nokia'nın 3310 ve 5110 cep telefonlarında kullandığına benzer küçük LCD'lerle arayüzleyebilirsiniz. bu ekranlar küçüktür (yalnızca 1,5 inç), ucuzdur, kullanımı kolaydır, oldukça düşük güçtedir (yalnızca 6 ila 7mA kadar düşük) ve metinlerin yanı sıra bitmapleri de görüntüleyebilir.

Bunlar 84×48 piksellik grafik ekranlardır. SPI'ye benzer bir seri veri yolu arabirimi aracılığıyla mikro denetleyicilere arabirim oluştururlar. LCD ayrıca kırmızı, yeşil, mavi ve beyaz gibi farklı renklerde bir arka ışık ile birlikte gelir. Arka ışık, ekranın kenarlarına yerleştirilmiş dört LED'den başka bir şey değildir.
Arduino ile arayüz oluşturan 8 pini vardır, pin çıkışı aşağıdaki gibidir:

- RST: ekranı sıfırlar. Aktif bir düşük pin anlamıdır; aşağı çekerek ekranı sıfırlayabilirsiniz. Bu pini de Arduino resetine bağlayarak ekranı otomatik olarak resetlemesini sağlayabilirsiniz;

- CE(Chip Enable): aynı SPI veri yolunu paylaşan birçok bağlı cihazdan birini seçmek için kullanılır;

- D/C(Veri/Komut): pin ekrana aldığı verinin komut mu yoksa görüntülenebilir veri mi olduğunu belirler;

- DIN: SPI arayüzü için bir seri veri pinidir;

- CLK: SPI arayüzü için bir seri saat pinidir;

- VCC: Arduino üzerindeki 3.3V volt pinine bağladığımız LCD'ye güç sağlar;

- BL(Arka Işık): ekranın arka ışığını kontrol eder. Parlaklığını kontrol etmek için bir potansiyometre ekleyebilir veya bu pini herhangi bir PWM özellikli Arduino pinine bağlayabilirsiniz;

- GND: bir topraklama pin'idir ve Arduino'nun toprağına bağlanmalıdır.

Veri iletim pinlerini herhangi bir dijital I/O pinine bağlayabilirsiniz. LCD'nin 3v iletişim seviyeleri vardır, bu yüzden bu pinleri doğrudan Arduino'ya bağlayamıyoruz. Bunun bir yolu, her bir veri iletim pinine inline dirençler eklemektir. Sadece CLK, DIN, D/C ve RST pinleri arasına 10kΩ direnç ve CE arasına 1kΩ direnç ekleyin. Arka ışık (BL) pin'i, 330Ω akım sınırlama direnci ile 3.3V'a bağlanır. Parlaklığını kontrol etmek istiyorsanız, bir potansiyometre ekleyebilir veya bu pini herhangi bir PWM özellikli Arduino pinine bağlayabilirsiniz.
Arduino topluluğu, bu NOKIA ekranlarını işlemek için Adafruit'in PCD8544 Nokia 5110 LCD kitaplığı gibi birkaç kitaplık geliştirdi. Kitaplığı kurmak için Çizim > Kitaplığı Dahil Et > Kitaplıkları Yönet'e gidin... Kitaplık Yöneticisinin kitaplıklar dizinini indirmesini ve kurulu kitaplıkların listesini güncellemesini bekleyin.

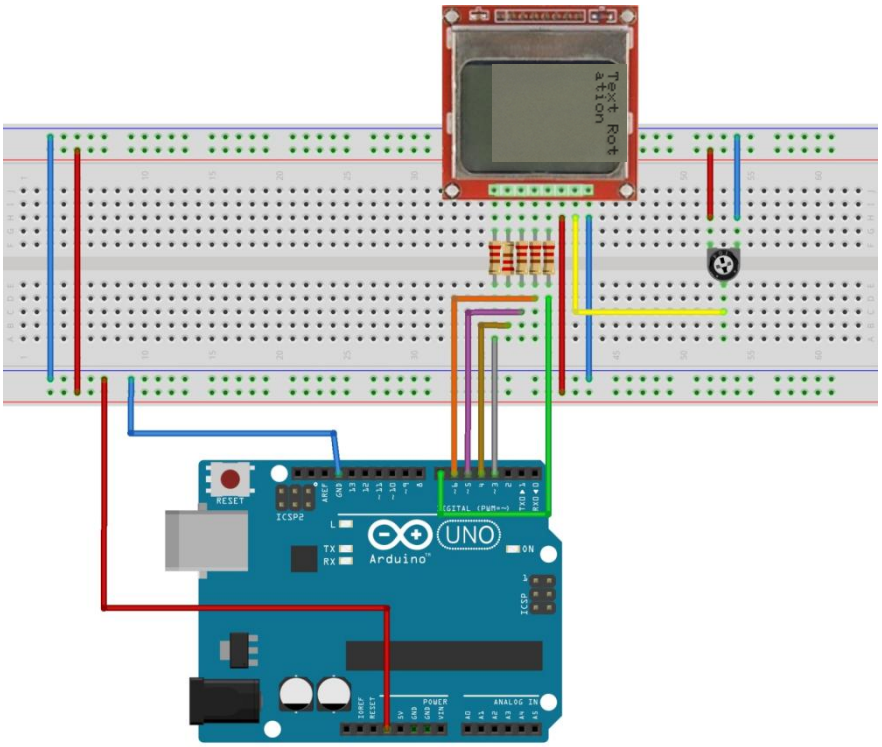
Devre 5:

Devre başlığı: "Metin Döndürme"

Devre Açıklaması: setRotation() işlevini çağırarak ekranın içeriğini döndürebilirsiniz. Ekranınızı portre modunda görüntülemenizi veya baş aşağı çevirmenizi sağlar.

Not: İşlev, 4 ana dönüşe karşılık gelen yalnızca bir parametreyi kabul eder. Bu değer, 0'dan başlayarak negatif olmayan herhangi bir tam sayı olabilir. Değeri her artırdığınızda, ekranın içeriği saat yönünün tersine 90 derece döndürülür. Örneğin:

• 0 – Ekranı standart yatay yönde tutar.
• 1 – Ekranı 90° sağa döndürür.
• 2 – Ekranı baş aşağı çevirir.
• 3 – Ekranı 90° sola döndürür.



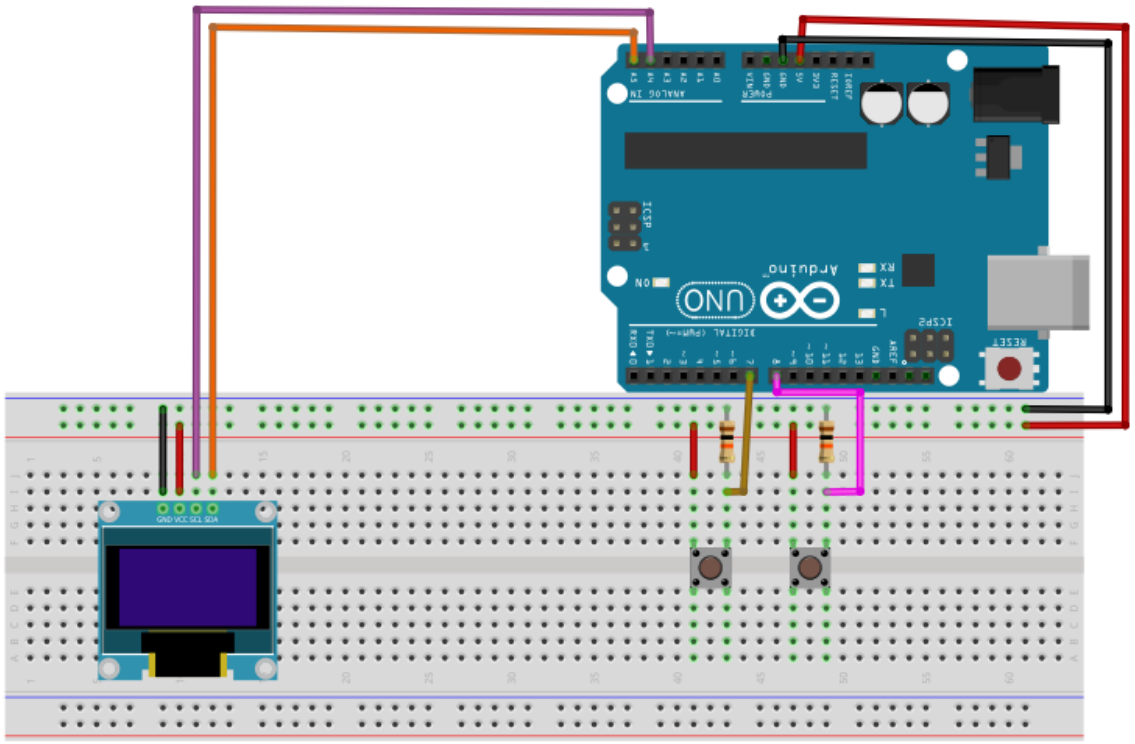

```
/*  Metin Döndürme */
#include <SPI.h>        // Bu kütüphane, ana cihaz olarak Arduino ile SPI cihazları ile iletişim
                        kurmanıza izin verir.
#include <Adafruit_GFX.h>           // OLED ekranın kitaplıkları
#include <Adafruit_PCD8544.h>
// Declare LCD object for software SPIAdafruit_PCD8544(CLK,DIN,D/C,CE,RST);
Adafruit_PCD8544 display = Adafruit_PCD8544(7, 6, 5, 4, 3);

void setup()  {
        Serial.begin(9600);
```

```
        //Initialize Display
        display.begin();

        display.setContrast(57);        // ekranı uyarlamak için kontrastı değiştirebilirsiniz

        display.clearDisplay();         // arabelleği temizle

// Text Rotation
while(1)
{
        display.clearDisplay();
        display.setRotation(rotatetext);
        display.setTextSize(1);
        display.setTextColor(BLACK);
        display.setCursor(0,0);
        display.println("Text Rotation");
        display.display();
        delay(1000);
        display.clearDisplay();
        rotatetext++;
}
}

void loop() {}
```

Devre 6:

Devre başlığı: "Düğme sayacı"

Devre Açıklaması: İki farklı düğmenin tıklanmasıyla artırılıp azaltılabilen bir sayı gösteren OLED ekran.



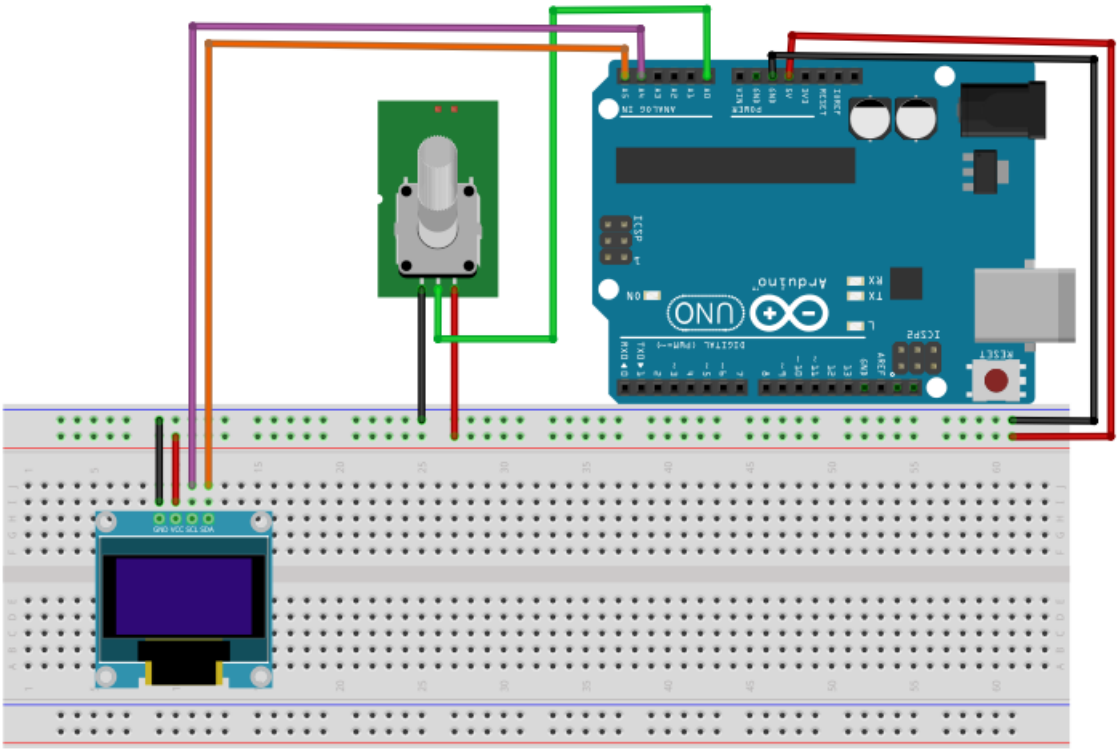
```
#include <U8glib.h> //lcd libraries
#include "U8glib.h"
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE|U8G_I2C_OPT_DEV_0);  //display model
int button_plus = 8, button_minus = 7; // bildirim pin düğmeleri
int state_plus, state_minus, number=0;
void setup() {
  pinMode(button_plus,INPUT);
  pinMode(button_minus,INPUT);
  Serial.begin(9600);
  u8g.setFont(u8g_font_fub25n);  // sayının yazılması için kullanılacak yazı tipi
}
void write_number(void) {
  u8g.setPrintPos(10,50);
```

```
  u8g.print(number);
}
void loop() {
  //buttons
  state_plus = digitalRead(button_plus);
  state_minus = digitalRead(button_minus);
  if(state_plus==1)
  {
     number++;
     delay(50);
  }
  if(state_minus==1)
  {
    number--;
    delay(50);
  }
  //write de number sul display
  u8g.firstPage();
  do {
    write_number();
  } while  ( u8g.nextPage() );
    u8g.firstPage();
}
```

Devre 7:

Devre başlığı: "Potansiyometreli sayaç"

Devre Açıklaması: Potansiyometre döndürüldükçe artan ve azalan bir sayı gösteren OLED ekran.



```
#include <U8glib.h> //lcd libraries
#include "U8glib.h"
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE|U8G_I2C_OPT_DEV_0);  //display model

int potentiometer = 0; //declaration and potentiometer
int state_pot, stop_pot, difference, number=0;

void setup() {
  Serial.begin(9600);
  u8g.setFont(u8g_font_fub25n);//font to be used for the write of the number
}

void write_number(void) {
  u8g.setPrintPos(10,50);
```

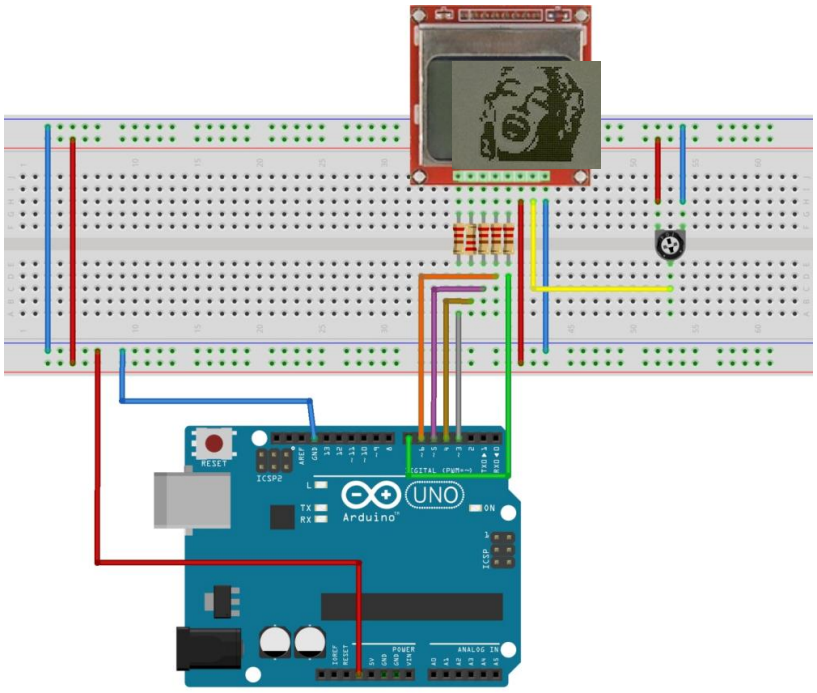
```
  u8g.print(number);
}

void loop() {
  state_pot = analogRead(potentiometer);

  //potentiometer
  stop_pot = state_pot;//save the value when it is stop to see if the potentiometer turns clockwise
or counterclockwise
  delay(100);
  state_pot = analogRead(potentiometer);
  difference = stop_pot-state_pot;
  if((difference>2)||(difference<2))// potansiyometre dururs a işlem yapmaktan kaçınmak için
kullanılır
  {
    number=number-difference/5;// fark 0'dan büyükse saat yönünde döner aksi halde çıkarır
    delay(100);
  }

  // ekrandaki numarayı yaz
  u8g.firstPage();
  do {
    write_number();
  } while
  ( u8g.nextPage() );
    u8g.firstPage();
}
```

Devre 8:

Devre başlığı: "Marilyn Bmp Image"

Devre Açıklaması: Nokia 5110 LCD Ekranına bitmap görüntüleri nasıl çizilir. Bu örnekte Marilyn Monroe'nun bir portresi var.

Not: Nokia 5110 LCD ekranının ekran çözünürlüğü 84×48 pikseldir, bu nedenle bundan daha büyük resimler düzgün görüntülenmeyecektir. Bitmap görüntüsünü Nokia 5110 LCD ekranında göstermek için drawBitmap() işlevini çağırmamız gerekiyor. Altı parametre alır:. sol üst köşe X koordinatı, sol üst köşe Y koordinatı, monokrom bitmap bayt dizisi, piksel cinsinden bitmap genişliği, piksel cinsinden bitmap yüksekliği ve Renk. Örneğimizde bitmap görüntüsü 84×48 boyutundadır. Bu nedenle, X ve Y koordinatları 0'a, genişlik ve yükseklik ise 84 ve 48'e ayarlanmıştır.



```
/*  Marilyn Bmp Görüntüsü */
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h>
Adafruit_PCD8544 display = Adafruit_PCD8544(7, 6, 5, 4, 3);
// 'Marilyn Monroe 84x48', 84x48px
const unsigned char MarilynMonroe [] PROGMEM = {
0x00, 0x00, 0x00, 0x7f, 0x00, 0x02, 0xfe, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xbe, 0x00,
0x00, 0x1f, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x00, 0x00, 0x3f, 0x80, 0x00, 0x00,
```

0x00, 0x00, 0x00, 0x00, 0xf0, 0x00, 0x00, 0x1f, 0xe1, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0xc0,
0x00, 0x00, 0x0f, 0xf1, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0e, 0xd8, 0xe0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0x80, 0x00, 0x07, 0xe0, 0x70, 0x00, 0x00, 0x00, 0x00, 0x03,
0x3f, 0xe0, 0x00, 0x07, 0xf0, 0x78, 0x00, 0x00, 0x00, 0x00, 0x01, 0xe0, 0x70, 0x00, 0x0f, 0xee,
0x7c, 0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x0f, 0xf7, 0x1c, 0x00, 0x00, 0x00, 0x00,
0x07, 0x80, 0x00, 0x0f, 0xc7, 0xf3, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x07, 0xc0, 0x00, 0x0f, 0xf3,
0xdf, 0x7f, 0x80, 0x00, 0x00, 0x00, 0x07, 0xfe, 0x00, 0x08, 0x7d, 0xef, 0xff, 0xc0, 0x00, 0x00,
0x00, 0x7f, 0xff, 0x80, 0x30, 0x0f, 0xfc, 0xe0, 0xc0, 0x00, 0x00, 0x01, 0x9e, 0x73, 0xc0, 0xe0,
0x07, 0xf8, 0xc1, 0xc0, 0x00, 0x00, 0x03, 0xfc, 0x00, 0x01, 0xc0, 0x0f, 0xfd, 0xe1, 0x80, 0x00,
0x00, 0x03, 0xf8, 0x00, 0x01, 0x9c, 0x0f, 0xff, 0xc1, 0xc0, 0x00, 0x00, 0x02, 0xc0, 0x00, 0x01,
0x9f, 0xbf, 0xfe, 0x01, 0x40, 0x00, 0x00, 0x02, 0x60, 0x00, 0x03, 0x07, 0xef, 0xff, 0x01, 0x40,
0x00, 0x00, 0x00, 0x60, 0x00, 0x07, 0x01, 0xf7, 0xff, 0x80, 0xc0, 0x00, 0x00, 0x00, 0x50, 0x01,
0xdf, 0x00, 0x7f, 0xff, 0x1c, 0x80, 0x00, 0x00, 0x00, 0x40, 0x01, 0xff, 0x00, 0x1f, 0xff, 0x1e,
0xe0, 0x00, 0x00, 0x02, 0x08, 0x00, 0x3f, 0x80, 0x07, 0xef, 0x03, 0xe0, 0x00, 0x00, 0x06, 0x08,
0x00, 0x03, 0xc0, 0x07, 0xdf, 0x07, 0xc0, 0x00, 0x00, 0x06, 0x08, 0x0f, 0x81, 0x80, 0x1f, 0xdf,
0x1f, 0x80, 0x00, 0x00, 0x03, 0x08, 0x1f, 0x98, 0x00, 0x3f, 0xfe, 0x19, 0x80, 0x00, 0x00, 0x18,
0x08, 0x3f, 0xfe, 0x00, 0x7f, 0xfe, 0x3f, 0x00, 0x00, 0x00, 0x08, 0x08, 0x30, 0x3f, 0x00, 0xff,
0xff, 0x3f, 0x00, 0x00, 0x00, 0x01, 0xe0, 0x76, 0x0f, 0x89, 0xff, 0xff, 0x9f, 0x00, 0x00, 0x00,
0x03, 0xe0, 0x7f, 0xc3, 0x81, 0xff, 0xfe, 0x9f, 0x80, 0x00, 0x00, 0x03, 0xf0, 0x7f, 0xf3, 0xc3,
0xff, 0xfe, 0x1f, 0x00, 0x00, 0x00, 0x03, 0xf0, 0x7f, 0xfd, 0xc3, 0xff, 0xfe, 0x5e, 0x00, 0x00,
0x00, 0x03, 0xf0, 0x7f, 0xff, 0xc3, 0xff, 0xf3, 0x1e, 0x00, 0x00, 0x00, 0x03, 0xf0, 0x71, 0xff,
0x87, 0xff, 0xe3, 0xff, 0x00, 0x00, 0x00, 0x07, 0xf0, 0x7c, 0x3f, 0x87, 0xff, 0xe3, 0xfe, 0x00,
0x00, 0x00, 0x0f, 0xf0, 0x3c, 0xff, 0x05, 0xff, 0xf3, 0xfc, 0x00, 0x00, 0x00, 0x0f, 0xf0, 0x0f,
0xfe, 0x09, 0xff, 0xf7, 0xfc, 0x00, 0x00, 0x00, 0x08, 0xf8, 0x01, 0xfc, 0x19, 0xff, 0xff, 0xf8,
0x00, 0x00, 0x00, 0x0c, 0x78, 0x00, 0x00, 0x13, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x0e, 0x78,
0x00, 0x00, 0x23, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x0e, 0xf8, 0x00, 0x00, 0x47, 0xff, 0xff,
0xf0, 0x00, 0x00, 0x00, 0x0c, 0xfa, 0x00, 0x01, 0x8f, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x08,
0x7b, 0x00, 0x03, 0x3f, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x0c, 0x3b, 0xf8, 0x0f, 0xff, 0xff,
0xff, 0xe0, 0x00, 0x00, 0x00, 0x0f, 0xbb, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00,
0x07, 0xfb, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x71, 0xff, 0xff, 0xff,

```
0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x41, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00};
void setup()   {
        Serial.begin(9600);
        display.begin();
        display.setContrast(57);
        display.clearDisplay();
        // Display bitmap
        display.drawBitmap(0, 0,  MarilynMonroe, 84, 48, BLACK);
        display.display();
        // Invert Display
        //display.invertDisplay(1);
}
void loop() {}
```

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+ KA-202

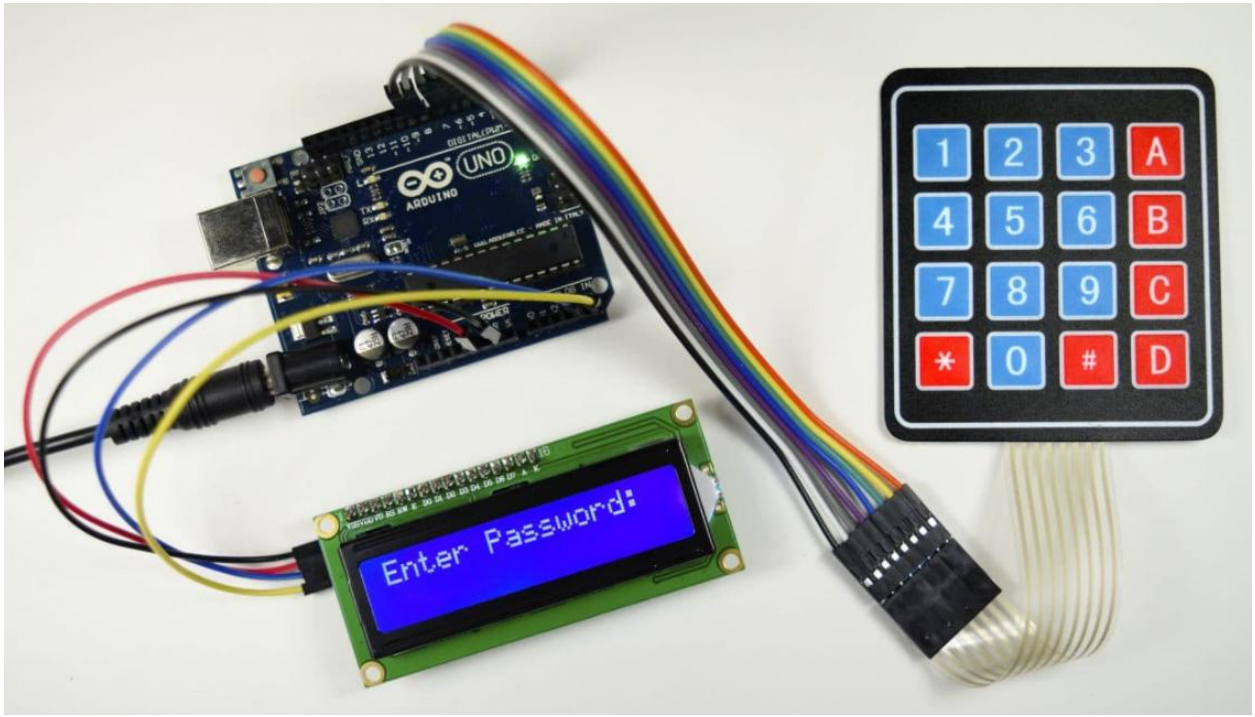
Mesleki Eğitim ve Öğretime Yönelik Stratejik Ortaklıklar Projesi

Proje Başlığı : "Mesleki Eğitimde Arduino'ları Öğretme ve Öğrenme"

Proje Kısaltması: " ARDUinVET "

Proje No: "2020-1-TR01-KA202-093762"

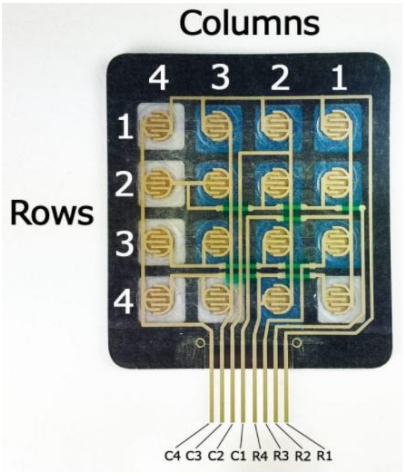
Keypad Modülü ve Eğitim Kiti



Keypad (Tuş takımı) nedir?

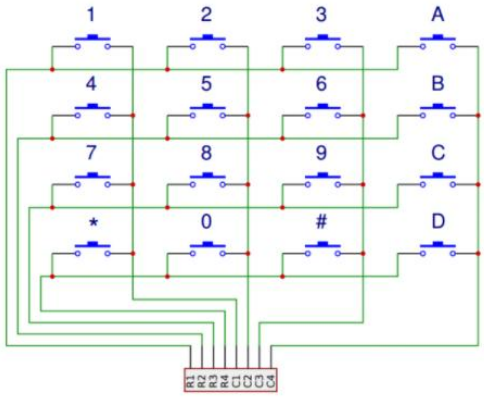
Tuş takımı, bir satır ve bir sütun arasında bağlantı sağlamak için bir anahtarlama elamanı olarak çalışan, bir matris içinde düzenlenmiş butonlar sistemidir.

Burada, 4X4 matrisli bir membran (ince) keypad kullanıyoruz. Çünkü, membran keypad incedir ve çoğu düz yüzeye yapıştırılabilmesi için yapışkan desteği vardır. Her tuşun altında bir membran anahtar bulunur. Sıradaki her bir tuş, yüzeyin altındaki iletken bir yol ile sıradaki diğer tuşlara bağlanır. Bir sütundaki her tuş aynı şekilde bağlanır - tuşun bir tarafı, iletken bir yol ile o sütundaki diğer tüm tuşlara bağlanır. Her satır ve sütun için yani toplamda, 4X4 tuş takımında toplam 8 pini için dışarıya bir bağlantı pini getirilir.	

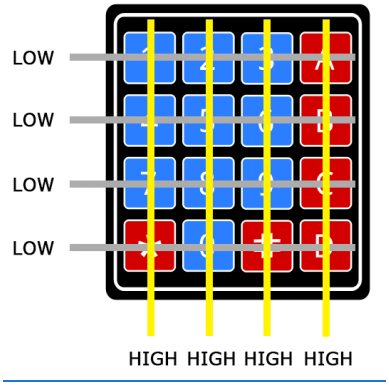
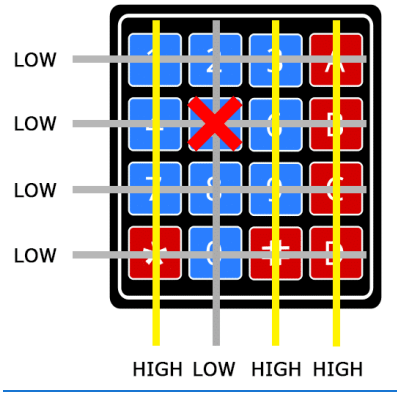


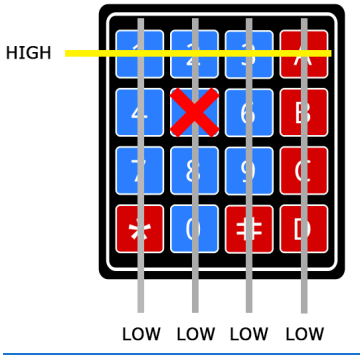
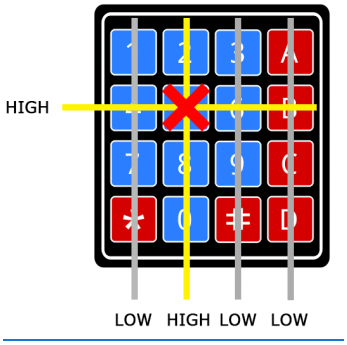
Bir tuşa basarak, bir sütun ve bir satır yolu arasındaki anahtar kapatılır akım bir sütun pini ile bir satır pini arasında akmasına izin verilir.

4X4 tuş takımı şeması, satırların ve sütunların nasıl bağlandığını gösterir:



Arduino, butona bağlı olan satır ve sütun pinini algılayarak hangi butona basıldığını algılar. Bu dört adımda gerçekleşir:

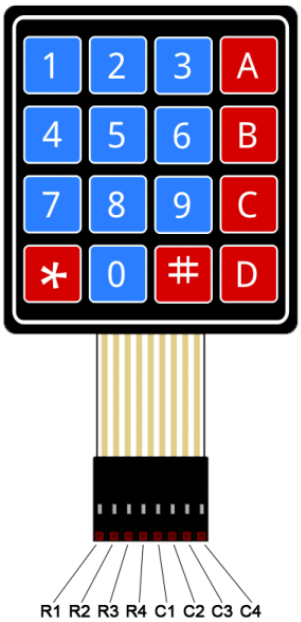
1. İlk olarak, hiçbir tuşa basılmadığında, tüm sütun pinleri YÜKSEK (HIGH) ve tüm satır pinleri DÜŞÜK (LOW) tutulur	
2. Bir tuşa basıldığında, YÜKSEK (HIGH) sütundan gelen akım, DÜŞÜK (LOW) satır pinine aktığı için, sütun pini DÜŞÜK'e (Lojik-0'a) çekilir:	

3. Arduino, artık butonun hangi sütunda olduğunu bilir. Bu yüzden şimdi sadece butonun bulunduğu satırı bulması gerekir. Bunu, satır pinlerinin her birini YÜKSEK olarak değiştirerek ve aynı zamanda YÜKSEK'e dönen sütün pinlerini tespit ederken, tüm sütun pinlerini okuyarak yapar.	
4. Sütun pini tekrar YÜKSEK'e gittiğinde, Arduino tuşa bağlı olan satır pinini bulur.	

Yukarıdaki şekilde, 2. satır ve 2. sütun kombinasyonunun yalnızca 5 numaralı tuşa basıldığı anlamına gelebileceğini görebiliriz.

TUŞ TAKIMINI ARDUINOYA BAĞLAMA

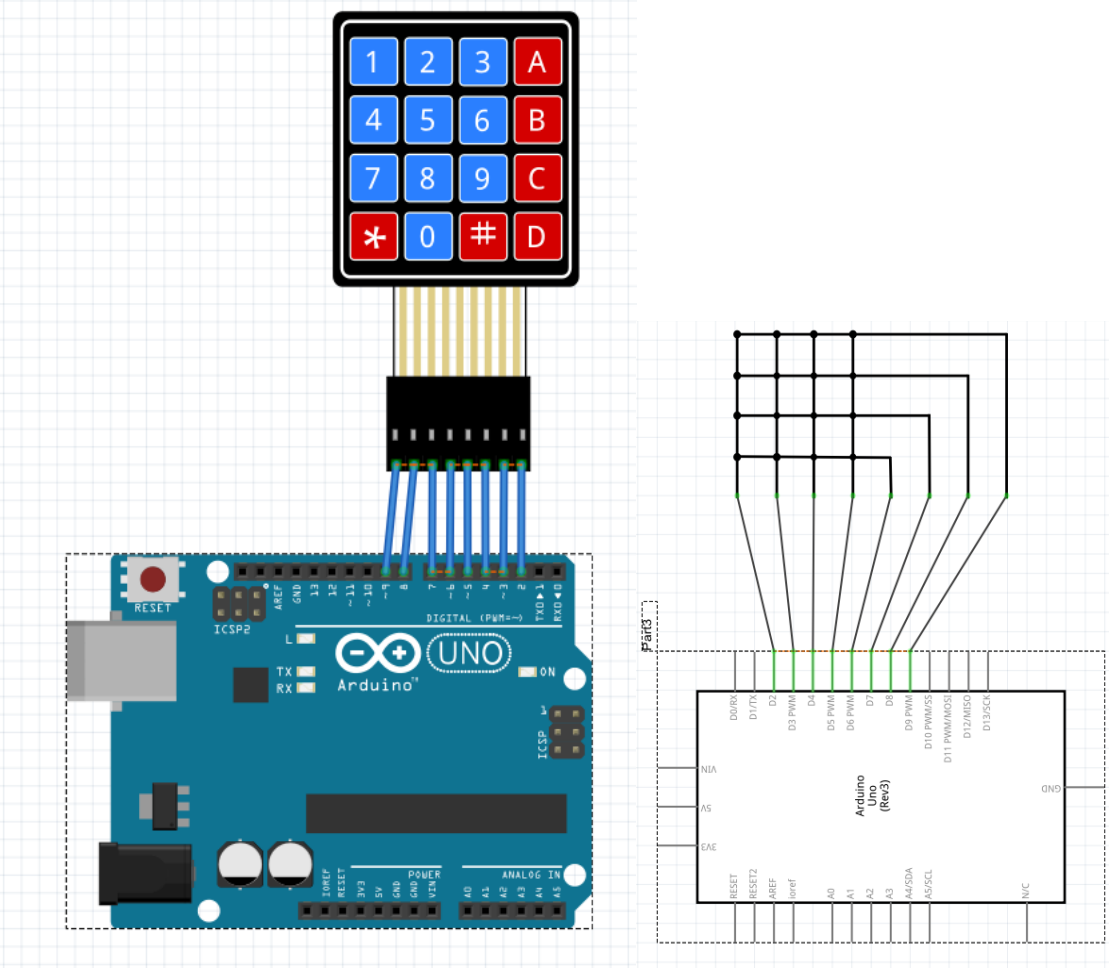
Çoğu membran tuş takımı için pin düzeni aşağıdaki gibidir:



Devre 1:

Devre Adı: Basılan tuşu Seri monitörde gösterir

Devre tanımı: Program bize basılan tuşun numarasının seri monitöre nasıl yazdırılacağını gösterir.

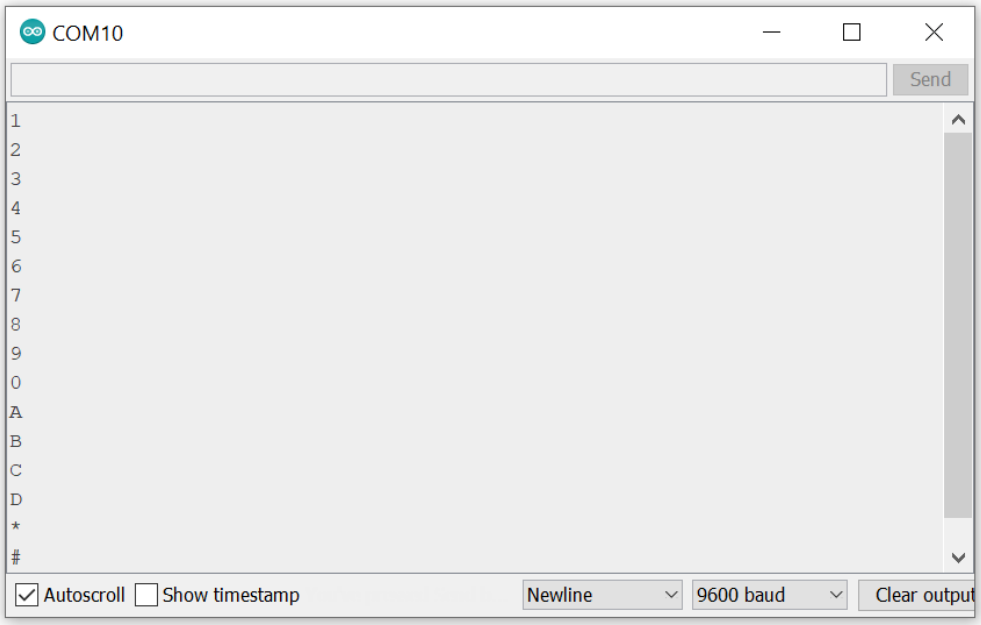


1-) Breadboard görünümü 2-) Şematik görünüm

/* "Basılan tuş Seri Monitörde gösterilir" */

```
#include <Keypad.h>
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
```

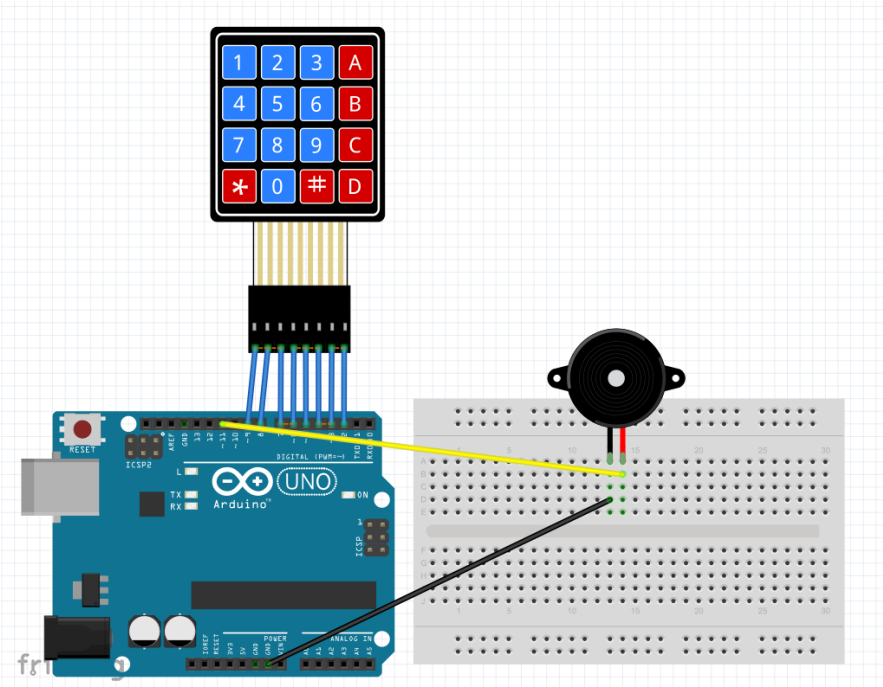
```
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {9, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};
Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS,
COLS);
void setup(){
  Serial.begin(9600);
}
void loop(){
  char customKey = customKeypad.getKey();
  if (customKey){
    Serial.println(customKey);
  }
}
```



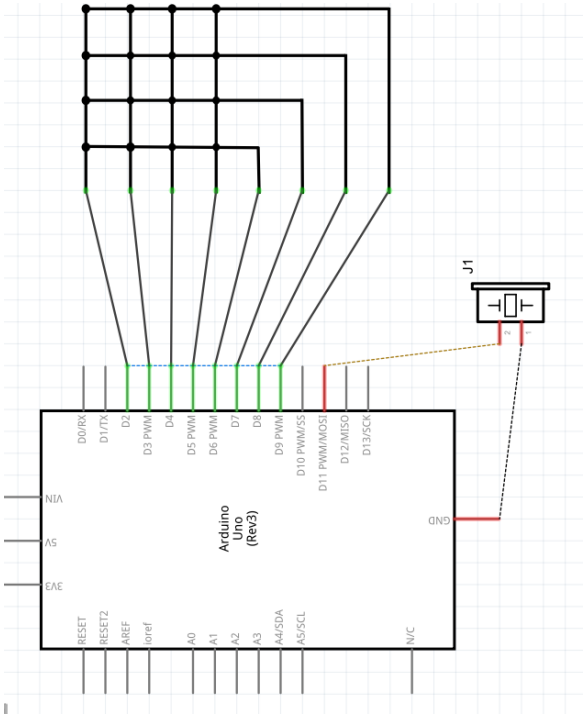
Devre 2:

Devre Adı: Arduino Keypad + Buzzer (Bee)p

Devrenin tanımı: Tuş takımındaki bir tuşa basıldığında, buzzer bip sesi çıkarır.



1-) Breadboard görünümü



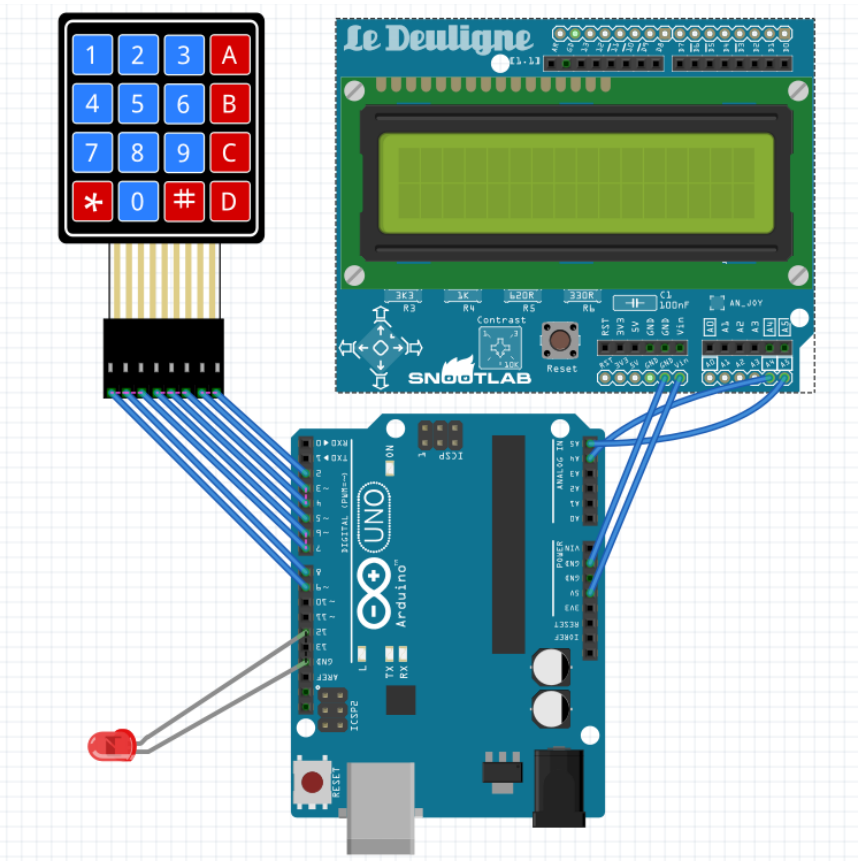
2-) Şematik görünüm

```
/*    "Arduino Keypad + Buzer ( Bip) "   */

#include <Keypad.h>
#include <ezBuzzer.h>
const int BUZZER_PIN = 11;
const int ROW_NUM    = 4; // four rows
const int COLUMN_NUM = 4; // four columns
char keys[ROW_NUM][COLUMN_NUM] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte pin_rows[ROW_NUM] = {9, 8, 7, 6};      // Tuş takımının satır pinlerine bağla
byte pin_column[COLUMN_NUM] = {5, 4, 3, 2}; // Tuş takımının sütün  pinlerine bağla

Keypad keypad = Keypad(makeKeymap(keys), pin_rows, pin_column, ROW_NUM,
COLUMN_NUM );
ezBuzzer buzzer(BUZZER_PIN);       // Pine bağlanan ezBuzzer nesnesini oluştur;
void setup() {
  Serial.begin(9600);
}
void loop() {
  buzzer.loop(); // MUST call the buzzer.loop() function in loop()
  char key = keypad.getKey();
  if (key) {
    Serial.print(key);                 // Basılan tuş numarasını Seri monitöre yazdır.
    buzzer.beep(100);                  // 100ms beep  sesi çıkart.
  }
}
```

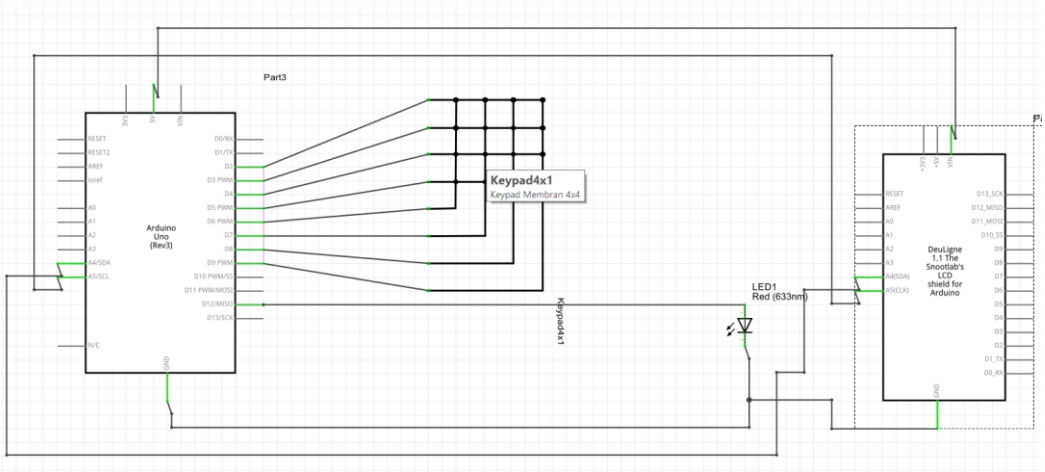
Devre 3:

Devre Adı: Arduino Kilit Açma kodu

Devrenin tanımı: Arduino bağlı bir tuş takımını ayarlayan program. Doğru kodu yazdığınızda bir LED yanacaktır.



1-) Breadboard görünümü



2-) Şematik görünüm

```
/* Arduino Kilit Açma Kodu */
#include <Keypad.h>     // Arduino IDE ile indirebileceğimiz kütüphaneler.
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#define Solenoid 12          //Gerçekte,  solenoidi kontrol eden transistörün Geyti (Beyzi),
                             //Biz bu durumumda, basit bir LED kullandık.
#include <liquidcrystal_i2c.h></liquidcrystal_i2c.h></lcd.h></wire.h></keypad.h>
#define I2C_ADDR 0x27 // LCD i2c Adress and pins
#define BACKLIGHT_PIN 3
#define En_pin 2
#define Rw_pin 1
#define Rs_pin 0
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7
LiquidCrystal_I2C
lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);
const byte numRows= 4;        //Keypad üzerinde ki satırların  sayısı
const byte numCols= 4;        // Keypad üzerinde ki sütünların  sayısı
int code = 1234;              //Şifre buradaki
int tot,i1,i2,i3,i4;
char c1,c2,c3,c4;
//keymap, basılan tuşu satır ve sütunlara göre tıpkı ekranda göründüğü gibi tanımlar.
keypad char keymap[numRows][numCols]=
{
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};
//Arduino terminallerine keypad bağlantılarını gösteren kod
```

```
byte rowPins[numRows] = {9,8,7,6}; //Rows 0 to 3
byte colPins[numCols]= {5,4,3,2}; //Columns 0 to 3
//initializes an instance of the Keypad class
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins, numRows,
numCols);
void setup()
        {
         lcd.begin (16,2);
         lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
         lcd.setBacklight(HIGH);
         lcd.home ();
         lcd.print("ROMANIA DoorLock");
         lcd.setCursor(9, 1);
         lcd.print("Standby");
          pinMode(Solenoid,OUTPUT);
         delay(2000);
        }
void loop()
{
    char keypressed = myKeypad.getKey();  //The getKey fucntion keeps the program
runing, as long you didn't press "*" the whole thing bellow wouldn't be triggered
      if (keypressed == '*')          // and you can use the rest of you're code simply
         {
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("Enter Code");              //when the "*" key is pressed you can enter
the passcode
              keypressed = myKeypad.waitForKey();  // here all programs are stopped until
you enter the four digits then it gets compared to the code above
              if (keypressed != NO_KEY)
               {
                c1 = keypressed;
                lcd.setCursor(0, 1);
```

```
  lcd.print("*");
  }
keypressed = myKeypad.waitForKey();
if (keypressed != NO_KEY)
 {
  c2 = keypressed;
  lcd.setCursor(1, 1);
  lcd.print("*");
  }
 keypressed = myKeypad.waitForKey();
if (keypressed != NO_KEY)
 {
  c3 = keypressed;
  lcd.setCursor(2, 1);
  lcd.print("*");
  }
  keypressed = myKeypad.waitForKey();
if (keypressed != NO_KEY)
 {
  c4 = keypressed;
  lcd.setCursor(3, 1);
  lcd.print("*");
  }
 i1=(c1-48)*1000;        //Basılan tuşlar karakterlerde saklanır, sonra onlar  Int'e
```

dönüştürülür, daha sonra kodun Int'sini xxxx'i yapmak için çarpma yapılır.

```
i2=(c2-48)*100;
  i3=(c3-48)*10;
  i4=c4-48;
  tot=i1+i2+i3+i4;
 if (tot == code)          //Kod doğruysa, burada ne istersen onu tetiklersin, ekrana bir
```

mesaj yazdırman yeterlidir.

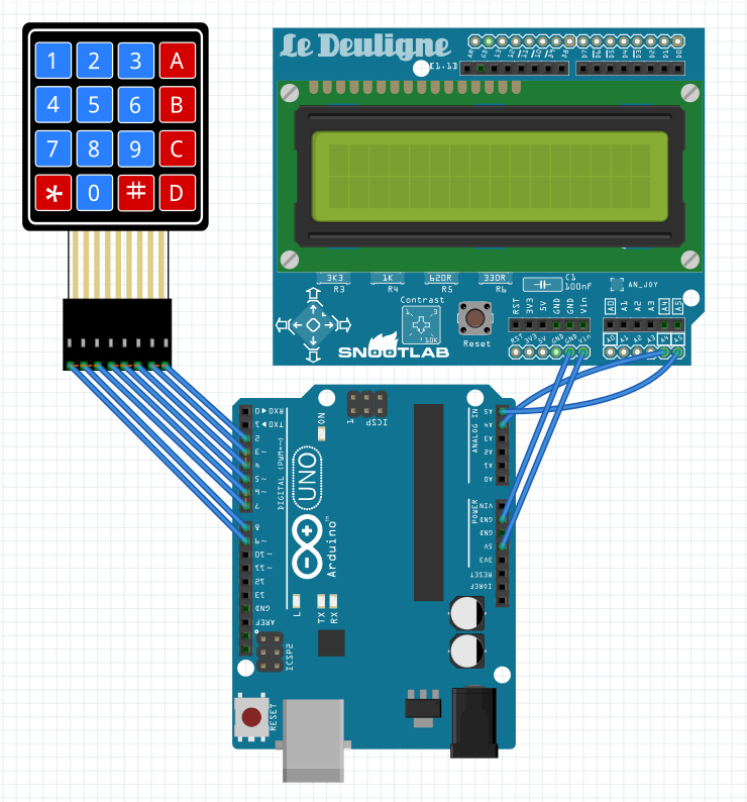
```
{
 lcd.clear();
```

```
        lcd.setCursor(0, 0);
        lcd.print("Welcome");
        digitalWrite(Solenoid,HIGH);
 delay(3000);
 digitalWrite(Solenoid,LOW);
        lcd.setCursor(7, 1);
        lcd.print("ROMANIA DoorLock");

        delay(3000);
        lcd.clear();
        lcd.print("ROMANIA DoorLock");
        lcd.setCursor(9, 1);
        lcd.print("Standby");

       }
       else          // Kod yanlışsa, başka bir şey alırsınız.
       {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("WRONG CODE");
        delay(3000);
        lcd.clear();
        lcd.print("ROMANIA DoorLock");
        lcd.setCursor(9, 1);
        lcd.print("Standby");
        }
          }
}
```

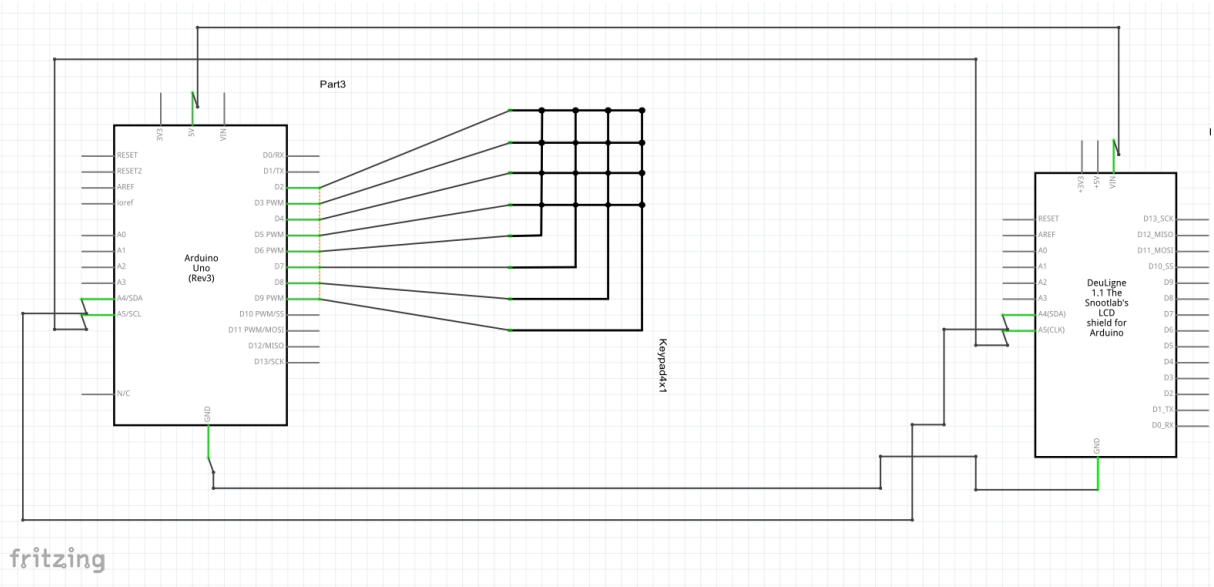
Devre 4:

Devre Adı: Arduino hesap makinası

Devrenin Tanımı: Program temel matematiksel hesaplamaları yapar.



1-) Breadboard görünümü



2-) Şematik görünüm

```
/* Arduino hesap makinası */
#include <Keypad.h>
#include <EEPROM.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#define I2C_ADDR 0x27                 // I2C adresi
#define BACKLIGHT_PIN 3               //  LCD Pinlerini tanımlama
#define En_pin 2
#define Rw_pin 1
#define Rs_pin 0
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7
const byte ROWS = 4;              // Dört satır
const byte COLS = 4;              // Dört sütün
// Define the Keymap
char keys[ROWS][COLS] = {
 {'1','2','3','A'},
 {'4','5','6','B'},
 {'7','8','9','C'},
 {'*','0','#','D'}
};
byte rowPins[ROWS] = {9,8,7,6};       //0-3  satırlar
byte colPins[COLS]= {5,4,3,2};          // 0-3 sütünlar
LiquidCrystal_I2C
lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);
Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
//  Keypadi oluşturma
long Num1,Num2,Number;
 char key,action;
 boolean result = false;
 void setup()
```

```
      {
       lcd.begin (16,2);
       lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
         lcd.setBacklight(HIGH);                //Arka ışığı yakma
      lcd.print("Calculator Ready");          // Ekran giriş mesajı
 lcd.setCursor(0, 1);                         // İmleci 0. kolon, 1. satıra ayarlama
 lcd.print("A=+ B=- C=* D=/");                //Giriş mesajını gösterme
  delay(6000);                                // Ekranın bilgileri göstermesini bekle
   lcd.clear();                               //Sonra mesajı sil.
                //       for(i=0 ; i<sizeof(code);i++){       // Kodu ilk kez yüklediğinizde, yoruma
devam edin
//          EEPROM.get(i, code[i]);        //Kodu yükleyin ve EEPROM'da saklamak için
değiştirin
//          }       // Ardından bunu döngü için kaldırın ve kodu yeniden yükleyin (Yalnızca bir kez
yapılır)
      }
void loop() {
  key = kpd.getKey();                 //Basılan tuş değerini bir karakterde saklamak
if (key!=NO_KEY)
DetectButtons();
if (result==true)
CalculateResult();
DisplayResult();
}
void DetectButtons()
{
    lcd.clear();                        //Sonra sil
   if (key=='*')                        //İptal Butonuna basılırsa,
   {Serial.println ("Button Cancel"); Number=Num1=Num2=0; result=false;}
      if (key == '1')                   //If Buton-1'e basılırsa
   {Serial.println ("Button 1");
   if (Number==0)
   Number=1;
```

```
else
Number = (Number*10) + 1;            //2 defa bas
}
    if (key == '4')                  //Şayet  Buton-4'e  basılırsa
{Serial.println ("Button 4");
if (Number==0)
Number=4;
else
Number = (Number*10) + 4;            //2 defa bas
}
    if (key == '7')                  //Şayet Buton-7'e  basılırsa
{Serial.println ("Button 7");
if (Number==0)
Number=7;
else
Number = (Number*10) + 7;            //2 defa bas
}
  if (key == '0')
{Serial.println ("Button 0");        //Buton-0'a basılırsa
if (Number==0)
Number=0;
else
Number = (Number*10) + 0;            //2 defa bas
}
    if (key == '2') //Button 2 is Pressed
{Serial.println ("Button 2");
 if (Number==0)
Number=2;
else
Number = (Number*10) + 2;            //2 defa bas
}
    if (key == '5')
{Serial.println ("Button 5");
```

```
 if (Number==0)
Number=5;
else
Number = (Number*10) + 5;                    //2 defa bas
}
    if (key == '8')
{Serial.println ("Button 8");
 if (Number==0)
Number=8;
else
Number = (Number*10) + 8;                    //2 defa bas
}
  if (key == '#')
{Serial.println ("Button Equal");
Num2=Number;
result = true;
}
    if (key == '3')
{Serial.println ("Button 3");
 if (Number==0)
Number=3;
else
Number = (Number*10) + 3;                          //2 defa bas
}
    if (key == '6')
{Serial.println ("Button 6");
if (Number==0)
Number=6;
else
Number = (Number*10) + 6;                          //Pressed twice
}
    if (key == '9')
{Serial.println ("Button 9");
```

```
  if (Number==0)
  Number=9;
  else
  Number = (Number*10) + 9; //Pressed twice
  }
    if (key == 'A' || key == 'B' || key == 'C' || key == 'D')     // Sütun 4'teki butonları algılama
 {
  Num1 = Number;
  Number =0;
  if (key == 'A')
  {Serial.println ("Addition"); action = '+';}
   if (key == 'B')
  {Serial.println ("Subtraction"); action = '-'; }
   if (key == 'C')
  {Serial.println ("Multiplication"); action = '*';}
   if (key == 'D')
  {Serial.println ("Division"); action = '/';}
  delay(100);
 }
}
void CalculateResult()
{
 if (action=='+')
  Number = Num1+Num2;
 if (action=='-')
  Number = Num1-Num2;
 if (action=='*')
  Number = Num1*Num2;
 if (action=='/')
  Number = Num1/Num2;
}
void DisplayResult()
{
```

```
 lcd.setCursor(0, 0);                  // İmleci 0. sütün, 0. satıra ayarla
 lcd.print(Num1); lcd.print(action); lcd.print(Num2);
   if (result==true)
 {lcd.print(" ="); lcd.print(Number);}          // Sonucu göster
   lcd.setCursor(0, 1);                        // İmleci 0. sütün, 1. satıra ayarla
 lcd.print(Number);                            // Sonucu göster
}
```

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+ KA-202

Mesleki Eğitim ve Öğretime Yönelik Stratejik Ortaklıklar Projesi

Proje Başlığı : "Mesleki Eğitimde Arduino'ları Öğretme ve Öğrenme"

Proje Kısaltması: " ARDUinVET "

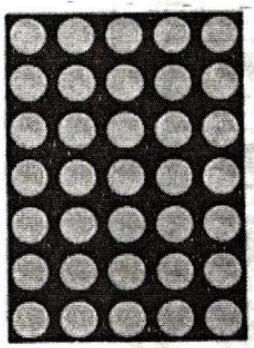
Proje No: "2020-1-TR01-KA202-093762"

Dot Matrix Modulü and Eğitim Kiti

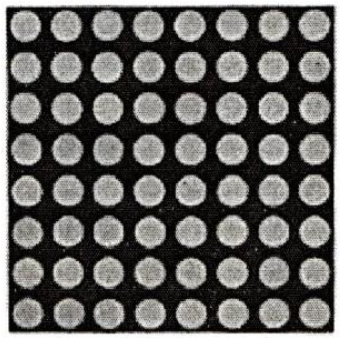
DotMatrix Display Nedir?

DotMatrix display is a group of LEDs arranged in a specific system. A standard LED array is made up of 5x7 or 8x8 LEDs arranged in rows and columns as shown in the figure below. A 5*7 DOT matrix have 5 rows and 7 columns of LEDs and an 8*8 DOT matrix have 8 rows and 8 columns of LEDs which are connecting together.

DotMatrix display, belirli bir sistemle düzenlenmiş bir grup LED'dir. Standart bir LED dizisi, aşağıdaki şekilde gösterildiği gibi sıralar ve sütunlar halinde düzenlenmiş 5x7 veya 8x8 LED'lerden oluşur. 5*7 DOTmatrix'te 5 sıra ve 7 sütun LED vardır ve 8*8 DOTmatrix'te birbirine bağlanan 8 sıra ve 8 sütun LED vardır.



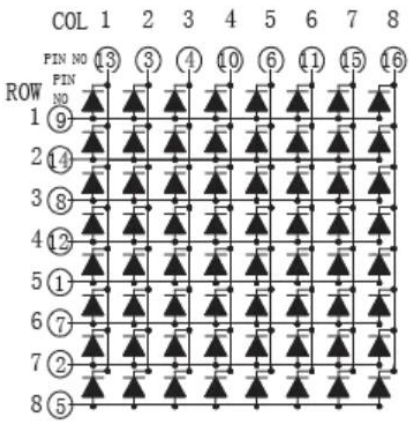
5*7 DOT Matrix Display

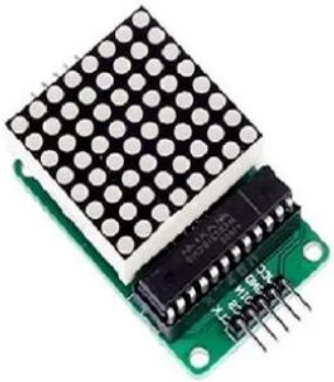


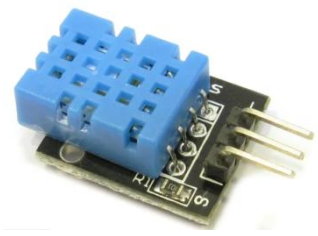
8*8 DOT Matrix Display

8x8 Dotmatrisinin bir parçasına bakacak olursak, 8'i satırlar, 8'i sütunlar için kullanılan 16 pin içerir. Bu, satırlarda ve sütunlarda toplam 64 LED anlamına gelir. Pin # 1'den pin # 8'e sıralamaya başlanır. 1 numaralı pin R5 (Satır-5) ve 8 numaralı Pin R3 (Satır-3) olarak aşağı yönde sıralanır.

Üst tarafta Pin9'dan (Satır-1) Pin16'ya (sütun-1) kadar pinler bulunur. Ancak bu yeni başlayanlar için her zaman kafa karıştırıcıdır çünkü onlar saymaya sıfırdan başlar. Pin bilgilerini çoğu zaman bir kaynaklardan buluruz. Ayrıca ortak katot/anot türünden +VE ve -VE hangi tip Dotmatrixin kullanıldığını da belirlememiz gerekir.



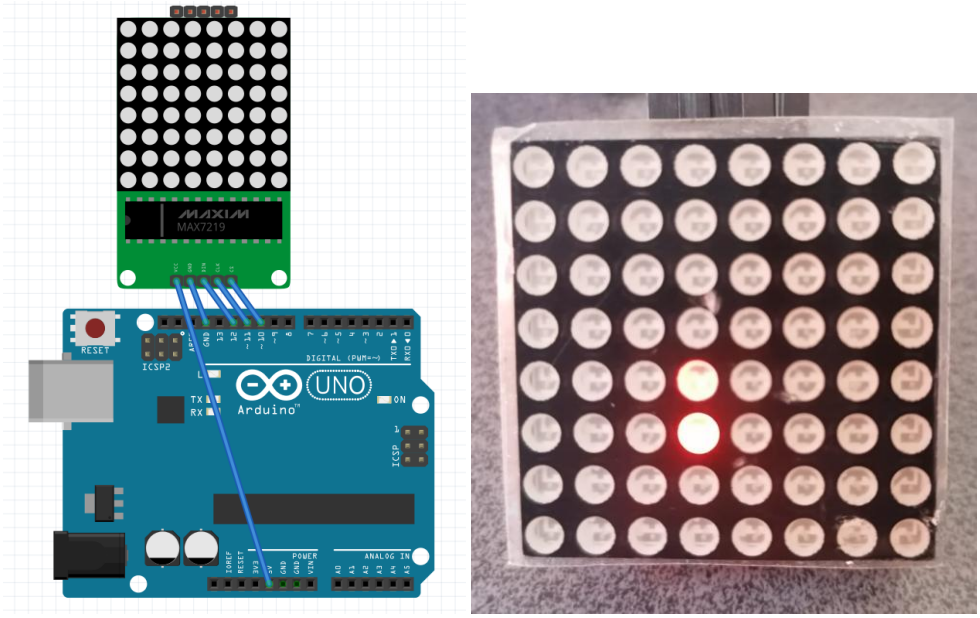
MAX7219 LED Domatrix display piyasada bulunan ve öğrenciler, elektronik meraklıları ve özellikle endüstriyel display uygulamalarında kullanılan popüler displaylerden biridir. Genel olarak kullanılabilen iki tür modülü vardır. Bunlar genel modüldür. Her modül iki üniteden oluşmaktadır. Biri 8X8 LED domatrix, diğeri ise MAX7219 IC. MAX7219, seri girişlere ve çıkış pinlerine sahip ortak katotlu bir ekran sürücüsü IC'dir. Sadece bir harici direnç kullanılarak ayarlanabilir akım özelliğine sahiptir. Ayrıca tüm mikroişlemcilere kolayca bağlanabilen dört telli (yollu) seri arayüze sahiptir. Arduino kullanarak sadece 4 kablo kullanarak çıkış pinlerine bağlı 64 ayrı LED'i çalıştırabilir. Ayrıca, DotMatrix displayleri, 7 Segment displayleri ve çubuk grafikleri (bar graph) çalıştırabilir. Bunun da ötesinde, MAX7219, yedi segmentli sayısal display kullanımı kolaylaştıran bir yerleşik BCD kod çözücüye sahiptir. Ek olarak, sayıları saklamak için kullanabileceğimiz 8×8 statik RAM'e sahiptir. En popüler ekran sürücüsü IC'lerinden biridir.	

	DHT11, bir dijital sıcaklık ve nem sensörüdür. Çevredeki havayı ölçmek için kapasitif bir nem sensörü ve bir termistör kullanır ve veri pinine dijital bir sinyal gönderir (analog giriş pinlerine gerek yoktur).

Devre 1:

Devre Adı: Tüm LED'leri tek tek görüntüleme

Devrenin açıklaması: Dot Matrix, tüm LED'leri tek tek gösterir.



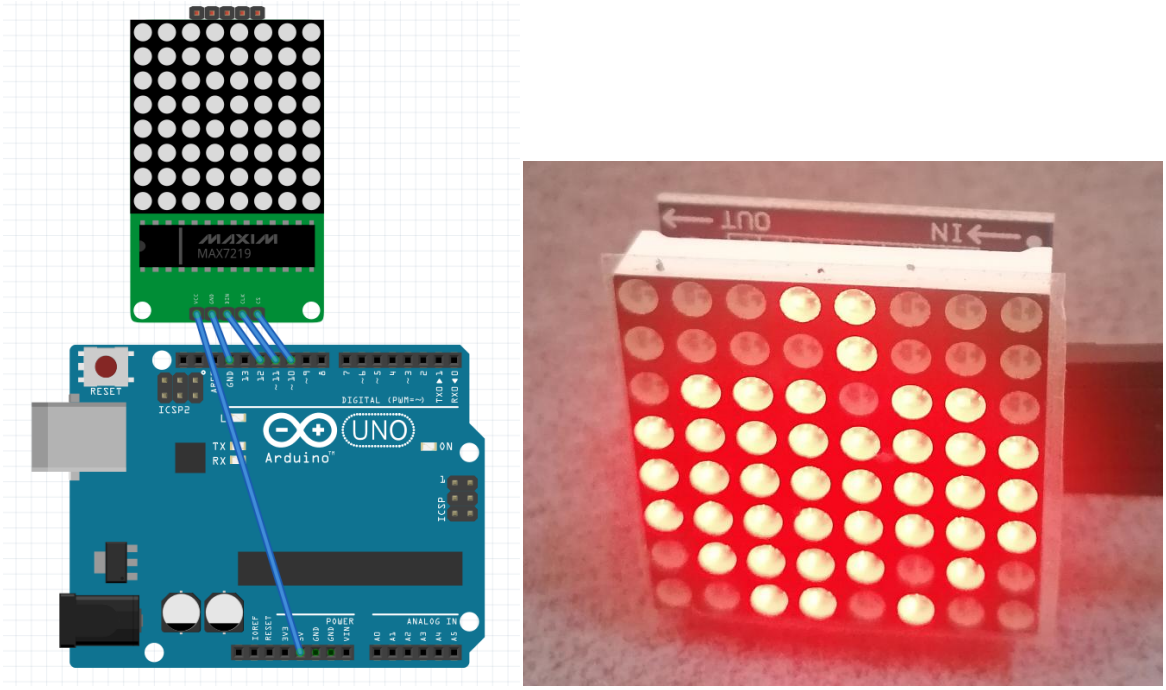
1-) Breadboard görünümü

```
/*    Tüm LED'leri tek tek görüntüleme */
#include <LedControl.h>
int DIN = 12;
int CS = 10;
int CLK = 11;
LedControl lc=LedControl(DIN, CLK, CS,0);
void setup() {
  lc.shutdown(0,false);
  lc.setIntensity(0,0);
  lc.clearDisplay(0);}
void loop() {
  for(int j=0;j<8;j++){
     for(int i=0;i<8;i++){
       lc.setLed(0,j,i,true);
       delay(100);
       lc.setLed(0,j,i,false); } }}
```

Devre 2:

Devre Adı: Apple iconunu (simgesini) gösterme

Devrenin açıklaması: Dot Matrix üzerinde, Apple simgesi görüntülenir.



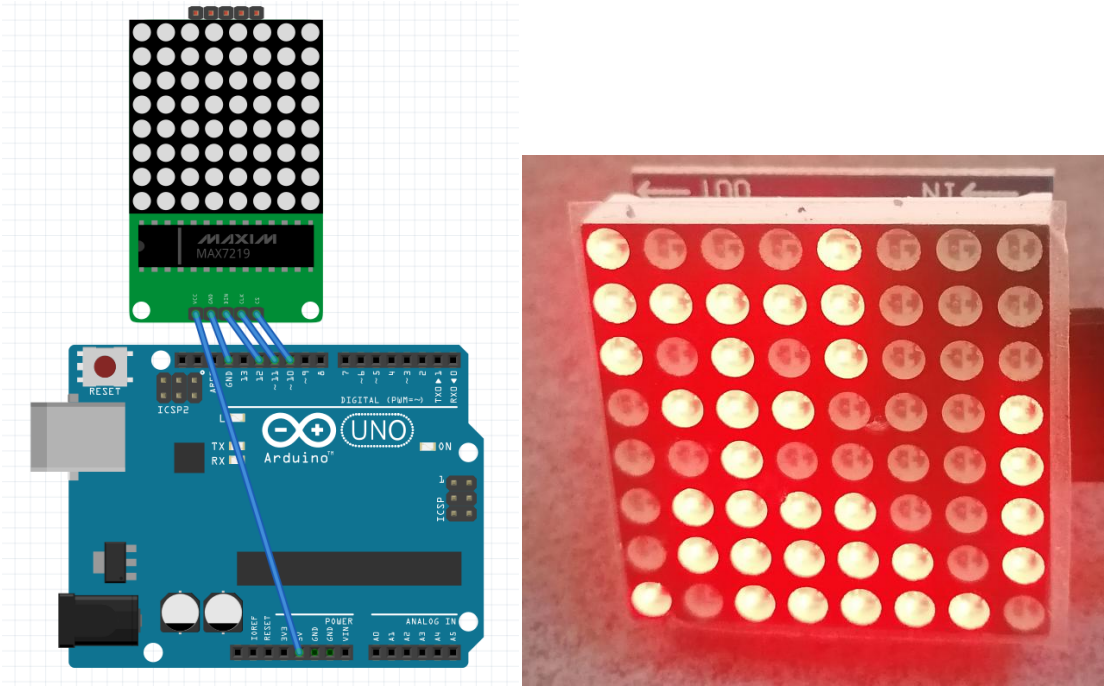
1-) Breadboard görünümü

```
/*    Apple iconunu, simgesini gösterme*/
#include <LedControl.h>
int DIN = 12;
int CS = 10;
int CLK = 11;
LedControl lc=LedControl(DIN, CLK, CS,0);
byte Apple
[8]={B00011000,B00001000,B01110110,B11111111,B11111111,B11111111,B01111010,B001
10100};
void setup() {
  lc.shutdown(0,false);
  lc.setIntensity(0,0);
  lc.clearDisplay(0);  }
  void loop(){
  for(int i=0;i<8;i++) lc.setRow(0,i,Apple[i]);   }
```

Devre 3:

Devre Adı : Bir kedi simgesini gösterme

Devrenin açıklaması: Dot Matrix arayüzü bir kedi simgesini gösterir.



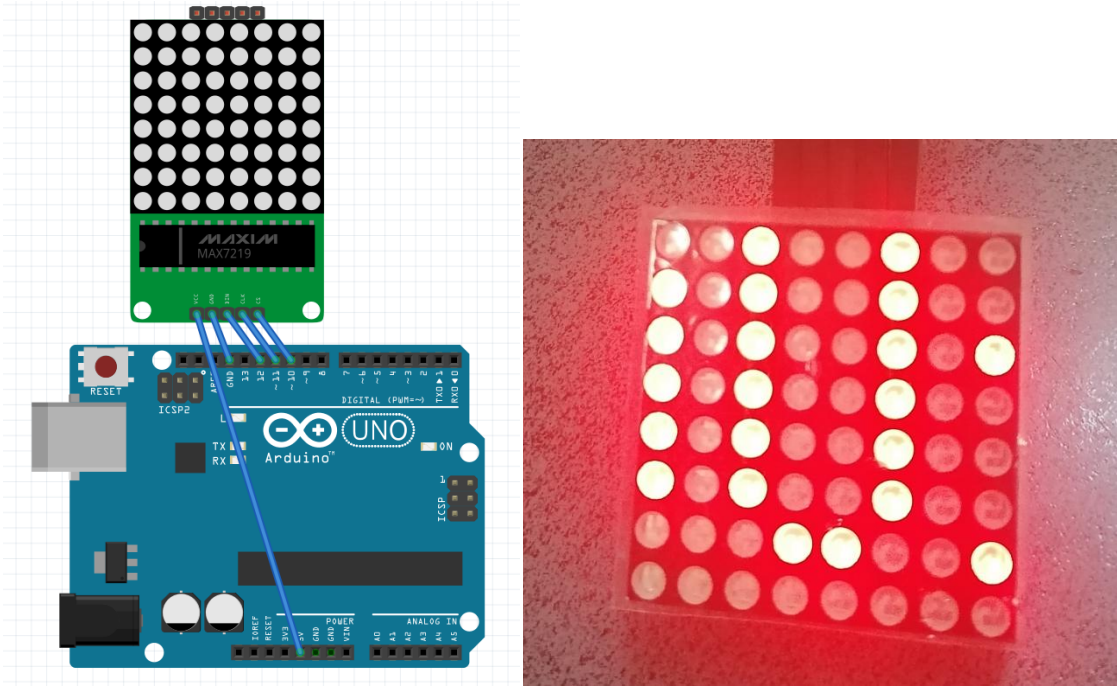
1-) Breadboard görünümü

```
/*  Display a cat icon*/
#include <LedControl.h>
int DIN = 12;
int CS = 10;
int CLK = 11;
LedControl lc=LedControl(DIN, CLK, CS,0);
int Cat[8]
={B10001000,B11111000,B10101000,B01110001,B00100001,B01111001,B01111101,B10111
110 };
void setup() {
  lc.shutdown(0,false);
  lc.setIntensity(0,0);
  lc.clearDisplay(0);  }
  void loop(){
    for(int i=0;i<8;i++) lc.setRow(0,i,Cat[i]); }
```

Devre 4:

Devre Adı: Kayan ARDUINVET metnini gösterme

Devrenin açıklaması: Dot Matrix arayüzünde, kayan ARDUNVET metni görüntülenir.



1-) Breadboard görünümü

```
/*  Kayan ARDUINVET metnini  gösterme   */
//D10=CS
//D11=CLK
//D12=DIN
#include <MaxMatrix.h>              // Matrix Kütüphanesi yükle
#include <avr/pgmspace.h>
#include <stdlib.h>
PROGMEM const unsigned char CH[] = {
3, 8, B00000000, B00000000, B00000000, B00000000, B00000000, // space
1, 8, B01011111, B00000000, B00000000, B00000000, B00000000, // !
3, 8, B00000011, B00000000, B00000011, B00000000, B00000000, // "
5, 8, B00010100, B00111110, B00010100, B00111110, B00010100, // #
4, 8, B00100100, B01101010, B00101011, B00010010, B00000000, // $
5, 8, B01100011, B00010011, B00001000, B01100100, B01100011, // %
5, 8, B00110110, B01001001, B01010110, B00100000, B01010000, // &
```

```
1, 8, B00000011, B00000000, B00000000, B00000000, B00000000, // '
3, 8, B00011100, B00100010, B01000001, B00000000, B00000000, // (
3, 8, B01000001, B00100010, B00011100, B00000000, B00000000, // )
5, 8, B00101000, B00011000, B00001110, B00011000, B00101000, // *
5, 8, B00001000, B00001000, B00111110, B00001000, B00001000, // +
2, 8, B10110000, B01110000, B00000000, B00000000, B00000000, // ,
4, 8, B00001000, B00001000, B00001000, B00001000, B00000000, // -
2, 8, B01100000, B01100000, B00000000, B00000000, B00000000, // .
4, 8, B01100000, B00011000, B00000110, B00000001, B00000000, // /
4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // 0
3, 8, B01000010, B01111111, B01000000, B00000000, B00000000, // 1
4, 8, B01100010, B01010001, B01001001, B01000110, B00000000, // 2
4, 8, B00100010, B01000001, B01001001, B00110110, B00000000, // 3
4, 8, B00011000, B00010100, B00010010, B01111111, B00000000, // 4
4, 8, B00100111, B01000101, B01000101, B00111001, B00000000, // 5
4, 8, B00111110, B01001001, B01001001, B00110000, B00000000, // 6
4, 8, B01100001, B00010001, B00001001, B00000111, B00000000, // 7
4, 8, B00110110, B01001001, B01001001, B00110110, B00000000, // 8
4, 8, B00000110, B01001001, B01001001, B00111110, B00000000, // 9
2, 8, B01010000, B00000000, B00000000, B00000000, B00000000, // :
2, 8, B10000000, B01010000, B00000000, B00000000, B00000000, // ;
3, 8, B00010000, B00101000, B01000100, B00000000, B00000000, // <
3, 8, B00010100, B00010100, B00010100, B00000000, B00000000, // =
3, 8, B01000100, B00101000, B00010000, B00000000, B00000000, // >
4, 8, B00000010, B01011001, B00001001, B00000110, B00000000, // ?
5, 8, B00111110, B01001001, B01010101, B01011101, B00001110, // @
4, 8, B01111110, B00010001, B00010001, B01111110, B00000000, // A
4, 8, B01111111, B01001001, B01001001, B00110110, B00000000, // B
4, 8, B00111110, B01000001, B01000001, B00100010, B00000000, // C
4, 8, B01111111, B01000001, B01000001, B00111110, B00000000, // D
4, 8, B01111111, B01001001, B01001001, B01000001, B00000000, // E
4, 8, B01111111, B00001001, B00001001, B00000001, B00000000, // F
4, 8, B00111110, B01000001, B01001001, B01111010, B00000000, // G
```

4, 8, B01111111, B00001000, B00001000, B01111111, B00000000, // H

3, 8, B01000001, B01111111, B01000001, B00000000, B00000000, // I

4, 8, B00110000, B01000000, B01000001, B00111111, B00000000, // J

4, 8, B01111111, B00001000, B00010100, B01100011, B00000000, // K

4, 8, B01111111, B01000000, B01000000, B01000000, B00000000, // L

5, 8, B01111111, B00000010, B00001100, B00000010, B01111111, // M

5, 8, B01111111, B00000100, B00001000, B00010000, B01111111, // N

4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // O

4, 8, B01111111, B00001001, B00001001, B00000110, B00000000, // P

4, 8, B00111110, B01000001, B01000001, B10111110, B00000000, // Q

4, 8, B01111111, B00001001, B00001001, B01110110, B00000000, // R

4, 8, B01000110, B01001001, B01001001, B00110010, B00000000, // S

5, 8, B00000001, B00000001, B01111111, B00000001, B00000001, // T

4, 8, B00111111, B01000000, B01000000, B00111111, B00000000, // U

5, 8, B00001111, B00110000, B01000000, B00110000, B00001111, // V

5, 8, B00111111, B01000000, B00111000, B01000000, B00111111, // W

5, 8, B01100011, B00010100, B00001000, B00010100, B01100011, // X

5, 8, B00000111, B00001000, B01110000, B00001000, B00000111, // Y

4, 8, B01100001, B01010001, B01001001, B01000111, B00000000, // Z

2, 8, B01111111, B01000001, B00000000, B00000000, B00000000, // [

4, 8, B00000001, B00000110, B00011000, B01100000, B00000000, // \ backsplash

2, 8, B01000001, B01111111, B00000000, B00000000, B00000000, //]

3, 8, B00000010, B00000001, B00000010, B00000000, B00000000, // hat

4, 8, B01000000, B01000000, B01000000, B01000000, B00000000, // _

2, 8, B00000001, B00000010, B00000000, B00000000, B00000000, // `

4, 8, B00100000, B01010100, B01010100, B01111000, B00000000, // a

4, 8, B01111111, B01000100, B01000100, B00111000, B00000000, // b

4, 8, B00111000, B01000100, B01000100, B00101000, B00000000, // c

4, 8, B00111000, B01000100, B01000100, B01111111, B00000000, // d

4, 8, B00111000, B01010100, B01010100, B00011000, B00000000, // e

3, 8, B00000100, B01111110, B00000101, B00000000, B00000000, // f

4, 8, B10011000, B10100100, B10100100, B01111000, B00000000, // g

4, 8, B01111111, B00000100, B00000100, B01111000, B00000000, // h

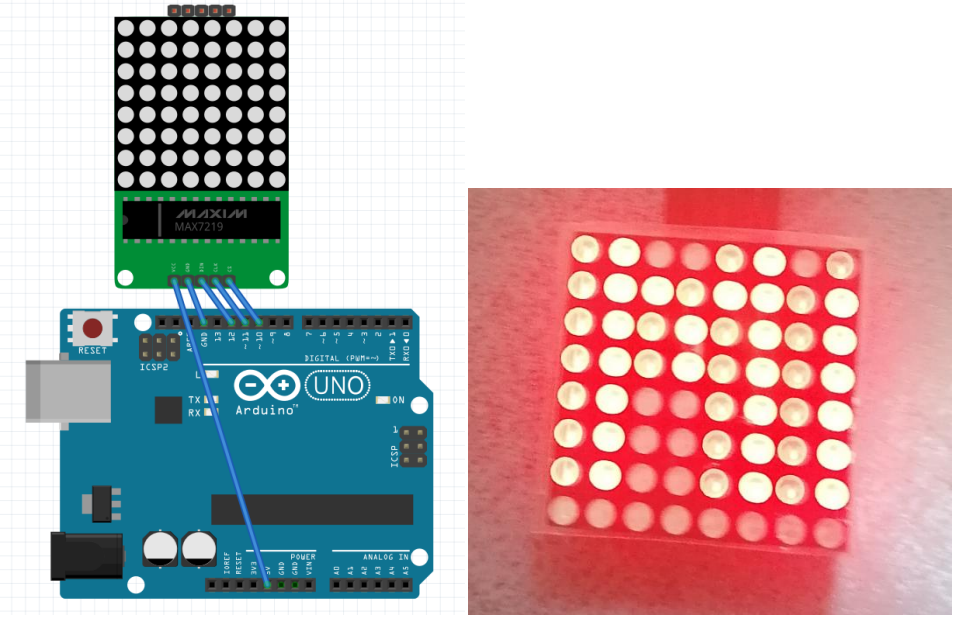
```
3, 8, B01000100, B01111101, B01000000, B00000000, B00000000, // i
4, 8, B01000000, B10000000, B10000100, B01111101, B00000000, // j
4, 8, B01111111, B00010000, B00101000, B01000100, B00000000, // k
3, 8, B01000001, B01111111, B01000000, B00000000, B00000000, // l
5, 8, B01111100, B00000100, B01111100, B00000100, B01111000, // m
4, 8, B01111100, B00000100, B00000100, B01111000, B00000000, // n
4, 8, B00111000, B01000100, B01000100, B00111000, B00000000, // o
4, 8, B11111100, B00100100, B00100100, B00011000, B00000000, // p
4, 8, B00011000, B00100100, B00100100, B11111100, B00000000, // q
4, 8, B01111100, B00001000, B00000100, B00000100, B00000000, // r
4, 8, B01001000, B01010100, B01010100, B00100100, B00000000, // s
3, 8, B00000100, B00111111, B01000100, B00000000, B00000000, // t
4, 8, B00111100, B01000000, B01000000, B01111100, B00000000, // u
5, 8, B00011100, B00100000, B01000000, B00100000, B00011100, // v
5, 8, B00111100, B01000000, B00111100, B01000000, B00111100, // w
5, 8, B01000100, B00101000, B00010000, B00101000, B01000100, // x
4, 8, B10011100, B10100000, B10100000, B01111100, B00000000, // y
3, 8, B01100100, B01010100, B01001100, B00000000, B00000000, // z
3, 8, B00001000, B00110110, B01000001, B00000000, B00000000, // {
1, 8, B01111111, B00000000, B00000000, B00000000, B00000000, // |
3, 8, B01000001, B00110110, B00001000, B00000000, B00000000, // }
4, 8, B00001000, B00000100, B00001000, B00000100, B00000000, // ~
};
int data = 12;            // MAX7219 modulünün  DIN pini
int load = 10;            // MAX7219 modulünün  CS pini
int clock = 11;           // MAX7219 modulünün CLK pini
int maxInUse = 1;     // Kaç tane MAX7219 kullanacağınızı ayarlamak için bu değişkeni kullan
MaxMatrix m(data, load, clock, maxInUse);      // Modulü tanımla
byte buffer[10];
void setup(){
  m.init();                        // Modul başlatılıyor
  m.setIntensity(2);               // Dot matix yoğunluğu;  0-15
  Serial.begin(9600);              // Seri iletişim başlatılıyor
```

```
  Serial.println("ARDUinVET");  }
void loop(){
     printStringWithShift("ARDUinVET    ", 100);
   m.shiftLeft(false, true);  }
void printCharWithShift(char c, int shift_speed){
  if (c < 32) return;
  c -= 32;
  memcpy_P(buffer, CH + 7*c, 7);
  m.writeSprite(32, 0, buffer);
  m.setColumn(32 + buffer[0], 0);
  for (int i=0; i<buffer[0]+1; i++)
  {    delay(shift_speed);
    m.shiftLeft(false, false);  }
}
void printStringWithShift(char* s, int shift_speed){
  while (*s != 0){
    printCharWithShift(*s, shift_speed);
    s++;  }
}
void printString(char* s)
{  int col = 0;
  while (*s != 0)
  {    if (*s < 32) continue;
    char c = *s - 32;
    memcpy_P(buffer, CH + 7*c, 7);
    m.writeSprite(col, 0, buffer);
    m.setColumn(col + buffer[0], 0);
    col += buffer[0] + 1;
    s++;  }        }
```

Devre 5:

Devre Adı: Alfabenin her harfini gösterme

Devrenin açıklaması: Dot Matrix displayde, bir saniye aralıklarla alfabenin her harfi görüntülenir.



1-) Breadboard görünümü

/* Alfabenin her harfini gösterme */

#include <MaxMatrix.h> // Buradan indir: https://code.google.com/archive/p/arudino-maxmatrix-library/downloads

int DIN = 12; // MAX7219 modulünün DIN pini

int CLK = 11; // MAX7219 modulünün CLK pini

int CS = 10; // CS pin of MAX7219 module

int maxInUse = 1;

MaxMatrix m(DIN, CS, CLK, maxInUse);

char A[] = {8, 8,

B00111000,B01000100,B01000100,B01000100,B01111100,B01000100,B01000100,B01000100};

char B[] = {8, 8,

B01111000,B01000100,B01000100,B01111000,B01000100,B01000100,B01111000,B00000000};

char c[] = {8, 8,

```
B00000000,B00111100,B01000000,B01000000,B01000000,B01000000,B00111100,B00000000};
char d[] = {8, 8,
B00000000,B01111000,B01000100,B01000100,B01000100,B01000100,B01111000,B00000000};
char e[] = {8, 8,
B00000000,B01111100,B01000000,B01000000,B01111100,B01000000,B01000000,B01111100};
char f[] = {8, 8,
B00000000,B01111100,B01000000,B01000000,B01111100,B01000000,B01000000,B01000000};
char g[] = {8, 8,
B00000000,B00111100,B01000000,B01000000,B01001110,B01000010,B00111110};
char h[] = {8, 8,
B00000000,B01000100,B01000100,B01111100,B01000100,B01000100,B01000100,B00000000};
char i[] = {8, 8,
B00000000,B01111100,B00010000,B00010000,B00010000,B00010000,B01111100,B00000000};
char j[] = {8, 8,
B00000000,B00000100,B00000100,B00000100,B00000100,B01000100,B00111000,B00000000};
char k[] = {8, 8,
B00000000,B00100100,B00101000,B00110000,B00101000,B00100100,B00100010,B00000000};
char l[] = {8, 8,
B00000000,B01000000,B01000000,B01000000,B01000000,B01000000,B01111100,B00000000};
char n[] = {8, 8,
B00000000,B01000010,B01100010,B01010010,B01001010,B01000110,B01000010,B00000000};
char o[] = {8, 8,
```

```
B00111100,B01000010,B01000010,B01000010,B01000010,B01000010,B01000010,B0011110
0};
char p[] = {8, 8,
B00000000,B00111000,B00100100,B00100100,B00111000,B00100000,B00100000,B0000000
0};
char q[] = {8, 8,
B00111100,B01000010,B01000010,B01000010,B01001010,B01000110,B00111110,B0000000
1};
char r[] = {8, 8,
B01111000,B01000100,B01000100,B01111000,B01100000,B01010000,B01001000,B0100010
0};
char s[] = {8, 8,
B00111100,B01000000,B01000000,B00111000,B00000100,B00000100,B01111000,B0000000
0};
char t[] = {8, 8,
B00000000,B01111100,B00010000,B00010000,B00010000,B00010000,B00010000,B0000000
0};
char u[] = {8, 8,
B00000000,B01000010,B01000010,B01000010,B01000010,B01000010,B00111100,B0000000
0};
char v[] = {8, 8,
B00000000,B00100100,B00100100,B00100100,B00100100,B00100100,B00011000,B0000000
0};
char w[] = {8, 8,
B00000000,B01010100,B01010100,B01010100,B01010100,B01010100,B00111000,B0000000
0};
char x[] = {8, 8,
B00000000,B01000100,B00101000,B00010000,B00101000,B01000100,B00000000,B0000000
0};
char y[] = {8, 8,
B10000001,B01000010,B00100100,B00011000,B00011000,B00011000,B00011000,B0001100
0};
char z[] = {8, 8,
```

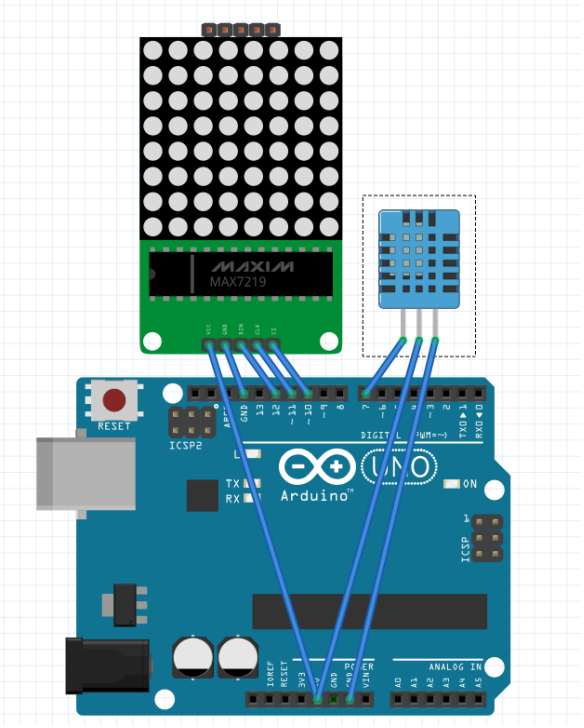
```
B00000000,B01111100,B00001000,B00010000,B00100000,B01111100,B00000000,B0000000
0};void setup() {
  m.init();                    // MAX7219' başlatma
  m.setIntensity(5);           // Başlangıç   led matrix yoğunluğu, 0-15
}  void loop() {
  // x,y veya satır,kolon pozisyonunda LED'lerin Açık veya Kapalı olarak ayarlanması
m.setDot(6,2,true);
  delay(1000);
  m.setDot(6,3,true);
  delay(1000);
  m.clear(); // Clears the display
  for (int i=0; i<8; i++){
    m.setDot(i,i,true);
    delay(300);}
  m.clear();
          // Karakteri x,y'de görüntüleme (karakterin sol üst köşesi)
  m.writeSprite(0, 0, A);
  delay(1000);
  m.writeSprite(0, 0, B);
  delay(1000);
  m.writeSprite(0, 0, c);
  delay(1000);
  m.writeSprite(0, 0, d);
  delay(1000);
  m.writeSprite(0, 0, e);
  delay(1000);
  m.writeSprite(0, 0, f);
  delay(1000);
m.writeSprite(0, 0, g);
  delay(1000);
m.writeSprite(0, 0, h);
 delay(1000);
m.writeSprite(0, 0, i);
```

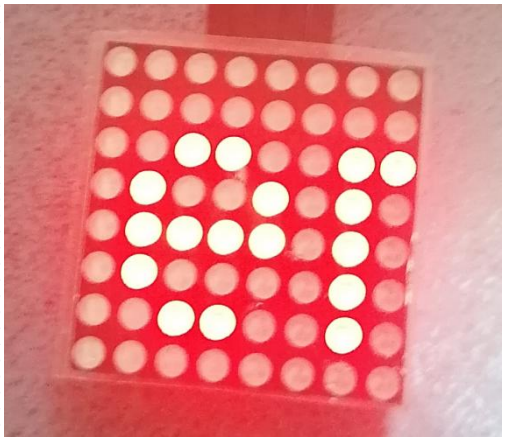
```
 delay(1000);
m.writeSprite(0, 0, j);
 delay(1000);
m.writeSprite(0, 0, k);
 delay(1000);
 m.writeSprite(0, 0, l);
 delay(1000);
m.writeSprite(0, 0, n);
 delay(1000);
m.writeSprite(0, 0, o);
 delay(1000);
  m.writeSprite(0, 0, p);
 delay(1000);
  m.writeSprite(0, 0, q);
 delay(1000);
  m.writeSprite(0, 0, r);
 delay(1000);
  m.writeSprite(0, 0, s);
 delay(1000);
  m.writeSprite(0, 0, t);
 delay(1000);
  m.writeSprite(0, 0, u);
 delay(1000);
  m.writeSprite(0, 0, v);
 delay(1000);
  m.writeSprite(0, 0, w);
 delay(1000);
  m.writeSprite(0, 0, x);
 delay(1000);
m.writeSprite(0, 0, y);
 delay(1000);
m.writeSprite(0, 0, z);
 delay(1000);}
```

Devre 6:

Devre Adı: Arduino ile Dot Matrix Display

Devrenin açıklaması: Dot Matrix Display ekranında, DHT11 sıcaklık sensörü kullanılarak ölçülen sıcaklık görüntülenir.





1-) Breadboard görünümü

```
/*    Arduino ile  Dot Matrix Display    */

//D7= temp
//D10=CS
//D11=CLK
//D12=DIN
#include <MaxMatrix.h>              // Matrix kütüphanesini dahil et, al
#include <avr/pgmspace.h>
#include <stdlib.h>
#include "DHT.h"                    //Temp sensor kütüphanesini dahil et, al
#define DHTPIN 7                    // Hangi pine bağlıyoruz
#define DHTTYPE DHT11               // DHT11 sıcaklık nem sensörü
DHT dht(DHTPIN, DHTTYPE);
PROGMEM const unsigned char CH[] = {
```

3, 8, B00000000, B00000000, B00000000, B00000000, B00000000, // space (boşluk)

1, 8, B01011111, B00000000, B00000000, B00000000, B00000000, // !

3, 8, B00000011, B00000000, B00000011, B00000000, B00000000, // "

5, 8, B00010100, B00111110, B00010100, B00111110, B00010100, // #

4, 8, B00100100, B01101010, B00101011, B00010010, B00000000, // $

5, 8, B01100011, B00010011, B00001000, B01100100, B01100011, // %

5, 8, B00110110, B01001001, B01010110, B00100000, B01010000, // &

1, 8, B00000011, B00000000, B00000000, B00000000, B00000000, // '

3, 8, B00011100, B00100010, B01000001, B00000000, B00000000, // (

3, 8, B01000001, B00100010, B00011100, B00000000, B00000000, //)

5, 8, B00101000, B00011000, B00001110, B00011000, B00101000, // *

5, 8, B00001000, B00001000, B00111110, B00001000, B00001000, // +

2, 8, B10110000, B01110000, B00000000, B00000000, B00000000, // ,

4, 8, B00001000, B00001000, B00001000, B00001000, B00000000, // -

2, 8, B01100000, B01100000, B00000000, B00000000, B00000000, // .

4, 8, B01100000, B00011000, B00000110, B00000001, B00000000, // /

4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // 0

3, 8, B01000010, B01111111, B01000000, B00000000, B00000000, // 1

4, 8, B01100010, B01010001, B01001001, B01000110, B00000000, // 2

4, 8, B00100010, B01000001, B01001001, B00110110, B00000000, // 3

4, 8, B00011000, B00010100, B00010010, B01111111, B00000000, // 4

4, 8, B00100111, B01000101, B01000101, B00111001, B00000000, // 5

4, 8, B00111110, B01001001, B01001001, B00110000, B00000000, // 6

4, 8, B01100001, B00010001, B00001001, B00000111, B00000000, // 7

4, 8, B00110110, B01001001, B01001001, B00110110, B00000000, // 8

4, 8, B00000110, B01001001, B01001001, B00111110, B00000000, // 9

2, 8, B01010000, B00000000, B00000000, B00000000, B00000000, // :

2, 8, B10000000, B01010000, B00000000, B00000000, B00000000, // ;

3, 8, B00010000, B00101000, B01000100, B00000000, B00000000, // <

3, 8, B00010100, B00010100, B00010100, B00000000, B00000000, // =

3, 8, B01000100, B00101000, B00010000, B00000000, B00000000, // >

4, 8, B00000010, B01011001, B00001001, B00000110, B00000000, // ?

5, 8, B00111110, B01001001, B01010101, B01011101, B00001110, // @

4, 8, B01111110, B00010001, B00010001, B01111110, B00000000, // A

4, 8, B01111111, B01001001, B01001001, B00110110, B00000000, // B

4, 8, B00111110, B01000001, B01000001, B00100010, B00000000, // C

4, 8, B01111111, B01000001, B01000001, B00111110, B00000000, // D

4, 8, B01111111, B01001001, B01001001, B01000001, B00000000, // E

4, 8, B01111111, B00001001, B00001001, B00000001, B00000000, // F

4, 8, B00111110, B01000001, B01001001, B01111010, B00000000, // G

4, 8, B01111111, B00001000, B00001000, B01111111, B00000000, // H

3, 8, B01000001, B01111111, B01000001, B00000000, B00000000, // I

4, 8, B00110000, B01000000, B01000001, B00111111, B00000000, // J

4, 8, B01111111, B00001000, B00010100, B01100011, B00000000, // K

4, 8, B01111111, B01000000, B01000000, B01000000, B00000000, // L

5, 8, B01111111, B00000010, B00001100, B00000010, B01111111, // M

5, 8, B01111111, B00000100, B00001000, B00010000, B01111111, // N

4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // O

4, 8, B01111111, B00001001, B00001001, B00000110, B00000000, // P

4, 8, B00111110, B01000001, B01000001, B10111110, B00000000, // Q

4, 8, B01111111, B00001001, B00001001, B01110110, B00000000, // R

4, 8, B01000110, B01001001, B01001001, B00110010, B00000000, // S

5, 8, B00000001, B00000001, B01111111, B00000001, B00000001, // T

4, 8, B00111111, B01000000, B01000000, B00111111, B00000000, // U

5, 8, B00001111, B00110000, B01000000, B00110000, B00001111, // V

5, 8, B00111111, B01000000, B00111000, B01000000, B00111111, // W

5, 8, B01100011, B00010100, B00001000, B00010100, B01100011, // X

5, 8, B00000111, B00001000, B01110000, B00001000, B00000111, // Y

4, 8, B01100001, B01010001, B01001001, B01000111, B00000000, // Z

2, 8, B01111111, B01000001, B00000000, B00000000, B00000000, // [

4, 8, B00000001, B00000110, B00011000, B01100000, B00000000, // \ backsplash

2, 8, B01000001, B01111111, B00000000, B00000000, B00000000, //]

3, 8, B00000010, B00000001, B00000010, B00000000, B00000000, // hat

4, 8, B01000000, B01000000, B01000000, B01000000, B00000000, // _

2, 8, B00000001, B00000010, B00000000, B00000000, B00000000, // `

4, 8, B00100000, B01010100, B01010100, B01111000, B00000000, // a

```
4, 8, B01111111, B01000100, B01000100, B00111000, B00000000, // b
4, 8, B00111000, B01000100, B01000100, B00101000, B00000000, // c
4, 8, B00111000, B01000100, B01000100, B01111111, B00000000, // d
4, 8, B00111000, B01010100, B01010100, B00011000, B00000000, // e
3, 8, B00000100, B01111110, B00000101, B00000000, B00000000, // f
4, 8, B10011000, B10100100, B10100100, B01111000, B00000000, // g
4, 8, B01111111, B00000100, B00000100, B01111000, B00000000, // h
3, 8, B01000100, B01111101, B01000000, B00000000, B00000000, // i
4, 8, B01000000, B10000000, B10000100, B01111101, B00000000, // j
4, 8, B01111111, B00010000, B00101000, B01000100, B00000000, // k
3, 8, B01000001, B01111111, B01000000, B00000000, B00000000, // l
5, 8, B01111100, B00000100, B01111100, B00000100, B01111000, // m
4, 8, B01111100, B00000100, B00000100, B01111000, B00000000, // n
4, 8, B00111000, B01000100, B01000100, B00111000, B00000000, // o
4, 8, B11111100, B00100100, B00100100, B00011000, B00000000, // p
4, 8, B00011000, B00100100, B00100100, B11111100, B00000000, // q
4, 8, B01111100, B00001000, B00000100, B00000100, B00000000, // r
4, 8, B01001000, B01010100, B01010100, B00100100, B00000000, // s
3, 8, B00000100, B00111111, B01000100, B00000000, B00000000, // t
4, 8, B00111100, B01000000, B01000000, B01111100, B00000000, // u
5, 8, B00011100, B00100000, B01000000, B00100000, B00011100, // v
5, 8, B00111100, B01000000, B00111100, B01000000, B00111100, // w
5, 8, B01000100, B00101000, B00010000, B00101000, B01000100, // x
4, 8, B10011100, B10100000, B10100000, B01111100, B00000000, // y
3, 8, B01100100, B01010100, B01001100, B00000000, B00000000, // z
3, 8, B00001000, B00110110, B01000001, B00000000, B00000000, // {
1, 8, B01111111, B00000000, B00000000, B00000000, B00000000, // |
3, 8, B01000001, B00110110, B00001000, B00000000, B00000000, // }
4, 8, B00001000, B00000100, B00001000, B00000100, B00000000, // ~
};
int data = 12;            // MAX7219 modulünün  DIN pini
int load = 10;            // MAX7219 modülünün  CS pini
int clock = 11;           // MAX7219 modulünün  CLK pini
```

```
int maxInUse = 1;    //Kaç tane MAX7219 kullanacağınızı belirlemek  için bu değişkeni kullan
MaxMatrix m(data, load, clock, maxInUse);      // Modulü tanımla
byte buffer[10];
void setup(){
  m.init();                          // Modülü başlatma
  m.setIntensity(2);                 // Dot matix yoğunluğu, 0-15
  Serial.begin(9600);                // Seri iletişim başlatma
  Serial.println("DHTxx test!");
  dht.begin();
}
void loop(){
  int t = dht.readTemperature();
  char temp[4];
  itoa(t,temp,10);                   // INT'i  CHAR'a  çevir!!!!
  Serial.println(temp);
   printStringWithShift("BRAILA ROMANIA", 100);
  printStringWithShift("  temp: ", 100);
  printStringWithShift(temp, 100);
  printStringWithShift(" C   ", 100);
  m.shiftLeft(false, true);
  }
void printCharWithShift(char c, int shift_speed){
  if (c < 32) return;
  c -= 32;
  memcpy_P(buffer, CH + 7*c, 7);
  m.writeSprite(32, 0, buffer);
  m.setColumn(32 + buffer[0], 0);
  for (int i=0; i<buffer[0]+1; i++)
  {
    delay(shift_speed);
    m.shiftLeft(false, false);
  }
}
```

```
void printStringWithShift(char* s, int shift_speed){
  while (*s != 0){
    printCharWithShift(*s, shift_speed);
    s++;
  }
}
void printString(char* s)
{
  int col = 0;
  while (*s != 0)
  {
    if (*s < 32) continue;
    char c = *s - 32;
    memcpy_P(buffer, CH + 7*c, 7);
    m.writeSprite(col, 0, buffer);
    m.setColumn(col + buffer[0], 0);
    col += buffer[0] + 1;
    s++;
  }
}
```

Erasmus+ KA-202

Mesleki Eğitim ve Öğretime Yönelik Stratejik Ortaklık Projesi

Proje Adı: "Mesleki Eğitimde Arduinoları Öğrenme ve Öğretme"

Proje Kısaltması: " ARDUinVET "

Proje Numarası: "2020-1-TR01-KA202-093762"

Arduino Motor Modülü ve Eğitim Kiti

(DC, Step, Servo)

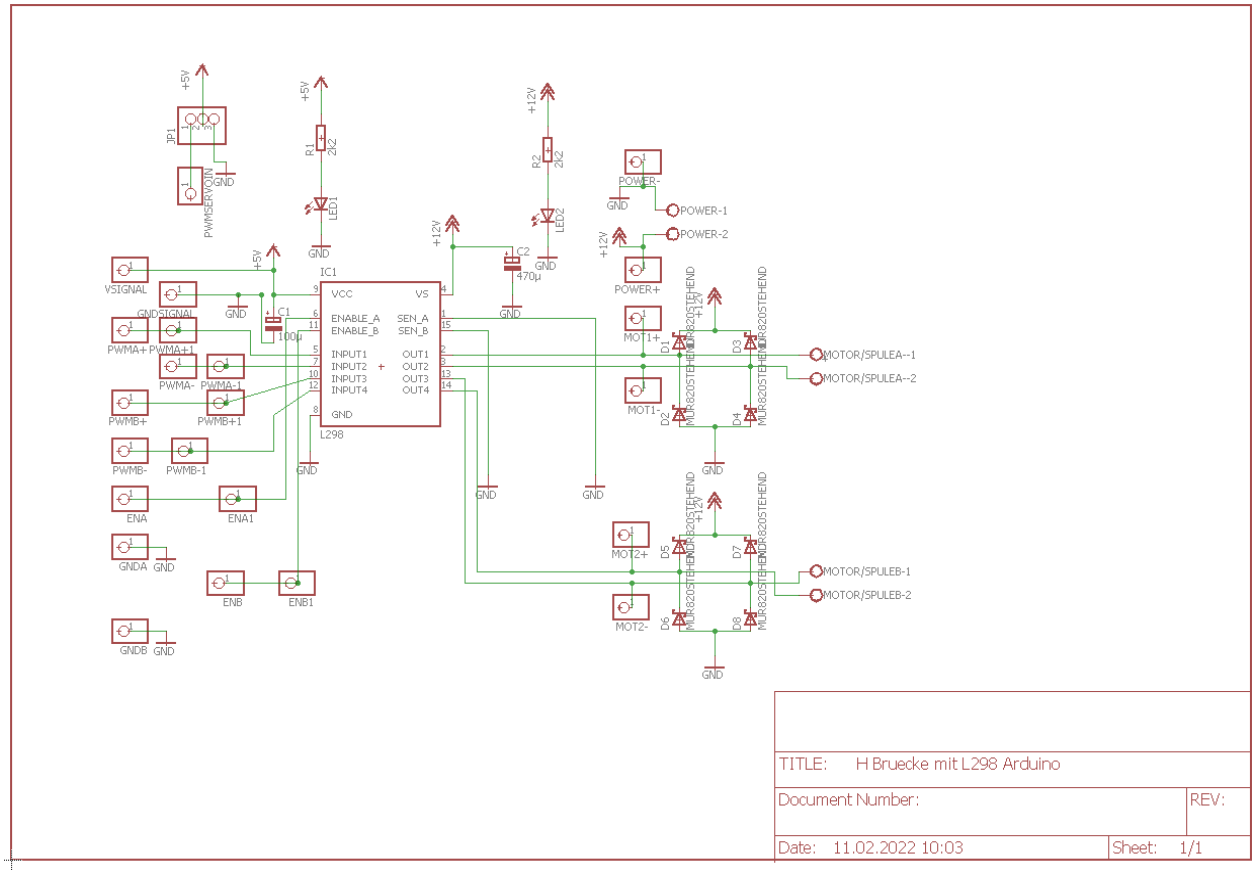
Arduino Motor Modülü ve Eğitim Kiti

Motor modülü eğitim seti, Arduino ile kontrol edilen farklı tipteki motorların (Servo, Step ve DC) işleyişini basit bir şekilde açıklamak amacıyla ARDUinVET projesi kapsamında geliştirilmiş didaktik bir araçtır.

Arduino programlama platformunu kullanarak, geliştirilen donanımın kullanımı ile farklı motorları çalıştırmak için Arduino'ların programlanmasına basit bir şekilde yaklaşmayı amaçlıyoruz.

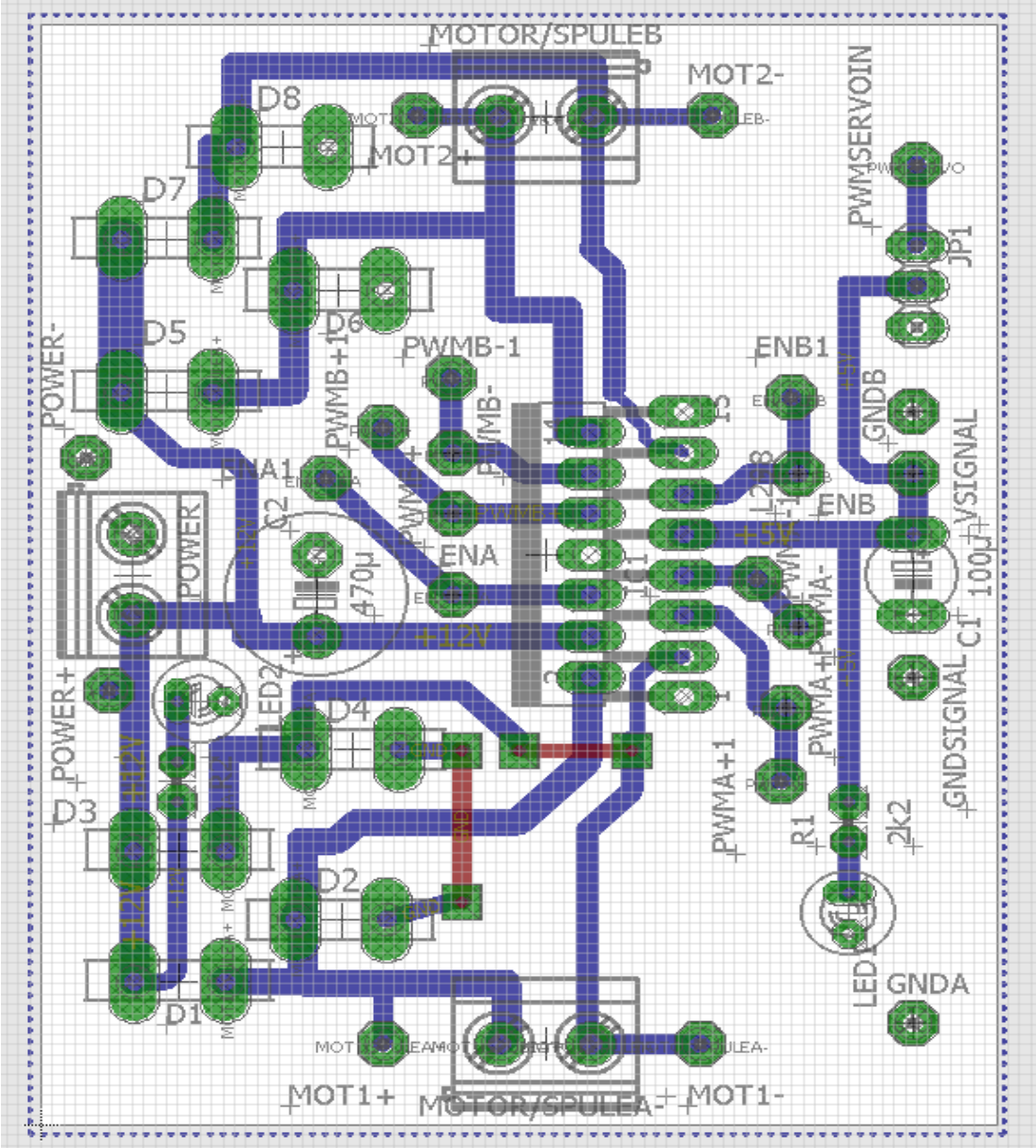
Amaç, öğrencilere bir baskılı devre kartının nasıl oluşturulacağını ve programlanacağını (resim 1) ve bunun uygulanabilirliğini, bu özel durumda motorların çalışmasını ve farklı görevleri yerine getirmesini göstermektir.

Motor Devre Şeması



Resim: 1

Motor Baskı Devre Bağlantısı



Resim: 2

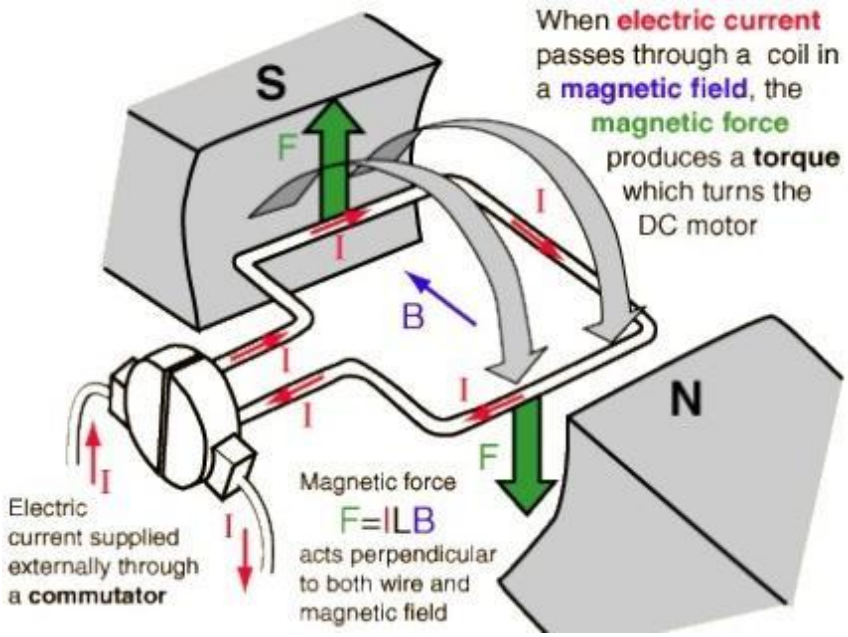
Motorlar

DC Motorlar

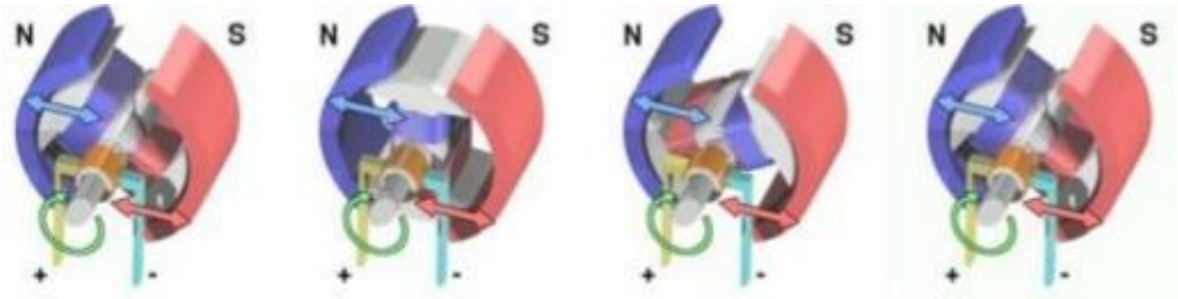
DC Motorlar, (Doğru Akım motorları, resim 3), mıknatısların oluşturduğu çekme ve itme kuvvetleriyle çalışan elektronik cihazlardır.

Picture SEQ Figura * ARABIC 2: Motor DC

Örnek:



Resim 3: DC Motorları Çalışması

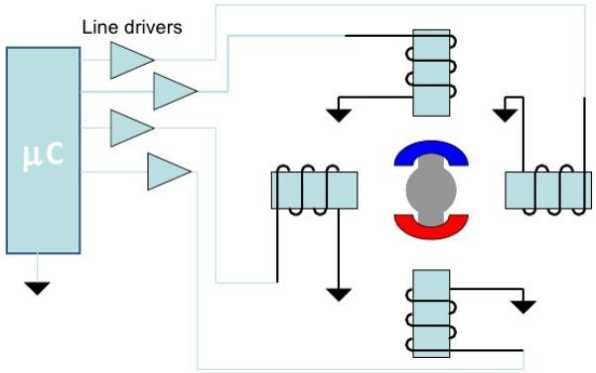


1 **2** **3** **4**

1. Bobin beslendiğinde, rotor etrafında bir manyetik alan üretilir, bu da mıknatıslar arasında bir itmeye neden olur ve motorun saat yönünde dönmesini sağlar;
2. Rotor dönmeye devam ediyor;
3. Rotor yatay olarak hizalandığında, manyetik alan ters çevrilir, hareketi saat yönünde sürdürür;
4. Motor beslenirken süreç kendini sürekli olarak tekrar eder.

Adım Motorları

Step motor, bir kontrolörden alınan darbeye göre hareket eden bir elektrik motorudur. Motorun verdiği adım sayısı, alınan pals sayısı ile tam olarak aynıdır ve motor hızı, pals frekansı ile aynıdır. Bu nedenle basit bir cihazdır (fırçasız, anahtarsız ve kodlayıcısız). Robotik, Kartezyen eksen hareketi gibi çeşitli elektronik ve otomasyon alanlarında kullanılan motorlardır.



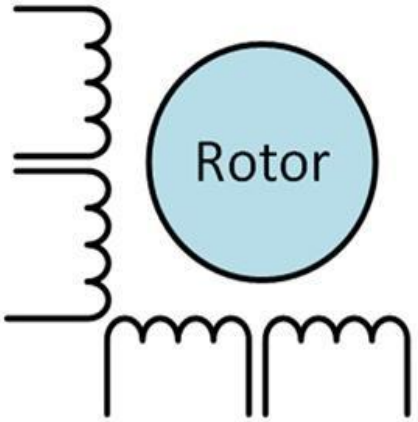
Resim 4: Bir Step Motorun basit diyagramı

Resim 5: Adım Motoru

Adım motorlarına örnekler

Tek kutuplu motor

Tek kutuplu bir motor, her bir elektrik akımı yönü için bir tane olmak üzere, faz bazında iki bobine sahiptir. Bu konfigürasyonda, elektrik akımı yönü değiştirilmeden bir manyetik kutup ters çevrilebilir, anahtarın devresi her bobin için çok basit bir şekilde yapılabilir.



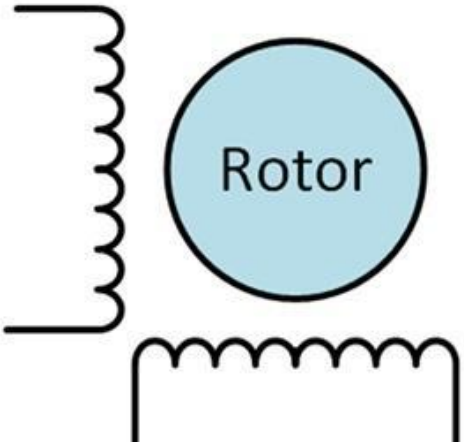
Resim 6: Tek Kutuplu Motor

Cycle	C1	C2	C3	C4
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

Tablo 1: Durum Tablosu.

Motor Bipolar

The bipolar motors have a unique coil by phase. The electrical current in a coil needs to be inverted to invert a magnetic pole. So the electrical circuit is a little more complicated, requesting the need for an H bridge. There are two connections by phase and none is in common.

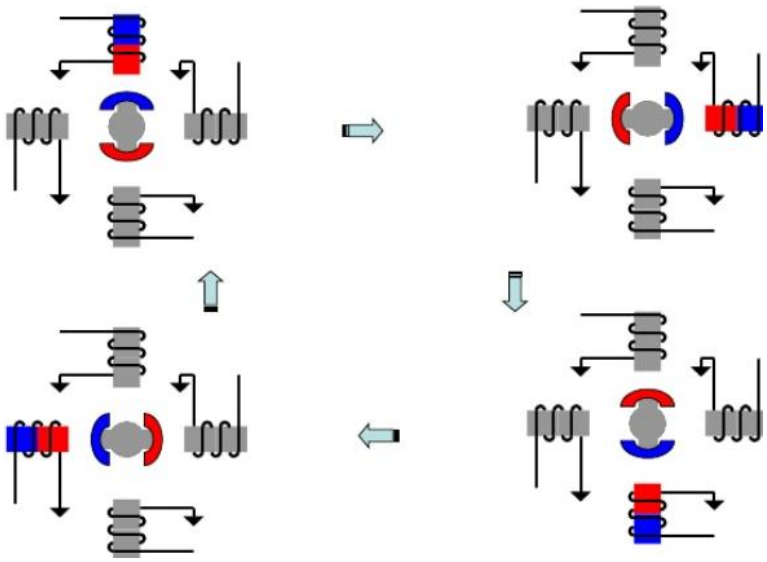


Picture 7: Motor Bipolar

Cycle	C1	C2	C3	C4
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

Table 2: State Table

Step motor functioning



Resim 8: Adım Motor Şeması

Servo motorlar

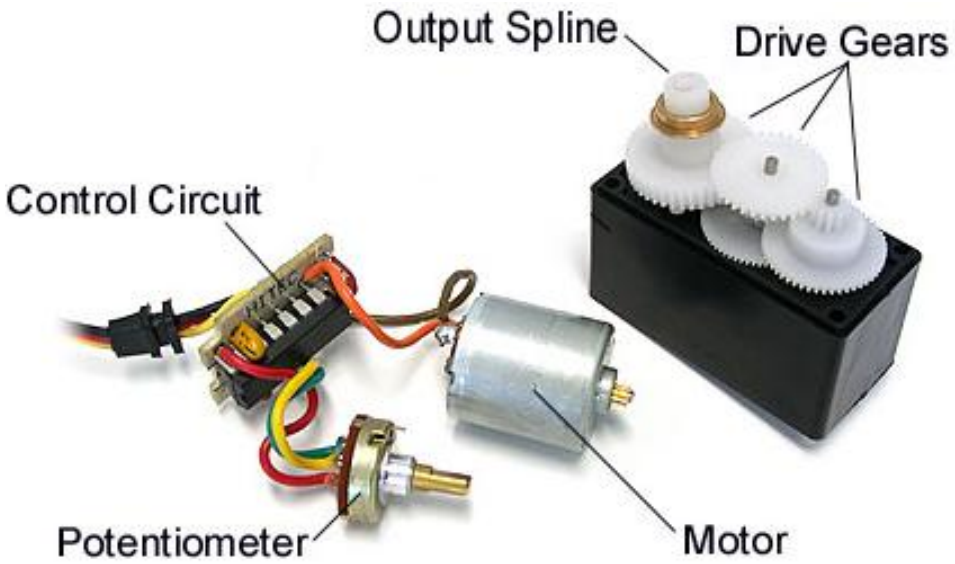
Servo motor, bir elektrik sinyali aracılığıyla ekseni açısal bir konuma yerleştiren elektromekanik bir cihazdır. Normalde bu motor tipi kompakttır ve kendi ekseninin kesin konumunu sağlar. Şu anda robotik ve modellemede kullanılan servo motorlar.

Resim 1: Servo Motor

Özellikleri:

Bir servo motor dört bölüme ayrılabilir:

● Kontrol Devresi – PWM sinyallerini ve enerjisini almaktan sorumludur. Potansiyometrenin konumunu kontrol eder ve alınan PWM sinyaline göre motoru kontrol eder.
● Potansiyometre – Konumlandırarak servo çıkış eksenine bağlanır.
● Motor – Dişliyi ve servonun ana eksenini hareket ettirir.
● Tahrik dişlileri – Ana eksende üstün bir tork uygulamak için gücü azaltarak motorun dönüşünü azaltır. Potansiyometreyi eksenle birlikte hareket ettirirler

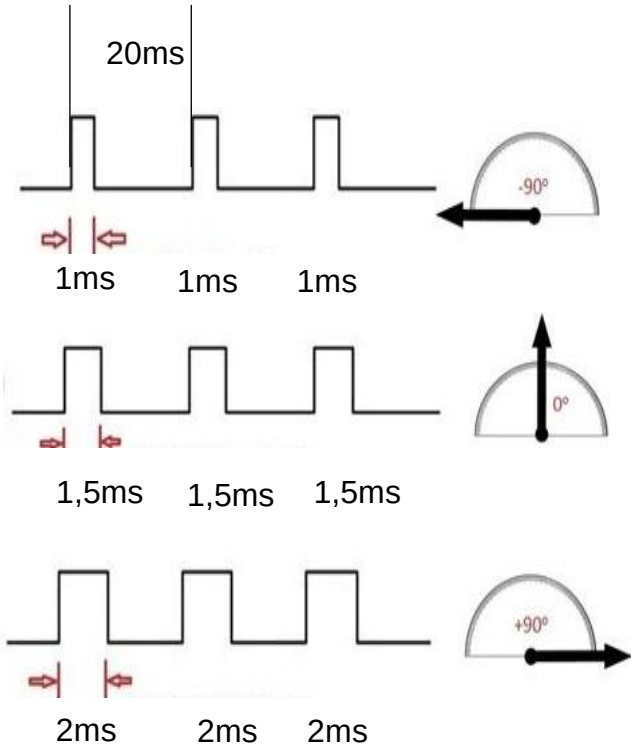


Resim 10: Servo motor parçaları

Servo Motor Konumları

Bir servo motorun kontrolü, konumunu belirleyen TTL voltaj seviyelerini sunan bir giriş sinyali ile elde edilir. Bu sinyal formatı, resim 11'de gösterildiği gibi PWM (Darbe Genişliği Modülasyonu) modülasyonunu takip eder.

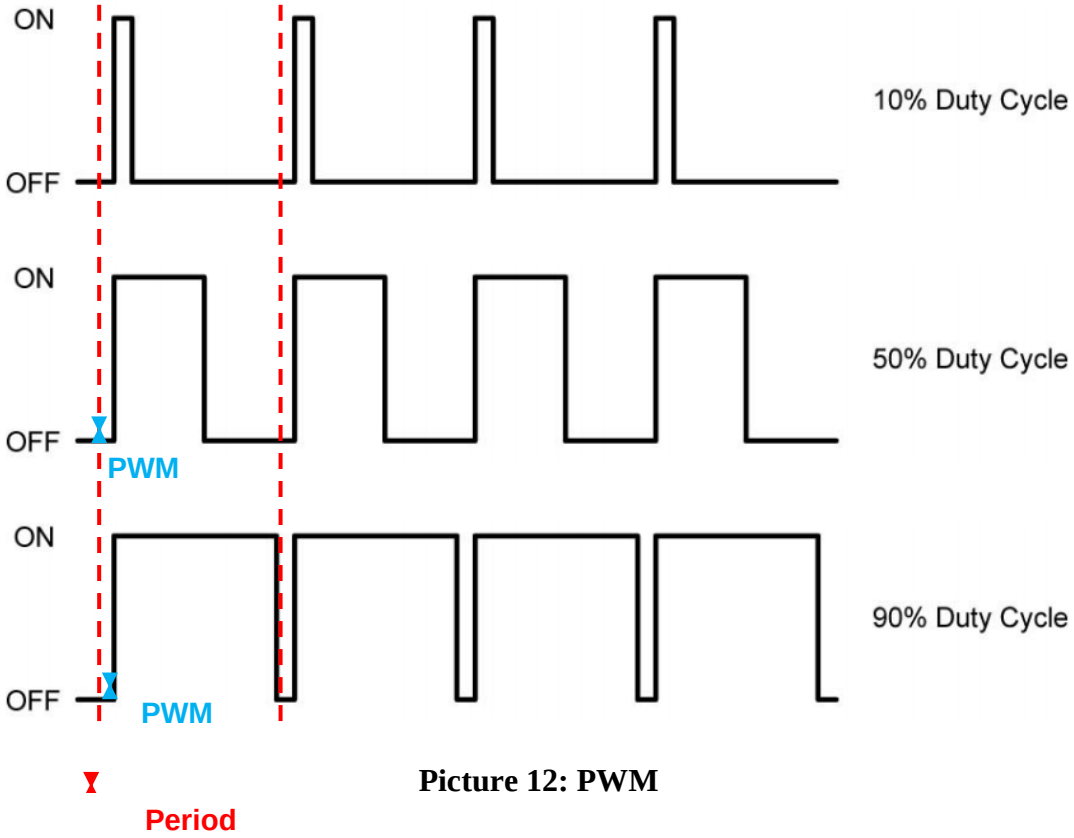
Bu özel durumda, 20 milisaniyede 1 milisaniye yüksek seviyede ise, servo motor resim 11'de gösterildiği gibi minimum konumda olmalıdır.



Resim 11: PWM servo motor

PWM (Darbe Genişlik Modülasyonu)

PWM, elektroniklerin çeşitli alanlarında uygulanabilir. Kullanımlarından biri güç kaynağı, DC motor hız kontrolü, ışık kontrolü, servo motor kontrolü ve diğer birçok uygulamadır. PWM sayesinde hızı ve gücü kontrol edebiliriz.



Picture 12: PWM

PWM işletimi

Bir kare dalga düşünüldüğünde, PWM'nin doğru çalışmasını elde etmek için dalganın darbe genişliğini değiştirmeliyiz. Hesaplamak için periyoda ve darbe genişliğine ihtiyacımız var ve sonucu denklemle tanımlandığı gibi görev döngüsü olarak adlandırılır:

$$Duty\ Cycle = 100\frac{Pulse\ Width}{Period}$$

Görev döngüsü: yüzde değeri;
Darbe Genişliği: sinyalin yüksek seviyede olduğu zaman sırası;
Periyot: bir dalga döngüsünün süresi.

NOT: Aşağıdaki deneyler öğrencilerin kendi yapacakları Arduino Motor Kiti kullanılarak yapılacaktır.

Devre_1 (DC Motorlu): "Tek DC Motorlu basit"

Devre Açıklaması: Bu program bir dc motorun açma ve kapama anahtarlama dizilerini gösterir.Program:

```
// One DC-Motor simple
/*
    Connection:
Arduino   | MotorShielt
PIN       | PIN
-------------------------
+5V       | VSIGNAL
GND       | GNDSIGNAL
4         | ENA
5         | PWMA+
6         | PWMA-
*/
//-------------------------------------------------------------
//  variables
//-------------------------------------------------------------
int ena = 4;    // ENA'yı Arduino kartındaki pin 4'e bağlı
int right = 5;  // PWMA+'yı Arduino kartındaki pin 5'e bağlı
int left = 6;   // Arduino pin 5 connected to PWMA-

//-------------------------------------------------------------
//  setup function
//-------------------------------------------------------------
void setup()
{
  pinMode(ena, OUTPUT);
  pinMode(right, OUTPUT);
  pinMode(left, OUTPUT);
  digitalWrite(ena, HIGH);  // ENA açık durumda
  delay(900);
}
```

```
//-------------------------------------------------------------
// loop Function
//-------------------------------------------------------------
void loop()
{
  analogWrite(right, 255);  // maksimum hızla sağa dön
  delay(1000);
  analogWrite(right, 0);    // Kapalı
  delay(1000);
  analogWrite(left, 255);   // maksimum hızla sola dön
  delay(1000);
  analogWrite(left, 0);     // Kapalı
  delay(1000);
  analogWrite(right, 127);  // Yarım hızla sağa dön
  delay(1000);
  analogWrite(right, 0);    // Kapalı
  delay(1000);
  analogWrite(left, 127);   // Yarım hızla sola dön
  delay(1000);
  analogWrite(left, 0);     // Kapalı
  delay(1000);
}
```

Devre _2: "Bir DC-Motor ilerlemesi"

Devre Açıklaması: Bu program bir DC motoru kontrol eder. Kontrol komutları seri monitör aracılığıyla verilir.

Kontrol komutları:

Seri Monitöre Yazın	*Faaliyeti*
stop	The motor durur
go right	Motor sağa döner
go left	Motor sola döner
+	Hızı artır
-	Hızı azalt

Program:

```
// One DC-Motor advance

/*
    Connection:
Arduino   | MotorShielt
PIN       | PIN
-------------------------
+5V       | VSIGNAL
GND       | GNDSIGNAL
4         | ENA
5         | PWMA+
6         | PWMA-
*/
//-------------------------------------------------------------
//  variables
//-------------------------------------------------------------
int en_a = 4;              // Pin'i etkinleştirmek için Arduino Pin 4 bağlı
int right_a = 5;           // PWMA+ Arduino Pin 5'e bağlı
int left_a = 6;            // PWMA- Arduino Pin 6'ya bağlı
String message = "";       // seri arayüzün mesajını kaydet
int val = 80;              // ız için varsayılan değer
int del = 10;              // "+" ve "-" komutlarıyla değeri azaltın
int minimumVal = 70;       // minimum hız / değer
int maximumVal = 250;      // maksimum hız / değer
bool flag_go_right = false; // "sağa git" komutu verildiğinde doğru olur
```

```
bool flag_go_left = false;  // "sola git" komutu verildiğinde doğru olur

//-------------------------------------------------------------
//  setup function
//-------------------------------------------------------------
void setup()
{
  Serial.begin(9600);
  pinMode(en_a, OUTPUT);
  pinMode(right_a, OUTPUT);
  pinMode(left_a, OUTPUT);
  digitalWrite(en_a, LOW);
  analogWrite(right_a, 0);
  analogWrite(left_a, 0);
  delay(900);
}
//-------------------------------------------------------------
// loop Function
//-------------------------------------------------------------
void loop()
{
  if(Serial.available()>0)
  {
    message = Serial.readString();
    Serial.println(" --> Incoming message: " + message ); // gelen mesajı görüyorsun

    if(message.equals("stop\n"))
    {
      digitalWrite(en_a, LOW);
            analogWrite(right_a, 0);
      analogWrite(left_a, 0);
      flag_go_right = false;
      flag_go_left = false;
      Serial.println(" <-- Outgoing message: stop \n");
    }
    else if(message.equals("go right\n"))  // "sağa git" mesajını kontrol et
    {
      digitalWrite(en_a, HIGH);     // Modülün etkinleştirme Pin'ini YÜKSEK olarak ayarlayın
```

```
        analogWrite(right_a, val);    // değeri sağa dönüş PWM Pin'ine yazın
    analogWrite(left_a, 0);           // sola dönüşün PWM Pin'ine sıfır yaz
    flag_go_left = false;             // bayrakları ayarla
    flag_go_right = true;
    Serial.println(" <-- Outgoing message: go right\n");  // geribildirim
   }
   else if(message.equals("go left\n"))
   {
    digitalWrite(en_a, HIGH);
        analogWrite(right_a, 0);
    analogWrite(left_a, val);
    flag_go_right = false;
    flag_go_left = true;
    Serial.println(" <-- Outgoing message: go left\n");
   }
   else if(message.equals("+\n"))      // "+" ise mesajı kontrol edin
   {
    if(val >= maximumVal)          // maksimum değere ulaşmak
    {
     Serial.println(" <-- Outgoing message: Maximum value! \n");
    }
    else if (val < maximumVal)
    {
     val = val + del;    // değeri artırmak
     if(flag_go_right)   // hangi yönün arttırılması gerektiğini kontrol edin
     {
      analogWrite(right_a, val);
     }
     else if(flag_go_left)
     {
      analogWrite(left_a, val);
     }
     Serial.print(" <-- Outgoing message: value = "); Serial.println(val); Serial.println("");
//feedback
    }
   }
   else if(message.equals("-\n"))
   {
```

```
    if(val <= minimumVal)                    // Minimum Hıza ulaşmak
    {
      Serial.println(" <-- Outgoing message: Minimum value! \n");
    }
    else if (val > minimumVal)
    {
      val = val - del;
      if(flag_go_right)
      {
        analogWrite(right_a, val);
      }
      else if(flag_go_left)
      {
        analogWrite(left_a, val);
      }
      Serial.print(" <-- Outgoing message: value = "); Serial.println(val); Serial.println("");
    }
  }
  else //generate message if the message is unknown
  {
    Serial.println("<<-- Incoming message " + message + "      IS UNKNOWN \n\n");
  }
 }
}
```

Devre _3: "İki DC Motor ilerler"

Devre Açıklaması: Bu program iki dc motoru kontrol eder. Kontrol komutları seri monitör aracılığıyla verilir.

Kontrol komutu:

Seri Monitöre Yazın	*Faaliyeti*
stop	Motorları durdur
stop a	A Motorunu durdur
stop b	B Motorunu durdur
go right a	A Motorunu sağa döndür
go left a	A Motorunu sola döndür
go right b	B Motorunu sağa döndür
go left b	B Motorunu sola döndür
+	Hızı artır
-	Hızı azalt

Program:

```
// two DC-Motors advance
/*
 Connection:
Arduino  | MotorShielt
PIN      | PIN
-------------------------
+5V      | VSIGNAL
GND      | GND
4        | ENA
5        | PWMA+
6        | PWMA-
8        | ENB
9        | PWMB+
10       | PWMB-
```

```
*/

//-----------------------------------------------------------
// variables
//-----------------------------------------------------------
String message = "";

int en_a = 4;
int right_a = 5;
int left_a = 6;
bool flag_go_right_a = false;
bool flag_go_left_a = false;

int en_b = 8;
int right_b = 9;
int left_b = 10;
bool flag_go_right_b = false;
bool flag_go_left_b = false;

int val = 255;
int del = 10;
int minimumVal = 70;
int maximumVal = 250;

//-----------------------------------------------------------
// setup Function
//-----------------------------------------------------------
void setup()
{
  Serial.begin(9600);

  pinMode(en_a, OUTPUT);
  pinMode(right_a, OUTPUT);
  pinMode(left_a, OUTPUT);

  pinMode(en_b, OUTPUT);
  pinMode(right_b, OUTPUT);
  pinMode(left_b, OUTPUT);
```

```
  digitalWrite(en_a, LOW);
  analogWrite(right_a, 0);
  analogWrite(left_a, 0);

  digitalWrite(en_b, LOW);
  analogWrite(right_b, 0);
  analogWrite(left_b, 0);

  delay(900);
}

//-------------------------------------------------------------
// loop Function
//-------------------------------------------------------------
void loop()
{
  if(Serial.available()>0)
  {
    message = Serial.readString();
    Serial.println(" --> Incoming message: " + message);

    if(message.equals("stop\n"))                    // both motors stop
    {
      digitalWrite(en_a, LOW);
            analogWrite(right_a, 0);
      analogWrite(left_a, 0);
      flag_go_right_a = false;
      flag_go_left_a = false;

      digitalWrite(en_b, LOW);
      analogWrite(right_b, 0);
      analogWrite(left_b, 0);
      flag_go_right_b = false;
      flag_go_left_b = false;

      Serial.println(" <-- Outgoing message: stop all\n");
    }
```

```
else if(message.equals("stop a\n"))                    // a motoru durdurma
{
  digitalWrite(en_a, LOW);
  analogWrite(right_a, 0);
  analogWrite(left_a, 0);
  flag_go_right_a = false;
  flag_go_left_a = false;
  Serial.println(" <-- Outgoing message: stop a\n");
}
else if(message.equals("stop b\n"))                    // b motoru durdurma
{
  digitalWrite(en_b, LOW);
  analogWrite(right_b, 0);
  analogWrite(left_b, 0);
  flag_go_right_b = false;
  flag_go_left_b = false;
  Serial.println(" <-- Outgoing message: stop b\n");
 }
else if(message.equals("go right a\n"))
{
  digitalWrite(en_a, HIGH);
        analogWrite(right_a, val);
  analogWrite(left_a, 0);
  flag_go_left_a = false;
  flag_go_right_a = true;
  Serial.println(" <-- Outgoing message: go right a\n");
}
else if(message.equals("go right b\n"))
{
  digitalWrite(en_b, HIGH);
  analogWrite(left_b, 0);
  analogWrite(right_b, val);
  flag_go_left_b = false;
  flag_go_right_b = true;
  Serial.println(" <-- Outgoing message: go right b\n");
}
else if(message.equals("go left a\n"))
{
```

```
  digitalWrite(en_a, HIGH);
        analogWrite(right_a, 0);
  analogWrite(left_a, val);
  flag_go_right_a = false;
  flag_go_left_a = true;
  Serial.println(" <-- Outgoing message: go left a\n");
 }
else if(message.equals("go left b\n"))
{
  digitalWrite(en_b, HIGH);
  analogWrite(right_b, 0);
  analogWrite(left_b, val);
  flag_go_right_b = false;
  flag_go_left_b = true;
  Serial.println(" <-- Outgoing message: go left b \n");
}
else if(message.equals("+\n"))
{
  if(val >= maximumVal)                    // maksimum değere ulaşmak
  {
    Serial.println(" <-- Outgoing message: Maximum value! \n");
  }
  else if (val < maximumVal)
  {
    val = val + del;
    if(flag_go_right_a) analogWrite(right_a, val);
    if(flag_go_right_b) analogWrite(right_b, val);
    if(flag_go_left_a) analogWrite(left_a, val);
    if(flag_go_left_b) analogWrite(left_b, val);
    Serial.print(" <-- Outgoing message: value = "); Serial.println(val); Serial.println("");
  }
}
else if(message.equals("-\n"))
{
  if(val <= minimumVal)                    // Minimum Değere ulaşmak
  {
    Serial.println(" <-- Outgoing message: Minimum value! \n");
  }
```

```
    else if (val > minimumVal)
    {
      val = val - del;
      if(flag_go_right_a) analogWrite(right_a, val);
      if(flag_go_right_b) analogWrite(right_b, val);
      if(flag_go_left_a) analogWrite(left_a, val);
      if(flag_go_left_b) analogWrite(left_b, val);

      Serial.print(" <-- Outgoing message: value = "); Serial.println(val); Serial.println("");
    }
   }
   else
   {
    Serial.println("  <<-- Incoming message " + message + "      IS UNKNOWN \n\n");
   }
  }
}
```

Devre _4: "Basit adımlı Motor programı"

Devre Açıklaması: " Bir step Motor saat yönünde ve saat yönünün tersine döner

Program:

```
// Basit adım Motor programı
/*
   Bağ:
Arduino  | MotorShielt
PIN      | PIN
-------------------------
+5V      | VSIGNAL
GND      | GNDSIGNAL
4        | ENA
5        | PWMA+
6        | PWMA-
8        | ENB
9        | PWMB+
10       | PWMB-
*/

//------------------------------------------------------------
//  librarys
//------------------------------------------------------------
#include <Stepper.h>
// libraty for the stepper
//------------------------------------------------------------
//  constants
//------------------------------------------------------------
#define STEPS 200
// Step Angle of used stepmotor = 1,8 degrees
// 360 / 1,8° = 200

//------------------------------------------------------------
//  class
//------------------------------------------------------------
Stepper Motor(STEPS, 5,6,9,10);
  // Stepper sınıfının yeni bir örneğini oluştur
  // Parametre:
  // STEPS -> Motorun bir devrindeki adım sayısı
```

```
 //  5,6,9,10 -> Kullanılan Pinler

//---------------------------------------------------------------
// variables
//---------------------------------------------------------------
int spe = 50;

//---------------------------------------------------------------
// setup function
//---------------------------------------------------------------
void setup()
{
 Serial.begin(9600);
 pinMode(4, INPUT);
 pinMode(8, INPUT);
 digitalWrite(4 ,HIGH);    // MotorShielt'in ENA Pin'ini etkinleştirin
 digitalWrite(8 ,HIGH);    // MotorShielt'in ENB Pin'ini etkinleştirin

 Motor.setSpeed(spe);  // Object Motor, hızı "bir dakikada dönüş" olarak alır
}

void loop()
{
 Motor.step(100);    // Motoru belirli sayıda adım döndürür
                     // setSpeed()'e yapılan en son çağrı tarafından belirlenen bir hızda
                     // bu durumda saat yönünde 100 adım
 delay(200);

 Motor.step(-50);    // 50 adım saat yönünün tersine
 delay(200);
}
```

Devre _5: "Gelişmiş adım Motor programı"

Devre Açıklaması: Bu program step motorları kontrol eder. Kontrol komutları seri monitör aracılığıyla verilir.

Örnek: Seri Monitör üzerinden 100'ü Arduino'ya gönderin Step-Motor 100 Adım yapar. -100 gönderirseniz Step-Motor diğer yönde 100 Adım yapar.

Program:

```
// Gelişmiş adım Motor programı
/*
   Bağ:
Arduino  | MotorShielt
PIN      | PIN
-------------------------
+5V      | VSIGNAL
GND      | GNDSIGNAL
4        | ENA
5        | PWMA+
6        | PWMA-
8        | ENB
9        | PWMB+
10       | PWMB-
*/

//-------------------------------------------------------------
//  librarys
//-------------------------------------------------------------
#include <Stepper.h>
//-------------------------------------------------------------
//  constants
//-------------------------------------------------------------
#define STEPS 200
//-------------------------------------------------------------
//  class
//-------------------------------------------------------------
Stepper Motor(STEPS, 5,6,9,10);
//-------------------------------------------------------------
// variables
//-------------------------------------------------------------
```

```
int spe = 100;
String message = "";

//------------------------------------------------------------
// setup function
//------------------------------------------------------------

void setup()
{
  Serial.begin(9600);
  pinMode(4, INPUT);
  pinMode(8, INPUT);
  digitalWrite(4 ,HIGH);
  digitalWrite(8 ,HIGH);
  Motor.setSpeed(spe);
}

void loop()
{
  if(Serial.available()>0)
  {
    message = Serial.readString();
    Serial.println("\n\n --> Incoming message: " + message );

    if(message.toInt())
    {
      Serial.print(" <-- Outgoing message:  Steps -> " + message);
      Motor.step(message.toInt());
    }
    else
    {
      Serial.println("  <<-- Incoming message " + message + "  !!! -->  is not a number <-- !!!
\n\n");
    }
  }
}
```

Devre _6 (Servo Motorlu): ” Basit servo motor programı ”

Devre Açıklaması: Bu programda bir servo motor bir pwm sinyali ile kontrol edilmektedir. for döngüsü kullanılır.

// Basit servo Motor programı

```
/*
   Bağ:
Arduino  | MotorShielt
PIN      | PIN
-------------------------
+5V      | VSIGNAL
GND      | GNDSIGNAL
3        | PWMSERVOIN
*/
//-------------------------------------------------------------
// variables
//-------------------------------------------------------------
int del = 100;  // time for delay

//-------------------------------------------------------------
// setup function
//-------------------------------------------------------------
void setup()
{
 Serial.begin(9600);
 pinMode(3, OUTPUT);
}

void loop()
{
 for(int i = 0; i<255; i++)
   {
    analogWrite(3,i);
    Serial.println(i);
    delay(del);
  }
  delay(1000);
}
```

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+ KA-202

Mesleki Eğitim ve Öğretime Yönelik Stratejik Ortaklıklar Projesi

Proje Adı: "Mesleki Eğitimde Aurdino' yu Öğrenme ve Öğretme"

Proje Kısaltması: " ARDUinVET "

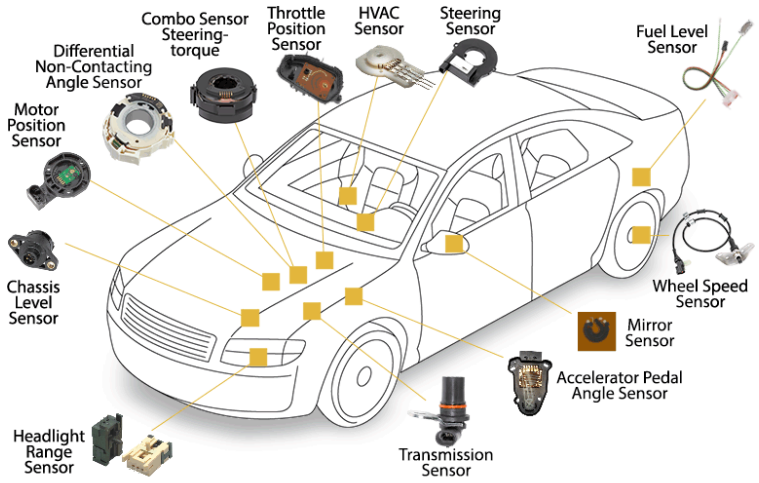
Proje Numarası: "2020-1-TR01-KA202-093762"

Sensör Modülü ve Eğitim Kiti

Sensörler

Sensör, amacı elektonik devre ile çevre arasında bir arayüz oluşturmak olan, elektronik cihaz, modül veya alt sistemdir. Çevre veya fiziksel dünyadan aldığı bilgiyi bir değer olarak elektronik devreye gönderir. Aslında ortamda meydana gelen olayları veya değişiklikleri algılar. Bu olay ya da değişiklikleri algılamak için sensörün belirli değerleri fiziksel ölçekte ölçmesi gerekir. Basınç, sıcaklık veya hızlanma gibi değerleri ölçer ve bir geri bildirim verir. Daha sonra bu değerleri –genellikle- elektriksel sinyallere dönüştürür.

Bu nedenle bir sensör her zaman diğer elektronik aygıt veya elemanlarla birlikte kullanılır. Sensörler, dokunmaya veya harekete duyarlı cihazlar, ışık ve sesle çalışan cihazlar vb.. gibi pek çok günlük nesnede bulunabilir. Sensör kullanımı geleneksel sıcaklık, basınç veya akış ölçümü gibi alanların ötesine geçmiştir. Örn: GPS Sensörleri. Analog sensörler hala yaygın olarak kullaanılmaktadır. Sensör uygulamaları imalat, makine, uçaklar ve havacılık, arabalar, tıp, robotic ve günlük hayatımızın diğer bir çok alanını içerir.



Şekil 1. Otomobil Sensörleri

Sensör Özellikleri

İyi bir sensör aşağıdaki kurallara uyar:

- Ölçülen özelliğe duyarlı olmalıdır.
- Uygulamada karşılaşılması muhtemel herhangi bir özelliğe karşı duyarsız olmalıdır.
- Ölçülen özelliği etkilememelidir.

Sensörlerin temel özellikleri:

- Doğrusallık: Sensörün değeri değişebilen bir ayarı ya da özelliği vardır. Ölçüen fiziksel nicelik değiştiğinde sensor özelliğinde kayda değer değişikliklerin olması beklenir. Bu özelliğe doğrusallık denir ve temel derecede öneme sahiptir.
- Kesinlik. Çıkış değerinin, giriş değerine yakınlığı.
- Hata: Ölçülen değer ile gerçek değer arasındaki fark.
- Tolerans: Sensörün üretebileceği maksimum hata.
- Tam Ölçekli Giriş: Sensörün ölçülen fiziksel boyutlardaki hangi karelerde kullanılabileceğini belirtir.
- Tam Ölçekli Çıkış: Sensör çıkışında akımın veya gerilimin alabileceği değerler.
- Duyarlılık: Ölçülen fiziksel boyutun her birimi için çıkış sinyalinin sensöre ne kadar yüksek çıkış verdiğini ifade eder.
- Çözünürlük: Sensörün algılayabileceği fiziksel boyuttaki en küçük değişikliği ifade eder ve buna göre çıkış değerini değiştirir.
- Histerezis: Histerezis hatası, çıkış değerinin önceki giriş değerlerine bağlı olarak değişmesine neden olur. Bir sensörün çıkışı farklıysa, girişi artırarak veya azaltarak belirli bir giriş değerine ulaşılıp ulaşılmadığına bağlı olarak, sensörde histerezis hatası vardır.
- Gecikme: Giriş değişikliğinden sonra çıkış değerindeki değişikliğinin gerçekleşmesinin arasındaki süredir.

- Ölü Bölge: Çıkış değerini etkilemeyen maksimum giriş değeri değişikliği miktarı.

Sensör Kategorileri

Bazıları aşağıda listelenen sensörleri kategorize etmenin birkaç yolu vardır.

Birincisi, sensörleri analog ve dijital sensörleri bölerek çıkış değerinin biçimine göre sınıflandırır.

İkinci sınıflandırma, doğal ve kimyasal sensörler arasında daha önemli bir ayrım ile bir sensörün neyi ölçebileceğiyle ilgilidir. Doğal sensörler konum, kütle, akım, zaman ve bunların göreceli boyutları gibi fiziksel boyutları kontrol ederken, kimyasal sensörler belirli bir atmosferdeki farklı gazların varlığını kontrol eder.

Diğer bir yol, sensörün çalıştığı fiziksel özelliklerdeki malzemelerle ilgilidir. Ana kategorideki sensörlerle birlikte iletken, yarı iletken, dielektrik, manyetik ve süper iletken malzemelere sahip sensörler.

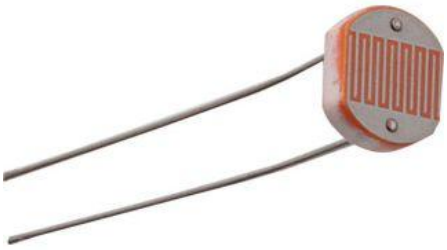
Son olarak, bir sınıflandırma yöntemi daha, endüstriyel, tıbbi, askeri, çevresel sensörler gibi ana kategorilerin yanı sıra nakliye ve otomasyon uygulamaları için kullanılan sensörleri ifade eder.

Şekil 2: Değişik Sensörler

Temel Sensörler

Genel uygulamalarda, yayın olarak kullanılan bazı sensörler:

Işığa Bağlı Direnç LDR: Direnç, üzerine düşen ışığa bağlı olarak değeri değişen bir dirençtir.



Şekil 3: Işığa bağlı direnç

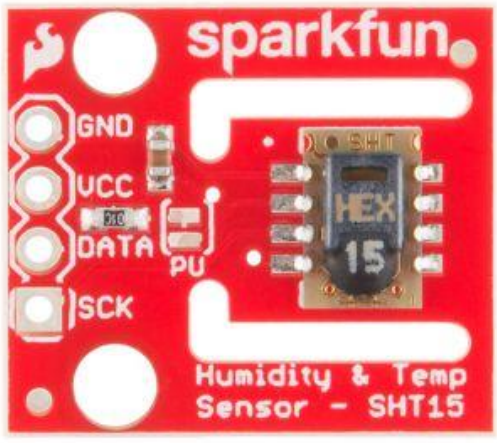
Dijital Parlaklık / Lux / Işık Sensörü: TSL2561 parlaklık sensörü, çok çeşitli ışık koşullarında kullanım için ideal olan gelişmiş bir dijital ışık sensörüdür. Bu sensör daha hassastır, tam lüks hesaplamalarına izin verir ve anında 0,1 - 40,000 + Lux'a kadar ışık aralıklarını algılamak için farklı kazanç / zamanlama aralıkları için yapılandırılabilir. Hem kızılötesi hem de tam spektrumlu diyotlar içerir! Bu, kızılötesi, tam spektrumlu veya insan tarafından görülebilen ışığı ayrı ayrı ölçebileceğiniz anlamına gelir.

Şekil 3: Dijital Parlaklık / Lux / Işık Sensörü

Sıcaklık Sensörü (analog çıkışlı): Ölçtüğü sıcaklık aralığı genellikle ± 1 ° C hassasiyetle -40 ° C ila + 100 ° C'dir.

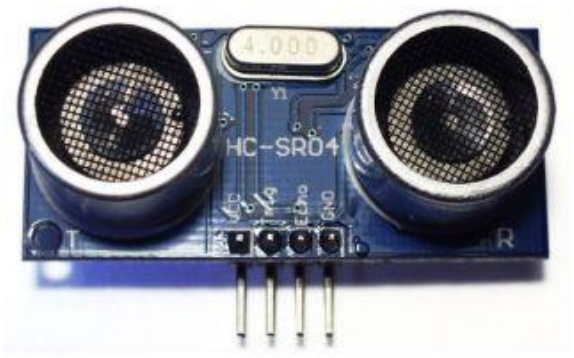
Şekil 4: Sıcaklık Sensörü

Nem ve Sıcaklık Sensörü: Son derece hassas, dijital sıcaklık ve nem sensörü. 25 ° C'de +/- 0,3 ° C sıcaklık doğruluğuna sahip% 0-100 bağıl nem ölçüm aralığına sahiptir.



Şekil 5: Nem ve Sıcaklık Sensörü

Infrared Sensör: Mesafeyi hesaplamak için kullanılır. 1 cm' den 400 cm' ye kadar 1cm hassasiyetle ölçüm yapılabilir.



Şekil 6: Ultrasonik sensör

Hareket sensörü: Bir insanın veya evcil hayvanın bir odadaki hareketini altı metre içinde tespit etme yeteneğine sahiptir. Sensörün, hareketi algıladığı andan itibaren hassasiyet ve aktivasyon süresinin ayarlanabileceği iki düzeltici direnci vardır.

Şekil 7: Hareket Sensörü

Titreşim Sensörü: Sensör titreşimi algıladığında, 1Mohm direnç kullanılarak bir voltaj üretilir

Şekil 8: Titreşim Sensörü

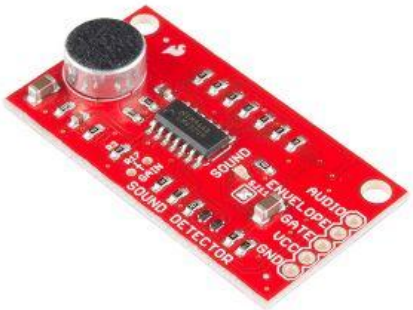
Üç Eksen İvmeölçer ve Jiroskop Sensörü: Sensör, 3 eksenli bir jiroskop ve 3 eksenli bir ivmeölçeri aynı silikon kalıp üzerinde, karmaşık 9 eksenli birleştirilmiş hareket algoritmalarını işleyebilen yerleşik bir Dijital Hareket İşlemcisi (DMP) ile birleştirerek tüm eksenler arası hizalama değerlerini döndürebilir.

Şekil 9: Üç Eksen İvmeölçer ve Jiroskop Sensörü

Gaz Sensörü: LPG, doğalgaz, kömür gazına duyarlıdır. Ölçülen gazların konsantrasyonu arttıkça çıkış voltajı da artar.

Şekil 11: Gaz Sensörü

Ses Dedektörü: Bu sensör bir ses çıkışı sağlar, aynı zamanda sesin varlığının ikili bir göstergesi ve genliğinin analog bir gösterimini sağlar.



Şekil 12: Ses Dedektörü

PROJE 1- Adı: ULTRASONİK SENSÖRLÜ TRAFİK SİNYALİZASYONU

Projenin Amacı: Bu projede öğrenciler, harici bir pull-up direnci kullanarak trafik sinyalinde ultrasonik sensörü nasıl kullanabileceğimizi görecekler.

Bu proje sayesinde öğrenciler, ses dalgalarını kullanarak bir nesneye olan mesafeyi ölçen bir cihaz olan ultrasonik bir sensörle deney yapabilecekler.

Metodoloji

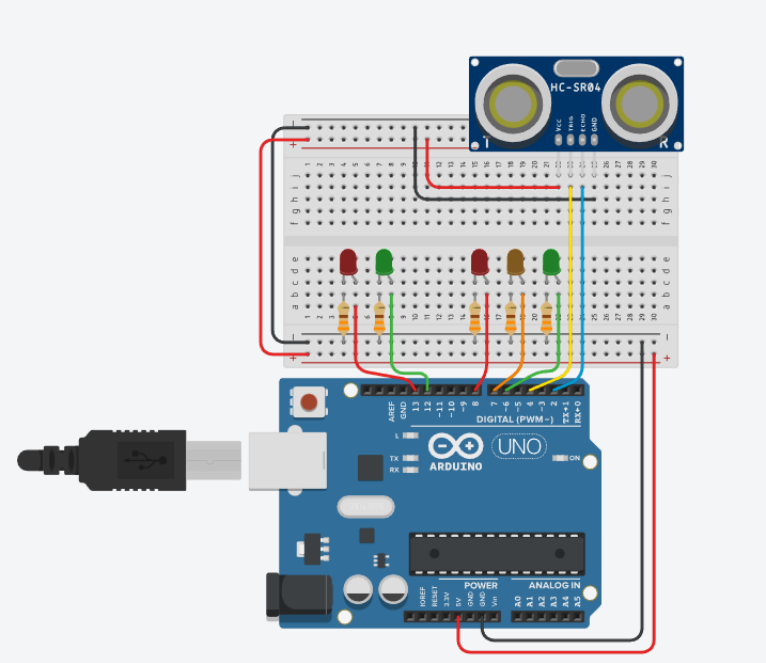
İlk olarak, projenin istenen iş akışını açıklıyoruz. Ardından öğrencilerden gerekli bileşenleri seçmelerini ve devreyi tasarlamak ve çizmek için Tinkercad'i kullanmalarını istiyoruz. Kodu yazmak için Tinkercad Kod Aracını kullanacaklar.

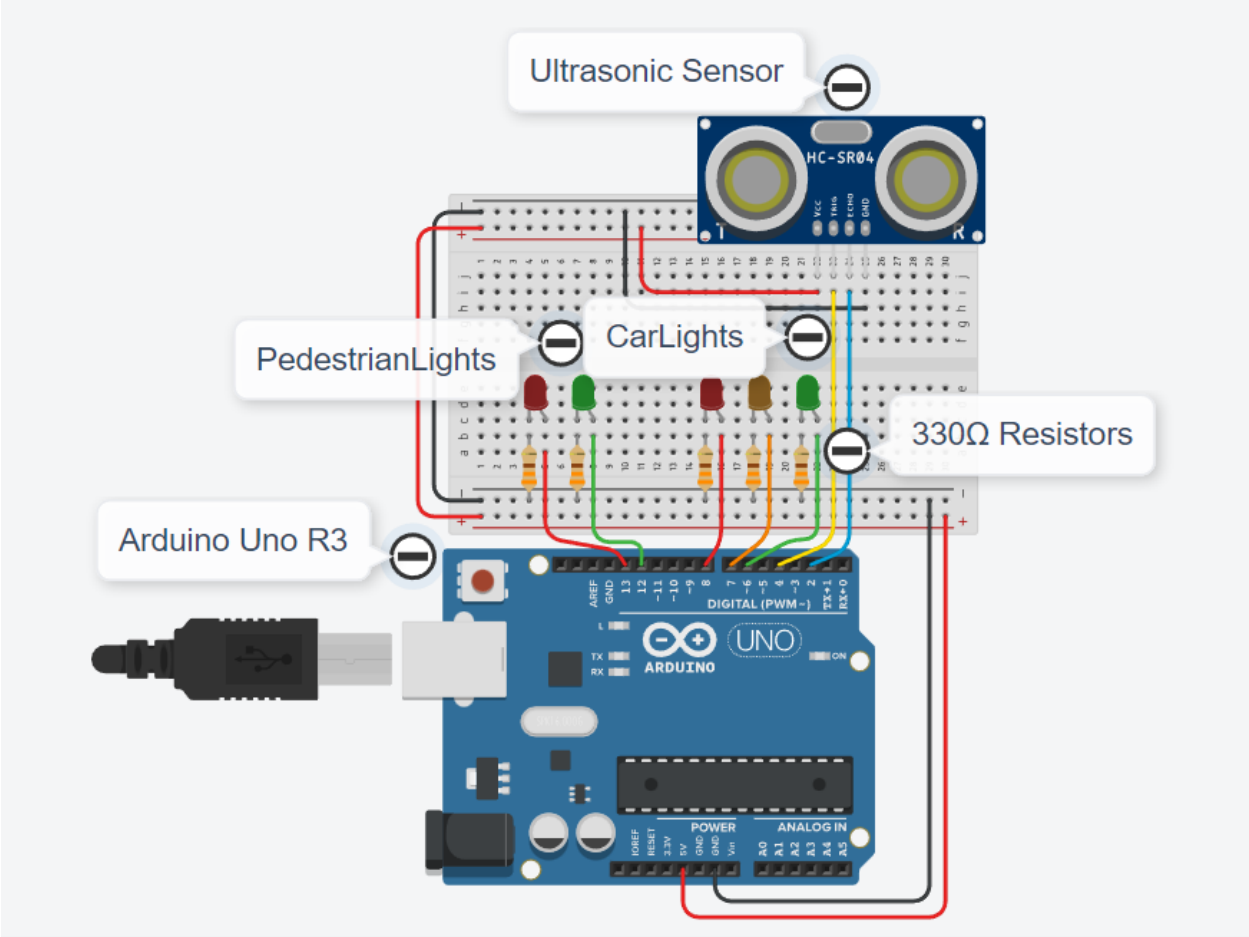
Ardından devreyi oluşturacak, kodu Arduino yazılım aracına yazacak ve Arduino kartına yükleyecekler.

Tinkercad'deki simülasyon, devrenin doğru şekilde oluşturulduğunu doğrulayacaktır.

Bu proje, Erasmus+KA-202 Mesleki Eğitim ve Öğretimde Stratejik Ortaklıklar projemizden "ArduinVET" adı verilen sonuç ve sonuçlardan yararlanmaya başlayan temel düzey öğrenciler içindir.

Proje Devresi





Nasıl Çalışır:

Ultrasonik sensör, ultrasonik frekansta bir ses dalgası göndererek ve nesneden geri sıçramasını bekleyerek çalışır. Sesin iletimi ile sesin alınması arasındaki zaman gecikmesi daha sonra mesafeyi hesaplamak için kullanılır.

Öğrenciler, ellerini sensörün önüne koyarak ses iletimi ile ses alımı arasındaki zaman gecikmesinin azalmasına neden olarak sensörü etkinleştirebilirler. Mesafe seri bağlı ekranda görüntülenecektir. Öğrenciler ışıkları gözlemleyerek trafik ışıklarının aktivasyonunu görebilirler.

Devrede ultrasonik sensör ile birlikte harici bir yukarı çekme direnci kullanıyoruz.

* Pin 3'ü kalıcı olarak YÜKSEK (+ 5V) durumda tutan yukarı çekme direncini korur.

* Ultrasonik sensör etkinleştirildiğinde Pin 3 topraklandığında (DÜŞÜK durum) anlık olarak topraklanır.

Zaman Tablosu

Fazlar	Araçlar	Yayalar	Zamanlar	Açıklama
0				Sensör aktif değil.
1			3sn	Sensör etkinleştirildiğinde, faz 1 ila 4 etkinleştirilir ve ardından cihaz önceki durumuna döner (Faz 0)
2			3sn	
3			4sn	
4			3sn	
0				Sensör tekrar aktif olana kadar

Malzeme Listesi:

Malzemelerimiz:

- Arduino Kartı
- Ultrasonik Sensör HC-SR04
- Breadboard ve Atlama Kabloları
- USB Kablosu
- Arduino UNO R3
- 5XLED
- 5X330 ohm

Projenin Arduino Kodu:

```
//Project Adı: ULTRASONİK SENSÖRLÜ TRAFİK SİNYALİZASYONU
// Yaya ışıklı ve ultrasonic sensörlü trafik sinyalizasyonu
// Sabitlerin tanımlanması
  const byte greenCar = 6;
  const byte orangeCar = 7;
  const byte redCar = 8;
  const byte greenPedestrian = 12;
  const byte redPedestrian = 13;
```

```
// Parametrelerin tanımlanması
  boolean buttonOn=false;

// Parametrelerin tanımlanması Ultrasonic Sensor HC-SR04 * ******
// pin numaralarının tanımlanması
  const int trigPin = 4;  //D4
  const int echoPin = 2;  //D3


void setup() {
 // Sabitlerin başlatılması Ultrasonic Sensor HC-SR04 * ******
  pinMode(trigPin, OUTPUT); // trigPin' in çıkı yapılması
  pinMode(echoPin, INPUT); // echoPin' in giriş yapılması
  Serial.begin(9600); // seri haberleşmenin başlatılması
 // parametrelerin başlatılması
  pinMode(greenCar,OUTPUT);
  pinMode(orangeCar,OUTPUT);
  pinMode(redCar,OUTPUT);

  pinMode(greenPedestrian,OUTPUT);
  pinMode(redPedestrian,OUTPUT);

  // yaya sinyalizasyon değerlerinin başlatılması
  digitalWrite(greenPedestrian,LOW);
  digitalWrite(redPedestrian,HIGH);

  // araç sinyalizasyon değerlerinin başlatılması
  digitalWrite(greenCar,HIGH);
  digitalWrite(orangeCar,LOW);
  digitalWrite(redCar,LOW);
  pinMode (2, INPUT_PULLUP);
}

void loop() {
 //****** * 3. Code Ultrasonic Sensor HC-SR04 * ******
 // trigPin leri temizleme
  delay(500);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

// trigPinleri 10usn için 1' e çekme
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

// echoPin'i okur, ses dalgasının seyahat süresini mikrosaniye cinsinden döndürür
  const long duration = pulseIn(echoPin, HIGH);
  Serial.print("Duration: ");
  Serial.println(duration);
```

```
// Mesafe hesaplama
  const long distance= duration/58.2;
//Seri Monitöre çıktı alma
  Serial.print("Distance: ");
  Serial.println(distance);

  delay(2000);
  if (distance<20){
     buttonOn=true;
  }
  if(buttonOn == true)
    {
     buttonOn = false;

     delay(3000);
     digitalWrite(greenCar,LOW);
     digitalWrite(orangeCar,HIGH);

     delay(3000);
     digitalWrite(orangeCar,LOW);
     digitalWrite(redCar,HIGH);

     digitalWrite(greenPedestrian,HIGH);
     digitalWrite(redPedestrian,LOW);

     delay(4000);
     digitalWrite(greenPedestrian,LOW);
     digitalWrite(redPedestrian,HIGH);

     delay(3000);
     digitalWrite(greenCar,HIGH);
     digitalWrite(redCar,LOW);

     delay(5000);
    }else{
     digitalWrite(greenPedestrian,LOW);
     digitalWrite(redPedestrian,HIGH);

     digitalWrite(greenCar,HIGH);
     digitalWrite(orangeCar,LOW);
     digitalWrite(redCar,LOW);
     }
}
```

PROJE 2 - ADI: ALARM SİSTEMİ

Projenin Amacı: Bu projede öğrenciler, sıvı gazı, kapalı alandaki hareketi ve dengesiz sıcaklık artışını algılayabilen bir alarm sisteminin nasıl çalıştığını görecekler. Alarm sistemi bir düğme ve bir fotodirenç sensörü ile kontrol edilecektir. Düğmeye basarak öğrenciler alarm sistemini etkinleştirebilirler. Bir düğmeye basarak bir zil çaldığında, onu durdurabilirler. Etkinleştirilen bir sensörü düğmeye basarak çalıştırdıklarında alarm sistemini devre dışı bırakabilirler. Foto direnç, alarm sisteminin etkin olmaması ve aydınlatmanın düşük olması durumunda alarm sistemini etkinleştirir.

Bu proje sayesinde öğrenciler şunları deneyimleyebilecekler:

- ✓ Sabit bir ses frekansı üreten, yerleşik bir osilatör devresine sahip aktif bir zil sensörü modülü, bir Arduino pini ile açılır ve kapatılır.
- ✓ Ultrasonik sensör kullanarak bir nesneye olan mesafeyi ses dalgaları ile ölçer.
- ✓ Bir gaz sensörü kullanır.
- ✓ İki gösterge ışığı ve bir buton,
- ✓ Bir foto direnç,
- ✓ Bir sıcaklık sensörü kullanır.

Metodoloji

İlk olarak, geliştirilecek projenin istenen iş akışını adım adım açıklıyoruz. Öğrencilerden daha sonra devreyi tasarlamak ve çizmek için gerekli bileşenleri seçmeleri istenir. Kodu yazmak için "Arduino" uygulamasını kullanacaklar.

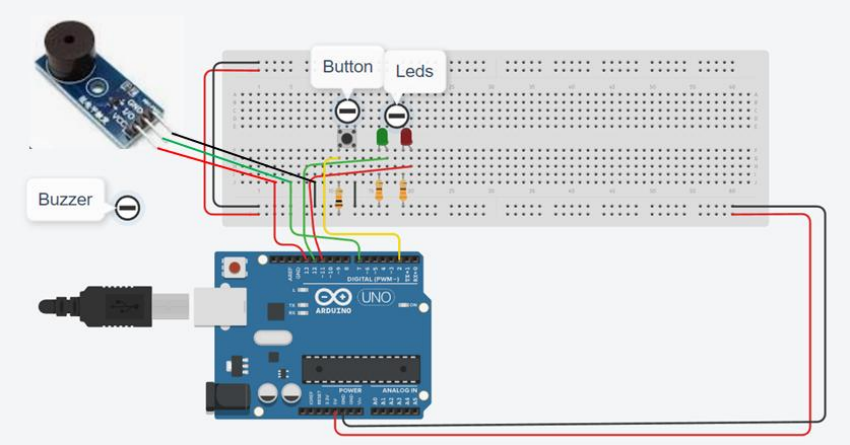
Ardından devreyi oluşturacak, kodu Arduino yazılım aracına yazacak ve Arduino kartına yükleyecekler.

Öğrenciler devrenin doğru yapıldığını doğrulayacaktır.

Bu proje, Erasmus+KA-202 Mesleki Eğitim ve Öğretimde Stratejik Ortaklıklar projemizden “ArduinVET” adı verilen sonuç ve sonuçlardan yararlanmaya başlayan temel düzey öğrenciler içindir.

Adım 1 (Alarm Sisteminde Farklı Ses Tonlu Buzzer Kullanımı)

Proje Devresi



Nasıl Çalışır:

Aktif bir zil sensörü ünitesinde yerleşik bir salınım devresi bulunur, bu nedenle sesin frekansı sabittir. Sesin kendisini üretebilir.

Bu çalışma içinde ses tonu, kod aracılığıyla aynı sese sahip olmaktan çıkar. Düğmeye basılır basılmaz bir Arduino pimi ile bip sesi çıkarmaya başlar, bekleyen kırmızı Led söner ve yeşil Led yanar.

Devrede ultrasonik sensör ile birlikte harici bir yukarı çekme direnci kullanıyoruz.

Pim 3'ü kalıcı olarak YÜKSEK (+ 5V) durumda tutan yukarı çekme direncini korur.

Düğmeye basıldığında Pim 3 topraklanır (DÜŞÜK durum).

Malzeme Listesi:

Komponentlerimiz:

- Buzzer
- Breadboard ve Atlama Kabloları
- USB Kablosu
- Arduino UNO R3
- 2XLED
- 2X330 ohm
- Buton
- 1X10Kohm

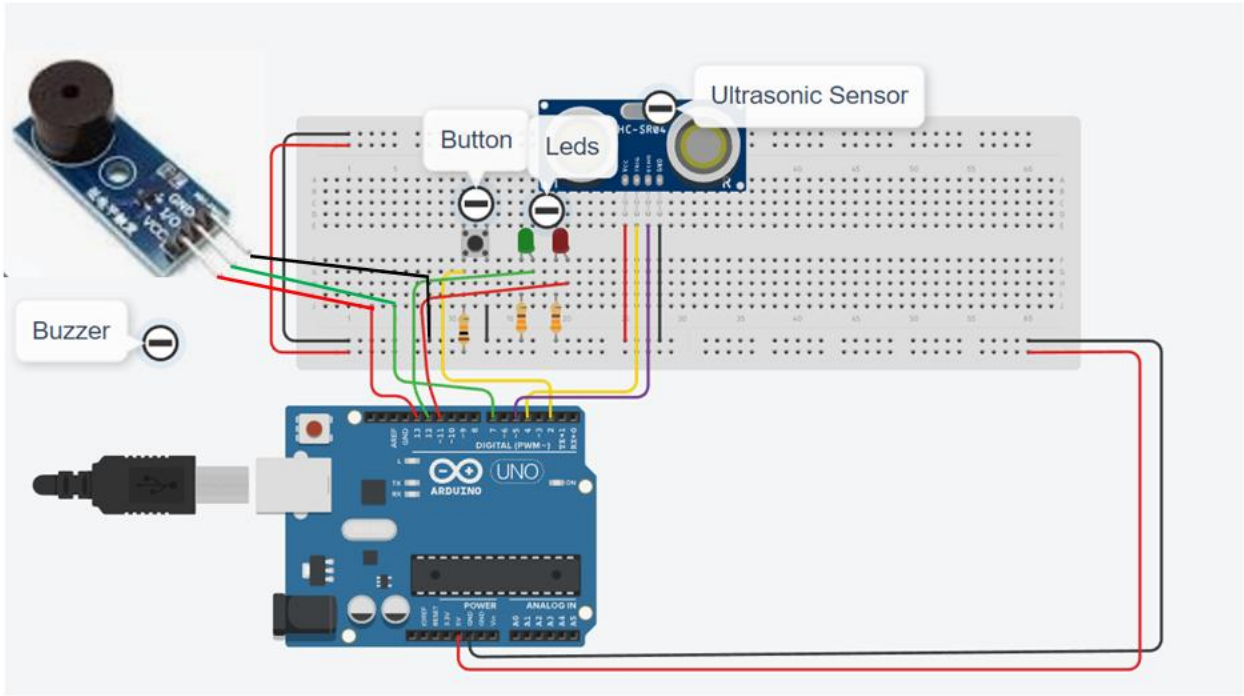
Adım 1 için Arduino Kodu:

```
//Proje Adı: Alarm Sistemi

// Adım 1  (farklı tonda buzzer)

//sabit ve değişkenlerin tanımlanması
int buzz= 7; // buzzerden gelen giriş-çıkış pini bağlanacak.
int buzzVcc= 13; //buzzerdan gelen besleme pini bağlanacak.
const int wpm = 20; // WPM mors hızı
const int dotL = 1200/wpm; // hesaplanmış nokta uzunluğu
const int dashL = 3*dotL; // Dash = 3 x dot
const int sPause = dotL; // Sembol bekleme = 1 dot
const int lPause = dashL; // Harf bekleme = 3 dots
const int wPause = 7*dotL; // Kelime bekleme = 7 dots

int red_led=12;
int green_led=11;

boolean buttonOn=false;
void setup()
{
  // değişken ve sabitlerin tanımlanması
  pinMode(buzz,OUTPUT);
  pinMode(buzzVcc,OUTPUT);

  pinMode(red_led,OUTPUT);
  pinMode(green_led,OUTPUT);

  pinMode (2, INPUT_PULLUP);
}

void loop(){
  if (digitalRead(2)== LOW){
      buttonOn=true;
      }

    //Buzzer
    if(buttonOn==true){

      digitalWrite(red_led, HIGH);
      digitalWrite(green_led, HIGH);

      digitalWrite(buzz, LOW); // Ton aktif
      delay(dashL); // Ton uzunluğu
      digitalWrite(buzz, HIGH); // Ton kapalı
      delay(sPause); // Sembol bekleme
```

```
    digitalWrite(buzzVcc, HIGH);
    digitalWrite(buzz, LOW); // Ton aktif
    delay(dotL); // Ton uzunluğu
    digitalWrite(buzz, HIGH); // Ton kapalı
    delay(sPause); // Sembol bekleme

    digitalWrite(buzz, LOW); // Ton aktif
    delay(dashL); // Ton uzunlığu
    digitalWrite(buzz, HIGH); // Ton kapalı
    delay(sPause); // Sembol bekleme

    digitalWrite(buzz, LOW); // Ton aktif
    delay(dotL); // Ton uzunluğu
    digitalWrite(buzz, HIGH); // Ton kapalı
    delay(sPause); // Sembol bekleme

    delay(lPause-sPause); // Zaten alınmış olan duraklamayı çıkarır

    digitalWrite(buzz, LOW); // Ton aktif
    delay(dashL); // Ton uzunluğu
    digitalWrite(buzz, HIGH); // Ton kapalı
    delay(sPause); // Sembol bekleme
    digitalWrite(buzz, LOW); // Ton aktif
    delay(dashL); // Ton kapalı
    digitalWrite(buzz, HIGH); // Ton kapalı
    delay(sPause); // Sembol bekleme

    digitalWrite(buzz, LOW); // Ton aktif
    delay(dotL); // Ton uzunluğu
    digitalWrite(buzz, HIGH); // Ton kapalı
    delay(sPause); // Sembol bekleme

    digitalWrite(buzz, LOW); // Ton aktif
    delay(dashL); // Ton uzunluğu
    digitalWrite(buzz, HIGH); // Ton kapalı
    delay(sPause); // Sembol bekleme
    delay(wPause-sPause); // Zaten alınmış olan duraklamayı çıkarır
    delay(5000);
    // Orijinal duruma değişken dönüş
    buttonOn=false;
    digitalWrite(buzz, LOW); // Ton aktif
    digitalWrite(buzzVcc, LOW);
    digitalWrite(red_led, LOW);
    digitalWrite(green_led, LOW);
  }
}
```

Adım 2 (HC-SR04 ultrasonik sensörün Alarm Sistemin eklenmesi)

Ultrasonic Sensor HC-SR04' ün bağlanması



Nasıl Çalışır:

Ultrasonik sensör, bir frekansta bir ses dalgası gönderir ve nesneden yansıma bekler. Sesin iletimi ile sesin alınması arasındaki zaman gecikmesi daha sonra mesafeyi hesaplamak için kullanılır. Öğrenciler düğmeye basarak alarm sistemini etkinleştirebilir, ardından yeşil led yanar ve ellerini sensörün önüne koyarlarsa ses iletimi ile ses alımı arasındaki zaman gecikmesi azalır ve yeşil led söner, kırmızı led'i yanar ve zil sesi başlayacaktır. Mesafe seri bağla ekranda görüntülenecektir. Öğrenciler düğmeye tekrar basarak alarm sistemini devre dışı da bırakabilirler.

Adım 2' nin Arduino Kodu:

```
// code of Step 2 - Added Ultrasonic Sensor HC-SR04
int red_led=12;
int green_led=11;
int sensorThres=400;

//buzzer
int buzz= 13; // buzzer' ın giriş çıkış pinleri buraya bağlanacak.
int buzzVcc= 7; //buzzer besleme pini buraya bağlanacak.
const int wpm = 20; // WPM Mors hızı
const int dotL = 1200/wpm; // Hesaplanmış nokta uzunluğu
```

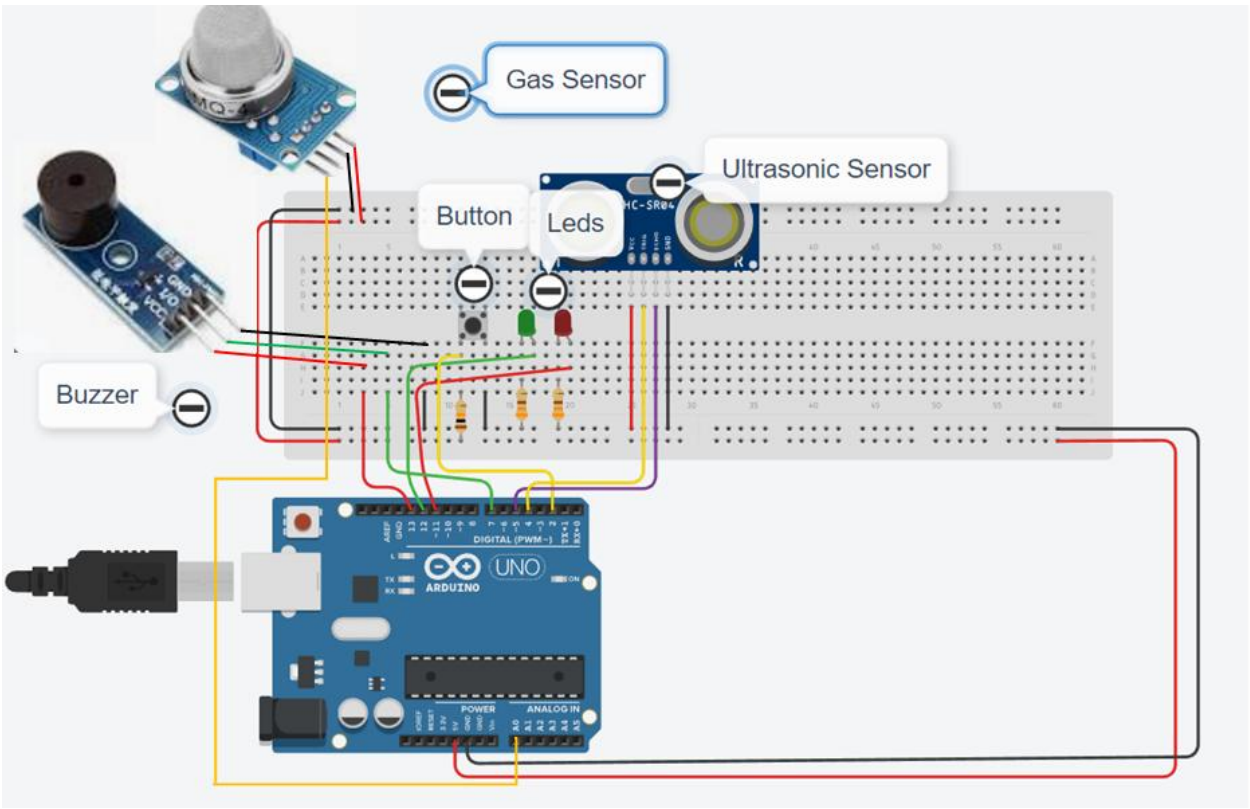
```
const int dashL = 3*dotL; // Dash = 3 x dot
const int sPause = dotL; // Sembol bekleme = 1 dot
const int lPause = dashL; // Harf bekleme = 3 dots
const int wPause = 7*dotL; // Kelime Bekleme = 7 dots

//Ultrasonik Sensör
#define trigPin 4
#define echoPin 5

//Buton
boolean buttonOn=false;

void setup()
{
pinMode(red_led,OUTPUT);
pinMode(buzz,OUTPUT);
pinMode(green_led,OUTPUT);

pinMode(buzz,OUTPUT); // Buzzer pinini çıkış olarak ayarlama
pinMode(buzzVcc,OUTPUT); //Buzzer besleme

//Ultrasonik Sensor
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);

pinMode (2, INPUT_PULLUP);

Serial.begin(9600);
}



void loop()
{
//Buton
 if (digitalRead(2)== LOW){
     buttonOn=true;
     digitalWrite(green_led, HIGH);
     }

//Ultrasonik Sensör kodu
long duration, distance;
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
```

```
digitalWrite(buzzVcc, LOW);

if (distance < 20)
  {
  while(buttonOn==true){

   if (digitalRead(2)== LOW){
      buttonOn=false;
      digitalWrite(red_led, LOW);
      digitalWrite(green_led, LOW);
      digitalWrite( buzz, LOW);
  }else{
  digitalWrite(red_led, HIGH);
  digitalWrite(green_led, LOW);
  digitalWrite( buzz, HIGH);



  //buzzer
  digitalWrite(buzzVcc, HIGH);
  digitalWrite(buzz, LOW); // Ton aktif
  delay(dashL); // Ton uzunluğu
  digitalWrite(buzz, HIGH); // Ton kapalı
  delay(sPause); // Sembol bekleme

  digitalWrite(buzz, LOW); // Ton aktif
  delay(dotL); // Ton uzunluğu
  digitalWrite(buzz, HIGH); // Ton kapalı
  delay(sPause); // Sembol bekleme

  digitalWrite(buzz, LOW); // Ton aktif
  delay(dashL); // Ton uzunluğu
  digitalWrite(buzz, HIGH); // Ton kapalı
  delay(sPause); // Sembol bekleme

  digitalWrite(buzz, LOW); // Ton aktif
  delay(dotL); // Ton uzunluğu
  digitalWrite(buzz, HIGH); // Ton kapalı
  delay(sPause); // Sembol bekleme

  delay(lPause-sPause); // Subtracts pause already taken

  digitalWrite(buzz, LOW); // Ton aktif
  delay(dashL); // Ton uzunluğu
  digitalWrite(buzz, HIGH); // Ton kapalı
  delay(sPause); // Sembol bekleme
  digitalWrite(buzz, LOW); // Ton aktif
  delay(dashL); // Ton uzunluğu
```

```
  digitalWrite(buzz, HIGH); // Ton kapalı
  delay(sPause); // Sembol bekleme

  digitalWrite(buzz, LOW); // Ton aktif
  delay(dotL); // Ton uzunluğu
  digitalWrite(buzz, HIGH); // Ton kapalı
  delay(sPause); // Sembol bekleme

  digitalWrite(buzz, LOW); // Ton aktif
  delay(dashL); // Ton uzunlupu
  digitalWrite(buzz, HIGH); // Ton kapalı
  delay(sPause); // Sembol bekleme
  delay(wPause-sPause); // Zaten alınmış olan duraklamayı çıkarır

  }  //end else
  } //end while
} //end if

delay(100);
}
```

Adım 3 (Gaz Sensörünün Alarm Sistemin eklenmesi)

Gaz Sensörü bağlantısı



Nasıl Çalışır:

Gaz dedektörü gazı algıladığında ve alarm sistemi etkinleştirilmiş olur, yeşil ışık söner, kırmızı ışık yanar ve zil çalar.

Adım 3' ün Arduino Kodu:

```
//Çalışma : Alarm Sistemi
// Gaz Sensörünün alarm sistemine eklenmesi
int red_led=11;
int green_led=12;
int gas_value;
int sensorThres=400;

//buzzer
int buzz= 13; // Buzzer giriş-çıkış pini buraya bağlanacak.
int buzzVcc= 7; //Buzzer besleme pini buraya bağlanacak.
const int wpm = 20; // WPM Mors hızı
const int dotL = 1200/wpm; // Hesaplanmış nokta uzunluğu
const int dashL = 3*dotL; // Dash = 3 x dot
```

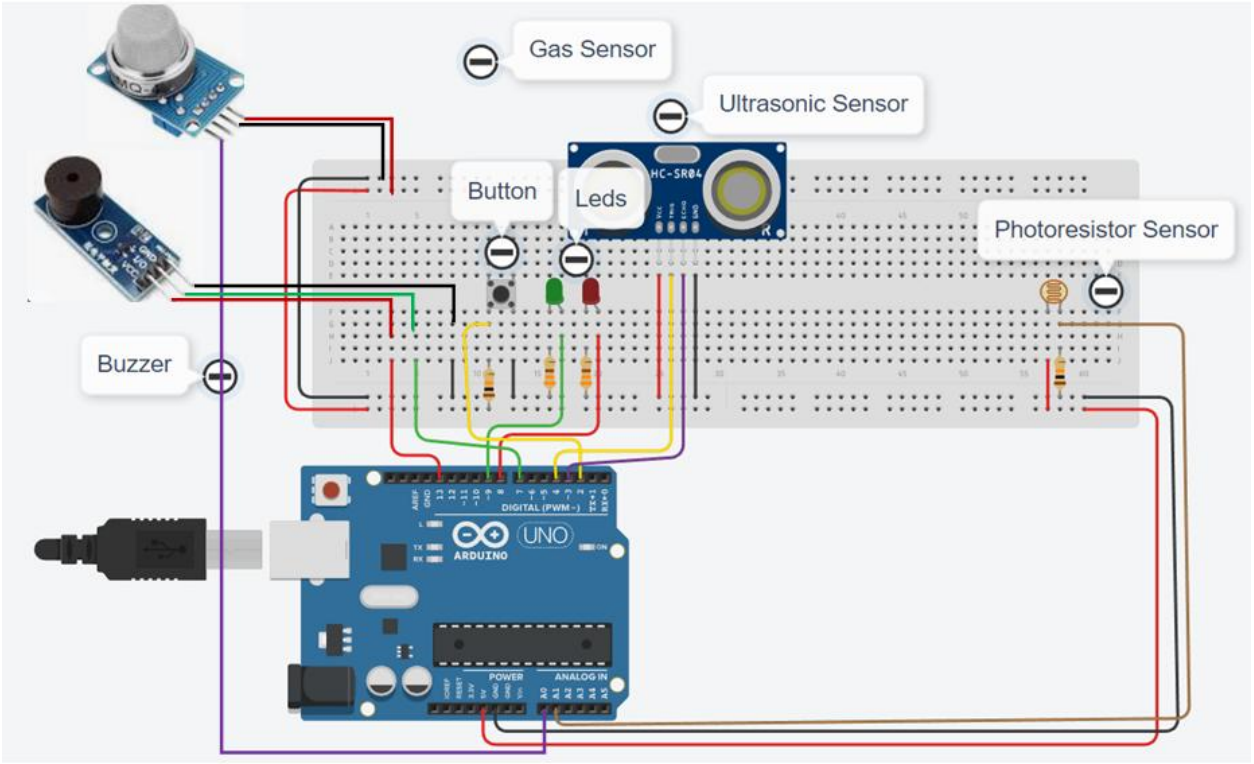
```
const int sPause = dotL; // Sembol bekleme = 1 dot
const int lPause = dashL; // Harf bekleme = 3 dots
const int wPause = 7*dotL; // Kelime bekleme = 7 dots

//Ultrasonik Sensör
#define trigPin 4
#define echoPin 5

//Buton
boolean buttonOn=false;

void setup()
{
pinMode(red_led,OUTPUT);
pinMode(green_led,OUTPUT);
pinMode(A0,INPUT); //A0 Gaz

pinMode(buzz,OUTPUT); // buzzer pinini çıkış olarak tanımlama
pinMode(buzzVcc,OUTPUT); // buzzer bekleme

//Ultrasonik Sensör
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);

pinMode (2, INPUT_PULLUP);

Serial.begin(9600);
}

void loop()
{
//Buton
 if (digitalRead(2)== LOW){
     buttonOn=true;
     digitalWrite(green_led, HIGH);
     Serial.println("The alarm system is enabled!");
     }

    //Ultrasonik Sensör kodu
long duration, distance;
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
```

```
digitalWrite(buzzVcc, LOW);

gas_value=analogRead(A0);

if (gas_value > sensorThres or distance < 20)
 {
  while(buttonOn==true){
  if (digitalRead(2)== LOW){
     buttonOn=false;
     digitalWrite(red_led, LOW);
     digitalWrite(green_led, LOW);
     digitalWrite( buzz, LOW);
     Serial.println("The alarm system is disabled!");
 }else{
 digitalWrite(red_led, HIGH);
 digitalWrite(green_led, LOW);
 digitalWrite( buzz, HIGH);
 Serial.println("DANGER!!!!");
 Serial.println(gas_value);

 //buzzer
 digitalWrite(buzzVcc, HIGH);
 digitalWrite(buzz, LOW); // Ton aktif
 delay(dashL); // Tone uzunluğu
 digitalWrite(buzz, HIGH); // Ton kapalı
 delay(sPause); // Sembol bekleme

 digitalWrite(buzz, LOW); // Ton aktif
 delay(dotL); // Ton uzunluğu
 digitalWrite(buzz, HIGH); // Tone kapalı
 delay(sPause); // Sembol bekleme

 digitalWrite(buzz, LOW); // Ton aktif
 delay(dashL); // Ton uzunluğu
 digitalWrite(buzz, HIGH); // Ton kapalı
 delay(sPause); // Sembol bekleme

 digitalWrite(buzz, LOW); // Ton aktif
 delay(dotL); // Ton uzunluğu
 digitalWrite(buzz, HIGH); // Ton kapalı
 delay(sPause); // Sembol bekleme

 delay(lPause-sPause); // Zaten alınmış olan duraklamayı çıkarır

 digitalWrite(buzz, LOW); // Ton aktif
 delay(dashL); // Ton uzunluğu
 digitalWrite(buzz, HIGH); // Ton kapalı
 delay(sPause); // Sembol bekleme
 digitalWrite(buzz, LOW); // Ton aktif
```

```
 delay(dashL); // Ton uzunluğu
 digitalWrite(buzz, HIGH); // Ton kapalı
 delay(sPause); // Sembol bekleme

 digitalWrite(buzz, LOW); // Ton aktif
 delay(dotL); // Ton uzunluğu
 digitalWrite(buzz, HIGH); // Ton kapalı
 delay(sPause); // Sembol bekleme

 digitalWrite(buzz, LOW); // Ton aktif
 delay(dashL); // Ton uzunluğu
 digitalWrite(buzz, HIGH); // Ton kapalı
 delay(sPause); // Sembol bekleme
 delay(wPause-sPause); // Zaten alınmış olan duraklamayı çıkarır
       } //end else
  }//end while
 } //end if

delay(100);
}
```

Adım 4 (Fotodirenç' in Alarm Sistemine eklenmesi)

FotoDirenç bağlantısı



Nasıl Çalışır:

Parlaklık azaldığında alarm sistemi devreye girer.

Adım 4 Arduino Kodu:

```
//Çalışma : ALARM SİSTEMİ
//FOTODİRENÇ' İN ALARM SİSTEMİNE EKLENMESİ
int red_led=8;
int green_led=9;
int gas_value;
int sensorThres=400;

//------------------buzzer-------------------------
int buzz= 13; // buzzer giriş-çıkış pini buraya bağlanacak.
int buzzVcc= 7; //buzzer besleme gerilimi buraya bağlanacak.
const int wpm = 20; // WPM Mors hızı
const int dotL = 1200/wpm; // Hesaplanmış nokta uzunluğu
const int dashL = 3*dotL; // Dash = 3 x dot
const int sPause = dotL; // Sembol bekleme = 1 dot
const int lPause = dashL; // Harf bekleme = 3 dots
const int wPause = 7*dotL; // Kelime bekleme = 7 dots

//------------Ultrasonik Sensör--------------------
```

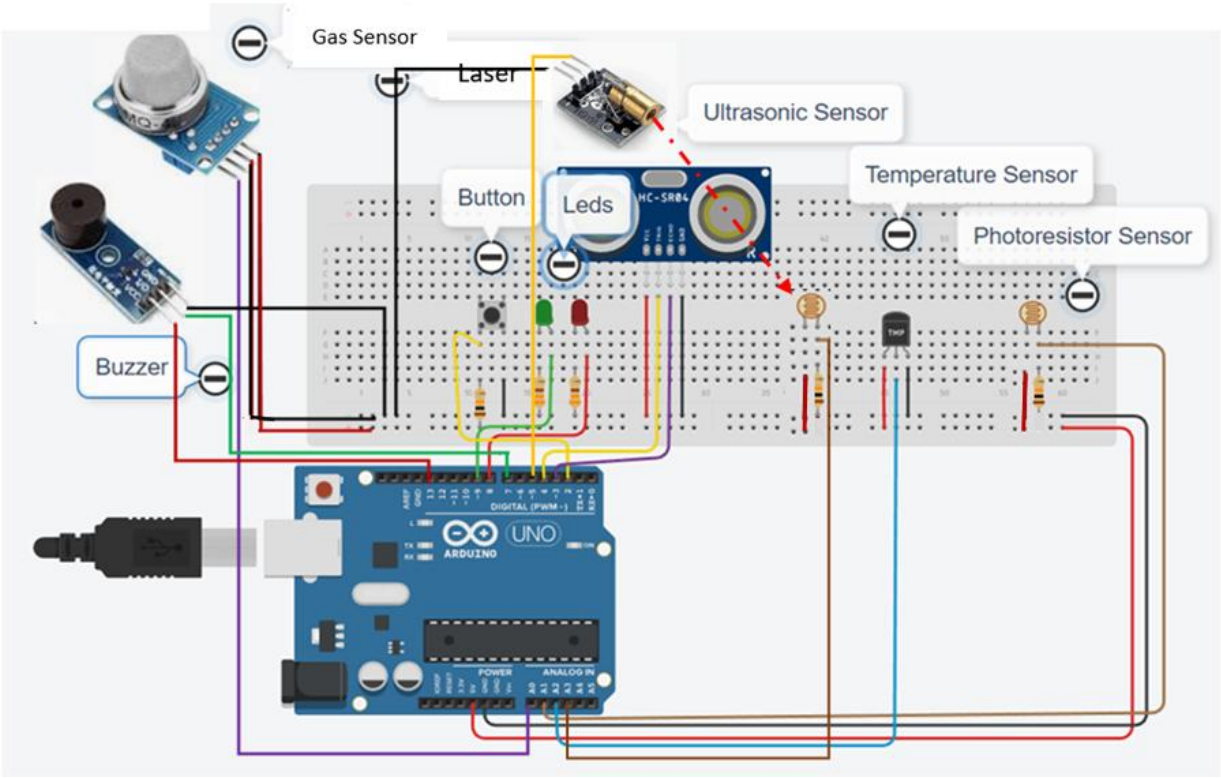
```
#define trigPin 4
#define echoPin 5

//------------------Buton-------------------------
boolean buttonOn=false;

void setup()
{
pinMode(red_led,OUTPUT);
pinMode(green_led,OUTPUT);
digitalWrite(red_led, LOW);
digitalWrite(green_led, LOW);
pinMode(A0,INPUT); //A0 Gass

pinMode(buzz,OUTPUT); // buzzer pinin çıkış olarak ayarlama
pinMode(buzzVcc,OUTPUT); //Buzzer besleme

//Ultrasonik Sensör
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);

pinMode (2, INPUT_PULLUP);

Serial.begin(9600);
}
void loop()
{
  //-------------fotodirenç---------------
  int valuePhotor = analogRead(A1);
   Serial.println("Analog valuePhotor : ");
   Serial.println(valuePhotor);
   delay(250);

  //------------------Buton------------------
   if (digitalRead(2)== LOW or valuePhotor<110){
       buttonOn=true;
       digitalWrite(red_led, LOW);
       digitalWrite(green_led, HIGH);
       }

  //-------------Ultrasonik Sensör--------------
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
```

```
Serial.println("Distance");
Serial.println(distance);

digitalWrite(buzzVcc, LOW);

gas_value=analogRead(A0);

if (gas_value > sensorThres or distance < 20){
  while(buttonOn==true){
    if (digitalRead(2)== LOW){
      buttonOn=false;
      digitalWrite(red_led, LOW);
      digitalWrite(green_led, LOW);
      digitalWrite( buzz, LOW);
      Serial.println("NO LEAKAGE");
      Serial.println(gas_value);
   }else{
   digitalWrite(red_led, HIGH);
   digitalWrite(green_led, LOW);
   digitalWrite( buzz, HIGH);
   Serial.println("DANGER!!!!");
   Serial.print("Gas Value: ");
   Serial.println(gas_value);
   Serial.print("Distance: ");
   Serial.println(distance);

   //buzzer
   digitalWrite(buzzVcc, HIGH);
   digitalWrite(buzz, LOW); // Ton aktif
   delay(dashL); // Ton uzunluğu
   digitalWrite(buzz, HIGH); // Ton kapalı
   delay(sPause); // Sembol bekleme

   digitalWrite(buzz, LOW); // Ton aktif
   delay(dotL); // Ton uzunluğu
   digitalWrite(buzz, HIGH); // Ton kapalı
   delay(sPause); // Sembol bekleme

   digitalWrite(buzz, LOW); // Ton aktif
   delay(dashL); // Ton uzunluğu
   digitalWrite(buzz, HIGH); // Ton kapalı
   delay(sPause); // Sembol bekleme

   digitalWrite(buzz, LOW); // Ton aktif
   delay(dotL); // Ton uzunluğu
   digitalWrite(buzz, HIGH); // Ton kapalı
   delay(sPause); // Sembol bekleme

   delay(lPause-sPause); // Zaten alınmış olan duraklamayı çıkarır
```

```
    digitalWrite(buzz, LOW); // Ton aktif
    delay(dashL); // Ton uzunluğu
    digitalWrite(buzz, HIGH); // Ton kapalı
    delay(sPause); // Sembol bekleme
    digitalWrite(buzz, LOW); // Ton aktif
    delay(dashL); // Ton uzunluğu
    digitalWrite(buzz, HIGH); // Ton kapalı
    delay(sPause); // Sembol bekleme

    digitalWrite(buzz, LOW); // Ton aktif
    delay(dotL); // Ton uzunluğu
    digitalWrite(buzz, HIGH); // Ton kapalı
    delay(sPause); // Sembol bekleme

    digitalWrite(buzz, LOW); // Ton aktif
    delay(dashL); // Ton uzunluğu
    digitalWrite(buzz, HIGH); // Ton kapalı
    delay(sPause); // Sembol bekleme
    delay(wPause-sPause); // Zaten alınmış olan duraklamayı çıkarır
    } //end else
  } //end while
 } //end if
 delay(100);
}
```

Adım 5 (Sıcaklık Sensörü ve Lazer Verici Modülünün Alarm Sistemine eklenmesi)

Sıcaklık Sensörü bağlantısı



Bu adımda bir lazer vericiye, alıcı için bir fotorezistöre ve onu korumak için 1KOhm dirence ihtiyacımız olacak.

Nasıl Çalışır:

Bir alıcı olarak çalışan fotodirenç sensörü ile birlikte lazer verici, Arduino'ya herhangi bir nedenle alım kesintiye uğradığında sistemi açması talimatını verir. Ayrıca sıcaklık sensörünün değeri belirli bir değeri aştığında Alarm Sistemi devreye girer.

Adım 5 Arduino Kodu:

```
//Çalışma : ALARM SİSTEMİ
//SICAKLIK SENSÖRÜNÜN SİSTEME EKLENMESİ
int red_led=8;
int green_led=9;
int gas_value;
int sensorThres=400;

//----------------buzzer--------------------------
int buzz= 13; // buzzer giriş-çıkış pinleri buraya bağlanacak.
int buzzVcc= 7; //Buzzer besleme pini buraya bağlanacak.
```

```
const int wpm = 20; // WPM Mors Hızı
const int dotL = 1200/wpm; // Hesaplanmış nokta uzunluğu
const int dashL = 3*dotL; // Dash = 3 x dot
const int sPause = dotL; // Sembol bekleme = 1 dot
const int lPause = dashL; // Harf bekleme = 3 dots
const int wPause = 7*dotL; // Keline bekleme = 7 dots

//-------------Ultrasonik Sensör------------------
#define trigPin 3
#define echoPin 4

//----------------Buton--------------------------
boolean buttonOn=false;

//-----------Sıcaklık Sensörü-------------------
float tempf;
int tempPin = A2;

//-----------Lazer-------------------
int Laser = 5; // Laser isimli bir değişken oluştur ve 5. Pine ata
int valueLaserPh=0;// valueLaserPh isminde bir değişken oluştur ve değerini 0 ata.
float valueLaserPhF=0;// valueLaserPhF isminde bir değişken oluştur ve değerini 0 ata.

// oda sıcaklığı
const float baselineTemp = 100.0;

void setup()
{
  pinMode(red_led,OUTPUT);
  pinMode(buzz,OUTPUT);
  pinMode(green_led,OUTPUT);
  pinMode(A0,INPUT); //A0 Gas

  pinMode(buzz,OUTPUT); // buzzer pinini çıkış pini olarak ata.
  pinMode(buzzVcc,OUTPUT); // Buzzer beslemesi

  //Ultrasonik Sensör
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

//-----------Lazer-------------------
  pinMode (Laser,OUTPUT); // çıkış için 5 numaralı pini ata
```

```
  digitalWrite(Laser,LOW); // sadece lazerin başlangıçta veya sıfırlamada kapalı olduğundan emin olun

  pinMode (2, INPUT_PULLUP);

  //sıcaklık sensörü
  pinMode(tempPin,INPUT);

  Serial.begin(9600);
}


void loop()
{
  digitalWrite(red_led, LOW);
  digitalWrite(Laser,HIGH);
  //-----------------Laser-------------------------------
  valueLaserPh=analogRead(A4); // A4 üzerindeki voltajı okuma ve alınan değeri "voltaj" olarak kaydetme
  valueLaserPhF = valueLaserPh * (5.0 / 1023.0); // "voltaj"da saklanan değeri okunabilir bilgiye dönüştürmek

  //----------sıcaklık sensörü--------------------------
  tempf = analogRead(tempPin);
  tempf=(tempf*5)/10;
  // analog voltajı sıcaklık eşdeğerine dönüştür.
  Serial.print("TEMPERATURE = ");
  Serial.print(tempf); // sıcaklık değerini görüntüle
  Serial.print("*F");
  Serial.println();
  delay(1000); // her bir saniyede sensör verisini güncelle.

 //------------fotoresistör-----------------
  int valuePhotor = analogRead(A1);
  Serial.print("Photoresistor value : ");
  Serial.println(valuePhotor);
  delay(250);

//----------------Buton----------------------
 if (digitalRead(2)== LOW or valuePhotor<20){
     buttonOn=true;
     digitalWrite(red_led, LOW);
```

```
    digitalWrite(green_led, HIGH);
    digitalWrite(Laser,HIGH); // lazeri aç
    Serial.println("The alarm system is enabled!");
    }

//------------Ultrasoc Sensor -----------
long duration, distance;
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
Serial.print("distance : ");
Serial.println(distance);
digitalWrite(buzzVcc, LOW);
gas_value=analogRead(A0);
Serial.print("Gas Value : ");
Serial.println(gas_value);
Serial.print("Laser value : ");
Serial.println(valueLaserPhF);

if (gas_value > sensorThres or distance < 10 or tempf>baselineTemp or valueLaserPhF<1)
 {
  while(buttonOn==true){
   if (digitalRead(2)== LOW){
     buttonOn=false;
     //digitalWrite(red_led, LOW);
     digitalWrite(green_led, LOW);
     digitalWrite( buzz, LOW);
     Serial.println("The alarm system is disabled!");
     distance=100;
     tempf=0.0;
  }else{
       digitalWrite(red_led, HIGH);
       digitalWrite(green_led, LOW);
       digitalWrite( buzz, HIGH);
       Serial.println("DANGER!!!!");
       Serial.print("Gas Value: ");
       Serial.println(gas_value);
       Serial.print("Distance: ");
```

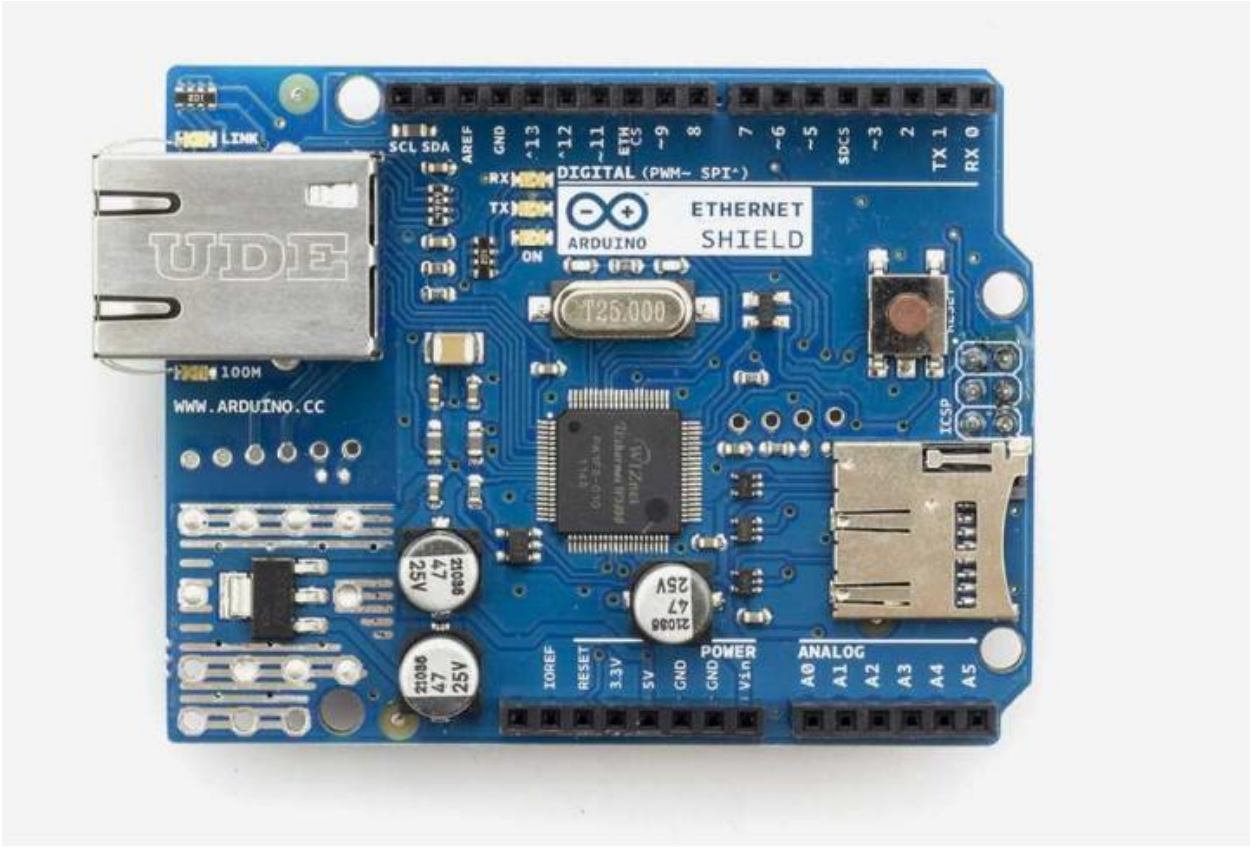
```
Serial.println(distance);
Serial.print("Temperature: ");
Serial.println(tempf);
Serial.print("Photoresistor value : ");
Serial.println(valuePhotor);
Serial.print("Laser value : ");
Serial.println(valueLaserPhF);

//--------------buzzer--------------
digitalWrite(buzzVcc, HIGH);
digitalWrite(buzz, LOW); // Ton aktif
delay(dashL); // Ton uzunluğu
digitalWrite(buzz, HIGH); // Ton kapalı
delay(sPause); // Sembol bekleme

digitalWrite(buzz, LOW); // Ton aktif
delay(dotL); // Ton uzunluğu
digitalWrite(buzz, HIGH); // Ton kapalı
delay(sPause); // Sembol bekleme

digitalWrite(buzz, LOW); // Ton aktif
delay(dashL); // Ton uzunluğu
digitalWrite(buzz, HIGH); // Ton kapalı
delay(sPause); // Sembol bekleme

digitalWrite(buzz, LOW); // Ton aktif
delay(dotL); // Ton uzunluğu
digitalWrite(buzz, HIGH); // Ton kapalı
delay(sPause); // Sembol bekleme

delay(lPause-sPause); // Zaten alınmış olan duraklamayı çıkarır

digitalWrite(buzz, LOW); // Ton aktif
delay(dashL); // Ton uzunluğu
digitalWrite(buzz, HIGH); // Ton kapalı
delay(sPause); // Sembol bekleme
digitalWrite(buzz, LOW); // Ton aktif
delay(dashL); // Ton uzunluğu
digitalWrite(buzz, HIGH); // Ton kapalı
delay(sPause); // Sembol bekleme

digitalWrite(buzz, LOW); // Ton aktif
```

```
        delay(dotL); // Ton uzunluğu
        digitalWrite(buzz, HIGH); // Ton kapalı
        delay(sPause); // Sembol bekleme

        digitalWrite(buzz, LOW); // Ton aktif
        delay(dashL); // Ton uzunluğu
        digitalWrite(buzz, HIGH); // Ton kapalı
        delay(sPause); // Sembol bekleme
        delay(wPause-sPause); // Zaten alınmış olan duraklamayı çıkarır
      }//end else
   }//end while
  } //end if
delay(100);
}
```

PROJE 3- ADI: Web Sunucu

Projenin Amacı: Bu projede öğrenciler bir arduino'yu ethernet kalkanı kullanarak yerel bir ağa nasıl bağladığımızı görecekler. Sensörler ile deneyler yapacaklar ve elde ettikleri değerleri yerel ağ üzerinden bilgisayar ekranlarında görebilecekler.

Bu projede öğrencilerin kullanarak deneyimleyecekleri malzemeler:



- ✓ Bir adet gaz sensörü
- ✓ Bir adet fotodirenç
- ✓ Bir adet sıcaklık sensörü

Bu Ethernet kalkanı, bir Arduino kartının internete bağlanmasını sağlar.

Malzeme Listesi:

- Ethernet Koruması
- Ethernet Kalbosu
- Breadboard and Atlama Kablolaları
- USB Kablosu
- Arduino UNO R3
- Bir adet Fotodirenç

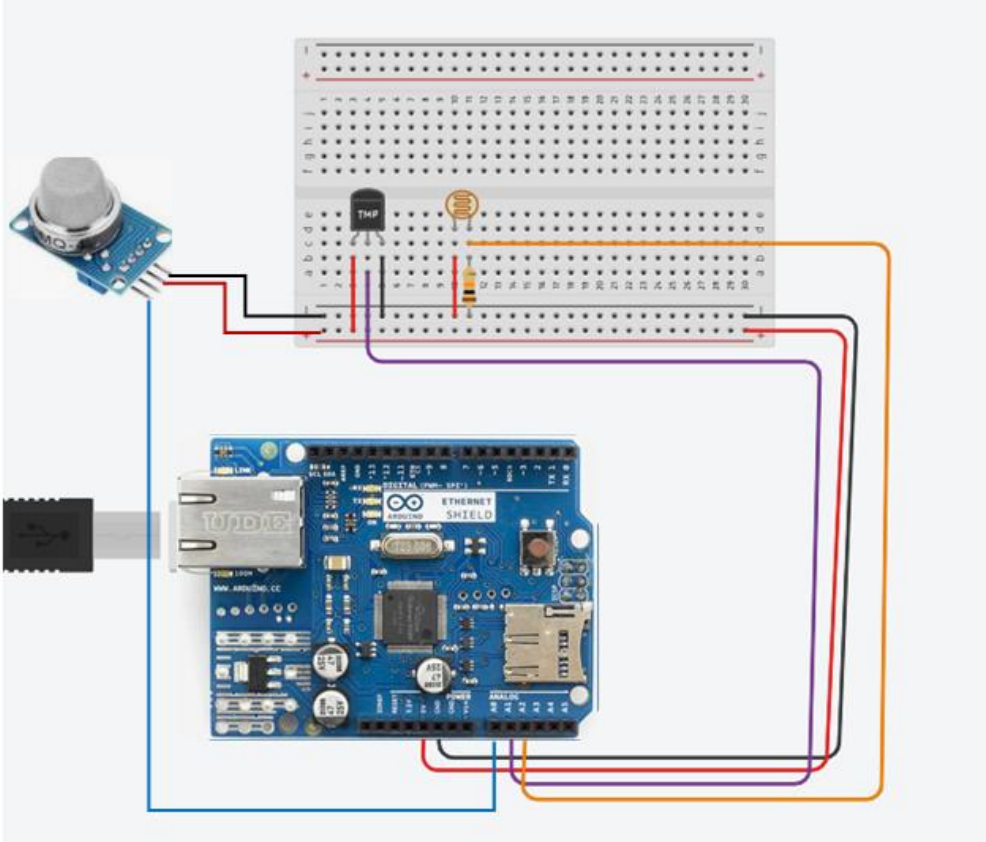
- Bir adet gaz sensörü
- Bir adet sıcaklık sensörü
- Bir adet 10Kohm direnç

Nasıl Çalışır:

Arduino Ethernet Shield 2, bir Arduino Kartının internete bağlanmasına izin verir. Bunu Wiznet W5500 Ethernet çipi sağlamaktadır. Wiznet W5500, hem TCP hem de UDP yapabilen bir ağ (IP) yığını sağlar. Sekiz adede kadar eşzamanlı soket bağlantısını destekler.

Bu örnekte öğrenciler ethernet kalkanının ıp'sini bir yerel ağ tarayıcısında arayacak ve ethernet kalkanına bağladıkları sensörlerin değerlerini dostça bir sayfada görebileceklerdir. Sayfa her 5 saniyede bir otomatik olarak yenilenir.

Proje Devresi



Metodoloji

İlk olarak, geliştirilecek projenin istenen iş akışını adım adım açıklıyoruz. Öğrencilerden daha sonra devreyi tasarlamak ve çizmek için gerekli bileşenleri seçmeleri istenir. Kodu yazmak için "Arduino" uygulamasını kullanacaklar.

Ardından devreyi oluşturacak, kodu Arduino yazılım aracına yazacak ve Arduino kartına yükleyecekler.

Öğrenciler devrenin doğru yapıldığını doğrulayacaktır.

Bu proje, Erasmus+KA-202 Mesleki Eğitim ve Öğretimde Stratejik Ortaklıklar projemizden "ArduİnVet" adı verilen sonuç ve sonuçlardan yararlanmaya başlayan temel düzey öğrenciler içindir.

Projenin Arduino Kodu:

```
/*
  Web Sunucu

 Arduino Wiznet Ethernet kalkanı kullanarak, analog giriş pinlerinin değerini gösteren
basit bir web sunucusu.


 Devre Şeması:

* 10, 11, 12, 13 pinlerine bağlı Ethernet kalkanı
 * A0 ile A5 arasındaki pinlere bağlı analog girişler (isteğe bağlı)
 */

#include <SPI.h>
#include <Ethernet.h>
unsigned long refreshCounter  = 0;
// Kontrol cihazınız için aşağıya bir MAC adresi ve IP adresi girin.
// IP adresi yerel ağınıza bağlı olacaktır
byte mac[] = {
  0xA8, 0x61, 0x0A, 0xAE, 0x63, 0xA8
};
IPAddress ip(169, 254, 122, 108);

// Ethernet sunucu kütüphanesini
// kullanmak istediğiniz port ve ip adresi ile yükleyin.
// (port 80 is default for HTTP):
EthernetServer server(80);

EthernetClient client;
void setup() {
  // You can use Ethernet.init(pin) to configure the CS pin
  //Ethernet.init(10);  // Most Arduino shields
```

```
 //Ethernet.init(5);   // MKR ETH shield
 //Ethernet.init(0);   // Teensy 2.0
 //Ethernet.init(20);  // Teensy++ 2.0
 //Ethernet.init(15);  // ESP8266 with Adafruit Featherwing Ethernet
 //Ethernet.init(33);  // ESP32 with Adafruit Featherwing Ethernet

 // Seri iletişimi açın ve bağlantı noktasının açılmasını bekleyin:
 Serial.begin(9600);
 while (!Serial) {
        ; // wait for serial port to connect. Needed for native USB port only
 }
 Serial.println("ARDUinVET in Ethernet WebServer ");

 // Ethernet bağlantısını ve sunucuyu başlat:
 Ethernet.begin(mac, ip);

 // Ethernet donanımını kontrol et.
 if (Ethernet.hardwareStatus() == EthernetNoHardware) {
   Serial.println("Ethernet shield was not found.  Sorry, can't run without hardware. :(");
        while (true) {
                delay(1); // do nothing, no point running without Ethernet hardware
        }
 }
 if (Ethernet.linkStatus() == LinkOFF) {
        Serial.println("Ethernet cable is not connected.");
 }
 // start the server
 server.begin();
 Serial.print("server is at ");
 Serial.println(Ethernet.localIP());
}

void loop() {
    // listen for incoming clients
   client = server.available();
   if (client) {
        Serial.println("ARDUinVET in client");
       // an http request ends with a blank line
       boolean currentLineIsBlank = true;
       while (client.connected()) {
           if (client.available()) {
              char c = client.read();
             Serial.write(c);
              // satırın sonuna geldiyseniz (yeni satır aldıysanız
             // karakter) ve satır boş, http isteği sona erdi,
             // bir cevap gönderebilmeniz için
                if (c == '\n' && currentLineIsBlank) {
                    // send a standard http response header
                    client.println("HTTP/1.1 200 OK");
```

```
                client.println("ARDinVET - HTTP");
                client.println("Content-Type: text/html");
                client.println("Connection: close");  // yanıtın tamamlanmasından sonar
bağlantıyı kapat
                client.println("Refresh: 5");  // Her 5 saniyede sayfayı otomatik olarak
yenile
                client.println();
                webpage(client);
                break;
                }
                if (c == '\n') {
                        // yeni satır başlat.
                        currentLineIsBlank = true;
                        } else if (c != '\r') {
                                // mevcut satırta bir karakter varsa
                                currentLineIsBlank = false;
                          }
                }
            }
            // web tarayıcıya veri alması için zaman ver
            delay(1);
            // bağlantıyı kapat:
            client.stop();
            Serial.println("client disconnected");
        }
}

void webpage(EthernetClient client) { /* function webpage */
////Send wbepage to client
client.println("ARDUinVET - HTTP");
//client.println("Content-Type: text/html");
client.println(""); //  do not forget this one
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head>");
client.println("<title>ARDUinVET</title>");
//client.println("<script src='https://smtpjs.com/v3/smtp.js'></script>");
//client.println("<script>");
//client.println("function sendEmail() {");
/*client.println("Email.send({");
client.println("  Host: 'smtp.gmail.com',");
client.println("  Username : '**********@gmail.com',");
client.println("  Password : '**********',");
client.println("  To : '**********@gmail.com',");
client.println("  From : '**********@gmail.com',");
client.println("  Subject : 'Data from Arduino',");
client.println("  Body : 'Sensors value....',");
client.println("  }).then(");
client.println("    message => alert('mail sent successfully')");
```

```
client.println("  );");*/
//client.println("}");
//client.println("</script>");
client.println("</head>");
client.println("<body bgcolor = '#cccc00'>");
client.println("<hr/><hr>");
client.println("<h1 style='color : #0000cc;'><center>  Alarm System </center></h1>");
client.println("<hr/><hr>");
client.println("<br><br>");
client.println("<br><br>");
client.println("<center>");
client.println("<br>Sensors Output<br>");
client.println("Gas.....:");
client.println("  <input value=" + String(analogRead(A0)) + " readonly></input>");
client.println("<br>");
client.println("  Photoresistor..:");
client.println("  <input value=" + String(analogRead(A1)) + " readonly></input>");
client.println("<br>");
client.println("  Temperature....:");
client.println("  <input value=" + String(analogRead(A2)) + " readonly></input>");
client.println("<br>");
client.println("  </center>");
client.println("<br><br>");
client.println("<center>");
client.println("<a style='color : #0000cc;'
href='https://www.arduinvet.com/'>www.arduinvet.com</a>");
client.println("</center>");
client.println("<br><br>");
client.println("</body></html>");
client.println();
delay(1);
}
```

Proje Sayfasının ekran görüntüsü

Erasmus+ KA-202

Mesleki ve Teknik Eğitime Yönelik Stratejik Ortaklıklar Projesi

Project Title: "Mesleki Eğitimde Arduino Öğrenme ve Öğretme"

Project Acronym: " ARDUinVET "

Project No: "2020-1-TR01-KA202-093762"

Serbest Arduino Projeleri

(Bu projeler, ARDUinVET projesi kapsamında, ortak okulların öğrencileri tarafından yapılmıştır.)

NO:	Project NAME	Country
1	SOSYAL MESAFELİ ŞAPKASI	TÜRKİYE
2	SICAKLIK ÖLÇER	YUNANİSTAN
3	ACİL FREN FLAŞÖRÜ – UYARLANABİLİR FREN LAMBASI	ROMANYA
4	ÇİZGİ TAKİP EDİCİ	AVUSTURYA
5	SERA OTOMASYONU	ITALYA

1_PROJE ADI: SOSYAL MESAFELİ ŞAPKASI (TÜRKİYE)

Projenin Amacı: Covid – 19 pandemisinin etkili olduğu bu günlerde "SOSYAL MESAFE" kuralına uymak, hastalığa yakalanmamanın temel koşullarından biridir.

"Sosyal Mesafe" konusuna dikkat çekmek için konuyla ilgili bir Arduino projesi tasarlamaya karar verdik.

Hazırladığımız "Sosyal Mesafeli Şapkası" bu kuralı hatırlatıyor. Sesli ve aynı zamanda ışıklı uyarı verecektir.

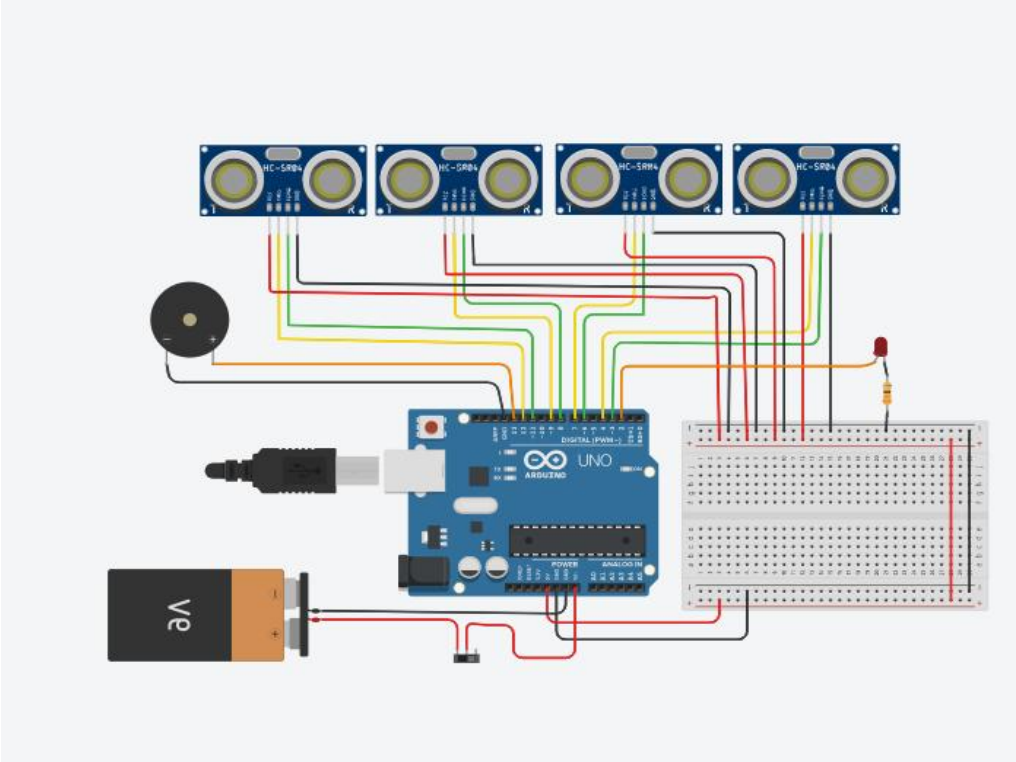
Proje Metodumuz:

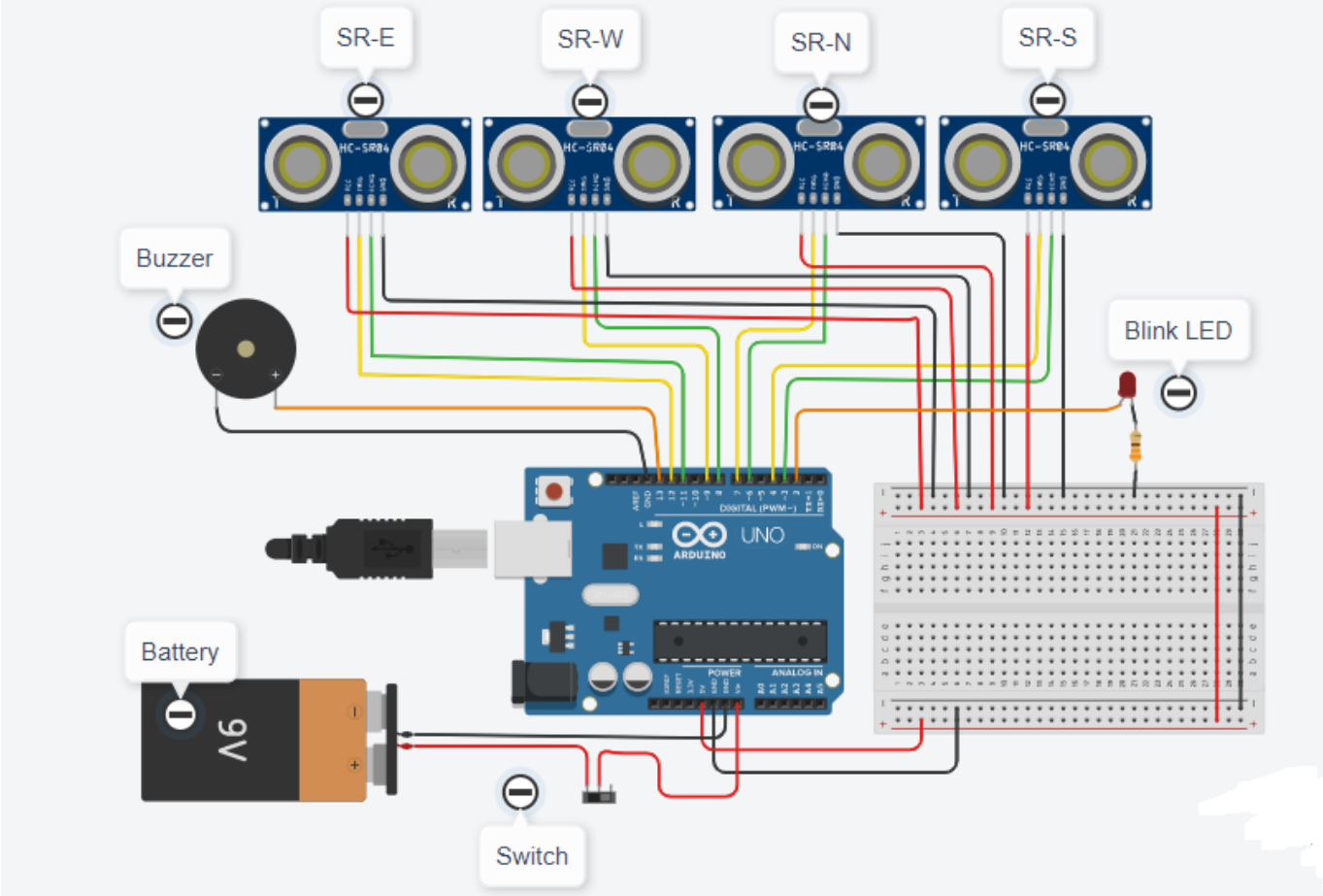
İlk olarak projenin algoritması hazırlandı. Hazırlanan bu algoritmaya göre mantıksal tasarım yapıldı ve Arduino programı yazıldı. Devre tasarımında Arduino kullanılmıştır. Öğrencilerimiz bu tasarımları yaparken "ArduinVET" olarak adlandırılan Erasmus+ KA-202 VET projesinin tüm sonuçlarından ve ürünlerinden yararlanmıştır.

Gerekli ekipmanları sağlanması ve Arduino' nun programlanmasından sonar kesinleşen devre tasarımı kuruldu. Kesinleşen devre tasarımı şapkanın üzerine yerleştirildi.

Fuarlarda sergilenen bu proje, insanlara "Sosyal Mesafe" kuralını ve önemini hatırlatıyor.

Proje Devresi





Nasıl Çalışır:

Şapkanın 4 yönüne 4 adet ultrasonik mesafe sensörü yerleştirilmiştir. Sensörler, 100 cm veya altına yaklaşan veya yakın birisini algıladığında sesli uyarı verir ve led yanıp söner. Devre 9V pil ile beslenecektir.

Malzeme Listesi:

Şapka, Arduino Uno R3, 4 Ultrasonik Mesafe Sensörü, Buzzer, 330 Ω Direnç, Blink Kırmızı LED, 9V Pil, Swiç Anahtar.

Projenin Aurdino Programı:

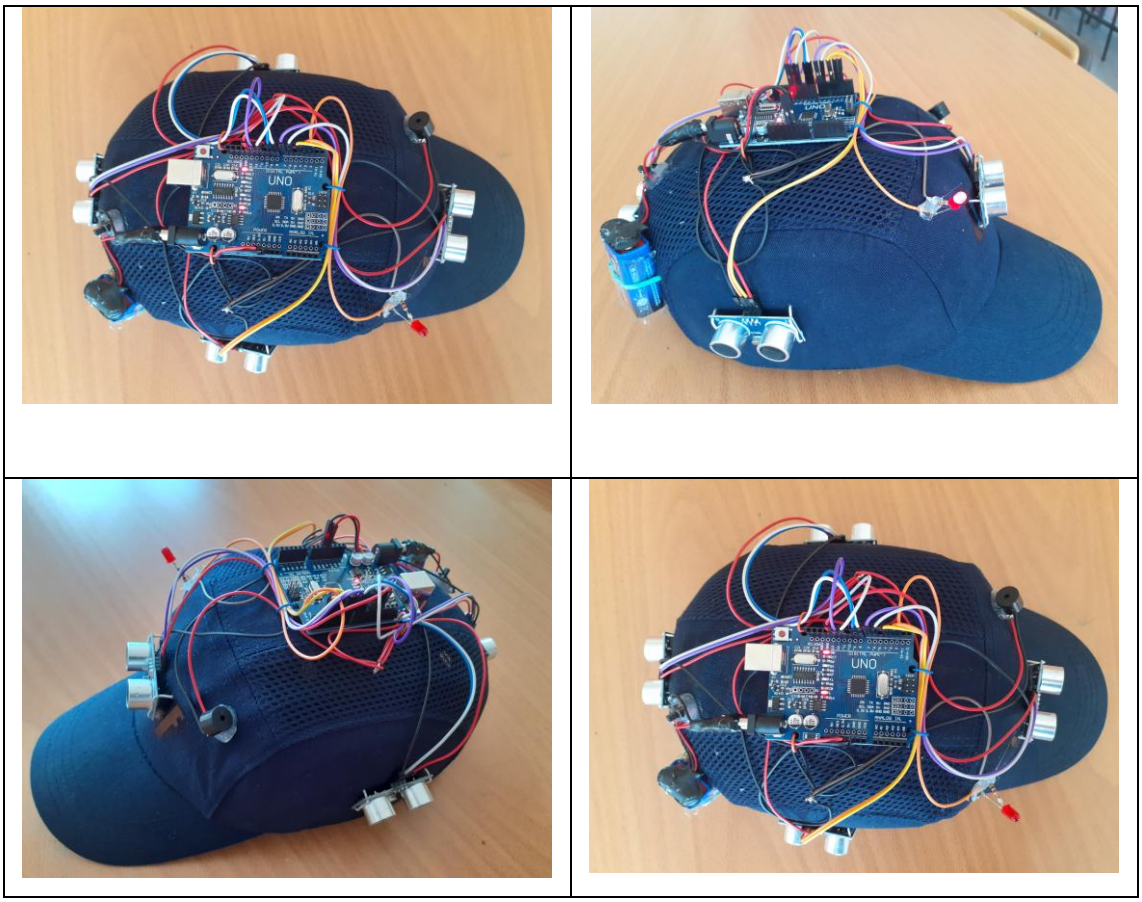
//Proje Adı: SOSYAL MESAFELİ ŞAPKA:

```
int trigPin=12;
int echoPin=11;
int trigPin2=9;
int echoPin2=8;
int trigPin3=7;
int echoPin3=6;
int trigPin4=4;
int echoPin4=3;
void setup()
{
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
 pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);
pinMode(trigPin3, OUTPUT);
pinMode(echoPin3, INPUT);
 pinMode(trigPin4, OUTPUT);
pinMode(echoPin4, INPUT);
pinMode(13,OUTPUT);
pinMode(2,OUTPUT);
 Serial.begin(9600);
}
void loop() {
```

```
 digitalWrite(trigPin, LOW);
delay(3);
digitalWrite(trigPin, HIGH);
delay(3);
digitalWrite(trigPin, LOW);
int time1 = pulseIn(echoPin, HIGH);
int distance1 = (time1/2) / 28.97;         //Sosyal Mesafe 100 cm' ye eşit veya daha kısa ise.
//-----------------------------------------------------
 digitalWrite(trigPin2, LOW);
delay(3);
digitalWrite(trigPin2, HIGH);
delay(3);
digitalWrite(trigPin2, LOW);

int time2 = pulseIn(echoPin2, HIGH);
int distance2 = (time2/2)  / 28.97;    // Sosyal Mesafe 100 cm' ye eşit veya daha kısa ise.
//-----------------------------------------------------
 digitalWrite(trigPin3, LOW);
delay(3);
digitalWrite(trigPin3, HIGH);
delay(3);
digitalWrite(trigPin3, LOW);
int time3 = pulseIn(echoPin3, HIGH);
int distance3 = (time3/2)  / 28.97;       // Sosyal Mesafe 100 cm' ye eşit veya daha kısa ise.
//-----------------------------------------------------
digitalWrite(trigPin4, LOW);
```

```
delay(3);
digitalWrite(trigPin4, HIGH);
delay(3);
digitalWrite(trigPin4, LOW);
int time4 = pulseIn(echoPin4, HIGH);
int distance4 = (time4/2)  / 28.97;         // Sosyal Mesafe 100 cm' ye eşit veya daha kısa ise.
//-----------------------------------------------------
if(distance1<75)
 {
  digitalWrite(13,HIGH);
  digitalWrite(2,HIGH);
  }
 else if(distance2<75)
 {
  digitalWrite(13,HIGH);
   digitalWrite(2,HIGH);
  }
 else if(distance3<75)
 {
  digitalWrite(13,HIGH);
   digitalWrite(2,HIGH);
  }
 else if(distance4<75)
 {
  digitalWrite(13,HIGH);
   digitalWrite(2,HIGH);
```

```
 }
 else
 {
 digitalWrite(13,LOW);
 digitalWrite(2,LOW);
 }
  Serial.println(distance1); // Seri monitörde hangi ultrasonik sensörün aktifleştiğini görmek için.
  Serial.println(distance2);
  Serial.println(distance3);
  Serial.println(distance4);
  delay(500);
}
```



Fotoğraf: Şapkanın son devre tasarımı

2_PROJE ADI: SICAKLIK ÖLÇER (YUNANİSTAN)

Projenin Amacı: Bu proje, birisinin orijinal Arduino el kitabında bulabileceği basit temel Love-o-Meter projesinin genişletilmiş bir versiyonudur. Bu genişletilmiş versiyonda, öğrenciler bir sensör (sıcaklık sensörü) ile uğraşabilecek, potansiyometre ile deney yapabilecek, bir RGB LED yakabilecek ve LCD ekranı tanıyabilecek (eğer mevcut değilse), aksi takdirde sadece Arduino yazılım aracındaki seri monitörde görüntüleyeceklerdir.

Öğrencilerin farklı türlerdeki birçok bileşeni denemelerine, girdi değerleri ve görüntüleme yöntemlerine aşina olmalarına olanak tanıyan bir projedir.

Metodoloji

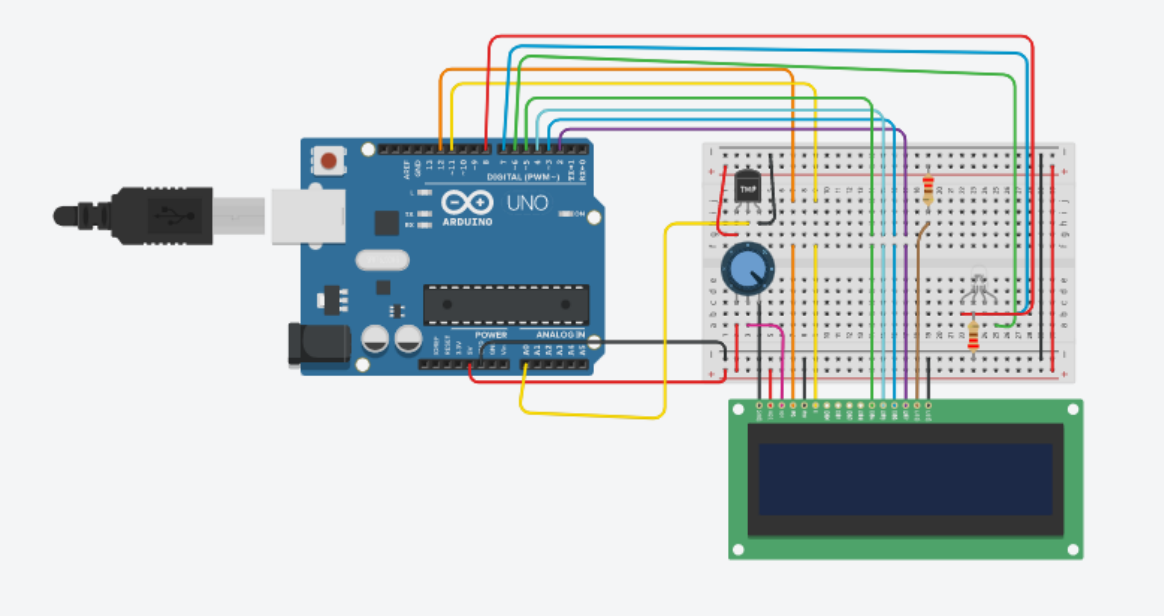
İlk olarak projenin gerekli iş akışını tanımlıyoruz. Daha sonra öğrencilerden gerekli bileşenleri seçerek ve Thinkercad kullanarak devreyi tasarlamalarını ve çizmelerini istiyoruz. Kodu yazmak için Thinkercad kod aracını kullanacaklar.

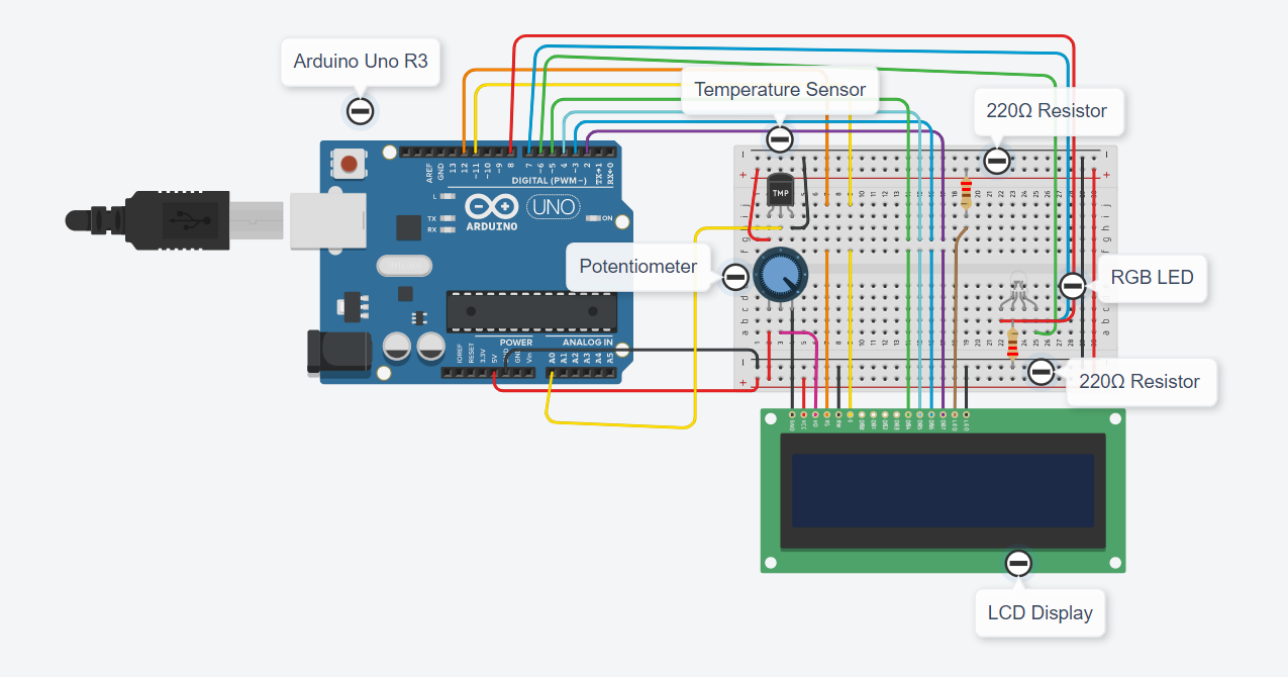
Sonra devreyi kuracaklar, Arduino yazılım aracına kodu yazacak ve Arduino kartına yükleyeceklerdir.

Thinkercad'de ki simülasyon, devrenin doğru yapıldığını doğrulayacaktır.

Bu proje, "ArduinVET" adlı Erasmus+ KA-202 Mesleki Eğitimde Stratejik Ortaklıklar projemizin sonuç ve çıktılarından yararlanmaya başlayan temel düzeydeki öğrenciler içindir.

Proje Devre Şeması.





Nasıl Çalışır:

Sıcaklık sensörü oda sıcaklığı değerlerini giriş olarak alır. Tabiki öğrenciler sensor sıcaklığını parmaklarını kullanarak değiştirebilirler. Sıcaklık, LCD Ekranda (varsa) ve Seri Monitörde görüntülenecektir. RGB LED, sıcaklık değerine göre farklı renklere (kapalı, yeşil, mavi, kırmızı) sahip olacaktır. Potansiyometre, LCD Ekran için bir anahtar olarak kullanılacaktır. Devre USB Kablosundan güç alacaktır (gerekirse bunun yerine bir pil kutusu kullanabilirsiniz).

Malzeme Listesi:

Bir adet Arduino Uno R3, bir sıcaklık sensörü, iki adet 220Ω direnç, bir RGB LED ve (opsiyonel) bir adet potansiyometre ve bir adet LCD Ekran (16*2).

Projenin Arduino Kodu:

//Proje Adı: SICAKLIK ÖLÇER

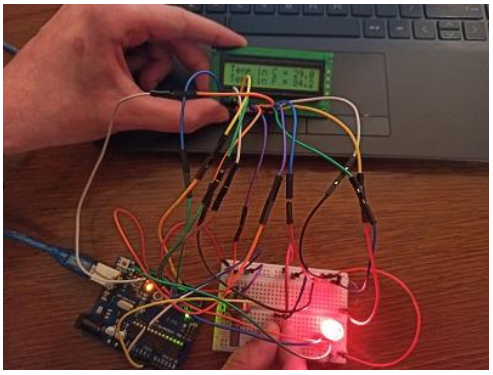
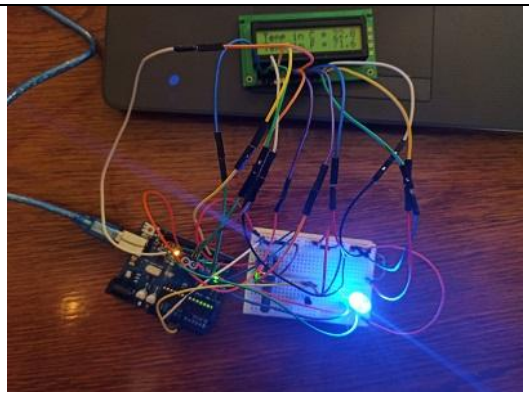
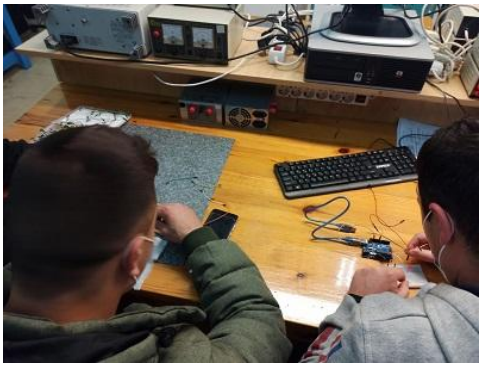
// Kütüphane kodune ekle:

```
#include <LiquidCrystal.h>
int t=0;
int sensor = A0;
float temp;
float tempc;
float tempf;
// Kütüphane arayüz pinlerinin numaraları ile başlar
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
//**********RGB**********
const int sensorPin = A0;
// Santigrat cinsinden oda sıcaklığı
const float baselineTemp = 20.0;
//**************************
void setup() {
  // LCD ekran satır ve sütun sayısı belirlenmesi:
pinMode(sensor,INPUT);
lcd.setCursor (0,0);
lcd.begin(16, 2);
  // LCD' de mesaj yazdırma.
lcd.print ("    ARDUINO     ");
lcd.setCursor (0,1);
lcd.print ("TEMPERATURE METER");
```

```
delay (3000);
Serial.begin(9600);
for(int pinNumber = 6; pinNumber<8; pinNumber++){
    pinMode(pinNumber,OUTPUT);  // pin6:Blue pin7:Green  pin8:Red
    digitalWrite(pinNumber, LOW);
   }
}
void loop() {
delay(2000);
t=t+2;
temp=analogRead(sensor);
//tempc=(temp*5)/10;
tempc=map(((temp-20)*3.04), 0, 1023, -40, 125);
tempf=(tempc*1.8)+32;
Serial.println("_______");
Serial.println("Temperature Logger");
Serial.print("Time in Seconds= ");
Serial.println(t);
Serial.print("Temp in deg Celcius = ");
Serial.println(tempc);
Serial.print("Temp in deg Fahrenheit = ");
Serial.println(tempf);
lcd.setCursor(0,0);
lcd.print("Temp in C = ");
lcd.println(tempc);
lcd.setCursor(0,1);
```

```
lcd.print("Temp in F = ");
lcd.println(tempf);


//*************code RGB***************
  // AnalogIn pin 0’ daki değeri okuma
  // ve değişkende depolama
  int sensorVal = analogRead(sensorPin);
  // 10 bit sensor değerini seri porttan gönder
  Serial.println("sensor Value: ");
  Serial.println(sensorVal);

  // okunan voltaj değerini ADC ile dönüştür.
  float voltage = (sensorVal/1024.0) * 5.0;
  // Gerilim seviyesini seri bağlantı noktasından gönder
  Serial.println(", Volts: ");
  Serial.println(voltage);
   if(tempc < 20){
   digitalWrite(6, LOW);
   digitalWrite(7, LOW);
   digitalWrite(8, LOW);
   }
   else if(tempc >= 20 && tempc < 80){
         digitalWrite(6, HIGH);
   digitalWrite(7, LOW);
   digitalWrite(8, LOW);
  }
```

```
 else if(tempc >= 80 && tempc < 140){
        digitalWrite(6, LOW);
  digitalWrite(7, HIGH);
  digitalWrite(8, LOW);
}
 else if(tempc >= 140){
        digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, HIGH);
}
delay(100);
}
```

Fotoğraf: Proje eğitim seti ve proje üzerinde çalışan öğrenciler

.

3_PROJE ADI: ACİL FREN FLAŞÖRÜ – UYARLANABİLİR FREN LAMBASI

(ROMANYA)

Projenin Amacı: Ani fren durumunda, uyarlanabilir fren lambası arkadan gelen trafiği uyarabilir ve böylece arkadan çarpmaların önlenmesine yardımcı olabilir. Frenleme sırasında ve frenleme basıncına ve hızına bağlı olarak kritik bir frenleme durumu tespit edildiğinde etkinleştirilirler.

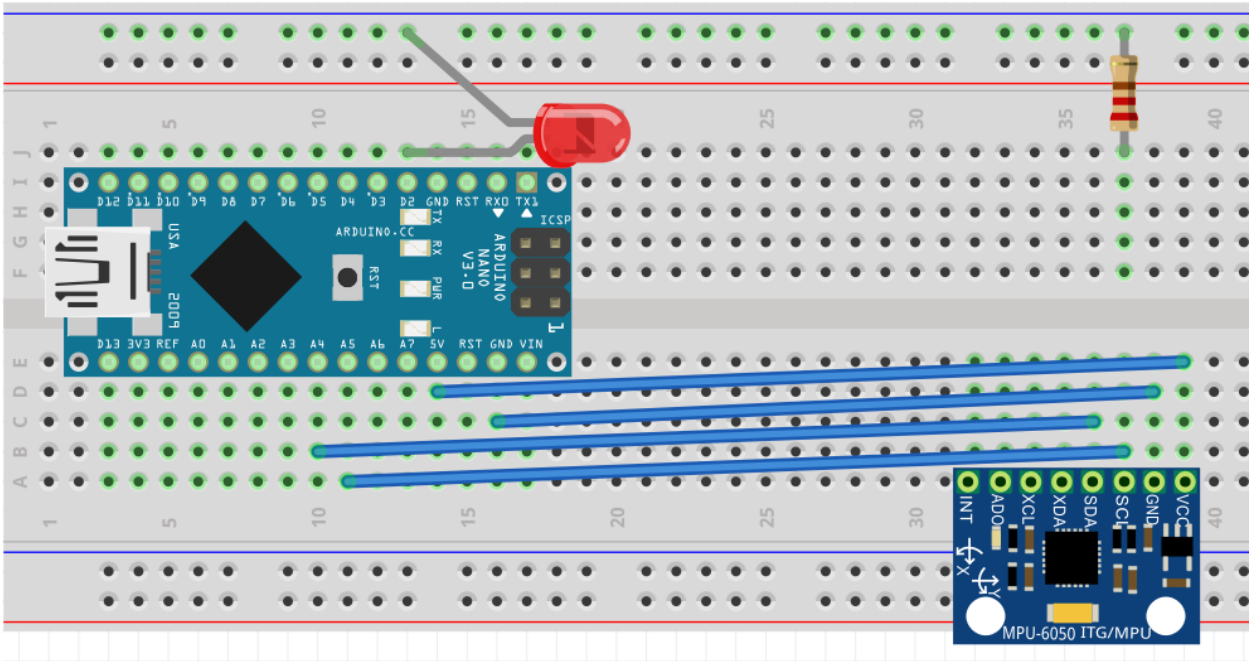
Proje Metodumuz:

İlk olarak projenin algoritması hazırlandı. Hazırlanan algoritmaya göre mantıksal tasarım yapıdı ve Aurdino programı yazıldı. Devre tasarımı için Arduino programı kullanıldı. Öğrencilerimiz bu tasarımları yaparken "ArduinVET" olarak adlandırılan Erasmus+ KA-202 VET projesinin tüm sonuçlarından ve ürünlerinden yararlanmaktadır.

Gerekli ekipmanların sağlanması ve Arduino' nun programlanmasından sonra kesinleşen tasarım devremiz kurulur. Son tasarım devremiz bir arabaya yerleştirildi.

Bu proje fuarlarda sergilenmekte ve ziyaretçilere trafik güvenliğinin ne kadar önemli olduğunu hatırlatmaktadır.

Projenin Devresi.



Nasıl Çalışır:

Arkadaki araçları sert frenleme konusunda uyarmak için acil durum fren lambaları etkinleştirilir. Fonksiyon, fren lambasının - normal frenlemede olduğu gibi - sabit bir parlama ile parlamak yerine yanıp sönmesi anlamına gelir. Ani fren durumunda 50 km/s üzerindeki hızlarda acil durum fren lambaları devreye girer.

Bir mikro denetleyici tarafından desteklenen bir elektronik ivmeölçer (MPU6050) üzerine kuruludur ve atalet kuvvetine dayalı ani frenlemeyi algılar ve üçüncü fren duruşunu hızlı bir şekilde yanıp sönerek otomobilin arkadan çok daha kolay fark edilmesini sağlar.

MPU6050, 3 eksenli bir jiroskop, 3 eksenli İvmeölçer ve tek bir çip üzerine entegre edilmiş bir Dijital hareket işlemcisine sahiptir. 3V-5V güç kaynağı ile çalışır. **MPU6050, iletişim ve veri aktarımı için I2C protokolünü kullanır**. Bu modül, büyük doğruluk sağlayan yerleşik bir 16 bit ADC' ye sahiptir.

Kalman filtresi, bir dizi gürültüye duyarlı ölçüme dayalı olarak dinamik bir sistemin durumunu değerlendiren verimli bir özyinelemeli filtredir. İçsel özellikleri nedeniyle, sıfır ortalamalı Gauss sistemleri üzerinde etkili olan gürültü ve bozulmalar için mükemmel bir filtredir.

Malzeme Listesi:

Bir adet Arduino Nano, bir adet MPU6050, bir adet 180 Ohm direnç, bir adet kırmızı LED.

Projenin Arduino Programı:

//Proje Adı: ACİL FREN FLAŞÖRÜ – UYARLANABİLİR FREN LAMBASI

```
#include<Wire.h>;
```

//https://www.codetd.com/en/article/11749279 MPU6050 ile nasıl daha hızlı iletişim kurulacağını çok iyi açıklayan bir makale.

// otomotiv koşullarında yol koşullarından, titreşimlerden vb. birçok gürültü geliyor ve çok iyi bir yazılım filtresi kullanacağım (Kalman)

```
const int MPU_addr = 0x68;
float kalman_old = 0;
float cov_old = 1;
```

```
float val1, val2, val3, val4, val5;
int med1_sort, med2_sort, med3_sort, med4_sort;
float avg;
int med;
int m;
int med_sort[5];
int c_avg = 1;
int c_med = 1;
int d = 0;
float old_x = 0;
float real_angle = 0;
float prev_angle = 0;
unsigned long previousMillis = 0;
const long interval = 50;
void setup() {
Serial.begin(9600); // for USB monitor
Serial.println ("Hit a key to start");     // sinyal başlatma tamamlandı
  while (Serial.available() == 1);
  //G sensörü bağlantısı
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B); // PWR_MGMT_1 register
  Wire.write(0x00); // sıfıra ayarla (MPU-6050' uyandır.)
  Wire.endTransmission(true);
  delay(50);
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x1C);
  Wire.write(0x00);
  Wire.endTransmission(true);
pinMode(2, OUTPUT); //  BİRİNCİ ve İKİNCİ aşama için çıkış sinyali.
digitalWrite (2, LOW);
```

//pinMode(7, OUTPUT);// İKİNCİ aşama için çıkış sinyali. Daha etkileyici işlevsellik istiyorsanız, bunu ve sonraki satırı etkinleştirin.

//digitalWrite (7, LOW); Bu seçeneği etkinleştirirseniz, tüm çizimde dijital çıkış pimi 7'yi etkinleştirmeniz gerekir!

// bu sinyal, devreyi tamamlamak için bir PNP transistörü veya P-Gate MOSFET ile kullanılabilir.

// gerçek devrede 5v mavi led kullanıyorum. Fritzing devresinde, ek dirençli normal bir 1.8 kırmızı led kullanın. - 180 ohm}

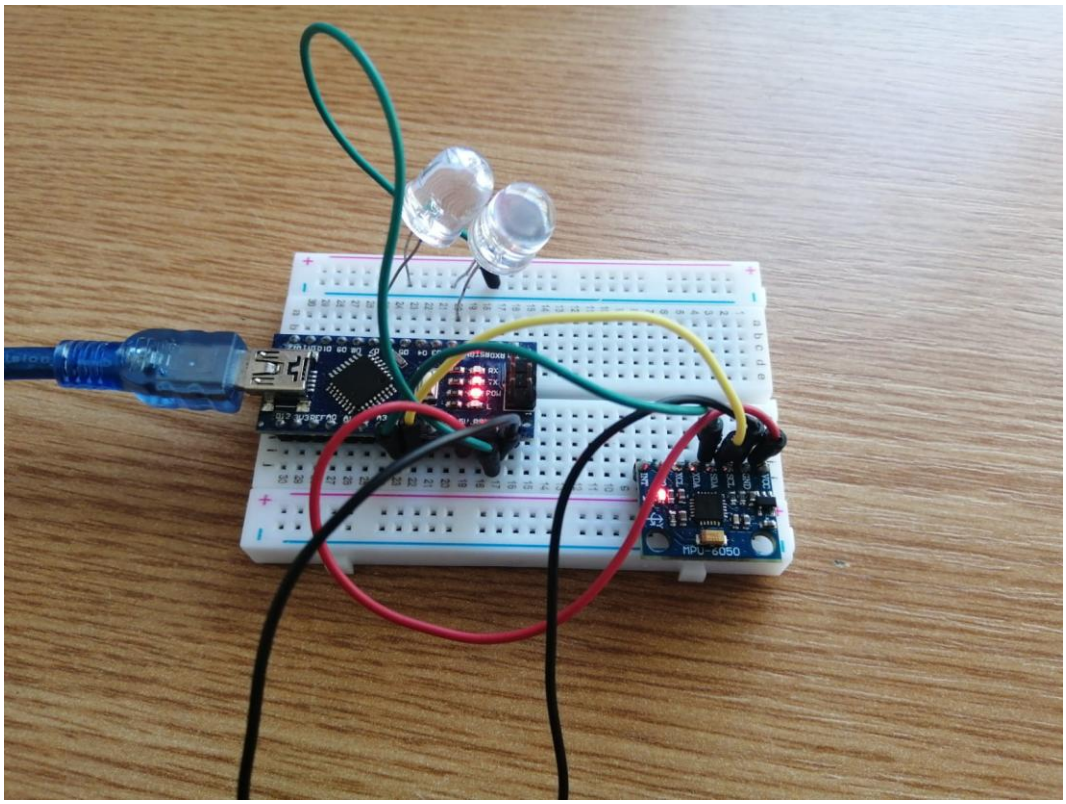
```
void loop() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    Wire.beginTransmission(MPU_addr);
    Wire.write(0x3D);
// 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
// 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
// 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_addr, 2, true);
    int16_t YAxisFull =  (Wire.read() << 8 | Wire.read());
    float YAxisFinal = (float) YAxisFull / 16384.0;
    int YAxis_Filtered = general_filter(c_avg, c_med, YAxisFull);
    int YAxis_kalman = kalman_filter(YAxisFull);
    int YAxis_movavg = mov_avg(c_avg, YAxisFull);
    int YAxis_median = median(c_med, YAxisFull);
    Serial.print("YAxis_Filtered");Serial.print(YAxis_Filtered/16384.0);
    Serial.print("\t");
   // Serial.print("YAxis_kalman");Serial.print(YAxis_kalman);
  //  Serial.print("\t");
  //  Serial.print("YAxis_movavg");Serial.print(YAxis_movavg);
```

```
  // Serial.print("\t");
  //  Serial.print("YAxis_median");Serial.print(YAxis_median);
  //  Serial.print("\t");
    Serial.print("YAxisFull");Serial.println(YAxisFull/16384.0);
   // Serial.print("\t");
   // Serial.print("YAxis_Final");Serial.println(YAxis_Final);
//BİRİNCİ Bölüm
if (YAxis_Filtered/16384.0 > 0.4 and YAxis_Filtered/16384.0 <= 0.7){
  for (int i = 1; i <= 4; i++) {
      digitalWrite (2, HIGH);
     // digitalWrite (7, HIGH);
      delay (75);
      digitalWrite (2, LOW);
     // digitalWrite (7, LOW);
      delay (50);}}
else {
  //İKİNCİ Bölüm – çok güçlü fren
if (YAxis_Filtered/16384.0 > 0.7){
   for (int i = 1; i <= 7; i++) {
      digitalWrite (2, HIGH);
     // digitalWrite (7, HIGH);
      delay (75);
      digitalWrite (2, LOW);
      //digitalWrite (7, LOW);
      delay (50);}}}
    c_avg = c_avg + 1;
    c_med = c_med + 1;
    if (c_avg > 5 && c_med > 5)
    {      c_avg = 6;
      c_med = 6;    }  }}
float kalman_filter (float input)
{  float kalman_new = kalman_old;
  float cov_new = cov_old + 0.50;
```

```
  float kalman_gain = cov_new / (cov_new + 0.9);
  float kalman_calculated = kalman_new + (kalman_gain * (input - kalman_new));
  cov_new = (1 - kalman_gain) * cov_old;
  cov_old = cov_new;
  kalman_old = kalman_calculated;
  return kalman_calculated;}
float mov_avg (int counter, float input)
{  avg = 0;
  m = 0;
  if (counter == 1) {
    val1 = input;
    avg = val1;  }
  else if (counter == 2) {
    val2 = input;
    avg = val2;  }
  else if (counter == 3) {
    val3 = input;
    avg = val3;  }
  else if (counter == 4) {
    val4 = input;
    avg = val4;  }
  else if (counter == 5) {
    val5 = input;
    avg = val5;  }
  else if (counter > 5 ) {
    counter = 6;
    if (val1 == 0) {
      m = m + 1;    }
    if (val2 == 0) {
      m = m + 1;    }
    if (val3 == 0) {
      m = m + 1;    }
    if (val4 == 0) {
```

```
    m = m + 1;    }
  if (val5 == 0) {
    m = m + 1;    }
  if (input == 0) {
    m = m + 1;    }
  d = 6 - m;
  if (d == 0)
  {     avg = input;
    counter = 1;    }
  else
  {     avg = (val1 + val2 + val3 + val4 + val5 + input) / d;    }
  val1 = val2;
  val2 = val3;
  val3 = val4;
  val4 = val5;
  val5 = input;  }
 return avg;}
float median (int counter, int input)
{  if (counter == 1) {
  med1_sort = input;
  med_sort[0] = med1_sort;  }
 else if (counter == 2) {
  med2_sort = input;
  med_sort[1] = med2_sort;  }
 else if (counter == 3) {
  med3_sort = input;
  med_sort[2] = med3_sort;  }
 else if (counter == 4) {
  med4_sort = input;
  med_sort[3] = med4_sort;  }
 else if (counter >= 5) {
  counter = 6;
  med_sort[4] = input;
```

```
    sort(med_sort, 5);
    med = med_sort[2];
    med1_sort = med2_sort;
    med2_sort = med3_sort;
    med3_sort = med4_sort;
    med4_sort = input;
    med_sort[0] = med1_sort;
    med_sort[1] = med2_sort;
    med_sort[2] = med3_sort;
    med_sort[3] = med4_sort;  }
  return med;}
void sort(int a[], int size) {
  for (int i = 0; i < (size - 1); i++) {
    for (int o = 0; o < (size - (i + 1)); o++) {
      if (a[o] > a[o + 1]) {
        int t = a[o];
        a[o] = a[o + 1];
        a[o + 1] = t;      }    }  }  }
//tüm filtrelere uygulanan fonksiyon
float general_filter (int counter_avg, int counter_med, float input) {
float input_movavg = mov_avg(counter_avg, input);    //Hareketli Ortalama Uygulaması
float input_med = median(counter_med, input_movavg); //Medyan Uygulaması
float input_filtered = kalman_filter(input_med);     //Kalman Filtre Uygulaması
  return input_filtered;}
```



Fotoğraf: Final Devre Tasarımı

4_PROJE ADI: ÇİZGİ TAKİP EDİCİ (Çizgi İzleyen Robot) (AVUSTURYA)

Çizgi takip edici (Çizgi İzleyen Robot) nedir? :

Çizgi izleyen robot, zemine çizilen çizgiyi algılayıp takip edebilen mobil bir makinedir. Genel olarak, yol önceden tanımlanmıştır ve yüksek kontrastlı bir renge sahip beyaz bir yüzey üzerinde siyah bir çizgi gibi görünebilir. Kesinlikle bu tür bir robot, robotun altına yerleştirilen kızılötesi ışın (IR) sensörleri ile çizgiyi algılamalıdır. Bundan sonra, veriler işlemciye iletilir. Böylece işlemci uygun tavsiyelere karar verecek ve daha sonra bunları sürücüye gönderecek ve böylece çizgi izleyen robot tarafından yol izlenecektir.

Çizgi izleyen robot yapımında önemli olan nokta, yolu olabildiğince hızlı takip etmeye yetecek kadar iyi bir kontroldür.

Çizgi Takip Eden Robotun Çalışması

Çizgi izleyen robot kavramı ışıkla ilgilidir. Burada ışığın siyah beyaz yüzey üzerindeki davranışını kullanıyoruz. Beyaz renk üzerine düşen tüm ışığı yansıtırken siyah renk ışığı emer.

Bu çizgi izleyen robotta IR vericileri ve alıcıları (fotodiyotlar) kullanıyoruz. Işıkları göndermek ve almak için kullanılırlar. IR ışınları beyaz bir yüzeye düştüğünde, IR alıcısına yansır ve bazı voltaj değişiklikleri oluşturur.

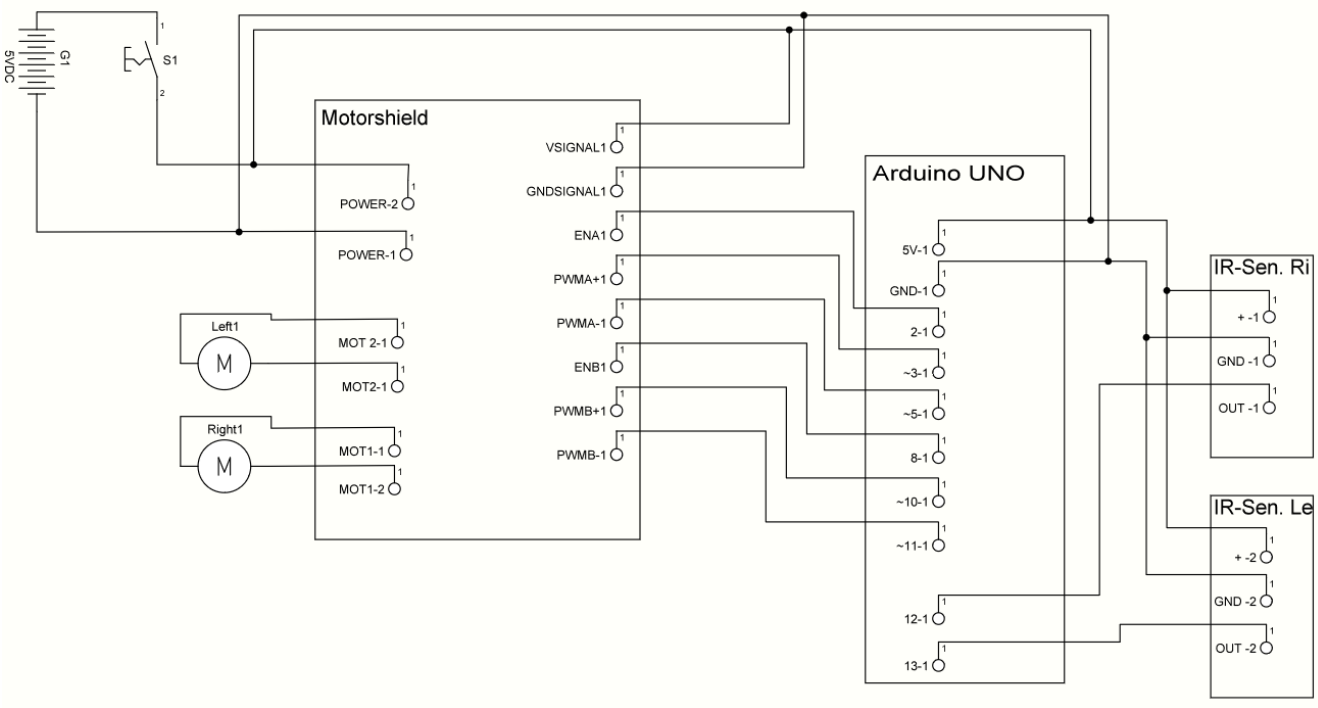
IR ışınları siyah bir yüzeye düştüğünde, siyah yüzey tarafından emilir ve hiçbir ışın yansıtılmaz; bu nedenle, IR alıcısı herhangi bir ışın almaz.

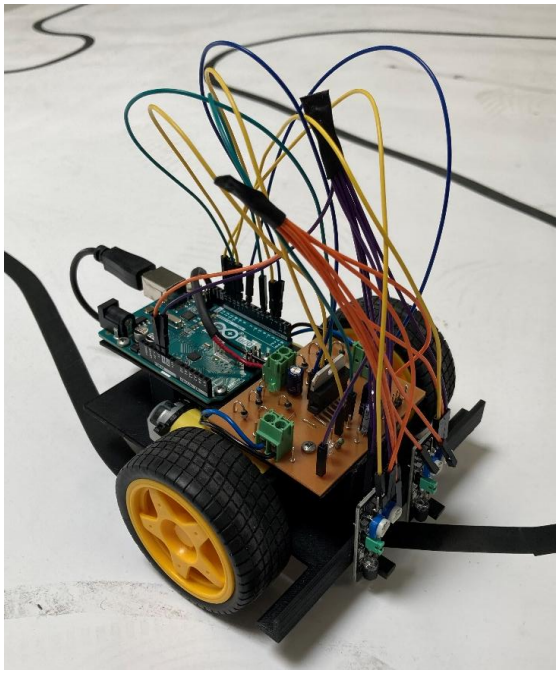
Bu projede, IR sensörü beyaz bir yüzey algıladığında, Arduino giriş olarak 1 (HIGH) alır ve siyah bir çizgi algıladığında, Arduino giriş olarak 0 (LOW) alır. Bu girdilere dayanarak, bir Arduino Uno, botu kontrol etmek için uygun çıktıyı sağlar.

Partlist:

parts	type	pieces
Arduino	Uno	1
Motor Kalkanı	by HTL Wolfsberg	1
DC-Motor	COM-Motor01, 3-9Volt DC, joy-it	2
Güç Bataryası	Flip12 PhonePowerBank, 3350mAh	1
IR-Sensörler	SEN-KY032IR, joy-it	2
Ana Şalter		1
Atlama teli	RB-CB3-025, joy-it	1
USB Kablo	Digitus,AK-300102-010-S, Typ A-B	s

Şematik Devre Diyagramı:





Fotoğraf: Final Devre Tasarımı

Program:

```
// Çizgi Takip Edici
const byte sr = 12;
const byte sl = 13;
const byte enr = 2;
const byte enl = 8;
const byte mrf = 3;
const byte mrb = 5;
const byte mlf = 10;
const byte mlb = 11;

void setup()
{
  Serial.begin(9600);

  pinMode(sr,INPUT);
  pinMode(sl,INPUT);

  pinMode(enr,OUTPUT);
  pinMode(enl,OUTPUT);
  digitalWrite(enr,HIGH);
  digitalWrite(enl,HIGH);

  pinMode(mrf,OUTPUT);
  pinMode(mrb,OUTPUT);
  pinMode(mlf,OUTPUT);
  pinMode(mlb,OUTPUT);

  analogWrite(mrf,0);
  analogWrite(mrb,0);
  analogWrite(mlf,0);
  analogWrite(mlb,0);
}
  bool senR;
  bool senL;

void loop()
{
  senR = digitalRead(sr);
  senL = digitalRead(sl);

  if(!senR && !senL) //staight
  {
    analogWrite(mrf,220);
    analogWrite(mrb,0);
    analogWrite(mlf,200);
    analogWrite(mlb,0);
```

```
  }

  //sağ sensör çizgide
  //sağa dön
  else if(senR && !senL)
  {
    analogWrite(mrf,0);
    analogWrite(mrb,100);
    analogWrite(mlf,255);//255
    analogWrite(mlb,0);
  }

  //sol sensör çizgide
  //sola dön
  else if(!senR && senL)
  {
    analogWrite(mrf,255);//255
    analogWrite(mrb,0);
    analogWrite(mlf,0);
    analogWrite(mlb,100);
  }

  // iki sensörde siyah
  // dur
  else if(senR && senL)
  {
    analogWrite(mrf,0);
    analogWrite(mrb,0);
    analogWrite(mlf,0);
    analogWrite(mlb,0);
    delay(3000);
  }
}
```

5_PROJE ADI: SERA OTOMASYONU (İTALYA)

Projenin Amacı: Neden okulda bir sera projesi?

Okuldaki öğrenciler, aktif vatandaşlık ve BM Sürdürülebilir Kalkınma Gündemi 2030'da enerjinin sürdürülebilirliği, su israfından kaçınmanın önemi ve derin iklim değişikliklerini dikkate alarak yenilikçi ve çevre dostu yetiştirme yöntemleri geliştirme ihtiyacı hakkında çeşitli yollar izlediler. Gençlerin ekolojik bilinci, Arduino' nun uygulama olanaklarının doğasında var olan okulda edindiği teorik ve laboratuvar bilgisinden başlayarak teknik çözümler ve yenilikçi fikirler arayışında ifade edilir.

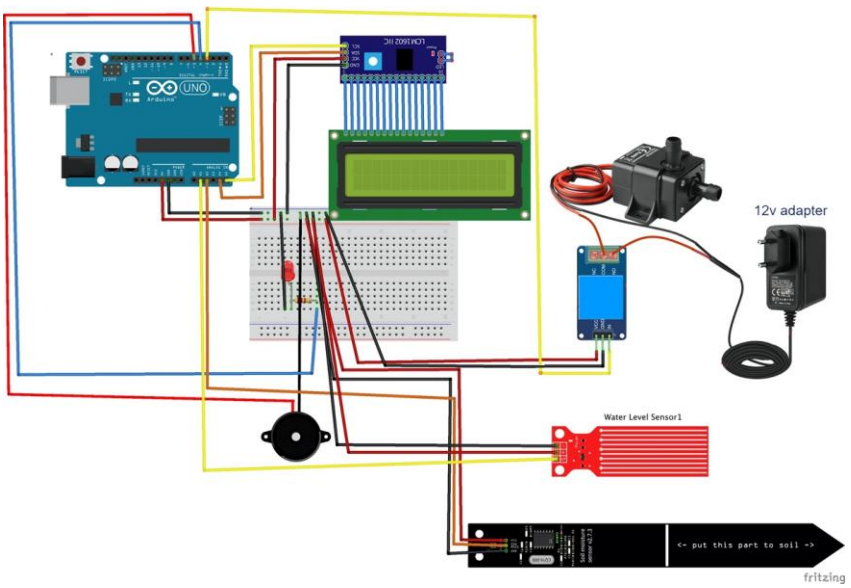
Öğretmenler tarafından desteklenen bazı öğrenciler, yerel meyve ve sebzelerin yetiştirilmesine uygun bir prototip seranın sulanmasını ve havalandırmasını yönetmek için toprak ve hava sıcaklığını, toprak nemini algılamak için bir sistem tasarladı.

Deney, umulan sonuçları verdi ve gelecekteki okul sebze bahçesine ev sahipliği yapması planlanan okulun çevresindeki alanda fiili uygulamaya katkıda bulunacak.

Proje Metodumuz

Öncelikle serayı çalıştırmak için kullanmak istediğimiz tüm parametrelerin bir listesi hazırlandı. Bundan yola çıkarak proje algoritmasının ilk versiyonu hazırlandı. Bu algoritmaya göre gerekli sensörler satın alınmış ve mantık projesi gerçekleştirilmiştir. Devre tasarımında Arduino kullanılmaktadır. Son olarak, nihai yazılım uygulandı. Çeşitli sensörlerin eşik değerlerinin doğru ayarını kontrol edildikten sonra seranın içine yerleştirildiler.

Project Circuit



Nasıl Çalışır:

Bir LCD ekran, havanın sıcaklık ve nem değerlerini, toprak nemini ve sulama kabında bulunan su miktarını görüntülemeyi sağlar. Değerler belirli bir eşiğin altına düştüğünde, bir röle tarafından kontrol edilen motorlu pompa, toprağın sulanmasını başlatır. Su seviyesi belirli bir eşiğin altına düştüğünde, ekranda sesli bir alarm eşliğinde yeniden doldurma uyarısı görüntülenir.

Malzeme Listesi:

Bir adet Sera, bir adet Arduino Uno R3, bir adet sıcaklık/nem DHT11 sensörü, bir adet buzzer, bir adet toprak nem sensorü, bir adet su seviye sensörü, bir adet kırmızı led, bir adet 12 V batarya, bir adet 12V pompa, bir adet 220 ohm direnç, bir adet Röle, bir adet LCD ekran.

Projenin Arduino Programı:

//Proje Adı: SERA OTOMASYONU:

```
#include <LiquidCrystal_I2C.h>

#include <Wire.h>

#include <DHT.h>

#define DHTPIN 11

#define DHTTYPE DHT11 // DHT 11

DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal_I2C lcd(0x27,20,4);  // 16 karakter ve 2 satır ekran için LCD adresini 0x27 olarak ayarlayın

int t, h, Lumen, lumin, water, igro, umdtr, wlevel;

const int pin_water = A1;

const int pin_igro = A2;

const int pin_pump = 3;

const int pin_led = 4;

const int pin_buzzer = 5;
```

```
void setup() {
   Serial.begin(9600);

   dht.begin();
   lcd.init();
   lcd.backlight();
   lcd.setCursor(5,0);
   lcd.print("School");
   lcd.setCursor(3,1);
   lcd.print("Greenhouse");
   delay (5000);
   lcd.clear();
   lcd.setCursor(2,0);
   lcd.print("IIS Einstein");
   lcd.setCursor(3,1);
   lcd.print("De Lorenzo");
   delay (5000);
   lcd.clear();

   pinMode(pin_pump,OUTPUT);
   pinMode(pin_led, OUTPUT);
   pinMode(pin_buzzer, OUTPUT);
   digitalWrite(pin_pump, HIGH);
}

void loop() {
   // su seviyesi
   water = analogRead (pin_water);
```

```
wlevel = map (water,0, 1023, 0, 100);
if(wlevel < 35){
   lcd.clear();
   tone(pin_buzzer, 800);
   digitalWrite(pin_led, HIGH);
   delay(300);
   noTone(pin_buzzer);
   digitalWrite(pin_led, LOW);
   delay(300);
   lcd.clear();
   //lcd.begin(16, 2);
   lcd.setCursor(2,0);
   lcd.print("Refill Water");
   delay (1000);
   lcd.clear();
} else {
   noTone(pin_buzzer);
   digitalWrite(pin_led, LOW);
}

// nem ölçer
igro = analogRead(pin_igro);
umdtr = map (igro, 0, 1023, 0, 100);

if(umdtr < 45){
   digitalWrite(pin_pump, LOW);
   delay(10000);
   digitalWrite(pin_pump, HIGH);
```

```
}

// Sensör nem ve sıcaklık okuması DHT11
h = dht.readHumidity();
t = dht.readTemperature();

// imleci sütun 0 ve satır 1'e yerleştir
// (not: 1. satır ve 2. satır, 0'dan başlayarak sayıldığı için):
lcd.setCursor(0, 0);
lcd.print("T:");
lcd.print(t);
lcd.print( (char) 223 );
lcd.print("C");

lcd.setCursor(10, 0);
lcd.print("H:");
lcd.print(h);
lcd.print("%");

lcd.setCursor(0, 1);
lcd.print("SH:");
lcd.print(umdtr);
lcd.print("%");

lcd.setCursor(10, 1);
lcd.print("WL:");
lcd.print(wlevel);
lcd.print("%");
```

}

Fotoğraf: Son Tasarım

EK:

Arduino'nun misyonu, herkesin erişilebilir elektronik ve dijital teknolojiler aracılığıyla hayatlarını iyileştirmesini sağlamaktır. Bir zamanlar elektronik, tasarım ve programlama dünyası ile dünyanın geri kalanı arasında bir engel vardı. Arduino bu bariyeri yıkmıştır. Arduinos hakkında daha fazla bilgi için aşağıdaki web sitesini ziyaret edebilirsiniz…

www.arduino.cc/

ARDinVET projemiz hakkında daha fazla bilgi için aşağıdaki proje web sitemizi ziyaret edebilirsiniz…

www.arduinvet.com

Co-funded by the Erasmus+ Programme of the European Union

"Bu proje Erasmus+ Programı kapsamında Avrupa Komisyonu tarafından desteklenmektedir. Ancak burada yer alan görüşlerden Avrupa Komisyonu ve Türkiye Ulusal Ajansı sorumlu tutulamaz."