

Yazar:Ali Birkan Bayram

ARDUINO OGRENIYORUM

Neden “Drone” diye sorabilirsiniz fakat daha 2019’da bir mühendislik öğrencisi arduino kullanarak (ESP 8266 kullanıyor ki şirketinin adı dahi bundan geliyor ama yapıları benzer diyebiliriz) bir drone şirketi kurdu. Metehan Emlik olan bu girişimciyi takip etmenizi öneririm. Bu başarı hikayesi inşallah sizin ufkunuzu açar ve daha çok ilgi duyarsınız. Buradan kendisine de başarılar dilerim.

Sizlere birkaç söz bırakayım:

“Her şey hiç olarak başladı.” – *Ben Weissenstein*

“Dostum, elle yapılan işler berbat.” – *Alex Fraiser*

Baskı

I.Basım

Ağustos 2020

Yazar : ALİ BİRKAN BAYRAM

Ali Birkan Bayram kimdir?

Ali Birkan Bayram 1998 Konya doğumlu teknoloji bağımlısı bir mühendis olup şu anda bir sürü projeleri olan birisidir. Kendisini bazen bir kaçık gibi görmekte çünkü şu anda her alanda neredeyse düşündüğü ve uğraştığı proje mevcut ve bu çılgınca. Hayvancılık üzerine yapmış olduğu proje ile ülke de çoğu kurumca kullanıma geçeceğini düşündüğü makinaları ile çiftlikleri mükemmel derecesine getiriyor. Bazı ürünlerin komple kullanılması ile %90'a varan çiftlik maliyetini düşürüyor. Tarımda da hızla makine ve yazılım projeleri gelişmekte olup ortağı Halil Erdem Güzel ile harika işler başarıyor. En azından kendileri öyle düşünüyor 😊 Yakın bir zaman sonra kendisini bir araba projesi ile duyacağınızın müjdesini verelim buradan. Ürettiğiniz kadar bu dünya da varsınız unutmanınız dileğiyle ...

Önsöz girişi yapmayacağım sadece hobi ya da profesyonel işler üretmek istiyorsanız buyurun ...

İçerik

- **Arduino Nedir? – 6.sayfa**
- **Nesnelerin interneti – 7.sayfa**
- **Arduino ile yapılanlar (Kısacası !) – 10.sayfa**
- **Arduino Çeşitleri ve Özellikleri – 11.sayfa**
- **Arduino Uno R3 – 12.sayfa**
- **Arduino IDE Kurulumu – 17.sayfa**
- **Arduino IDE Ayarları – 18.sayfa**
- **Arduino Kütüphane yapısı – 21.sayfa**
- **Arduino ile Programlama - 22.sayfa**
- **Serial Port Ekranı – 23.sayfa**
- **Arduino Pin Yapısı ve Tanımlamalar – 24.sayfa**
- **Arduino Buton Okuma (Dijital) – 25.sayfa**
- **Arduino Buton Okuma (Analog) – 27.sayfa**
- **Arduino Birden Fazla Buton Okuma – 29.sayfa**
- **TEMEL ELEKTRONİK FORMULÜ – 32.sayfa**
- **Arduino ile projeler – 32.sayfa**
- **Arduino Led Yakmak – 33.sayfa**
- **Breadboard Nedir? – 37.sayfa**

- **Kodlamaya başlıyoruz – 41.sayfa**
- **Kısaca ram ve rom nedir? – 47.sayfa**
- **Seri haberleşme – 50.sayfa**
- **Gelişmiş seri haberleşme – 53.sayfa**
- **74HC595 entegresi (Gelişmiş Seri Haber...) – 54.sayfa**
- **SSD (Seven Segment Display) kullanımı – 55.sayfa**
- **HC-SR04 (Mesafe Sensörü) kullanımı – 59.sayfa**
- **Dht11,dht22,lm35 ve tmp36 kullanımı – 64.sayfa**
- **16x2 lcd ekran kullanımı – 73.sayfa**
- **Arduino ile örnek projeler – 81.sayfa**
- **Kapanış (Son) – 85.sayfa**

Arduino Nedir?

Arduino bana göre herkesin ihtiyacını karşılayacak kadar güzel bir teknolojik araçtır. (Mühendislik işi değildir en azından basit düzeyde bilinmesi bir mühendislik değildir bence, bu konuya youtube sayfam üzerinde değineceğim)

Arduino kodlanabilir bir karttır yani kısmi bir şekilde yanımız da taşıyabileceğimiz ya da kodlama yapabileceğimiz bir basitleştirilmiş kodlama işine yarayan elektronik devre kartıdır ilk başta küçük bir monitörü olmayan bilgisayar diyebilirsiniz. (Normal bilgisayara göre daha basitleştirilmiş görevler için kullanılıyor ve her geçen gün daha da kullanım özellikleri artıyor)

Arduino ile ilgili yapılan ve yapılacaklar kitabın ilerleyen sayfalarında görülecektir sıfırdan başlayanlar için özellikle de yazılım bilgisi donanım bilgisi olmayan birisi için hiçbir şekilde atlamamasını tavsiye ederim. Bu bölümü kısa bırakıp tam anlamıyla arduinoyu sizin tanımlamanız daha doğru olacağına inanıyorum. İleride sizlerden duymak isterim arduino nedir deyince ama arduinoya fanatik olmayın çünkü her elektronik devrenin kartın bir kullanım alanı mevcuttur bir yerde takılı kalmayın.

Nesnelerin interneti

Son yıllar da sıkça karşımıza çıkan nesnelerin interneti kavramı bu cihazla birlikte sizin de karşınıza çıkacaktır. İsterseniz daha ayrıntılı bir şekilde nesnelerin interneti hakkında bilgi almak kodlamayı ona göre yapmak isterseniz yazmış olduğum “Nesnelerin İnterneti Dünyasına Giriş” adlı kitabımı da okuyabilirsiniz lakin öncelikle bu kitabı bitirmenizi tavsiye etmekteyim çünkü bu kısmı tamamlamadan oraya geçmeniz pek sağlıklı olmayacaktır.

Nesnelerin internetinin dünyası şu anda 20milyara yakın cihaz mevcuttur bu rakam yıl sonuna kadar (31 Aralık2020) 30 milyar olması tahmin edilmektedir. Rakamlar açıkçası bu sektörün çok hızlı ilerlediği gerçeğini gözler önüne sürmekte ve bu sektöre yönelecek insanların ise acele edip ilk olmak için yer ayırtması gerekmektedir. Sizin de böyle bir düşünceniz var ve ürün ortaya koymayı düşünüyorsanız kendinizi daha da fazla geliştirmenizi tavsiye ederim. Yeniliklere açık olun ve takip edin bunları yaparsanız artık siz takip ettiğiniz her şeyi bırakacaksınız ve yenilik sizi takip edecek (çok sevdiğim bir cümledir önemlidir ve dikkate alın !).

“Nesnelerin interneti ne için kullanılır?” dersiniz aşağıdaki örnekleri inceleyiniz:

- **Endüstriyel kolaylık sağlamak**
- **Akıllı evler ve daha fazlası ...**

Endüstriyel olarak işlerin takibi, işlerin aksaklıklarının giderilmesi, zamandan kazanç, parasal büyük bir kar, verim arttırma, ürün sayısı ve kalitesi arttırıcı gibi kolaylıklar sağlanır.

Akıllı evler olarak hayatımıza giren konfor alanlarımızı arttırmaya yarayan bu sektör ise gerçek bir konfor sağlayıcı olarak işimize yarar. Akıllı evler de kullanılan sistemler şunlardır:

- Ev güvenliğinin sağlanması (yangın, deprem anında uyarılar gibi) -sigorta şirketleri için bence zorunlu yapılmalı-
- Ev güvenliği (kapı otomatı gibi – sesli& görüntülü)
- Akıllı aydınlatmalar
- Akıllı süpürgeler
- Ev ihtiyaçlarının belirlenmesi alınması (Edelkrone)

Benim de en sevdiğim bir cihaz var ki o da akıllı süpürgeler. 😊
Akıllı süpürgeler ile ev daimî bir temizlenme sürecinde lakin kimse farkında değil. Efor harcayan birisi yok ama daimi bir temizlik hijyen mis gibi değil mi? Bence öyle. 😊

Akıllı evler hayatımıza daha da fazla girecek ve bu konu da geliştirecek çok şey var basit bile görülebilecek eksiklikler var aslında kimse farkında değil. Benim ve ekibimin bile bulduğu 20'ye yakın proje mevcut. Buradan da bir sitem olsun bir sigorta şirketinin

Hackathon una çağırılmadık bile bu güzel projeleri yazmamıza rağmen çok üzücü bir olay ama alıştım. Akıllı evler için üretilen nesnelerin daha önü çok açık ve kimsenin görmediğini görmemiz dileğiyle.

Arduino ile yapılmış güzel projeler (Kısaca)

- Güneşi takip eden güneş panelleri
- Hava istasyonu (Hava raporlama)
- Boy ölçme cihazları
- GPS takip sistemleri
- ¹Robot araçlar (Uzaktan kumanda edilebilen)
- Akıllı sulama sistemleri
- ²Parmak izi giriş sistemleri

Gibi bazı örneklendirme yaptım sizler için. Kendi sınırınızı kendiniz belirlemeniz için daha fazlası ve başka örnek vermeyeceğim sırf bu örnekler benim ufkumu açar diyor olabilirsiniz bunun için ikinci kitabımı da okumanızı ve ilerleyen bölümleri dikkat etmenizi öneririm. Hiç düşünmediğiniz sektörler için artık bir ticari elektronik kart olmuştur ve ticarileşmiştir. Arduino üzerinden bir sürü insan para kazanıyor ve bu işi para için bile yapıyor olsanız doğru yerdesiniz.

¹ Otonom sürüş için yeterli hızlara ve yeteneğe sahip değildir.

² Bu konuda da yeterli olmayabilmektedir basit olanını düşünüldü.

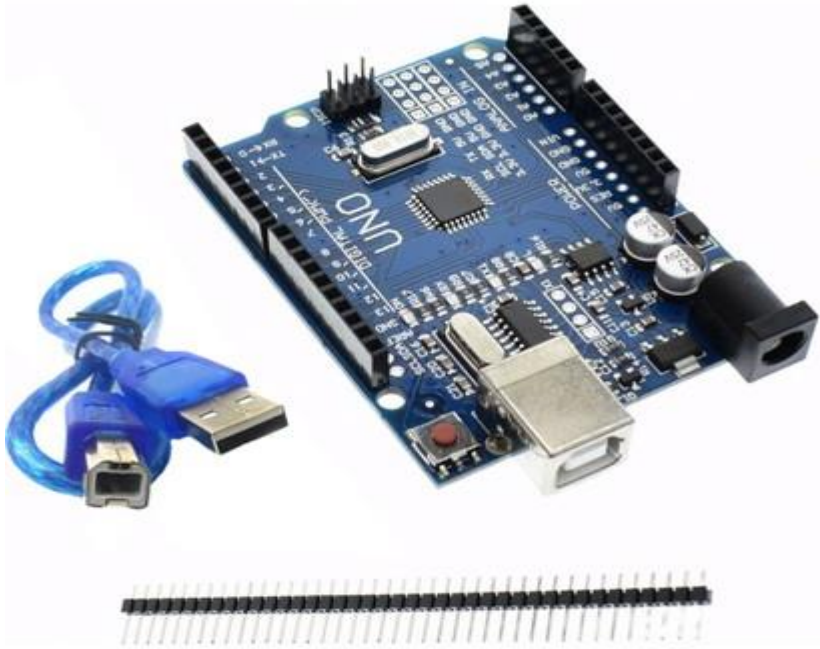
Arduino Çeşitleri ve Özellikleri

Arduino çeşitleri

- Arduino UNO R3
- Arduino Mega
- Arduino Leonardo
- Arduino Esplora
- Arduino Lilypad (Giyilebilir Teknoloji)
- Arduino Nano
- Arduino Pro mini

Daha fazlası da mevcuttur ama ilk etap için sizlere bu kadarını aktaracağım. Daha fazla arduino kartı için araştırmalar yapınız. İkinci kitabımda sizler ile ihtiyaçlara göre arduino tabanlı kartlar üreteceğim.

Arduino Uno R3



Arduino uno R3 en çok tercih edilen modellerdendir. Biz de bu eğitimi onun üzerinden devam ettireceğiz. Başka modeller için de neredeyse aynı hususlar geçerli olduğunu farz edeceğiz çok aykırı bir durum olur ise bunu zaten yazılı bir şekilde belirtirim.

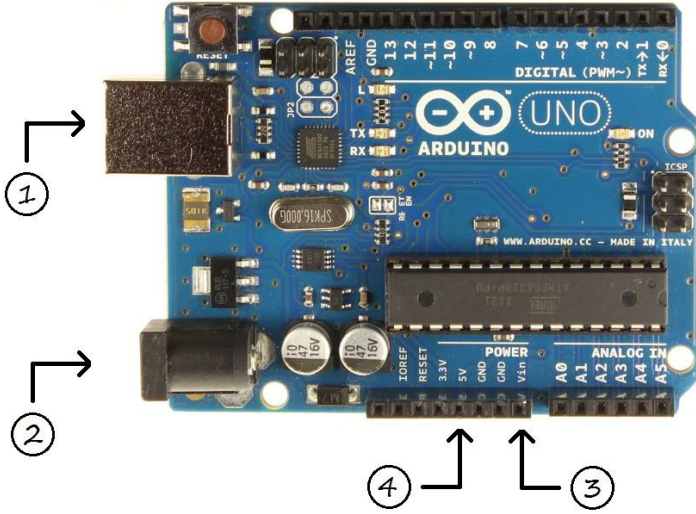
En çok tercih edilme nedeni kullanımı diğer modellere göre daha kolay olması olacaktır. Bu kart daha çok başlangıç paketi ürünü diyebiliriz ve bu model için ve boyutuna göre oldukça işlevsel olduğunu göreceksiniz. Arduino Uno R3 'ün bir de özelliklerine bakalım. Özellikleri diğer sayfada. →

- ¹ Mikrodenetleyici: ATmega328
- Çalışma Gerilimi: 5V
- Giriş Gerilimi (önerilen): 7-12V
- Dijital G/Ç Pinleri: 14 (6 tanesi PWM çıkışı)
- Analog Giriş Pinleri: 6
- Her G/Ç için Akım: 40 mA
- 3.3V Çıkış için Akım: 50 mA
- Flash Hafıza: 32 KB (ATmega328) 0.5 KB kadarı bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Saat Hızı: 16 MHz
- Uzunluk: 68.6 mm
- Genişlik: 53.4 mm
- Ağırlık: 25 g

¹ Mikrodenetleyici : Arduionun her şeyidir.

Bootloader, işletim sistemini önyükleyen, başlamasını sağlayan yazılımdır. Asıl ve tek amacı budur.

Arduino Uno R3 5V ile USB den beslenebilir bunun için 1 numaralı girişi kullanmanız gerekecektir.2 nolu girişten vermeye çalışırsanız bunun için 9-12V arası bir voltaj girişi olması gerekmektedir.(kablo ile board üzeri yani kart üzeri ¹pinden de beslenebilir)



¹ Pin: Tam bir tanımlı kafama oturtamasam da şöyle denebilir bir bilgisayarın içinde bulunan işlemcinin yüzeyindeki iğneler pindir. Pin haberleşme için gerekli ayaktır denebilir.

Girişler:

1. Bu giriş hem programlama için hem elektrik beslemesi için kullanılır. Giriş bu model için ¹Type B ama bazı arduino modelleri için bu giriş değişmektedir. Bazı modeller de ise hiçbir bu port kablo girişi mevcut değildir. Başka yolla kodlama yapacağız. (TTL ve arduino kullanarak diğer arduino-yu kodlamak gibi)
2. Bu giriş 9-12V sadece elektrik besleme yeridir ve sadece elektrik girişi mevcuttur.
3. 3 ile ve 4 ile gösterilen girişler ²jumper kablolar ile elektrik besleme yerleridir ve sadece elektrik beslemesi yapılır. Bazı kartlar da bu besleme RAW girişi üzerinden de yapılmaktadır. İlerleyen kısımlar da RAW girişi ile besleme yapacağımız projelerimiz olacaktır.

Biz elektrik beslemesi için ilk etapta 1. Girişten kablo ile yapılmasını öneriyoruz. Bu kablo ile besleme de bir sorun mevcuttur ve bunu çoğu kişi bunu pek önemsememektedir. Kayıp yaşamadan bir güç gereken projeler ile uğraşılacağımız zamanlarda bu göz önünde tutulmalıdır. Kısa yazmak istediğim cümle maalesef ki bu kar çok uzadı ama kimsenin(neredeyse) değinmediği bir durum olan bu durum bazen hayat kurtarabiliyor.

Mesela “bir adaptör kaç volt verir sizce?” diyeceksiniz ki “üstünde yazanı verir neden bu kadar saçma bir soru bu?”

¹ Type B kablo :Yazıcıların kablosu genellikle budur. İnternet üzerinden araştırabilirsiniz.

² Jumper Kablo: basit bir giriş çıkış başları olan kablo çeşididir bir sonraki sayfa da gösterilecek.

Bu soru biraz saçma bir soru ilk başta hakkınız diye düşünüyorum ama emin olun ki bir bildiğim var. Adaptör de yazılan 12V çıkış cidden 12V mi veriyor bir ölçün. Ben hemen avometresi olmayan arkadaşlar için konuyu açıklayayım ve bu işe hobi olarak veya profesyonel girmeyi düşünüyorsanız kesinlikle bir avometre temin etmenizi öneririm. Gelelim 12V cidden 12V mi? “Hayır” diye cevap veriyorum maalesef ki adaptörler hiçbir zaman tam değerini vermez ve dalgalanmadan etkilenmemek için birazcık (çok değil ama) yüksek vermektedir. Yine 12V bir adaptör üzerinden devam edelim ölçüm yaptık diyelim 12.3-12.5V gibi değer almanız muhtemeldir ve bu bezen başınıza büyük sorunlar açabilir. Adaptör konusunu ciddiye almanızı tavsiye ederim bir yangın çıkarma ihtimaliniz oluştirebilir. Fazla voltaj mikro denetleyicimize zarar verir fazla ısınmasını ve lehimlerinin kopmasına sebep olur bu da yangına neden olabilir kısaca. İçerisindeki devreler de bu sorundan dolayı ısınmadan dolayı yangın çıkarabilir. Elektrik işleri konusunda daha da fazla bilgi edinmenizi tavsiye ederim.



*Jumper kablo

Jumper kablo sizlerin ilk başta bir kadim dostunuz olacaktır. Bu kabloları basit prototipleme evresinde kullanacaksınız veya basit düzeyler de öğrenmek için kullanacaksınız. İlerleyen projeler de bu kablolar sadece görsel olarak kalıp artık kendiniz lehimleme yapmaya başlamış olmanız gerekecektir. Lehimleme işi çok zor değil ve bu konuda da sizlere yardımcı olacağım.

Takipte kalın 🙏

Arduino IDE Kurulumu

Arduino IDE arduinomuzu kodlamamızı sağlayan bir kodlama yardımcıdır diyebiliriz ilk etapta. Aslında kendisi de bir kodlanmış programdır ve C dilini makine diline çevirir. (Buralar biraz ağır kaçabilir herhangi bir kodlama bilgin olmadan okumaya başlamışsan sonra küsme kodlamaya daha yolun başındasın biraz daha sabır lütfen 🙏) Arduino IDE ile çalışacağımız için kısmen bir söylemde bulunayım. Daha büyük işler yapmak istiyorsanız ve bunu gerçek profesyonellik içinde yapmak istiyorsanız kendiniz C üzerinden kodlamaya bakabilirsiniz bu konuda yine sizler için videolar çekip kitap yazacağım ama bu kitabı bitirmişsen senin için olan kısım sadece 10-15 sayfalık olacaktır. Emin ol ki zor değil! Neyse gelelim Arduino IDE ye biz şimdilik bu IDE yi kullanacağız ve eğer ilerleyen kısımlarda daha da profesyonelleşeceksek kendimizi geliştirmek için C dilini tam anlamıyla öğreneceğiz. Arduino IDE'yi bu link üzerinden (Bilgisayar platformunu kullandığınızı farz ediyorum) açınız. Arduino IDE sadece bilgisayar üzerinde çalışıyor gibi düşünebilirsiniz ilk başta ama tam bir kodlamayı öğrendikten ve Arduinoyu tam bir kavradıktan sonra size daha da ileri gitmenizi tavsiye ediyorum.

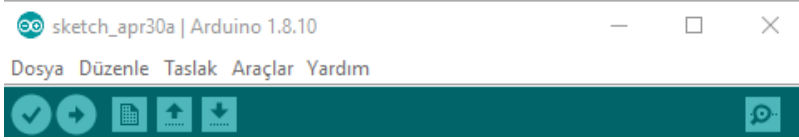
ARDUINO IDE LINK :

<https://www.arduino.cc/en/main/software>

Daha önceden de demiştim Arduino mühendislik değildir. İleriye dönük bir gömülü sistemler üzerine kodlama yaparsanız Atmel işlemcileri o zaman başka tabii.

Arduino IDE Ayarları

Biraz kurcalayalım mı ne dersin ?



Dosya : Yeni bir proje açmanızı sağlayacak kısımlar ayarlar ve hazır yazılmış kodlar barındıran kısımdır ilk etapta sıkça kullanılabilir.

Düzen: Kod içinde ki düzen işlemlerini yapmamıza yarayan kısımdır ve HTML'e dönüştürme filan bu kısımdan yapılıyor.

Çok kullandığım bir bölüm değil açıkçası ve siz de eliniz alıştıktan sonra pek kullanacağınızı düşünmüyorum.

Taslak: Doğrulama, derleme ve kütüphane ekleme gibi kısımları yaptığımız kısımdır. Çalışırken diğer kısımlara nazaran daha çok kullanacağımız bir kısımdır.

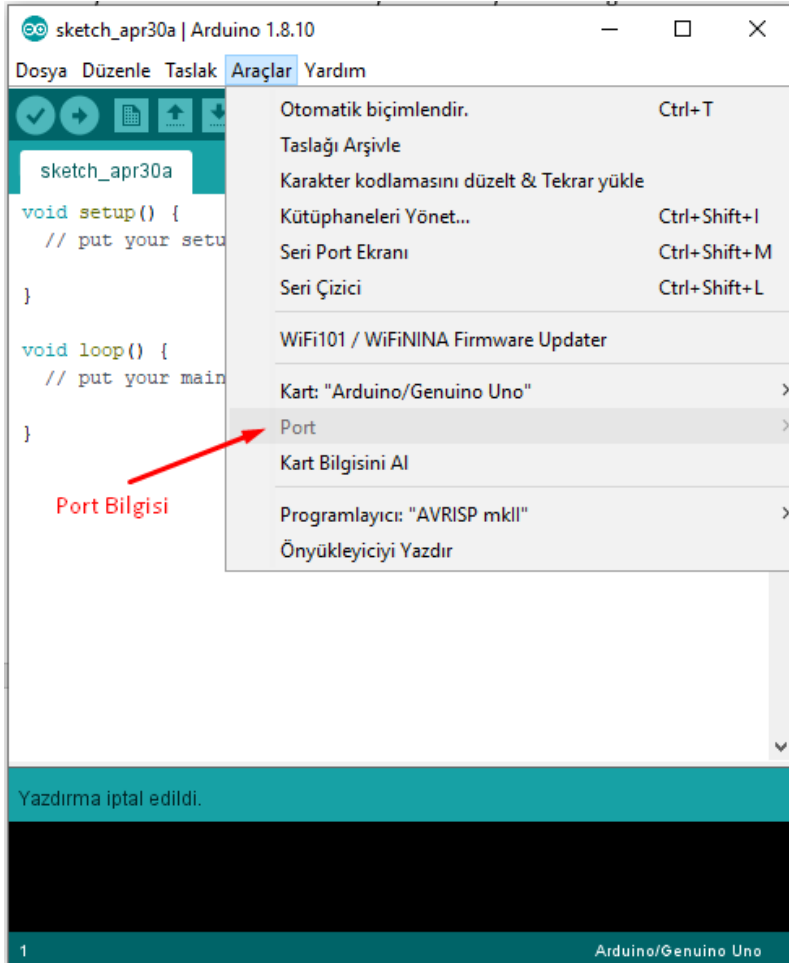
Araçlar: Kartın programlanması için gerekli kısımdır aslında diyebiliriz. Buradan denetleyici kartını olsun portunu olsun seçeceğimiz kısımdır ve burası diğer kısımlara nazaran daha çok kullanacağız buraya hakim olunması şarttır ve size yeteri kadar anlatılacak bu kitapta ve daha ilerisi için gerekli kısım ise diğer kitabımda yer vereceğim. Diğer kitapta daha çok detaya girmek istemem ve bu kitapta daha az bilgi bulunması hobi için uğraşacakların tamamen bir bilgiye boğmak yeni başlayan ya da başlamak isteyen arkadaşları sıkacaktır ve zorlayacaktır. Diğer kitabı eğer bu kitabı bitirmeden okumanız halinde kafanızda oturmayan yerler kalacaktır. Aslında bir nevi güzel bir durum ve sizi araştırmaya itebilir ama biz kötü olan ihtimali düşünelim ve size tavsiyem bu kitabı bitirmeniz.

Yardım: Bu kısımda gerekli olan şeyler (Makale anlayacak kadar bir İngilizce 😊) Neyse İngilizceniz yok ise bu hiç problem değil demek isterdim ama biz o neredeyse kısmını tamamlayacağız ve size İngilizce basit kaynakları projeleri de paylaşacağız. Hiç endişe yapmayın. Gelelim “Yardım” kısmının ne işe yaradığına şöyle başlayalım Arduino çok sade ve basit bir ara yüz yapmış olmasına rağmen belli sorunlar ya da sorular meydana gelebilir diye yardımcı olabilecek düşüncesi ile kendi sitesine yönlendiriyor. Kendi sitesinde Türkçe dil desteği mevcut olmadığı için bu konuya değinmeyeceğiz ama en güzel kaynaklar genellikle İngilizce olduğunu hatırlatmak isterim. Kaynak sayısı gün geçtikçe artmakta olan Arduino derslerini yeterli bulamayabilirsiniz ve size İngilizce kaynaklar öneririm ve bunlardan en iyisi Arduino’nun kendi sitesidir size olabildikçe iyi yardımcı olur ve bunun yanında aforum sitesi vardır ve oradan da sizlere yardımcı olacak konu başlıklarına bakabilir ve okuyabilirsiniz. Ben ise size bu kaynaklara en yakın olan eğitimi hazırlamayı hatta Türkiye de olmayan eğitim konularını ele alarak PİYASADAKİ EN İYİ KAYNAĞI sunacağım.

Şimdi gelelim kısmen bize yarayacak bilgiye:

Port: Bir iletişim uç birimidir yani Usb bir iletişim uç birimidir herhangi bir donanım taktığınız da işletim sistemi bun algılar ve ona göre işlem yapar. Biz Arduino ide de ise usb ye taktığımız arduino-muzu port seçerek kodlayabiliriz bazen bunu otomatik algılasa da seçimi her zaman el ile program (ide) üzerinden yapmak her zaman en iyisidir.

Bir sonraki sayfa da sizlere port kısmını göstereceğim devam etmenizi ve kafanızın karışmaması için sabırla beklemenizi tavsiye ediyorum.



*Port ayarlamak için gerekli kısım

Şimdilik sadece bir ön gösterim yaptım ve ilerleyen kısımlarda uygulama çalıştırırken tamamen bahsedeceğim ve beraber seçeceğiz.

Arduino Kütüphane yapısı

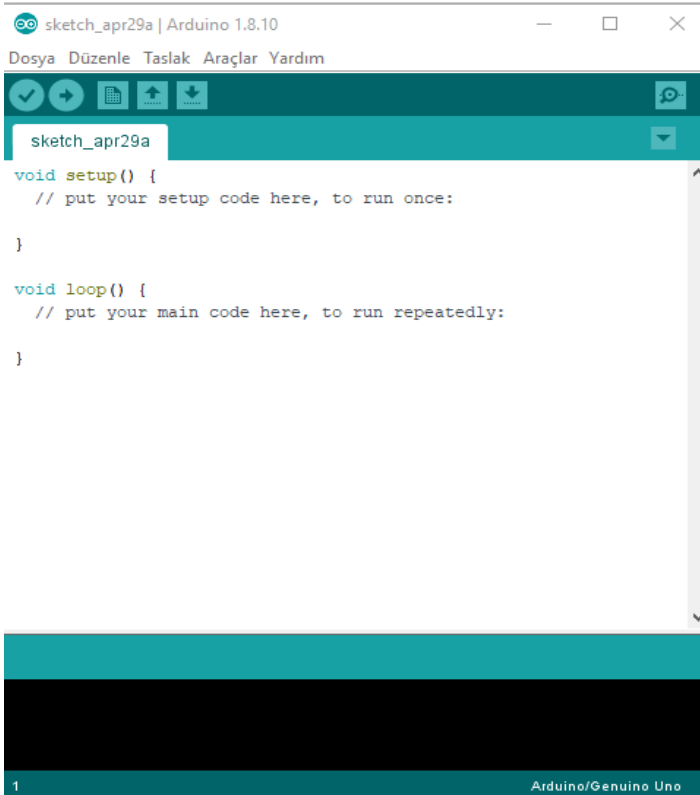
Daha önceden demişim ya arduino ide yazılım programıdır. Arduino ide bir C ile yazılmış aslında kütüphanedir diyeceğim şimdi de maalesef ki. Kütüphane bir kod için yazılmış kaynak kodudur .

Mesela bir led yakalım mı kodlama olmadan. Led yakmak için ledi elektrik ile buluşturmamız lazım ki yansın değil mi? Evet elektrik ver diye bir yapı yapalım kafamızda. Elektrik ver yapısına nasıl numara verirsek o numara yansın yani bir nevi anahtar olsun. Elektrik ver yapısı bir fonksiyondur artık ve farklı bir yerde barındırılıp programda çağırıldığında her yerde aynı şekilde kullanılabilecektir. Bu her yer de aynı şekilde çağırılabilmesi için kütüphane olacak şekilde header dosyası yapmanız gerekmektedir. Bir toplama fonksiyonu yazalım hemen oturduğumuz yerde o zaman çok basit değil mi? İki sayıyı alsın toplasın sürekli. Toplama kısmı zor bir durum değil ama her yer de evrensel kullanılıyor ise bu mantık aslında bir kütüphane olmuştur yani bizim bilgilerimiz de beynimizdeki kodlanmış bilgileri kütüphanemizden (beynimiz) çağırmış oluyoruz.

Şimdi biraz daha kodlama kısmına geçelim ki bu dediklerim tam anlamıyla anlaşılsın. Hepimiz rahat bir nefes alalım ve artık önyargılarımızdan kurtulmuş olalım. 😊

Arduino ile Programlama

Arduino ile programlama konusunda bilmeniz gereken iki ana unsur vardır. Birinci unsur arduino bir ön tanımlama kullanması zorunludur. İkinci unsur ise arduino sürekli ¹ döngüye girmektedir. (Elektrik olduğu sürece sonsuza kadar aynı işlemleri gerçekleştirebilirsiniz)



```
sketch_apr29a | Arduino 1.8.10
Dosya Düzenle Taslak Araçlar Yardım

sketch_apr29a
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}

1 Arduino/Genuino Uno
```

*Bir boş yapımızı görüyorsunuz ve bu kısım daimî aynıdır.

¹ Döngü : Sürekli tekrar eden.

Void setup () { } : Kurulum fonksiyonu diyebiliriz bu bloğun içine kodumuzda kullanacağımız giriş çıkış elemanlarını ve tanımlamalar için kullanacağız. Void setup kod bloğu bir defa okunur (bir defa çalıştırılır bunu sakın aklınızdan çıkarmayın).

Void loop() { } : Sürekli döngü fonksiyonudur. Bu kod bloğu içine yazılanlar arduino çalıştığı süre boyunca tekrar eder. Void loop kod bloğunu içerisine ¹ işlemlerimizi yazacağız ve sürekli olarak kontrollerimizi sağlayacağız.

Serial Port Ekranı

Serial port bizim seri haberleşmemiz için olan ekrandır. Serial port bizim aslında görsel anlamda çıktıları görmemize yarar diyebiliriz. Serial port ekranı hakkında “İleri seviye Arduino Kodlama” kitabında yazılar bulunmakta ve C# üzerinden seri haberleşme ile gelişmiş ara yüzler yapacağız. İkinci kitabı da takipte kalın

Serial port ekranı baud cinsinden seçilir bu baud veri aktarım hızı olarak karşımıza çıkar ve buna önem gösterilmesi gerekmektedir . Veri aktarım hızları uyumsuzluğu olursa kodunuz çalışmayacaktır veya hatalı çalışacaktır diyebiliriz buna önem göstermeniz gerekmektedir. (Kodunuz çalışır ama gözükmez!)

Serial print() : Serial ekran üzerinde görüntüleme yapmak için bu fonksiyon kullanılmaktadır. Yukarıda da gösterildiği gibi kullanımı olan bu yapı bize programladığımız cihazı Arduino IDE ile ² değer göstermemizi kolaylaştırır.

¹ Void setup içerisine de işlem yaptırabiliriz aslında bunu ilerleyen dersler de anlayacaksınız. Değer döndürmeme anlamına gelir diye bilin şimdilik.

² Değer : Değer herhangi bir çıktı veya bir tanımın değeridir.

Arduino Pin Yapısı ve Tanımlamalar

Arduino pin yapısı **Dijital** ve **Analog** olarak ayrılmıştır. Dijital pinler genellikle kullanacağımız pinlerdir bunu zaten farkedeceksiniz .

Arduino Pin Programlama



```
pin_yap_s_ | Arduino 1.8.10
Dosya Düzenle Taslak Araçlar Yardım

pin_yap_s_

byte pin = 1; // pin değişkenine 1 atandı
byte pin2 = 2; // pin2 değişkenine 2 atandı

bool okunanDeger; // okunanDeger diye bir değişken oluşturuldu
bool okunanDeger2; // okunanDeger diye bir değişken oluşturuldu

void setup() {
  // Arduino açıldığında :
  pinMode (pin, INPUT); // 1. pinin INPUT yani giriş söyleniyor
  pinMode (pin2, OUTPUT); // 2. pinin çıkış olacağı söyleniyor
}

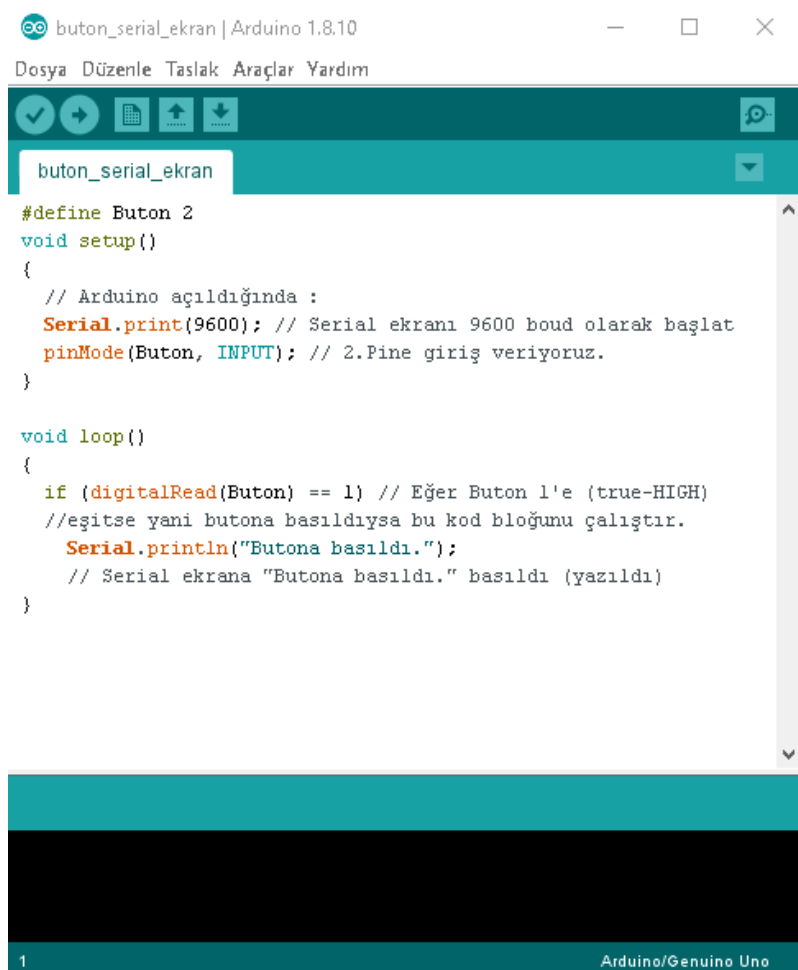
void loop() {
  // Arduino çalıştığı sürece :
  okunanDeger=digitalRead(pin); // 1.pinden değer okundu (T/F)
  okunanDeger2=analogRead(A1); // A1 pininden değer okundu (T/F)
  digitalWrite (pin2,HIGH); // 2.pine elektrik verildi
  delay (10000); // 10 saniye bekle
  digitalWrite (pin2,LOW); // 2.pine gelen elektrik kesildi
}
```

Kaydedildi.

21 Arduino/Genuino Uno

Tek değinilmesi gereken şu anda Analog pinler ön tanımlamaya ya da void setupta olmasına gerek yoktur.

Arduino Buton Okuma (Dijital)



Arduino dijital buton okuma işi bu kadar kısa bir kodla yapılıyor görüldüğü gibi şimdi bu kodu açıklayalım öğrenelim.

#define Buton 2 : Define yapısı bir tanımlama yapısıdır ve global yani her tarafta okunabilir yazılabilir hale getirir diyebiliriz şimdilik yani bura da “Buton” diye bir değişkene “2” diye bir sayı atandı.

Serial.begin(9600) : 9600 baud olarak ekrana baskı yapmak istediğimizi açıkladık eğer seri port ekranını da 9600 baud olarak ayarlarsınız çıktı aldığımızı göreceksiniz.

pinMode () : Pinlerin burada türünü açıkladığımızı belirtmek isterim bunu önceki bölümlerde açıkladığımız için bundan sonraki kodlarda bu konuda açıklama yapılmayacaktır.

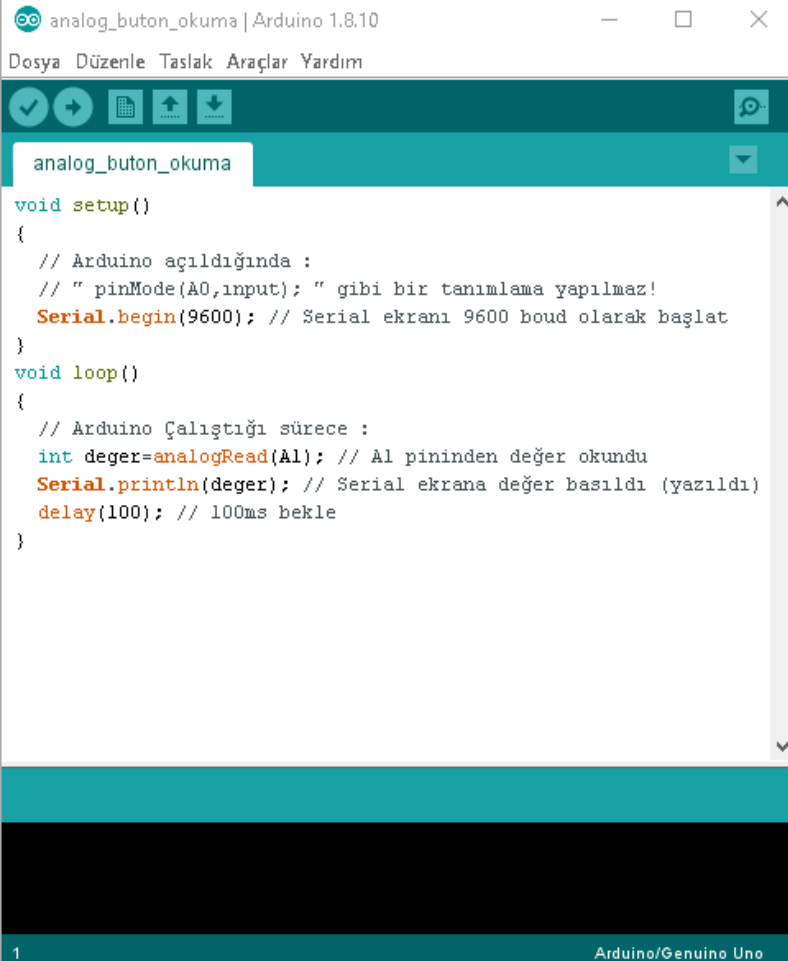
if() : If yapısı bizim en çok tercih edeceğimiz yapılardır “eğer” anlamına gelmektedir ve eğer bu ise dedikten sonra kod bloğunun içindekiler yapılır.

Serial.print() : 9600 baud olarak ayarladığımız ekrana baskı yapmamızı sağlar.

¹**Serial.println()** : 9600 baud olarak ayarladığımız ekrana baskı yapmamızı sağlar.

¹ Tek fark şudur “ln” takısı bir alt satıra geçmemizi söyler ve sürekli satır atlayarak baskı verdiğini göreceksiniz.

Arduino Buton Okuma (Analog)



```
void setup()
{
  // Arduino açıldığında :
  // " pinMode(A0,input); " gibi bir tanımlama yapılmaz!
  Serial.begin(9600); // Serial ekranı 9600 baud olarak başlat
}
void loop()
{
  // Arduino Çalıştığı sürece :
  int deger=analogRead(A1); // A1 pininden değer okundu
  Serial.println(deger); // Serial ekrana değer basıldı (yazıldı)
  delay(100); // 100ms bekle
}
```

Analog buton okuma denildiğinde ise analog okuma açıklama olarak ise :

Serial.begin(9600) : 9600 baud olarak ekrana baskı yapmak istediğimizi açıkladık eğer seri port ekranını da 9600 baud olarak ayarlarsınız çıktı aldığımızı göreceksiniz.

int deger : Integer yani tamsayı bir değişken yaptık ve bu değişkenin içine aşağıdaki fonksiyon ile atama (değer eşitleme de diyebiliriz) yapıldı.

analogRead() : Analog giriş tanımlayıp değer okuması için gereken fonksiyondur burada okunan değer ise “deger” denilen değişkene aktarılmıştır.

Serial.print() : 9600 baud olarak ayarladığımız ekrana baskı yapmamızı sağlar.

¹**Serial.println()** : 9600 baud olarak ayarladığımız ekrana baskı yapmamızı sağlar.

²**delay ()** : Milisaniye olarak bekleme sağlanır bekleme esnasında Arduino işlem yapmaz kısmi olarak çalışmıyor gibi düşünebilirsiniz.

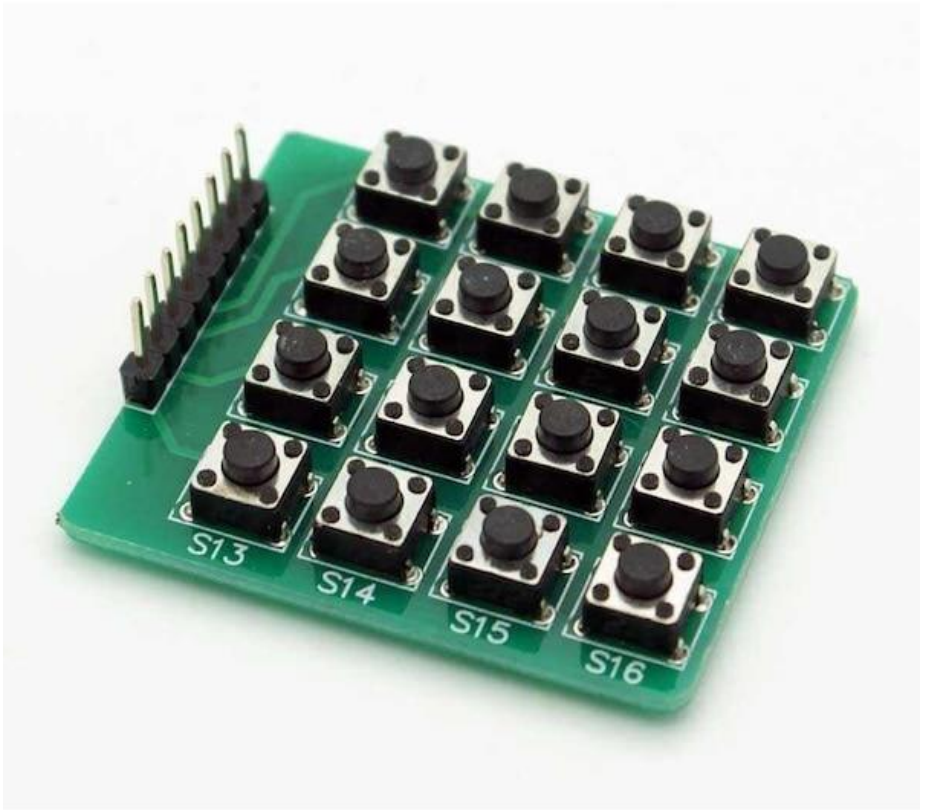
¹ Tek fark şudur “\n” takısı bir alt satıra geçmemizi söyler ve sürekli satır atlayarak baskı verdiğini göreceksiniz. İstenirse alt alta da yazdırabilirsiniz bu kodu düzenlemek isterseniz diye size seçenek sunulmuş oluyor.

² Delay yerine ileri de çalışmayı aksatmaması için bir başka fonksiyondan bahsedeceğiz ve bu fonksiyon ile arduino işlem yapmaya devam ederken sayım yaparak size istenen süreye kadar devam ettirir koşullar sağlandığında ise işleme alınır ve bekleme yapılmış olur.

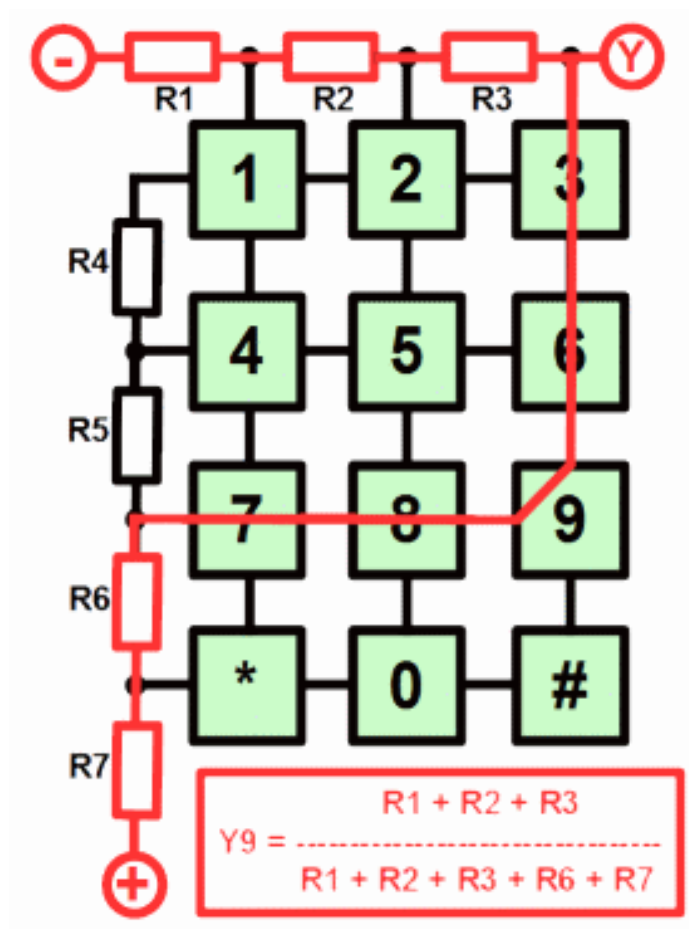
Arduino Birden Fazla Buton Okuma

Bu konuda ise dijital pinleri meşkul etmek yerine anlog deęerler okunup istenilen kadar buton tek pinden okunabilir buna matris okuma da denebilir.

Birden fazla buton okuma iři iin rnek verecek olursak ařaęıdaki donanımlar buna rnek verilebilir.

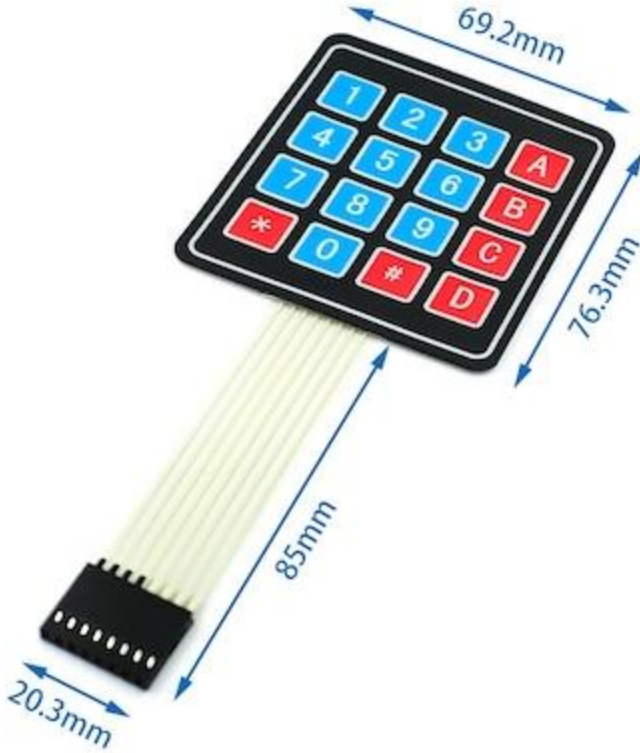


¹Bu matris butonunu çalışma mantığı ise şudur :



¹ Bu yapı biraz karmaşık gelebilir ilerleyen bölümler de daha net anlayacaksınız.

Bu yapının çalışması dirençler ile bölme işlemidir sadece bu bilgi sizlerin kafasındaki soru işaretlerini gidermek için verilmiştir. Şu anlık dirençler ile uğraşmayacağımız için anlamanız beklenmemektedir. İlerleyen bölümler de net bir şekilde anlatılacaktır.



Bu yapı özellikle “otomat” gibi makinalarda kullanılır 16 tane buton 8 pinden kontrol edilebilir. Bu key pad fazla pin kullanımına sebebiyet verir. Sizlerle diğer kitapta kendimizin tasarımı ile bir keybo-

ard yapacağız. 😊 Lütfen kendinizi bir şey bilmiyormuş gibi hissetmeyin çoğu kişi bu bilgiyi bilmez çünkü biraz daha da teknik bir yapıya giriyor. Diğer kitabı okuyarak bilinmeyen ya da az bilinen yöntemleri bilenler arasında olacaksınız. Ben bu kısmı geçiyorum şimdilik ve bu bilgileri hem tecrübe ile daha iyi hale getirmenizi hem de Youtube üzerinden takip ederek daha da uygulamalı görmüş olacaksınız. Kendiniz tecrübe etmeniz daha iyi anlamanızı sağlar. Unutmayın insan tecrübe ettiği şeyi asla unutmaz. (Bak o elin değdiğinde sobadan yaktığın elin yüzünden sobaya dokunmuyorsun değil mi ? 😊)

TEMEL ELEKTRONİK FORMULÜ

$$V = i \times R$$

Nereye yazarsan yaz ama sürekli gözünün önünde olsun. Özellikle beynine yaz ama bunu hemen yazabilir misin bilmiyorum tekrar etmen lazım kalıcı olabilmesi için.

Arduino ile projeler kısmına geldik hazır mısın?

Bence dünden hazırsın ve artık söylemekten usandım, sende okumaktan ama zor değil bunlar dostum. Zor dediğin şey senin basit düzey de kaldığını ilerisi için uğraşmadığını gösterir ama eğer ona ulaşmak istiyorsan “zor” deme “çabalıyorum” de. Umarım bu projeler senin ufkunu açacaktır.

Arduino Led Yakmak

Arduino da her zaman “Led Yakma” dersi ile başlar ben de bu geleneği sürdürmek istedim. Led yakmak aslında her şeyin en karmaşık halini sana en hızlı anlatan sürekli görmüş olduğun lambayı andırır. Led yakmak kafana daha iyi yatar çünkü insan beyni her şeyi birbirine bağdaştırıp bağlantı kurmayı sever. 😊

Şimdi size bir bilgi vereyim mi? Aslında o gördüğün pahalı işlemciler var ya “Evet!” işte o 2bin lira verilen işlemciler onlar bir sürü anahtardan oluşur. Doğru duydun anahtar en azından kafanda canlandırman için anahtar diyelim ilerde transistör diyeceğiz. İşlemci de aslında yaptığı bir yerlere elektriği iletmek ama bunu yaparken belli bir düzende ilettiği için daha çok, bir işlem yaparak iletiyor diyebiliriz. “Bana bunu niye anlatıyorsun” dersene de o lambayı açıp kapattığın lamba anahtarı var ya, işte o bilgisayarın atası diyebilirsin. Şimdi oldu mu? Her şey aslında neden basitmiş? Çünkü: sen geneli görmeden tek bir şeye odaklanıyorsun lütfen oku araştırmayı ve kendini geliştir. 😊 Ben de sana araştırmayı için kaynak sağlayayım olur mu? Beni takip et hadi gidiyoruz ne bekliyorsun...

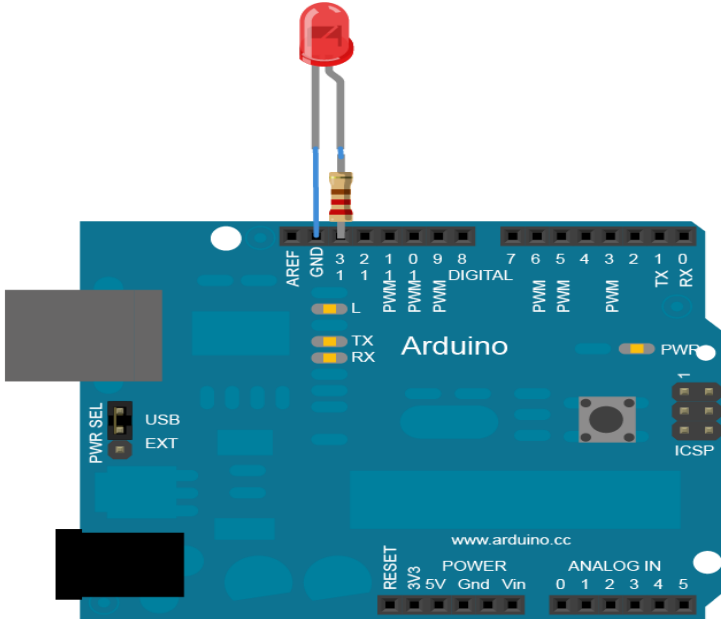
Led için öncelikle bilmemiz gereken formül $V=I.R$ 'dir . Bir önceki sayfa boşuna verilmedi.

Öncelikle bir arduino 5V çıkış verdiğini unutmayalım. Bir bacak en fazla 50mA çıkış sağlar ve ona göre yüksek değerli cihazlar bağlamayınız. Artık belli bir bilgi birikimine ulaştığımızı kabul ediyorum ve ona göre kademeli olarak projeye girişiyorum.

“ $V=I.R$ ” demiştik şimdi bir led ne kadar akım çeker ona bakalım.

Her yerde led bağlama ile yazılar var ama basit bir ledin kaç mA harcadığını yazan yer yok. İnternet sanıldığı kadar detay vermez maalesef ki. **Bir led 20mA civarı harcama yapar.** Voltaja göre şimdi size hesaplamalar yapalım ve ona göre bize direnç lazım mı onu görelim. Unutma arduino 50mah veriyor!

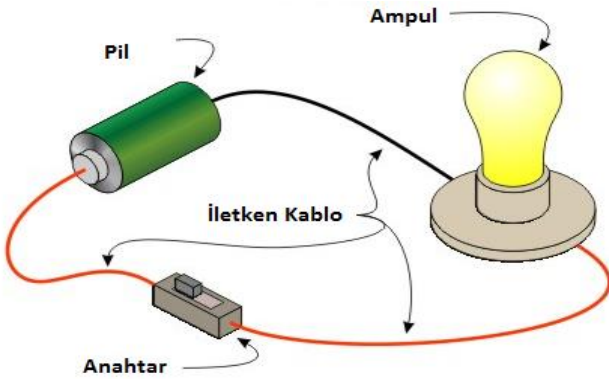
$5000\text{mV} = 20\text{mA} \times R \rightarrow$ Bu formüle göre 250ohm dirence gerek vardır. Direnç takmazsanız da lediniz çalışır fakat ciddi olarak bir ömür kaybı yaşatır bir önceki sayfa da dediğimiz gibi 50mA akıma maruz kalacak ve 2.5 katı bir amper ile ömrü çok kısılacaktır hatta patlayabilecektir. Direnç konusunu da çözdük ve artık bağlantı şemasına girip kodlama ile artık biraz daha yazıdan çok incelemele-re ve kısa kısa açıklamalara gireceğiz çünkü farkındayım ki sıkıyor bu durum.



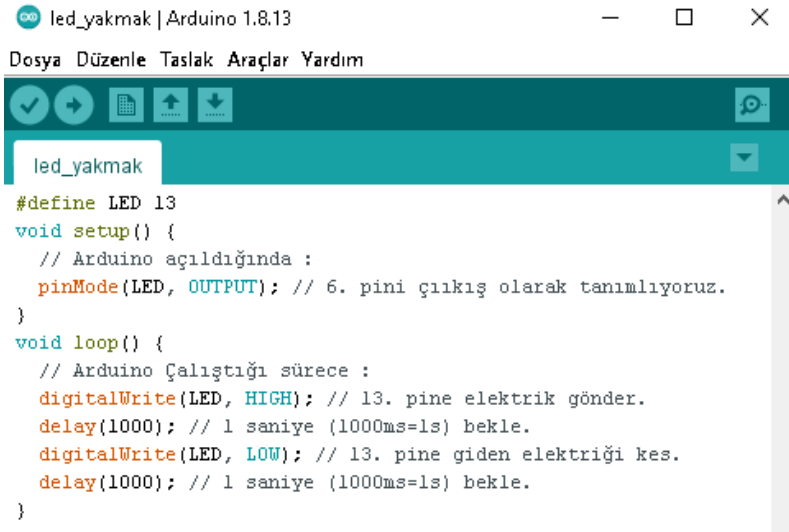
*Breadboard kullanımı için bilgilendirme yapıp öyle breadboard kullanımı yapacağız.



Uzun bacağı “+” (artı) bacağı olarak kabul ediyoruz ve diğer bacak ise “-” (eksi) bacağıdır. Elektrik devresi için “+” ve “-” birbirini daimi tamamlamalı. Bunu kısmi olarak şöyle göstereyim.



Bu devre ile daha iyi aklınıza oturur.



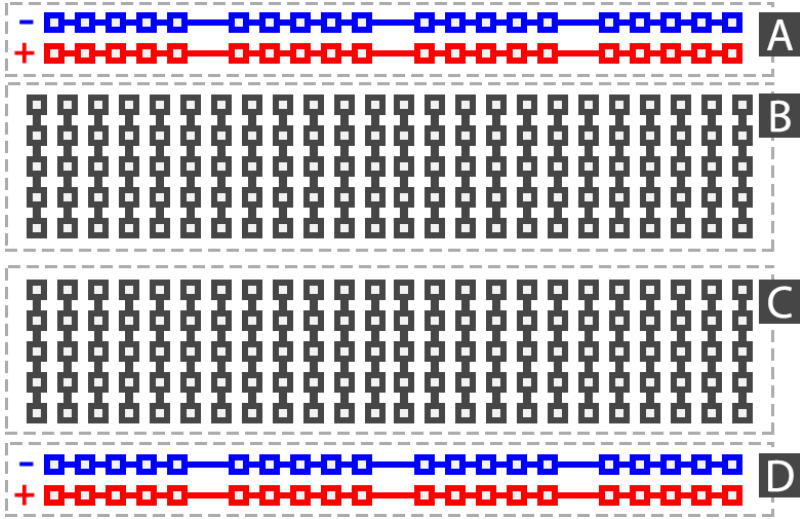
```
#define LED 13
void setup() {
  // Arduino açıldığında :
  pinMode(LED, OUTPUT); // 6. pini çıkış olarak tanımlıyoruz.
}
void loop() {
  // Arduino çalıştığı sürece :
  digitalWrite(LED, HIGH); // 13. pine elektrik gönder.
  delay(1000); // 1 saniye (1000ms=1s) bekle.
  digitalWrite(LED, LOW); // 13. pine giden elektriği kes.
  delay(1000); // 1 saniye (1000ms=1s) bekle.
}
```

Yukarıda açıklamaları ile verilen kodun üstüne pek düşmeyeceğiz . İleride bu açma kapama işlerini C dilinde yapmak daha hızlı olacaktır şimdiden belirteyim.

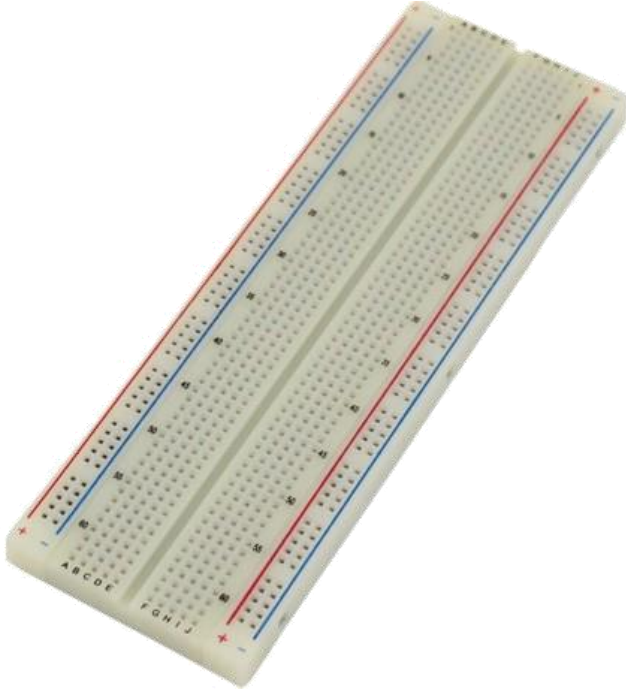
#define LED 13 -> Bu kod define komutu ile artık “LED” (tırnak işaretleri mevcut değil) çağrılan (yazılan) yerlerde 6 sayısını çağıracaktır.

delay(1000) -> Bir bekleme komutudur sizin de bu komutla çalışmalar yapmanızı tavsiye. Bu komut çalıştırıldığında cihaz hiçbir işlem yapmadan bekleyecektir ve bu komut size bazen yeterli gelmeyebilir. Delay komutu sürekli işlem gerektiren kodlarda yeterli değildir çünkü sürekli işlem yapılması gerekli olup bekleme işleri aksatacaktır. Aksatmalar yaşanmaması için belli çözümler var ve ilerleyen kısımda anlatılacaktır. Devam edelim o zaman.

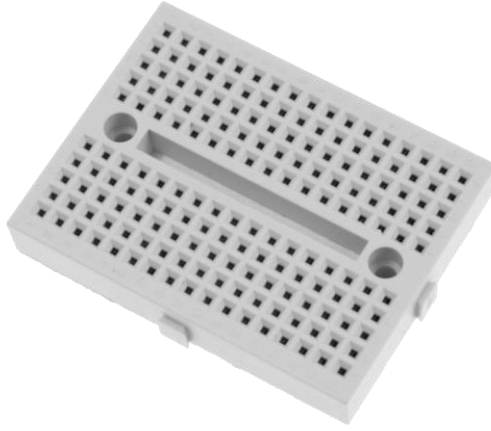
Breadboard Nedir?



Breadboard seri bağlantılar içeren elektronik devreleri basitleştirme aracıdır. “A” ve “D” kısımları genellikle güç beslemeleri için kullanılmaktadır sizin de böyle bir tercihte bulunmanızı öneririm. “B” ve “C” kısımları ise genellikle sensörler için filan kullanılan kısımdır ve güç besleme kısmına göre yatay değil dikey seri bağlantılar vardır. Az önceki örneği (LED Yakmak) breadboard üzerinden yapmayı deneyelim. Ama öncelikle breadboard görselleri verelim ve sizler de bunları bir inceleyin. Nasıl bir görünüme sahip bu alet bir sonraki sayfaya ilerle. →

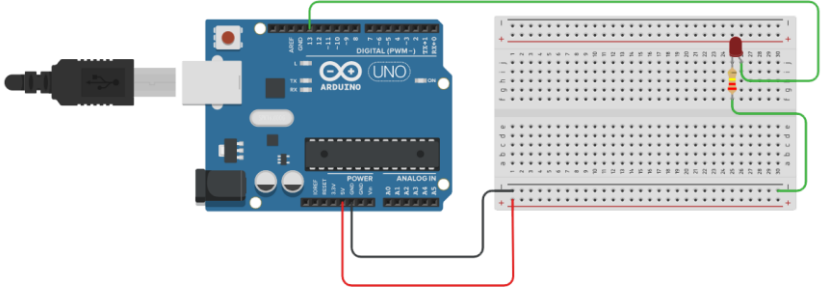


Breadboard denilince akla ilk gelen model budur ve bu model genel olarak herkes tarafından kullanılır. Çalışma yapısı önceki belirtmiş olduğumuz yapı ile aynı olup hiçbir fark yoktur. Sizlere kolaylık olsun diye artı ve eksi kısımları güzelce belirtilmiştir. Artı ve eksi kısımları güzelce kullanarak kafanız karışmadan projelerinizin prototiplerini yaparken kullanabilirsiniz. İleri seviyeler de bu kartı sadece protoip üretmek için kullanın dahası için stabil değildir. Bu kartlar da bazen belli yerlerinde kopmalar olabiliyor. Kart bağlantıları genel olarak fazla akım geçtiğinde, yanlış bağlamalar da kopmalar yaşatabiliyor bunu aklımızda tutalım. Daha küçük modelleri olan bu Breadboard'un küçük olanını da görelim. →



Bir önceki sayfa da ele aldığım breadboardun bir küçük modeli olan bu model daha küçük çalışma alanları için kullanılıyor. Burada biraz işler karışıyor açıkçası. Bu model de artı ve eksi kısmı mevcut değil bunu gözünüzle bile farkedebilirsiniz bunun yerine diğer kalan kısmı mevcut. Artı ve eksi kısmını siz bunu bilerek vereceksiniz sisteme bunu aklınızda bulundurun ve bunu bence pek gereksinim yoksa kullanmayın bile çünkü belki de benim başıma geldi fakat 5 adet almış olduğum ürünün 3 ünde bazı veri yolları iletim sağlamıyordu. Devam edelim mi?

Klasik bir eğitimde olsaydık büyük ihtimalle sizlere led yakmak üzerine daha da video ya da yazı hazırlardım ama bunu ilk iki sayfa da bile hallettik emin olabilirsiniz. Sadece ledi breadboarda bağlayacağım ondan sonra devam edeceğim.



Breadboard üzerinde duruşu bu şekilde ve artık gösterimini anlamışsınızdır ama yine de açıklama yazıp devam edeceğim.

13 nolu pin → Ledin uzun bacağına (Anot bacağı)

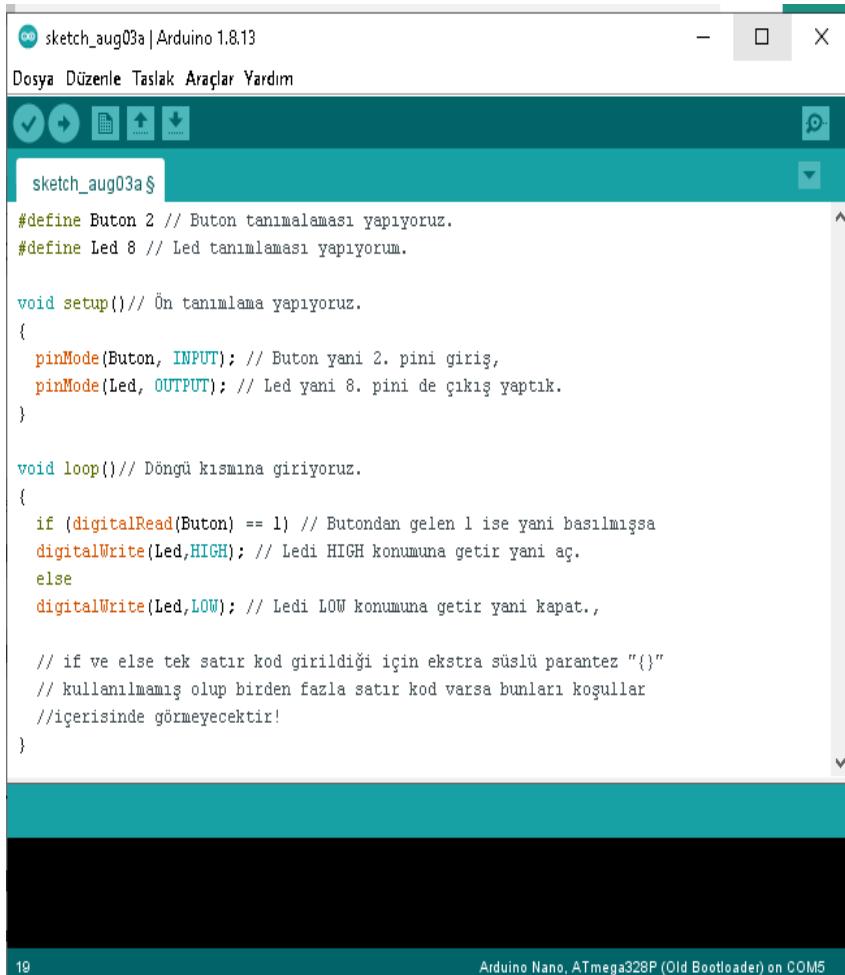
Gnd pini → Ledin kısa bacağına (Katot bacağına)

5v ve Gnd pinlerini daimi olarak sisteme bağlıyorum çünkü güç gerektiren zamanlarda uğraşmak istemiyorum tekrar.

Evet şimdi yazılım kısmına geleyim. Arduino ve çoğu dillerde bulunan genel terimler var bunlar: if-else, while, for, switch-case. Bunları tek tek inceleyelim biraz yazılım bilgisine girerek.

Kodlama Yapıyoruz

Kodlama kısmına geldik tam olarak artık. Daha önceden bir kısmı başlangıç yapmıştık ve bunun üzerine koymak gerektiğini düşünüyorum. Arduino kodlarken çok if else yapısı kullanacaksınız bunun için bence bu kısmı öğrenelim buton okumayı öğrenmiştik zaten.



```
sketch_aug03a | Arduino 1.8.13
Dosya Düzenle Taslak Araçlar Yardım

sketch_aug03a $

#define Buton 2 // Buton tanımlaması yapıyoruz.
#define Led 8 // Led tanımlaması yapıyoruz.

void setup() // Ön tanımlama yapıyoruz.
{
  pinMode(Buton, INPUT); // Buton yani 2. pini giriş,
  pinMode(Led, OUTPUT); // Led yani 8. pini de çıkış yaptık.
}

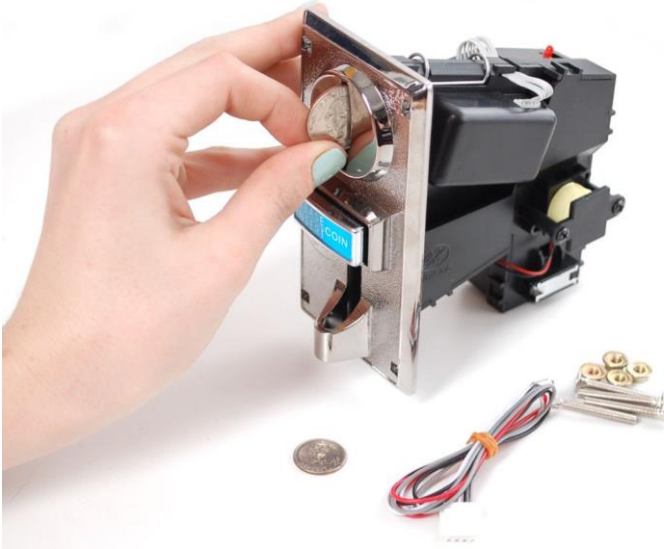
void loop() // Döngü kısmına giriyoruz.
{
  if (digitalRead(Buton) == 1) // Butondan gelen 1 ise yani basılmışsa
    digitalWrite(Led,HIGH); // Ledi HIGH konumuna getir yani aç.
  else
    digitalWrite(Led,LOW); // Ledi LOW konumuna getir yani kapat.,

  // if ve else tek satır kod girildiği için ekstra süslü parantez "{}"
  // kullanılmamış olup birden fazla satır kod varsa bunları koşullar
  //içerisinde görmeyecektir!
}
```

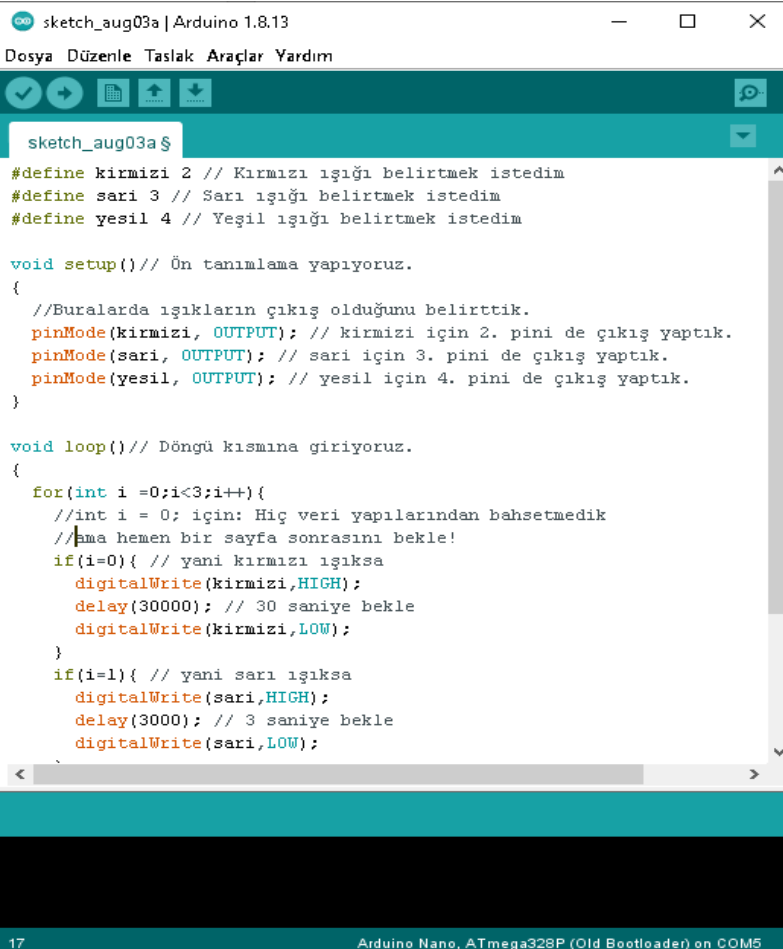
19 Arduino Nano, ATmega328P (Old Bootloader) on COM5

Bir önceki sayfa da istediğimiz kodu yazdık. Açıklamaları da içerisinde barındırıyorum fakat yine de buraya da yazayım.

İf ve else koşul olayı yani bana bir lira verirsen bir ekmek alırsın ama daha farklı bir para verirsen ekmek alamazsın. 2TL para verirsen 2 ekmek eder ama ben 1TL istediğim için 2TL benim koşulum değil ve benim koşulum dışında kaldığı için ekmek alamazsın. Bu örneği otomatları örnek göstermek için verdim misal veriyorum gelin bir otomat yapalım. Otomat içerisinde sizlere diyorum ki 1TL ile su alabilirsin onun dışındaki paraları kabul etmiyorum. Ben bu yapmış olduğumuz otomatı kullanırsam ve 25 kuruş, 50 kuruş atarsam çalışmayacak. Aslında verdiğim örneklerden sizlere bir koşul sundum ve bu yazılımın en büyük kısmıdır. Beraber bir sürü if-else komutları yazacağız bolca koşul ifadeleri yazacağız. Sizlere aslında örnek verirken bile fark etmeseniz de otomat mantığını anlattım size bunu yapmanızı sağlayan aleti de ileriki kitabıma saklayacak olsam da sadece modülünü göstereyim istedim.



Sizlere bazı bilgileri sunarken fark ettirmeden de olsa bazı teknolojilerin aslın da basitliğini söylemiş oluyorum. İnşallah bunu fark etmişsinizdir. 😊 Evet geldik for kısmına bu kodu anlamak için de sizlere bir trafik ışığı örneği yapayım aslında bunun için for döngüsüne ihtiyaç duymadan void loop üzerinden yapabiliriz ama sadece görün diye sizlere sadece yapıyorum.



```
sketch_aug03a | Arduino 1.8.13
Dosya Düzenle Taslak Araçlar Yardım

sketch_aug03a $
#define kirmizi 2 // Kırmızı ışığı belirtmek istedim
#define sari 3 // Sarı ışığı belirtmek istedim
#define yesil 4 // Yeşil ışığı belirtmek istedim

void setup()// Ön tanımlama yapıyoruz.
{
    //Buralarda ışıkların çıkış olduğunu belirttik.
    pinMode(kirmizi, OUTPUT); // kırmızı için 2. pini de çıkış yaptık.
    pinMode(sari, OUTPUT); // sari için 3. pini de çıkış yaptık.
    pinMode(yesil, OUTPUT); // yesil için 4. pini de çıkış yaptık.
}

void loop()// Döngü kısmına giriyoruz.
{
    for(int i =0;i<3;i++){
        //int i = 0; için: Hiç veri yapılarından bahsetmedik
        //ama hemen bir sayfa sonrasını bekle!
        if(i=0){ // yani kırmızı ışıkta
            digitalWrite(kirmizi,HIGH);
            delay(30000); // 30 saniye bekle
            digitalWrite(kirmizi,LOW);
        }
        if(i=1){ // yani sarı ışıkta
            digitalWrite(sari,HIGH);
            delay(3000); // 3 saniye bekle
            digitalWrite(sari,LOW);
        }
    }
}
```

17 Arduino Nano, ATmega328P (Old Bootloader) on COM5

Açıklama yine de yapmaya devam edelim. Bir trafik ışığı için üç renk mevcut ve bu renkler sırasıyla kırmızı, sarı ve yeşil olduğu için bu sıra ile sizlere belli aralıklarla bekletme yaptırıyoruz. Bir trafik ışığının çalışma mantığı bile bu kadar basit yani isterseniz bile kendi evinize kendi çocuğunuzun oyuncak arabasına bir trafik ışığı düzeneği kurabilirsiniz. Hemen bir veri tipleri diyelim ve bu yapmış olduğumuz projenin de bir breadboard üzerinde duruşunu görelim.



Veri Tipleri

TİP	BOYUT	ARALIK
int	2 bayt	-32768 ile 32767 arası
unsigned int	2 bayt	0 ile 65535 arası
long	4 bayt	-2147483648 ile 2147483647
unsigned long	4 bayt	0 ile 4294967295 arası
char	1 bayt	-128 ile 127 arası
byte	1 bayt	0 ile 255 arası

Veri tiplerinin kısmı bir boyut açıklaması bu ve bu int, unsigned int ve byte bir sayı tutar, char ise bir karakter tutar ve bunların önüne long getirerek daha da büyük bir boyuta kavuşturmuş olacağız yapıları.

int i = 10; → Bir “i” diye değişkenimiz var ve bu değişkenimizi nerede çağırırsak çağıralım integer tipinde yani tam sayı tipinde 10 sayısını getirecek. Bunu aslında bir rakamdan ziyade ikili tabanda kodlanmış olarak getirecek fakat siz onun 10 olduğunu bileceksiniz bu kısmın devamı da var ama ileri seviye olacağı için giremiyorum.

char harf= ‘1’; → Char yapısı sadece bir karakter ile çalışma yapar aslında bazı sim kartı modülleri filan bu yapıyı kullanır bunu bilmeniz gerekir ama bu kitap için söylemek gerekirse değinmeyeceğim.

String isimSoyisim= “Ali Birkan”; → String yapısı aslında bir char veri tipinin genişletilmiş halidir ve bazı diller de “String” diye bir veri tipi mevcut değildir. String olmayan diller kısmına girmeyeceğiz ama string aslında size büyük bir char alanı tanıyor yani bir ismi char ile yazmak için sürekli olarak char içine tanımlama yapmak gerekecektir (Basit düzey için bunu diyorum ileri düzey bilginiz mevcut ise bunu pointer ile rahatlıkla yapabildiğimizi biliyorsunuzdur).

Kısmen de olsa sizlere veri tiplerini gösterdim diyebilirim ama benim de çok kullandığım iki yapı halen duruyor.

bool→ Bool yapısı doğru ve yanlış tutar sadece ve bu yapı bize karar verme aşamasında çok yardımcı olacak bir yapı aslında. Bu veri tipini de bolca kullanırız.

byte→ Byte size küçük tamsayı tipleri için yardımcı olur ve ram denilen belleği doldurmamak için bolca kullanın.

For döngüsüne geri dönecek olursak aslında yapı sizlerin de anlayacağını düşünmüş olsam da açıklayayım. For yapısı:

for (başlangıç değeri; döngünün devam etme koşulu;
değerdeki değişim)

```
{  
    İç tarafta yapılacak işlemler...  
}
```

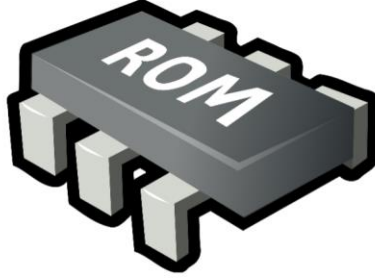
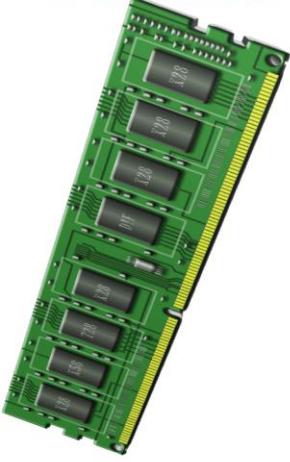
Bu başlangıç değeri kısmında veri tipi ile veri belirtebiliyoruz lakin bu da bazı diller için yapılamıyor. Siz şu yazım kuralına alışın bence:

```
int i=0;  
for(; i için bir koşul; i değerindeki değişim){  
    İç tarafta yapılacak işlemler...  
}
```

Evet ilk kısmı boş bırakıp i değerini sonraki koşul kısımlarına dahil ediyoruz. Bu kullanım dediğim gibi daha iyi bir kullanım sunar ve bu kullanım sizleri başka dillere geçerken daha rahat olmanızı sağlar ve gerektiği zaman bu i değerini rahat rahat ram denilen kısımdan silebilirsiniz yani size işinizi bitirince yer açıp açmadığınız konusunda kolaylık sağlar (Diğer türlü de i değişkenini serbest bırakabilirsiniz fakat benim size önerdiğim bu yazımda göze daha çok çarptığı için en azından serbest bırakıp bırakmadığınızı takip edebilirsiniz).

Kısaca Ram ve Rom:

RAM v/s ROM



Kısaca ram ve romdan bahsetmek istedim çünkü bu yapıları bilin ve ona göre aslında işlem yapacağız. Yazılım yaparken daima bunları göz önünde bulunduracağız. Bir led yakıp söndürmek için pek gereksinim yoktur fakat araya sensörler girince proje kapsamı büyüyünce mecburen bunlara dikkat etmemiz gerekiyor.

Ram Nedir?

Ram rastgele erişimli bellektir. Asıl adı Random Acces Memory olan bu bellekler tamamen okunabilir yazılabilir bellekler olup bu okuma yazma işini işletim sistemi ayarlar ve program izin verilen yeri alır kullanır. Ram aslında birincil bellektir bütün işler bura da döner. Programlar burada çalışır aslında ilk işlemler hafıza olarak buraya yerleşir ve bu yerleştirmeden sonra işleci bu bellekteki verileri işler. Şu anda kullandığınız hard disk, ssd den çok daha hızlı

olduğu için birincil hafıza olarak ram kullanılır ve bu hızdan dolayı da harddisk, ssd gibi depolama birimlerine göre birim fiyatı daha pahalıdır. 8Gb ram ile belki de 1TB harddisk aynı fiyat olabilir. Gelelim bizim için yarayan kısma biz bir yazılım yazarken (Arduino için bahsedilmiyor Arduino SRAM yapısını kullanır ama aynı da kabul edebileceğimiz kadar benzerdir) kullandığımız değişkenler, program aslında ilk olarak bu ram kısmında saklanıyor yani siz bir integer değer yazdınız kodda işte diyelim ki:

int sayi=42; → bu verilen sayı değişkeni ram'in bir kısmında saklanır ve değeri 42 olarak atanır. İleri de belki anlatırım ama araştırmak isterseniz pointer denilen bir yapı mevcut ve bu ram deki adreslere rahatlıkla uğraşabiliyoruz. Sizlerin daha da iyi anlaması için bir örnek vereyim belki küçükken oyunlara hile yapmış olabilirsiniz. Cheat engine diye bir program vardı bu programa oyunda ki değiştirmek istediğimiz sayıyı yazıyorduk ve buluyordu. Cheat engine'nin çalışma mantığı aslında budur yani sizden gelen değer ile ram de olan değerleri tarıyor ve sizin karşınıza çıkarıyor bulabilirsiniz değiştirebiliyorsunuz. Sizlere daha da fazla açıklama yapabilerim ama giriş için bu kadar bilgi yeteceğini düşünüyorum geri kalan kısmını kendiniz araştırmanızı tavsiye ederim. Son olarak ram i şöyle özetleyebilirim: veri depolanabilir, silinebilir, okunabilir, değiştirilebilir. Elektrik kesintisi veya makineyi kapatma durumunda Ram bellekteki tüm bilgiler silinir ve artık bu özetten sonra Arduino ile ilgili olana geçelim.

Arduino da kullanılan ise SRAM'dir ve bu yapı aslında ram ile özelliklere sahiptir ve bu yapı ram yapısını örnek alır "S" takısı static yani durağan anlamına gelmektedir. Durağan burada aslında güç verildikçe bilgilerini koruyan anlamında kullanılıyor. Bilgisayarları-

mızda ise Ram çeşidi olarak DRAM kullanıyoruz ve şimdi sizlere SRAM ile DRAM arasındaki farkları kısaca yazacağım.

SRAM'in avantajları

1. DRAM belleklere göre daha hızlı olmaları.
2. DRAM belleklere göre daha güvenilir olmaları.

SRAM'in dezavantajları

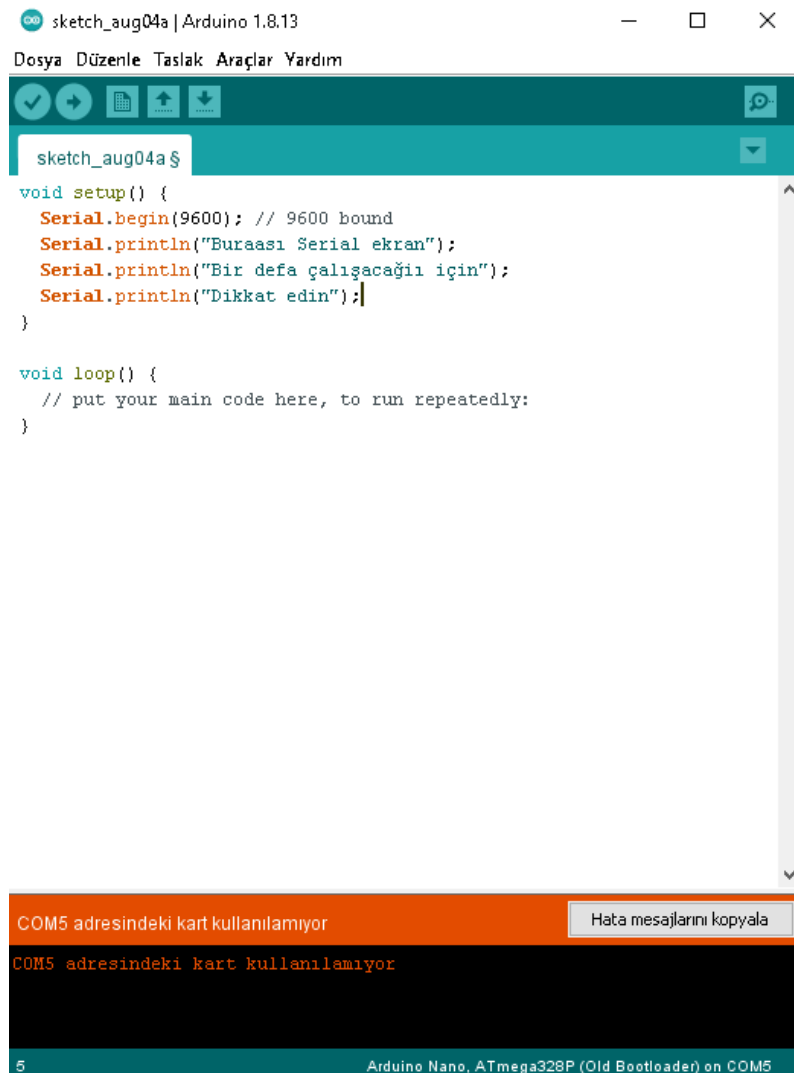
1. Yüksek maliyetlidir.
2. Pahalıdır.

Rom Nedir?

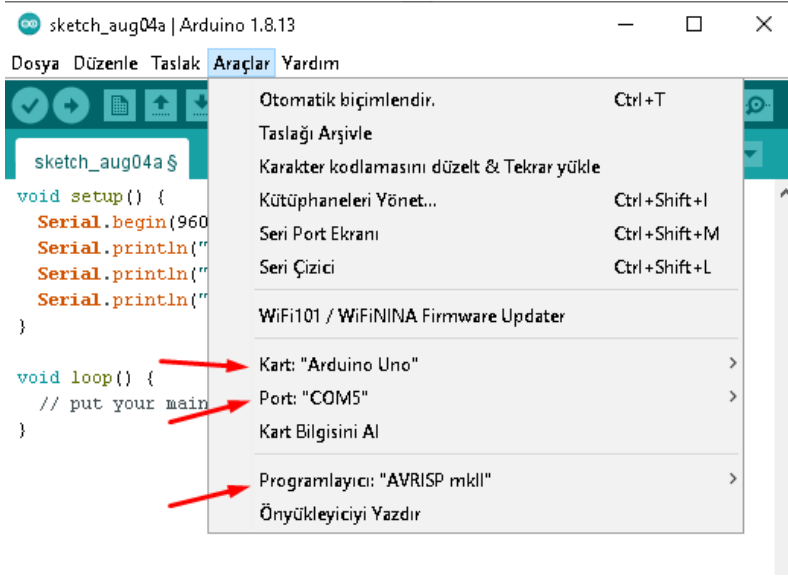
Rom sadece okuma belleğidir. Asıl adı Ready Only Memory olan bu bellek sadece okunabilir bir bellektir ve bir defa yazarsınız (bunu genel olarak firma yazar) ve sadece okunur yapıdadır. Bilgisayarı-mızda bulunan bios yapısı önceden öyle olsada artık hem okunabilir hem yazılabilir hale getirildi ama aklınızda rom denilince bios gelebilir. Rom aslında bir defa yazılır ve sürekli kullanılır olduğunu düşünün ve bunun gereği olduğu yerlerde kullanılır. Rom üzerinde pek durmayacağım çünkü pek işinize yarayacak başa bir bilgisi yok.

Veri tipleri aslında bizim için en fazla bilmemiz gereken kısım olduğunu söyleyebilirim çünkü bir arduino üzerinde bulunan miktar: SRAM: 2 KB (ATmega328) → İlk giriş sayfalarında bunu görmüştünüz fark ettiniz belki de evet çok küçük bir ram kapasitesine sahip ona göre kullanın ve daha fazlasına gerek varsa mikrodenetleyicimizi değiştireceğiz mesela burada Esp modellerine yönelebilirsiniz.

Seri Haberleşme



Evet burası artık bilgisayar ile haberleşme ve daha fazlası.devam →

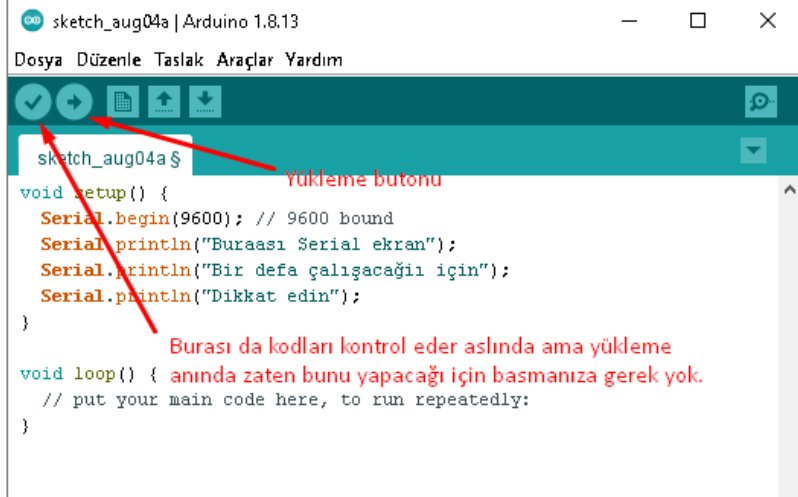


Evet artık bu kodu arduinomuza atmamıza geldik ve kısacası yine yazalım.

Araçlar → Kart : Burada hangi Arduino ile kodlama yapıyoruz seçtiğimiz yerdir eğer siz farklı bir kart ile yapıyorsanız o modeli seçiniz **fakat benim seçtiğim modelde mikrodenetleyici sürekli Atmega 328p kullanılıyor siz de bu değişiyorsa dikkat ediniz!**

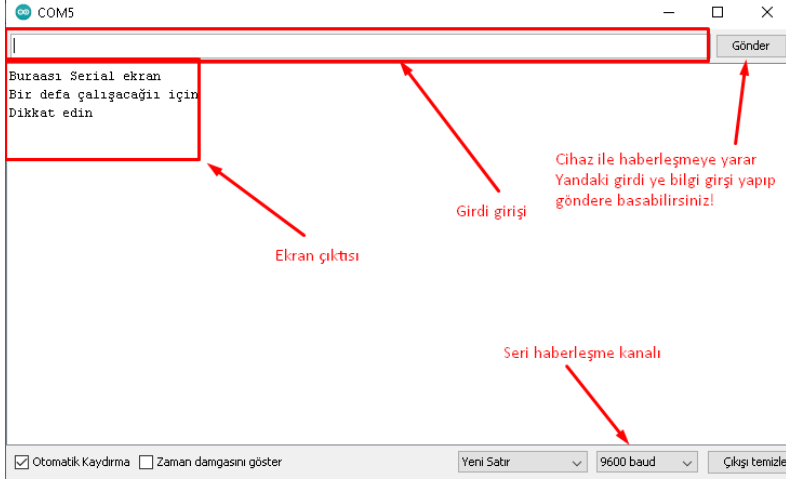
Araçlar → Port : Bilgisayarınıza bağladığınız arduino hangi portta yer alıyorsa onu seçiyoruz ve bu port bizim için veri alışverişini sağlıyor.

Araçlar → Programlayıcı: Burası biraz karışık ama sizinde arduinonuzla gelen kabloyu kullanıyorsanız seçmeniz gereken programlayıcı budur diyebilirim.



Evet şimdi yükleme yaptık diyelim fakat hani nerede bu yazdığımız ekran? Onun için de sağ köşede gösterilen büyüteç gibi olan simgeye tıklayın.





O kısma tıklayınca bu ekran gelecek. Seri haberleşme kısmında **9600 baud yazıyor ama ben açıklama da bound yazmışım yeni farkettim kusura bakmayın** küçük bir yanlışlık olmuş.

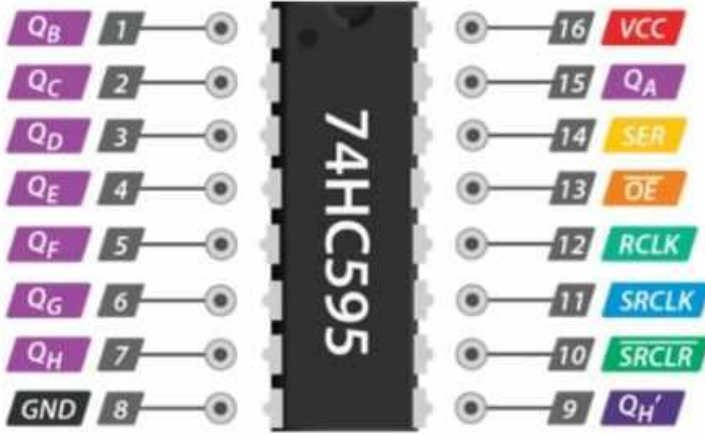
Ekran da görülen her türlü yazılanları okuyunuz!

Şimdi gelemim haberleşme de bir tık ileriye gideilim.

Gelişmiş Seri haberleşme

Her mikrodnetleyicide sınırlı sayıda G/Ç pini (GPIO) bulunduğundan bit kaydırma kayıtları, mikrodnetleyicideki pinleri kaydetmek amacıyla 74HC595 kullanılır. Mesela bir 16 led yakmak için 16 giriş çıkış pini gerekte ama başka herhangi bir pin kalmayacak ve herhangi bir modül takılamayacak kısacası yeterli gelemyecektir. Biz ise bu entegre ile 6 pin ile 16 pini yakabiliyoruz aslında iki entegre kullandığımız için böyle yoksa yine bir entegre ile 16 bit kodlayıcı ile yapılabilir ama küçükten başlayalım istedim. Bu entegre aslında motor sürücüler de kullanılıyor temeli hazırlıyorum.

74HC595 Entegresi

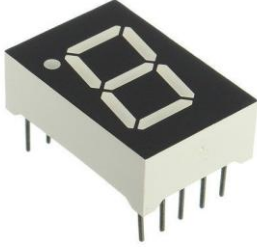


74HC595 Pinout

Bura da aslında kısaca sizlere yapısı veriliyor VCC ve Gnd kısmını zaten biliyorsunuz gerisini sonra öğreneceğiz biraz sabredin. 😊

Bazı sorunların çözümleri mevcut misal bu entegre bu işe yarıyor ama biraz kafa karıştırıcı zor diyebilirim. Bu entegreyi bir sonraki kitabımda işleyeceğim çünkü bu kitap için biraz ağır kaçmakta olduğunu biraz karaladıktan sonra farkettim.

SSD kullanımı (Seven Segment Display)

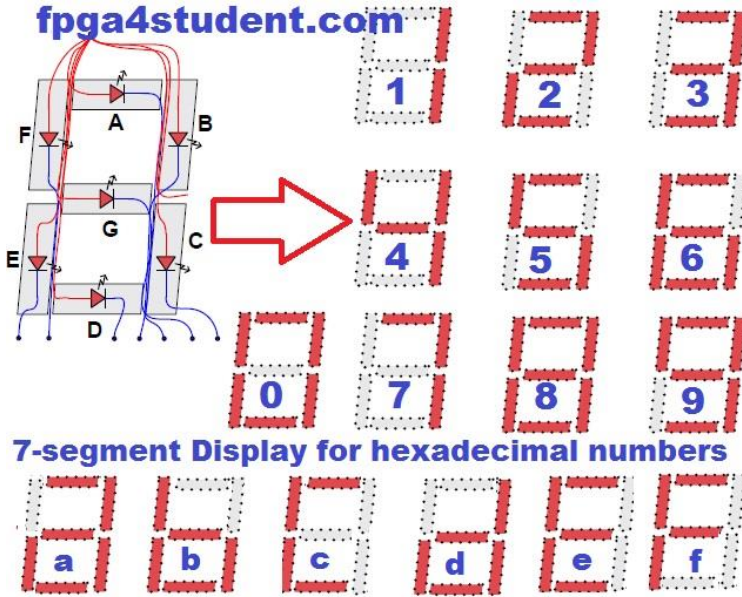


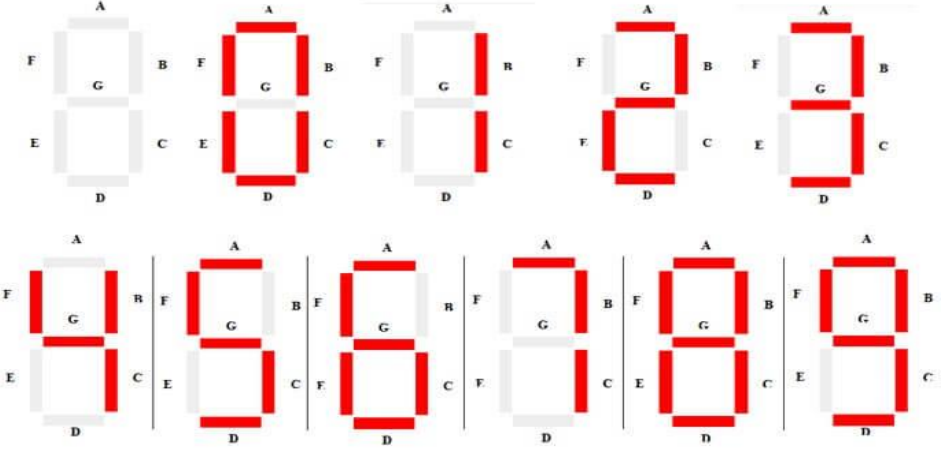
Yandaki resimde görülen entegrenin adı Seven Segment Display 'dır.

Bu yapı sizin istekleriniz doğrultusunda rakamlar göstermeye hatta dilerseniz sanırım yanlış bilmiyorsam ismini digit yazı stili ile yazıları da basabilirisiniz.

Bu yapıyı daha iyi anlamak için aslında en başlar da bir key pad sunmuştum hatırlarsanız o yapıya benzer bir şekilde çalışır.

Aşağıda ki görseli inceleyiniz ve diğer sayfaya geçiş yapınız. →



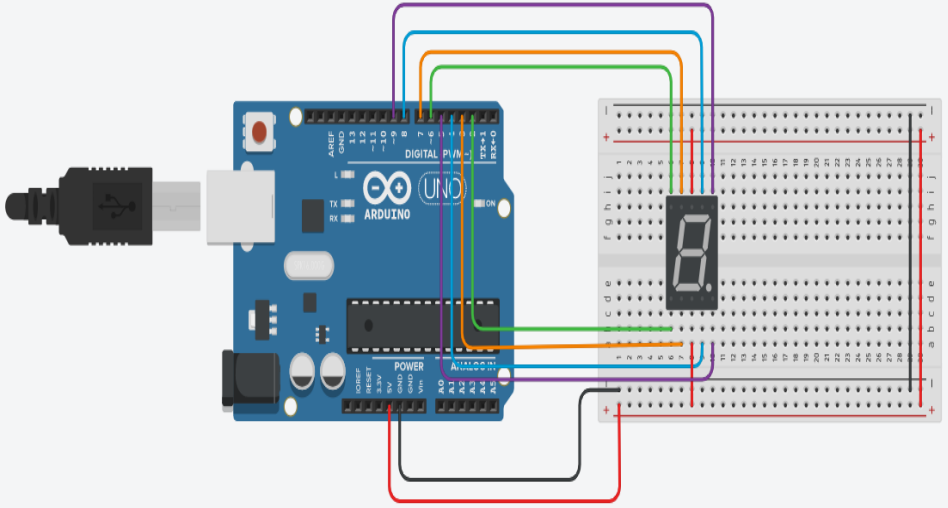


Aslında mantık olarak yapı olarak çalışma mantığı budur toplam da 7 adet ledimiz mevcut ve bu ledler ile istediğiniz yazabiliyorsunuz. İki adet neredeyse aynı fotoğrafı koyduğumun farkındayım fakat yapısını yeniden vurgulamak istedim sizlere. Şimdi de geelim kod kısmına ki en uzun gibi görünen kısım bu aslında ama korkulacak bir şey yok.

Bu kısmın devamı için sizlere bura da bir kod yazısı paylaşamaya-
cağım ama sizlere bir link paylaşayım.

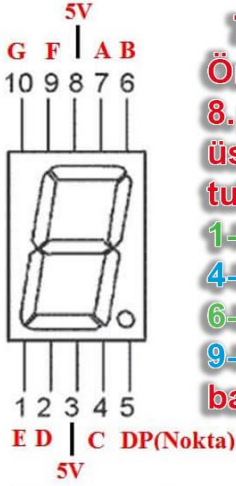
Link: <http://bc.vc/IsSCLWS>





Yukarıdaki bağlantılar çok basit bir şekilde yapılıyor. Devre üzerinde gösterimleri güzelce gösterdiğimi düşünüyorum ama daha da iyi görmek istiyorsanız Youtube kanalım (Ali Bayram) üzerinden inceleyebilirsiniz veya Udemy üzerinden de takip edebilirsiniz. Videolar aslında bu devrelerin daha da iyi görülmesi için birebirdir. Videoları da takip etmenizi biraz tekrara düşmüş gibi de olsam evet çok önemli bir olay. Yine de sizlere bu bağlantı şemasının mantığını tekrar olarak görsel olarak anlaşılır bir şekilde göstereceğim.

Diğer sayfaya ilerleyin →



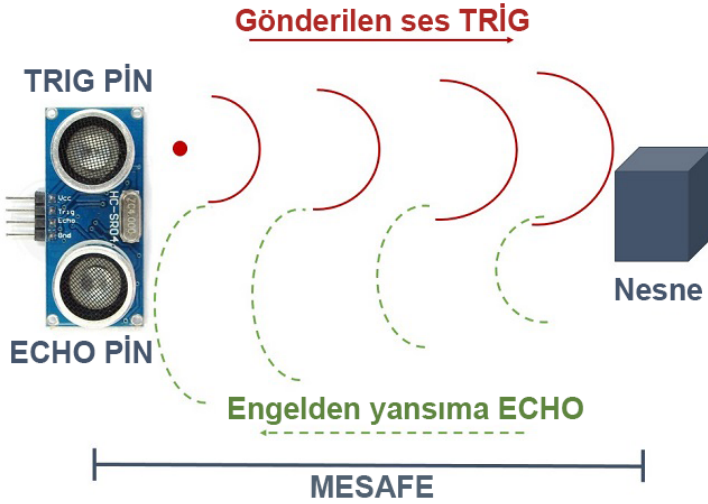
7Segment Display Bağlantısı
Öncelikle 5V bağlantılarımızı 3. ve 8. pinlerimize yapalım bu pinleri 2 üssü 3 =8 gibi kafanızda tutabilirsiniz.

1-> 2nolu pine, 2-> 3nolu pine,
 4-> 4nolu pine, 5-> 5nolu pine,
 6-> 6nolu pine, 7-> 7nolu pine,
 9-> 8nolu pine, 10-> 9nolu pine
 bağlantılarını gerçekleştiriniz.

Aslında yapı olarak bağlantı olarakta mantığı basit olan bu modül bir ledler bütündür diyebiliriz ve isterseniz kendiniz dahi bu bağlantıyı gerçekleştirebilirsiniz. Bu bağlantıları yaparken çalışma mantığını aklınız da canlandırırsanız sizler için daha pratik bir hal alır. Evet artık biraz da sensörler ile uğraşalım. **Sensörler hakkında kısaca bir bilgi** vermek istedim sadece bu sayfanın sonunu onun için ayarladım. Sensörler aslında sizler için bir el kol olacaktır yani bir koşulu sağlarken onlardan kontrol alabileceksiniz. Elinizi hareket ettirince algılayan ve sizlere kağıt havlu veren cihazlar mesela onu nasıl yapıyor? Sensörler aslında orada bir el olduğunu bildiriyor ve sizlere kağıt havlunuzu veriyor ki bunu bile yapmak çok zor değil belki sizlerin aklında öyle bir fikir vardır ki kağıt havlu makine-lerinin bir numaralı rakibi haline bile gelebilirsiniz. Her şey bir arduino ile hayata geçebilir ama daha da çok okuyun sizlere daha çok bilgi vereyim ve ona göre bilgilerinizi kullanarak daha gelişmişini dahi yapabilirsiniz.

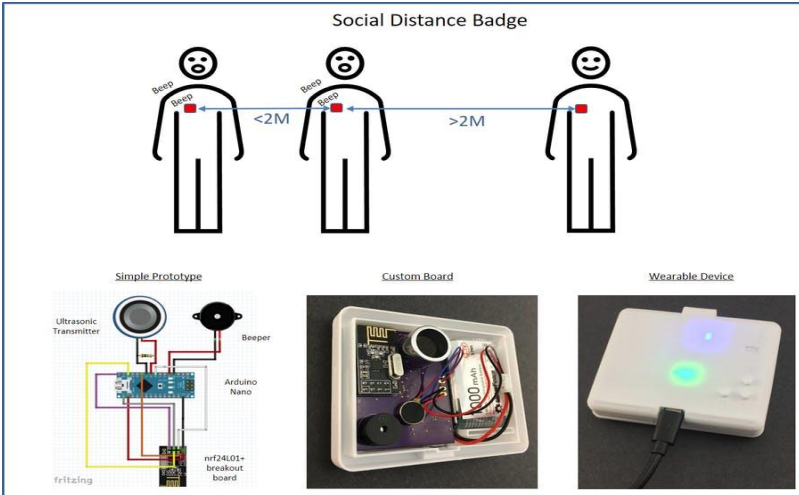
HC-SR04 kullanımı (Mesafe Sensörü)

Namı diyar mesafe sensörü ile artık tanışıyoruz. Bu sensör oldukça fazla kullanılıyor ki sizlerin de illaha bu cihaz ile ilgili projeniz mevcuttur. 😊 Mesafe sensörü ne işe yarar? En azından bu modeli için bahsederseniz sizlere 200cm ye kadar mesafe de herhangi bir cisim olup olmadığını öğrenebilirsiniz. Bu öğrenme işini güzel bir fikirle taçlandırırsanız daha da güzel olur. Şimdi gelelim bu sensör kısmen nasıl çalışıyor?

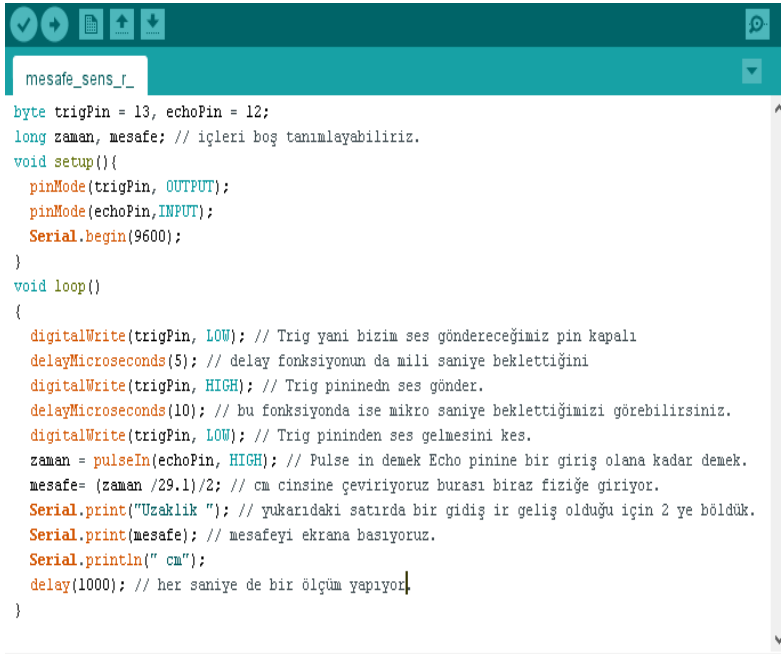


Çalışma mantığı aslında çok basit yıllar önce öğrendiğimiz bilgiler vardı ya okulda işte onlar ne zaman işinize yarayacak şimdi görmüşsünüzdür umarım. Bir dağa doğru bağırınca kaç saniye sonra sesinizi duyarsanız aslında dağ o kadar uzaklıkta gibi sorular çözerdik onların temelini alan bir sensör. 😊

Evet tam devam etmeden örnekler gösterip fırsatları kovalamanızı tavsiye ediyorum. Bu işleri hobi olarak ya da para için yapıyor olabilirsiniz ve eğer para için yapıyorsanız mesela korona sürecinde bir sürü böyle proje çıktı siz de kendinize bir proje bulabilirsiniz deyip sizi iki görsel ile bırakıyorum.



Devam ediyoruz artık kodlara ... →



```
mesafe_sens_r

byte trigPin = 13, echoPin = 12;
long zaman, mesafe; // içleri boş tanımlayabiliriz.
void setup(){
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}
void loop()
{
  digitalWrite(trigPin, LOW); // Trig yani bizim ses göndereceğimiz pin kapalı
  delayMicroseconds(5); // delay fonksiyonum da mili saniye beklettiğini
  digitalWrite(trigPin, HIGH); // Trig pininedn ses gönder.
  delayMicroseconds(10); // bu fonksiyonda ise mikro saniye beklettiğimizi görebilirsiniz.
  digitalWrite(trigPin, LOW); // Trig pininden ses gelmesini kes.
  zaman = pulseIn(echoPin, HIGH); // Pulse in demek Echo pinine bir giriş olana kadar demek.
  mesafe= (zaman /29.1)/2; // cm cinsine çeviriyoruz burası biraz fiziğe giriyor.
  Serial.print("Uzaklık "); // yukarıdaki satırda bir gidiş ir geliş olduğu için 2 ye böldük.
  Serial.print(mesafe); // mesafeyi ekrana basıyoruz.
  Serial.println(" cm");
  delay(1000); // her saniye de bir ölçüm yapıyor.
}
```

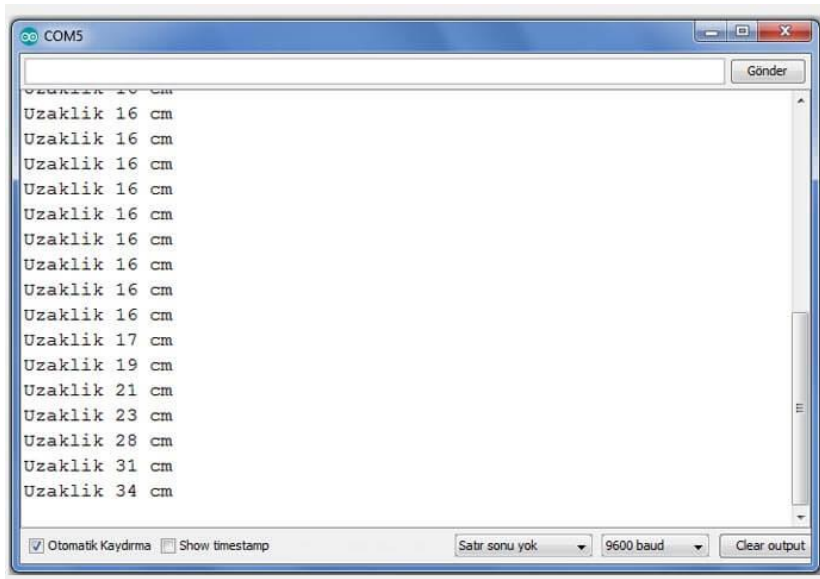
Yukarıda kodu detaylıca incelersek aslında her şeyin aslında bir led yakmak kadar neredeyse kolay olduğunu görebilirsiniz. Öncelikle burada ilk görüşümüz kısımları inceleyelim.

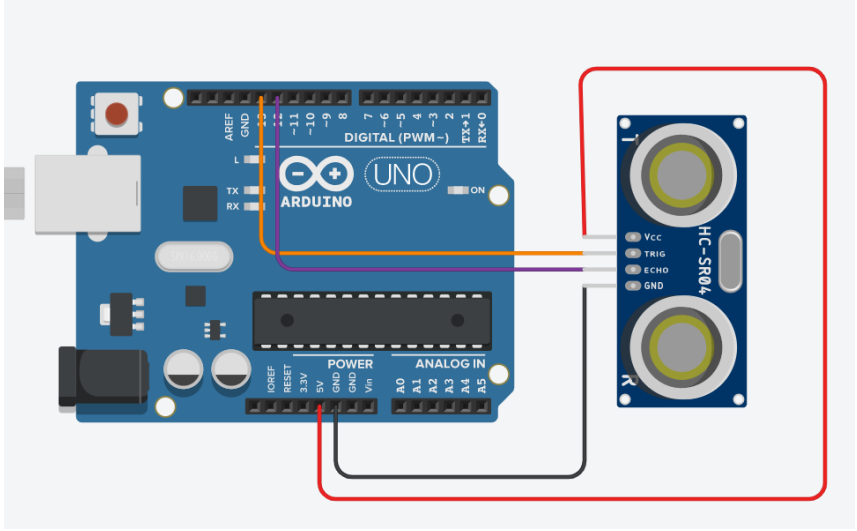
long zaman,mesafe; → İlk defa long kullanıyoruz. Burada long kullanmamızın sebebi değerın integer den büyük olabileceği ihtimalındendir. Yani bura da long kullanma nedenimiz vudur aslında.

delayMicrosecond; → İlk defa mikrosecond denilen yani türkçesi mikto saniye kullanımı gördük burada ki amaç Trig pininin ses üretimi için belli bir süre gecikme ile çalışmasıdır.

`pulseIn();` → Bu komut ise Belirlenen bir pin üzerindeki HIGH veya LOW sürelerini okur. Yani pin üzerindeki değer HIGH ise bu değer LOW olana kadar bekler ve darbenin **mikrosaniye** cinsinden uzunluğunu verir. Ayrıca Timeout süresi vardır ve bu süre **10 saniye** olarak tanımlanmıştır. Eğer bu süre içerisinde belirtilen pin üzerinde bir değişiklik yoksa sonucu 0 olarak geri döndürür ve pine bir de bir değişiklik olmadığını size bildirir. Bu sensörün 0 döndürmesi demek ya yansıyacak bir cisminin bulunmadığını ya da sensörün çalışmadığını gösterir.

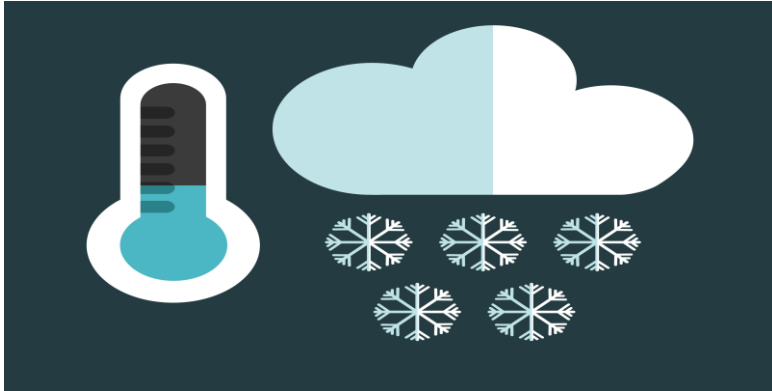
Aşağıda ki fotoğrafta verilen ise bu kodun çalıştırıp serial ekranın açıldığında gelecek görüntüdür.





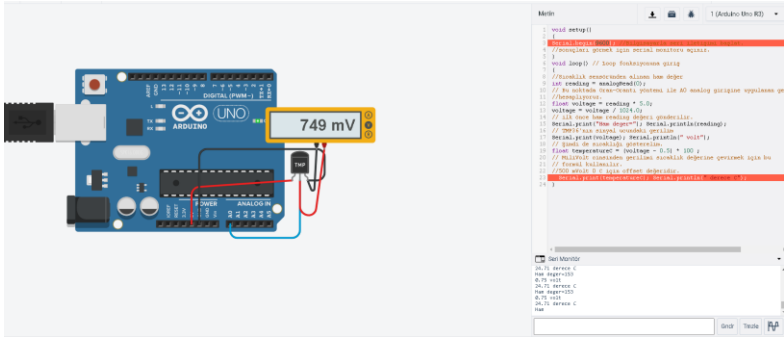
Artık pek eskisi kadar basit düzenekler için breadboard kullanmayacağım. Yapısal olarak bağlantılar bu şekildedir. HC-SR04 modülünü nerede kullanırsınız bilmiyorum ama çalışma mantığı bu kadardı. 😊

Şimdi gelelim bir tane daha sensörümüze ben size görsel göstereyim bakalım ne işe yaradığını söyleyebilcek misiniz?



DHT11,D22,LM35 ve TMP36 kullanımı

Evet geldik sıcaklık sensörlerimizi incelemeye ki burası birazcık uzun olacak benden belirtmesi ki tamamen anlayın diye zorlayacağım. Zorlamaktan kasıt biraz fazla bilgi içerecek belki bu sıkacaktır fakat gayet güzel bir şekilde anlayacaksınız.






Şimdilik üstte verdiğim gibi bir kısım ile TMP36 ile başlıyorum. Tmp ya da diğerlerinin aslında çalışma mantıkları şudur: bu entegrelere verdiğiniz gerilimlere göre size bir miliVolt değeri yani mV değeri döndürmesidir. Şöyle devam edelim o zaman bir tmp36 ya verilen gerilim 2.7-5V arasında oldukça size doğru hesaplamalar ile (sıcaklık hesaplamak için bir formül kullanacağız) sıcaklık değerimize ulaşabiliriz ve bu çok basit bir işlemdir. Bizim ise bura da yaptığımız olay ise 5v çıkış ile karşımıza çıkan sensörden ölçtüğümüz değer 749mV ve bunu basit bir analog giriş ile şu şekilde sıcaklık değerine dönüştürüyoruz.

(749/1042)x5000 ve bu volt cinsinden gelen değeri voltage değişkenine atayıp 0,5v çıkartıp 100 ile çarpıyoruz buradaki amaç 0 derecesi 0,5V a eşit olduğu içindir. 100 ile çarpım ise voltajı sıcaklığa çevirmek için virgülden sonra gelen iki kısmı almak için.

Buradaki örnek aslında kod olarak şundan oluşuyor.

Metin



1 (Arduino Uno R3)

```
1 void setup()
2 {
3   Serial.begin(9600); //Bilgisayarla seri iletişimi başlat.
4   //sonuçları görmek için serial monitörü açınız.
5 }
6 void loop() // Loop fonksiyonuna giriş
7 {
8   //Sıcaklık sensöründen alınan ham değer
9   int reading = analogRead(0);
10  // Bu noktada Oran-Orantı yöntemi ile AO analog girişine uygulanan ger
11  //hesaplıyoruz.
12  float voltage = reading * 5.0;
13  voltage = voltage / 1024.0;
14  // ilk önce ham reading değeri gönderilir.
15  Serial.print("Ham deger="); Serial.println(reading);
16  // TMP36'nın sinyal ucundaki gerilim
17  Serial.print(voltage); Serial.println(" volt");
18  // Şimdi de sıcaklığı gösterelim.
19  float temperatureC = (voltage - 0.5) * 100 ;
20  // MiliVolt cinsinden gerilimi sıcaklık değerine çevirmek için bu
21  // formül kullanılır.
22  //500 mVolt 0 C için offset değeridir.
23  Serial.print(temperatureC); Serial.println(" derece C");
24 }
```

Kod bundan ibaret ve açıklaması birebir verilmiştir. Tekrar gibi olacak ama fark etmişsinizdir bura da sıcaklık değeri için biz analog değerden değer okuma yapıp bu değeri işleme soktuk burada ki amaç şudur aslında herhangi bir kütüphane kullanmadan hesaplama yapmaktır. Eğer bu kısmı kütüphane yapsın dersek bu sefer kütüphane kısmı girecektir. Kütüphane kullanımı arduinonuzu şişirir. Ben olabildiğince kütüphane kullanmanızı önermem ki benim daha önceeden de dediğim bir olay vardı artık bura da az bir şekilde bunu açıklayabilirim diye düşünüyorum. Arduino bir kütüphanedir demiştim ki halen dediğimin arkasındayım. Şimdi sadece kısa olacak şekilde bu öne sürmüş olduğum söylemi sizlere göstereceğim. Hemen bir sonraki sayfa da bunu net bir şekilde görebilirsiniz. →

Sadece boş bir kod atalım ve ne kadar bir alan kullanıyor bakalım.

```
bo_
void setup() {}
void loop() {}

Yükleme tamamlandı.
Çalışmanız programın 444 bayt (1 %) saklama alanını kullandı. Maksimum 32256 bayt.
Global değişkenler belleğin 9 byte kadarını (0%) kullanıyor. Yerel değişkenler için 2039 byte
```

Evet ve görüyoruz ki 444bayt kullanım yapmış yani boş bir dosya sadece bu fonsiyonlar 44bayt yer kaplıyor. Burada küçük bir gösterim sadece olacak ama ayrıntıya öyle çok inmeyeceğim diğer kitabın ilk konularından birsi olacak bu artık diyor ve devam ediyorum.

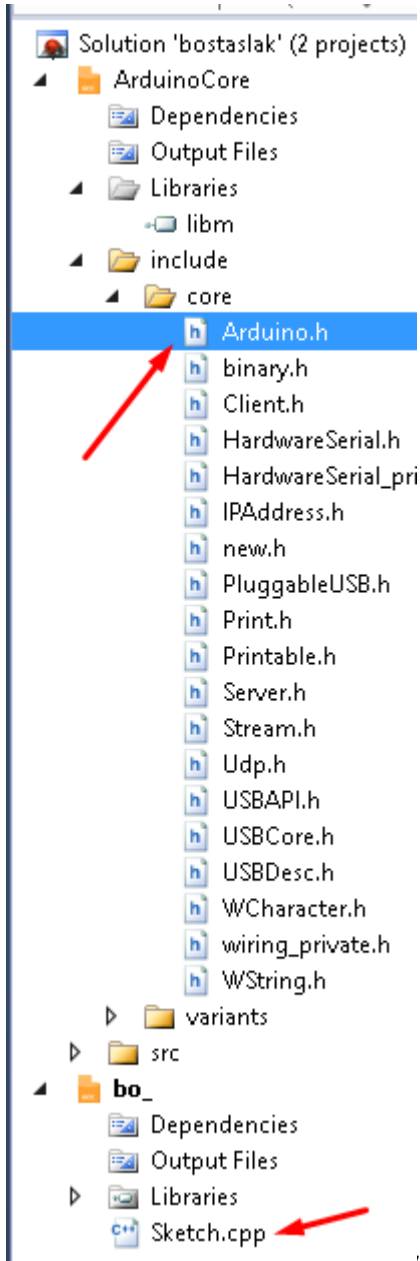
```
/*Beginning of Auto generated code by Atmel studio */
#include <Arduino.h>

/*End of auto generated code by Atmel studio */

//Beginning of Auto generated function prototypes by Atmel Studio
//End of Auto generated function prototypes by Atmel Studio

void setup() {}
void loop() {}
```

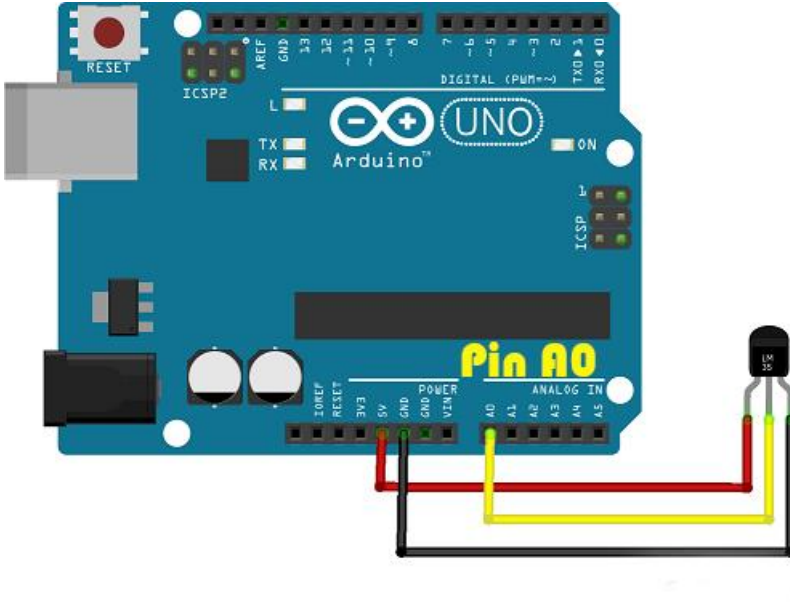
Evet bu kodu atmel studio ile açarsak üst tarafta bir kütüphane ile karşılaşacağız. Atmel studio ile de tabi ki kodlama yapacağız ama bu kitapta değil. Artık atmel studio gömülü sistemlere giriyorki bunu ayrı bir kitapta dahi yazabilirim çünkü çok gelişmiş bir konu. Profesyonel projeler çıkarmak istiyorsak kesinlikle bu ideyi kullanacağız fakat dediğim gibi o şimdilik kolay bir iş değil. Her şeyi anlayalım öyle girişelim bu işlere yoksa gömülü sistemler bir derya deniz. 😊 Kısaca tezimi gösterdim ve Arduino kütüphanesinin nelerden oluştuğunu sadece görüntü olarak göstereceğim. →



Sol tarafta görülen kısım o kısaca yazdığımız kodun arka planında ki işleyiş. Arduino.h kütüphanesinde ise bir sürü fonksiyonlar ve tanımlamalar mevcut. “Sketch.cpp” diye bir dosya mevcut ya işte o da bizim kodumuzun yazılı olduğu yer ve biz onu aslında .ino olarak görüyoruz ki aslında c++ dili olduğunu gördük. Aslında her şey bu kadar basitti ve neden arduino ide ile kodlama yapmak arduinmomuzu şişiridğini anlamışızdır. Benim diyeceklerim sadece bu kadar. Şu anlık arduino ide ile devam edeceğimiz projelerimiz ve çalışmalarımızı ileriki kitaplarda bu son açtığım Atmel Studio ile yeniden programlayacağız ve orada karşılaştırmalar ile devam edeceğiz.

Biraz ayrıntılara girdik ama yine de sizlere bu işlerin başlangıç olduğunu dahasının da olduğunu bildiriyorum ve ilerleyen kitap serilerinde bu daha da incelenecek olup daha iyi anlaşılacak.

Şimdi gelelim LM35'e bu kısımda da tmp36 ile aynı çalışmōa mantıkları hep aynı. Sıcaklık değeri direnç gibi düşünebilirsiniz ve bu direnç sayesinde sizlere mili volt değeri sunuyor.



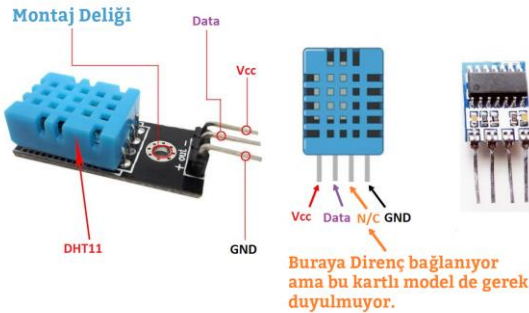
Dediğim gibi aslında herhangi bir numarası yok neredeyse her şey aynı ilerliyor birkaç değışiklik mevcut sadece. Kod üzerinden bunu inceleyebiliriz.

```

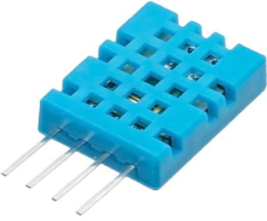
float gerilimDeger = 0, sensorDeger = 0 ,sicaklikDeger = 0;
void setup() {
  // Seri port ekranını 9600 baud değerinde başlatıyoruz.
  Serial.begin(9600);
}
void loop() {
  sensorDeger = analogRead(A0); // Bu fonksiyon ile A0 pininden değer alıyoruz.
  Serial.print("Sensör Değeri: "); // Aynı şekilde değerler devam ediyor.
  Serial.println(sensorDeger);
  // A0 pininden okunan değeri mV cinsinden gerilim değerine dönüştürüyoruz.
  gerilimDeger = (sensorDeger/1023)*5000;
  Serial.print("Gerilim Değeri: ");
  Serial.println(gerilimDeger);
  sicaklikDeger = gerilimDeger / 10.0; // Tmp36 daki gibi bir dönüşüm yapılmalı
  //Buradaki farklılık ise her 10mV 1 dereceyi ifade etmektedir.
  Serial.print("Sıcaklık Değeri: ");
  Serial.println(sicaklikDeger);
  // Her döngüde bir saniye bekliyoruz.
  delay(1000);
}

```

Tmp ile aralarında sadece son kısım ile çalışma voltajları farklılık göstermektedir. LM35 4-12V arası bir çalışma aralığına sahiptir bu değer TMP36 da ise 2.7-9V idi. Ölçüm değerlerinde mecburen sapmalar olacaktır. Her bir sensörün kullanım alanı buna göre değişir. Artık hazır kütüphaneler kullanabileceğimiz modellere Geçiyoruz. Karşınızda DHT11:



Dht modelinin çokça kullanılması diğer modellere göre bahsettiğimiz gibi bir uğraşınız yok karta jumper bağlayıp işinizi görebiliyorsunuz. Tabiki bunun bir bedeli var ve sizlere kimsenin sunmadığı kısımlar vereceğim demiştim ve sözümü tutmaya devam edeceğim bu kısımda da demek istediğim şu aslında siz bir ticari ürün yapacaksanız bu ürünü kartlı olacak şekilde almanız daha maliyetli olacak. Bir kusura neredeyse 2TL olan bu aradaki fark binlerce üründe sizler için önemsenecek duruma gelebilir. Sadece bir sensöre direnç lehimleyerek 2TL ye yakın bir kar elde edersiniz ve bunu diğer sensörler ile de aynı şekilde gerçekleştirebilerseniz çok fazlaca bir kar elde edebilirsiniz.



DHT11 Sıcaklık ve Nem Sensörü Modülü
Stoktan Aynı Gün Hızlı Kargo - 100% Müşteri Memnuniyeti
★★★★★ Ürün yorumu yok

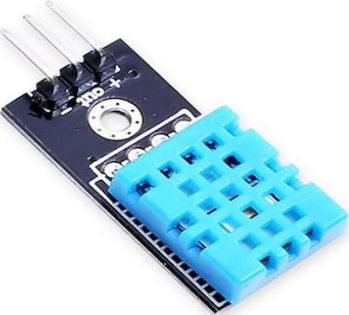
Adet

Teslimat Bilgisi : 8,90 TL
En geç 10 Ağustos Pazartesi günü İstanbul'dan kargoya verilir. ?

6,50 TL KDV DAHİL

Hemen AL **Sepete Ekle**

ORJİNAL ÜRÜNE %100 DESTEK ÖDEME KORUMA SİSTEMİ İNTERNETTE GÜVENLİ ALIŞVERİŞ



DHT11 Nem - Isı - Sıcaklık Sensörü
Sıfır - Stoktan - Faturalı - Garantili - Aynı Gün Kargo
★★★★★ Yorumları oku (2)

Adet

Teslimat Bilgisi : 8,90 TL
En geç 10 Ağustos Pazartesi günü İstanbul'dan kargoya verilir.

7,70 TL KDV DAHİL

Hemen AL **Sepete Ekle**

Hemen AL

ORJİNAL ÜRÜNE %100 DESTEK ÖDEME KORUMA SİSTEMİ İNTERNETTE GÜVENLİ ALIŞVERİŞ

Az önceki kısmı belki biraz garipsemiş olabilirsiniz fakat gerek hobi olarak uğraşıyor olun gerekse bundan ticari ürün yapacak durumda olun en ufak bir kar belki sizin için ekstra bir para oluşturabilecek.

1940 yılında Amerikan bir hava yolu şirketi ikram ettiği kahvaltı tabaklarından 1 adet zeytin eksilterek yıllık 40bin dolar kar elde etmeye başlamıştır.

Bu örneği vererek sizlere ne demek istediğimi daha iyi anlatmış olabilirim. Kodlama kısmında ise daha önce de dediğim gibi artık bizi ilk önce artık bir kütüphane karşılıyor.

```
1 anetili_duyuruuy
#include <dht11.h> // Dht11 kütüphanesini ekliyoruz.
#define DHT11PIN 2 // DHT11PIN olarak Dijital 2'yi belirliyoruz.
dht11 DHT11; // Bu tanımlama aslında artık dht11 yerine DHT11 diyerek ulaşmamızı sağlar.
void setup()
{
    Serial.begin(9600); // Seri iletişimi başlatıyoruz.
    //pinMode(DHT11PIN,INPUT); diyebilirdik.
}
void loop()
{
    int readData = DHT11.read(DHT11PIN);
    Serial.println("Nem (%): ");
    Serial.println((float)DHT11.humidity, 2); // Yüzdelik olarak nem verir sizlere.
    Serial.print("Sıcaklık (Celcius): ");
    Serial.println((float)DHT11.temperature, 2); // Dereceyi selsiyus cinsinden verir.
    Serial.print("Sıcaklık (Fahrenheit): ");
    Serial.println(DHT11.fahrenheit(), 2); // Dereceyi fahrenheit cinsinden verir.
    Serial.print("Sıcaklık (Kelvin): ");
    Serial.println(DHT11.kelvin(), 2); // Dereceyi kelvin cinsinden verir.
    delay(1000); // Bir saniye bekle.
}
```

Kod çok açıklayıcı olsa da yeniden açıklamakta fayda var. Dht11 modeli için kütüphane olarak “dht11.h” kütüphanesi, dht22 için ise “dht22.h” kütüphanesi dahil etmelisiniz. Kütüphanelerin aslında ne iş yaptığını biliyoruz çok farklı bir şey değil arkaplanlarda olan ama bizim şimdilik ön planla ilgilendiğimizi düşünürsek kodumuzu tek tek inceleyelim.

dht11 DHT11; → Bu tanımlama aslında her yer de “dht11” yazarak çağırmak yerine “DHT11” olarak çağırmaya yarar. Bu kullanım programlama dilleri ile uğraşmış olanlar var ise mecburen karşısına çıkmıştır ve aşınadır. Ama kısacası ben şu tanımlamayı bile yapsam “dht11 sensor ;” gibi bir tanımlama yaparsam kod artık her yerde “sensor” yazdığımda dht11 yazmışım gibi kabul edecek.

DHT11.read(DHT11PIN); → Bu tanımlama ise aslında bir fonksiyon çağırma kodudur. Bu kodun görevi bir önceki sensörlerde analog pin üzerinden aldığımız gibi pini oku demek. Ana amaç aslında kodda olan pinin ölçüm yapmak için değeri okumasını çalıştırması. Biraz karışık denebilir fakat arkaplanda çalışan kodu inceleyip anlamamız gerekmektedir diyip devam ediyorum.

DHT11.humidity → Bu tanım diğer “DHT11.” ile başlayanlar ile aynı. Bu kodu açıklarsak nem fonksiyonunu çalıştırıyor diyebiliriz. Diğer kodlar ise neyi çağırmak istiyorsa o fonksiyonu çağırıyor.

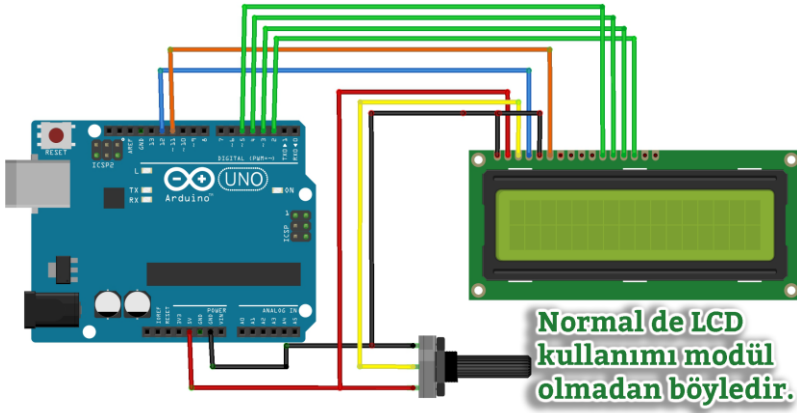
(float) DHT11.humidity,2 → Tekrar kodu açıklamayacağım ama bu float ve 2 ne dersiniz diye sadece bu kısmı açıklayacağım. Float yapısını öğrenmiştik float sayı olarak virgülden sonra bir sürü sayı alabileceğini biliyoruz ama bize sadece virgülden sonraki iki basamak kadar kısım lazımsa böyle yazılıyor. Sadece örnek olsun diye yazıyorum bunu :

36,3741 gibi bir derece ölçüm yapmışsa bunu yukarıdaki gibi yazarsak artık “36,37” olarak alacak bu aslında bir hatadır ama bu kadar küçük bir hata sizlerin umursayacağınız kadar büyük değildir.

Artık kodda 11 yazılan yerlere 22 yazarsanız dht22 içinde geçerli olacaktır.

16×2 LCD Ekran Kullanımı

Arduinoyu kullanırken bazen ekrandan değerler görmek isteyebileceğiz. Bunu göz ile takip edip sapmalar olmadığını takip etmek için özellikle kullanacağız. Çoğu kişinin evinin buzdolabında dereceyi ekrandan görebilirsiniz. Ekranda değer gözükmesinin amacı içerinin halen daha soğuk olup olmadığını tespit etmek ve herhangi bir sorun ile karşılaşılırsa kullanıcının bilmesi.

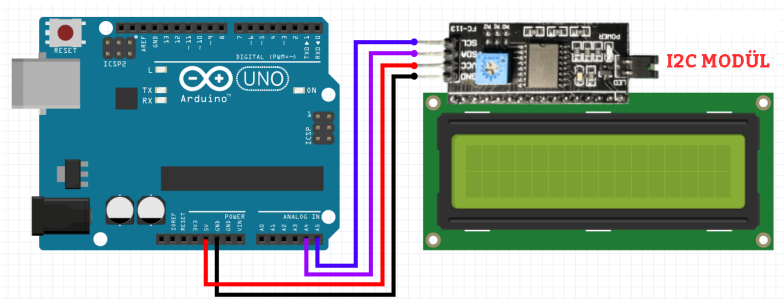


Normal lcd bağlantısı bu şekilde olmalıdır. Aslında bu tip bağlantı birazcık göze kötü görünmektedir ama arduino SPI olarak çalışmaktadır ve lcd I2C dediğimiz protokolü kullandığı için mecburuz. İlk etapta bu modeli potansiyometreli şekilde yapacağız ve dahasında I2C dediğimiz protokole göre bağlantılarını başka bir I2C kartı ile bağlantı kurup kolay bir şekilde yapacağız. Potansiyometre olmadan çalıştığını söyleyenler var ve bunları da ilk olarak potansiyometreli ve I2C kart ile yaptıktan sonra göreceğiz.

```
#include <LiquidCrystal.h> // LCD ekran için kütüphane.  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // 12,11,5,4,3,2 pinlerine  
void setup() {  
  lcd.begin(16, 2); // Satır, Sütun  
  lcd.print("hello, world!"); // Ekrana hello world yazdık.  
}  
void loop() {  
}
```

Lcd bağlantılarını 2,3,4,5,11,12 nolu pinlere bağladık. Bu bağlantıyı sağlarken ekstradan potansiyometremizin üçüncü bacağına da lcdnin ekranını göreceğimiz taraftan 3. Pinine sarı kablo ile bağlantı yaptık. Potansiyometre bize burada ayarlama yapmamızı sağladı. Sizler bu bağlantıyı yaptığınız da görüntü gelmezse potansiyometre ile oynayarak doğru noktayı bulabilirsiniz. Lcd bağlantısını bu şekilde yapmamanızı I2C modülü kullanmanızı tavsiye ederim. I2C modülünü de sizlere şimdi göstereceğim.

Aşağıda verilen I2C kartı sizler için SPI protokolünden dönüşüm sağlayarak ekran ile iletişiminizi kolaylaştırıyor.



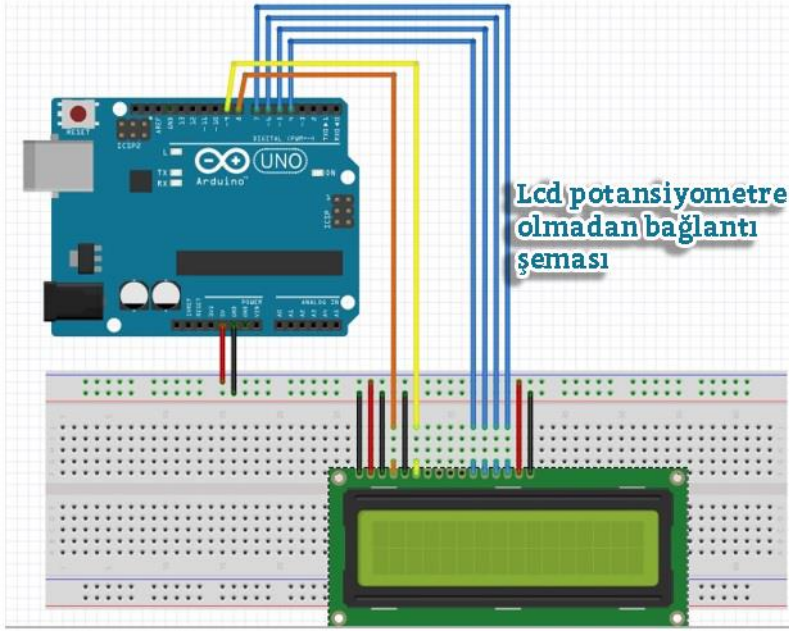
```
#include <Wire.h> //Arduino ile I2c haberleşmesinde Wire.h
//kütüphanesini kullanmaktayız
#include <LiquidCrystal_I2C.h> // I2C kütüphanesini indirin

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup()
{
  lcd.begin();// LCD yi çalıştır
  lcd.backlight();// LCD arkaplan ışığını açar.
  // Ekrana yazdırılacak metin giriyoruz.
  lcd.print("Hello World!");
}

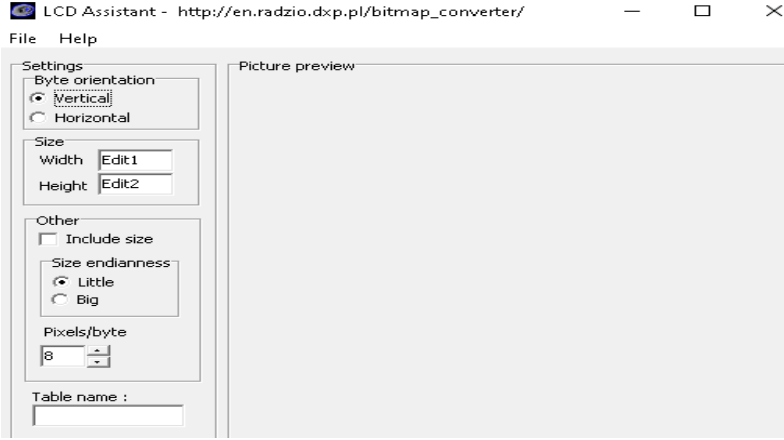
void loop()
{
  // Sürekli bir şey yazdırmak isterseniz burayı kullanın.
}
```

Burada farklılıklar hemen gözünüze çarpması gerekmektedir. Öncelikle bir “Wire.h” kütüphanesi karşımıza çıkıyor bu kütüphane Arduino ile I2C haberleşmesi için kullanılıyor ve bunu yazmamız gerekmektedir. Bir sonraki kütüphane de ekstra olarak kütüphane isminin sonuna “_I2C” ibaresi gelmiş gördüğünüz gibi. Sizler bunlara dikkat ederek kodlamanızı yapınız. Daha az pin kullanımı ile sizlere istediğiniz gibi yazmanızı sağlamakta ve daha az bağlantı ile sizlere temiz bir bağlantı sunuyor. Bir önceki yöntemden parasal ve donanımsal anlamda farklılıklar yaşatacağı bunu biliniz fakat daha temiz bir iş ortaya çıkarır. Sizlere şimdi bir de birkaç sayfa öncesinde söz ettiğim potansiyometresiz bağlantı şeklinin fotoğraflar ile göstereceğim önerdiğim bir yöntem olmadığı için sizlere doğru sonuç vereceğini garanti etmiyorum.

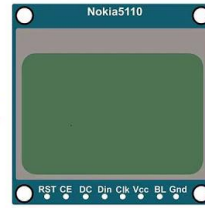
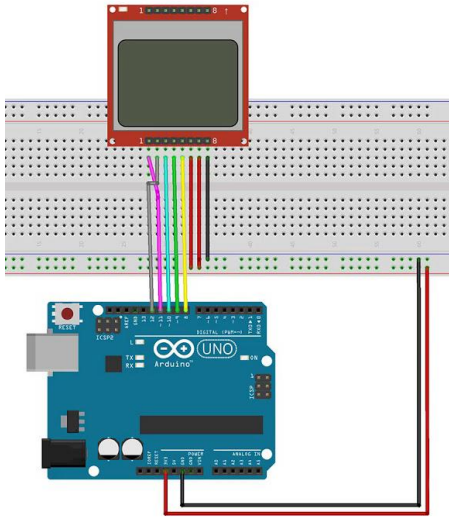


Lcd de potansiyometre olmadan bağlantı şeması bu şekilde olup bağlantı pinleri farklı seçilmiştir. Bağlantı şeması bu şekilde pek sağlıklı değildir açıkçası ve bu yüzden pek tavsiye etmem. Burada ekran kullanımı sonlanıyor sanıyorsanız maalesef ki yanılıyorsunuz.

Bu ekran çeşitliliği çok aşırı olup yeri geldiğinde küçük ekranlar kullanmamız gerekeceği için sizlere küçük ekran olarak nokia 5110 ekranını da tanıtır ekran olayına bu kitap için en azından nokta koyacağız. Nokia 51110 ekranı çok kullanıldığı için birinci eğitim kitabında bulunması zorunlu olacağını düşündüğüm bir modüldür fakat bu içerik çok fazla bilgi ve programlar içerdiği için tamamen resim çıktısı alma olayını sonraya bırakacağım. Resim çıktısını aldığımız programı gösterip o kısma değinmeyeceğim.



Böyle bir program ile sadece görüntü alındığını burada kod olarak adlandırmızı bilin şimdilik yeterli olacaktır. Aşağıdaki görselde bağlantıları güzelce belirtilmiş olup küçük bir “Hello World” Yazıp bırakacağım



NOKIA 5110 EKRAN KULLANIMI

1-RST: Bupine 0V uygulanırsa modül sıfırlanır.

2-CE: Birden fazla sı çevre birimi kullanıldığında kullanılır.

3-DC: Komut ya verimi işleneceğini belirten pindir.

4-DIN: Veri transferi pini.

5-VCC: 2/7 ile 3/3V arası gerilim uygulanan pin.

6-BL: Arkaplan ışığıdır. Daima yansın istersek 3/3V ye direkt bağlanır. Arkaplan ışığını kendiniz ayarlamak isterseniz aynı led gibi 220ohm direnç ile bağlamanız gerekmektedir.

7-GND: Toprak piniimizdir.

```
#include <LCD5110_Basic.h> // LCD kütüphanemizi ekliyoruz
LCD5110 myGLCD(8,9,10,11,12); // LCD pinlerimizi tanıtıyoruz
byte lcd_ledi = 6;
void setup()
{
    //Eğer ekranı kendiniz ayarlamak isterseniz aşağıdaki kodları aç.
    pinMode(lcd_ledi,OUTPUT); // Ekran ledini pine atadık.
    digitalWrite(lcd_ledi,HIGH); //ekran ışığını açıyoruz
    myGLCD.InitLCD(); //LCD ekranı başlatıyoruz
    myGLCD.clrScr();
    myGLCD.print("Hello World!",LEFT,10);
    // Ekranı "Hello World" yazısını basar.
}
void loop{}
```

Evet kısaca sizlerle bu kodla birlikte Hello World bakışı aldık ama tabiki de görseller filan basmak çok zor değil bu ekranda sadece bilmek lazım “Projelerle Arduino” kitabımda bu ekranla ilgili bilgiye fazlaca yer vereceğim ama şimdilik bu kodlara odaklanalım.

myGLCD.print(“metin”,X,Y); → Bu fonksiyon ile ekrana yazı yazdırıyoruz. İlk değişken yazdırılacak metin, ikinci ve üçüncü değişken ise metnin ekranda geleceği koordinatlardır. X ve Y için daha öncesinden tanımlı LEFT,RIGHT veya CENTER değişkenleri kullanılabilir bizde bu kodumuzda “LEFT” yani sola yaslanmış şekilde yazıyı çıkarmak istedik.

myGLCD.clrScr(); → Ekranı temizler sizlere boş bir ekran vermiş olur siz doldurana kadar. Kod da “Hello World” yazdırmamız ile ekran silinme fonksiyonu artık geçersiz oluyor bir daha çağırılana kadar.

myGLCD.setContrast(integer degisken) → Kontrast ayarını yapar. Değişken olarak 0-70 arasında bir değer girilmelidir. Bizim kodlarımız içerisinde yer almasa da yine de bu bilgiyi vermek istedim.

myGLCD.setFont(degisken_ismi) → LCD ekrana yazdırmak için kullanacağımız fontu ayarlamamızı sağlar. Bu fonksiyon içine SmallFont, MediumNumbers, BigNumbers yazılabilir. İsminden de anlaşılacağı gibi bu değişkenler yazı ya da rakam boyutunu değiştirir. Yine bu komutları kodumuzda kullanmasakta kullanmak istersek kodumuzun void setup kısmından önce bir tanımlama yapmamız gerekmektedir. Bu tanımlamalar şöyle gözükmekesi gerekli :

```
extern uint8_t BigNumbers[]; // Kullanacağımız fontlar için
```

```
extern uint8_t SmallFont[]; // Bu tanımlamaları yapabiliriz
```

```
#include <LCD5110_Basic.h> // LCD kütüphanemizi ekliyoruz
LCD5110 myGLCD(8,9,10,11,12); // LCD pinlerimizi tanıtıyoruz
extern uint8_t BigNumbers[]; // Kullanacağımız fontlar için
extern uint8_t SmallFont[]; // Bu tanımlamaları yapabiliriz
byte lcd_ledi = 6;
void setup()
{
    //Eğer ekranı kendiniz ayarlamak isterseniz aşağıdaki kodları
    pinMode(lcd_ledi,OUTPUT); // Ekran ledini pine atadık.
```

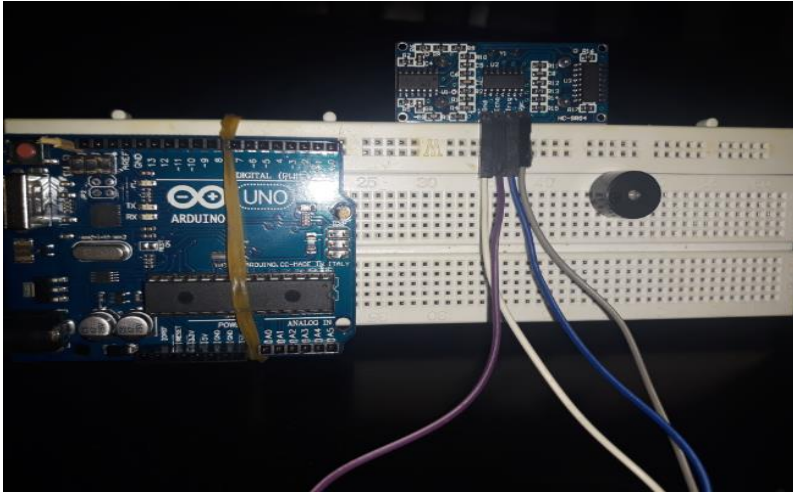
Yani bu şekilde olmalıdır. Kitabımız burada bitti artık normal bir şekilde sonlandırıyoruz. İlk basım olduğu için eksikler olabilir bu konuda belirtilen iletişim kanallarından iletişime geçebilirsiniz.

Sizleri bu yazılım dünyasına ısındırmak hırslandırmak için yapılan projelerle yalnız bırakıyorum ama öncelikle birkaç çözüm var. Bu kitabı yazmam çok uzun sürdü açıkçası belki ilk defa kitap yazdığımdan olabilir belki 10 gün sürdü tamamlanması fakat günde 7 saate yakın bir kısmını verdim açıkçası bunu sizlere 10 Ağustosta çıkaracağım diye söz verdim diye yaptım.

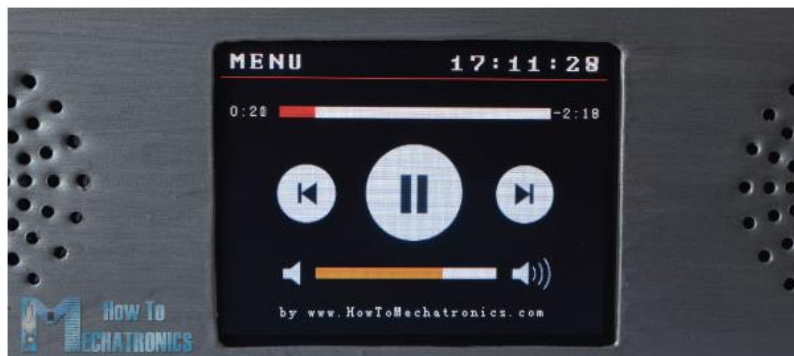
Diğer kitaplar için biraz beklemenizi istiyorum çünkü artık o kısım biraz daha da teknik olacak ve daha kapsamlı olacak tek tek sizlere o projeleri yaparak eğitim vermek istiyorum yoksa sadece kitap ya da sadece video ile eksik kalacağına inanıyorum. Kitabı hem yazıyor hem de videosunu çekiyorum kitap içeriğini bu süreç çok yorucu ilerliyor ve beni diğer işlerimden alıkoyuyor inşallah sizde bu verdiğim emeği takdir edersiniz. Bu süreçte bir emek verdim diye vidolarımı udemy üzerinden de satışa sunacağım. Udemy üzerinden videolara istekler doğrultusunda güncellemeler sağlayacağım. Biraz fazla konuştum ama arduino en basit elektronik kart programlama şekli lakin bunu gömülü sistem olarak kodlamaya yani c++ ile kodlamaya bu temeli öğrendikten sonra geçiş yapmanızı öneririm. C++ ile arduino gömülü sistemler olarakta videolar ve e kitap yayınlayacağım ama her birisi bir süreç ve basitten ileriye doğru sıralıyorum. Her şey için teşekkür ederim sağlıklıca kalın. 😊

Arduino ile örnek projeler

Sandalyeye koyulan sensör ile oturma bozukluğunu uyarı sistemi ile sizlere sunan proje. →



Alarm kurabileceğiniz ve müzik çalabileceğiniz bir proje. →



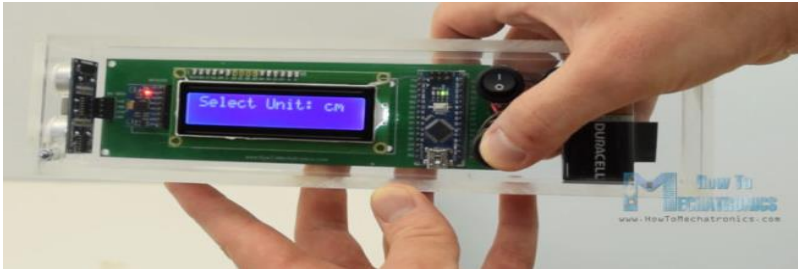
Basit bir otomat projesi. →



BGüzel sade bir hava istasyonu. →



Mesafe ölçer ve su terazisi →



Robot kol →



Ben size sadece birkaç daka arařtırmam ile bulduęum fotoęrafları paylařtıım daha fazlasını emin olun sizde bulursunuz.
Göstereceklerim de bu kadar ve gerisi sizde topu size attım
hořçakalın ve sadece bir kaynaktan ilerlemeyin. 🙌

Sözler:

Bilgisayar devrimi dünya tarihinde yaşanan en sesli devrimdir,
komik olan onun etkilerini hala küçümsememiz. Herman Kah

Artık insanlık deęil, teknoloji ilerliyordu. Uygarlıklar deęil,
teknolojiler geliřiyordu. Azra Kohen

Kişisel bir bilgisayar, zekamızın bir parçası olarak bize destek olan
araçtır. Steve Jobs

Yaratıcı bir işadamına bilgisayarlarla ilgili bir şey öğretmek, bir
programcıya iş hayatını öğretmekten çok daha kolaydır. Peter
Lawson

-SON-