

Ders 1 - Mikrodenetleyici Nedir?

Bir OR kapısının ne iş yaptığını hepimizin biliyor olması gerekir. OR kapısı lojik bir kapıdır ve iki adet girişi ile bir adet çıkışı kontrol edebilir. Siz de DIP kılıfa sahip 4 adet OR kapısını içeren bir entegre(74LS32) ile bu kapıların nasıl çalıştığını breadboard üzerinde gözlemleyebilirsiniz. DIP kılıfa sahip entegrelerde bir Vcc pini bir de GND pini olması gerekir. Elektrik sinyalleri Vcc ve GND arasında akarak entegrenin çalışması sağlanmış olur. Burada kullandığınız entegrenin nasıl yapıldığını, tasarlandığını bilmek zorunda değilsiniz. Şimdilik sadece girişlerin çıkışı nasıl etkilediğini bilmeniz yeterlidir. Bunun için lojik 1 vermek istediğiniz girişi Vcc ucuna, lojik 0 vermek istediğiniz girişi ise GND ucuna bağlayabilirsiniz. Eğer bu işlemler denemek için biraz zor gibi görünüyorsa merak etmeyin. Bu sistemin nasıl çalıştığını denemenin daha basit bir yolu var.

Bir mikrodnetleyici, OR kapısına benzemektedir. Aynı OR kapısındaki gibi mikrodnetleyicilerde de girişler ve çıkışlar vardır. Burada asıl mükemmel olan şey ise mikrodnetleyicilerde bu işlemlerin kod yazılarak yapılmasıdır. Örneğin A ve B girişlerinin C çıkışını nasıl etkilediğini aşağıdaki kod ile gözlemleyebilirsiniz.

```
if (A == 1 || B == 1)
```

```
C=1;
```

```
else
```

```
C=0;
```

Burada verilen kod bir C kodudur. Bu kodu derledikten sonra mikrodnetleyiciye yüklerseniz kodun nasıl çalıştığını gözlemleyebilirsiniz.

Mikrodnetleyiciler elektroniğin her alanında kullanılmaktadır. Mesela ; Mikrodalga fırınlar, TV Uzaktan kumandaları, cep telefonları, mouse, yazıcı gibi cihazların hepsinde mikrodnetleyiciler kullanılmaktadır.

Önceleri, mikrodnetleyiciler sadece bir kez programlanabilir ROM yapısındaydı. Bu da demek oluyor ki mikrodnetleyiciyi bir kez programladıktan sonra eğer kodunuz yanlış çalışıyorsa yeni kodunuz için yeni bir mikrodnetleyiciye ihtiyacınız vardı. Fakat yeni teknoloji ile yeni mikrodnetleyiciler ROM hafıza yerine 'flash' hafıza içerdiği için binlerce kez programlanabilmekte.

Flash hafızalı mikrodnetleyiciler bilgisayarlardan ve RAM'lerden biraz farklıdır. Bilgisayarlar çalışabilmek için çok fazla güce ihtiyaç duymaktadırlar. Ve çok da fazla ısınırlar. Bilgisayarlar, açılırken boot edilmek zorundadırlar fakat mikrodnetleyiciler sadece birkaç mili saniyede içerisinde flash hafızasındaki kodu çalıştırabilmektedirler.

Tekrar OR kapısına dönecek olursak; bu entegrenin giriş ve çıkışlar için kullanılabilen pinleri bulunmaktadır. Bir entegrenin içerisinde 4 adet OR kapısı olduğunu düşünürsek toplam 8 giriş ve 4 çıkış etmektedir. 2 pini de Vcc ve GND için eklersek toplam 14 pinli bir entegre ile 4 adet OR kapısını içeren bir yapıya sahip olabiliriz.

Bu işlemi bir mikrodnetleyici ile yapmayı düşünürsek bunun için GPIO pinlerini kullanmamız kaçınılmaz olacaktır. Bunun için aşağıdaki gibi bir kod yazılabilir.

```
if (PORTC.2 == 1)
```

kod bloğu...

Çıkış pinleri basit bir gecikme ile yanıp söndürülebilir. Örneğin ;

```
while (1)
```

```
{
```

```
RB3 = 1;
```

```
delay_ms(1000);
```

```
RB3 = 0;
```

```
delay_ms(1000);
```

```
}
```

Bu uygulama mikrodnetleyiciler dünyasının "Merhaba Dünya" uygulamasıdır. Küçük bir uygulama gibi görünse de mikrodnetleyicilerle program geliştirmeye ilk adımı atma niteliğini taşıdığından aslında önemli bir uygulamadır.

Piyasada yaygın olarak kullanılan mikrodnetleyicilerin genel özellikleri aşağıda verilmiştir.

- PIC - Microchip firmasının klasik mikrodnetleyicisidir. Piyasada en yaygın bulunan, kullanımı kolay bir mikrodnetleyicidir.
- AVR - PIC mikrodnetleyicilerin yaptığı tüm işleri yapabilmesine ek olarak PIC'ten daha hızlı, daha iyi ve daha ucuzdur.
- MSP430 - Texas Instruments firmasının ürettiği başarılı bir mikrodnetleyicidir. AVR'ler kadar güçlü değildir fakat düşük güç tüketme konusunda oldukça iddialıdır. Bir adet AA pil ile 5 yıla kadar dayanabilmektedir.
- ARM - PIC veya AVR gibi basit, 8 bit mikrodnetleyicilerin aksine ARM'lar 32 bit ve daha kompleks bir yapıya sahiptirler. Endüstrinin her alanında sıklıkla kullanılan bu mikrodnetleyiciler aslında 8 bit mikrodnetleyicilerden çok da fazla pahalı değillerdir. Fakat diğer mikrodnetleyicilere nazaran daha kompleks bir yapıya sahip olduğu için başlangıç için tercih edilmekte genellikle profesyonel uygulamalarda tercih edilmektedir.
- 8051 - 8051 mimarisi 1980'lerde Intel tarafından geliştirilmiştir. Intel'in o yıllarda sadece kontrol amaçlı olmak üzere bit bit işlem yapabilen bir sistemi geliştirmesinin ardından fazla popüler olan 8051 mimarisini birçok ünlü marka da satın alıp üretmişlerdir. Ve şu anda hala kullanılmaktadır. Ayrıca bu mimari ARM mimarisinin de temelini oluşturmaktadır.
- 68HC08/11 - Bu mikrodnetleyici ise Motorola'nın üretmiş olduğu önceden oldukça popüler olmuş, şimdi ise üniversitelerde temel mikroişlemci mimarisinin üzerinden işlendiği bir mikrodnetleyicidir.

Ben uygulamalarımda Atmega168 mikrodnetleyicisini kullanacağım. Neden mi ?

- 20 MIPS(Mega Insturction Per Second)
- Ucuz
- Gerekebilecek tüm çevre birimleri içerisinde barındırıyor.
- 16 KB program hafızası başlangıç projeleri için fazlasıyla yetiyor.
- Ücretsiz C derleyicisi.
- 20\$ gibi bir maliyetle programlayıcı ve debugger edinilebiliyor.

Herhangi bir entegre devre mutlaka üzerinden enerji alabileceği bir kaynağa ihtiyaç duymaktadır. Bu kaynak entegreyi Vcc ve GND uçlarından beslemek için kullanılır. Vcc, TTL(5 volt) seviyesinde çalışan entegreler için +5v'olu ifade eder. GND ise ground kelimesinin kısaltılmışıdır. Mikrodnetleyiciler genellikle TTL seviyesinde çalışan entegrelerdir. Dolayısıyla bu entegreleri beslemek için mutlaka bir 5v'luk kaynağımız olmalıdır.

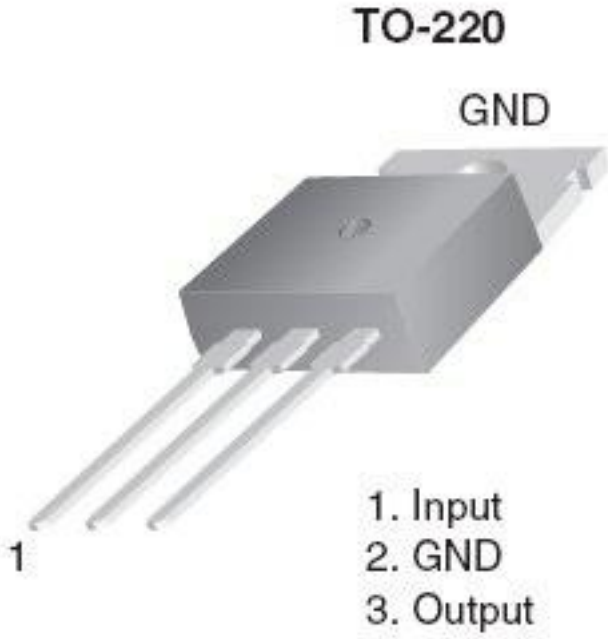
5 Voltluk Gerilim Elde Etmek

Bu ders için gerekli olan tüm parçaları burdan temin edebilirsiniz.

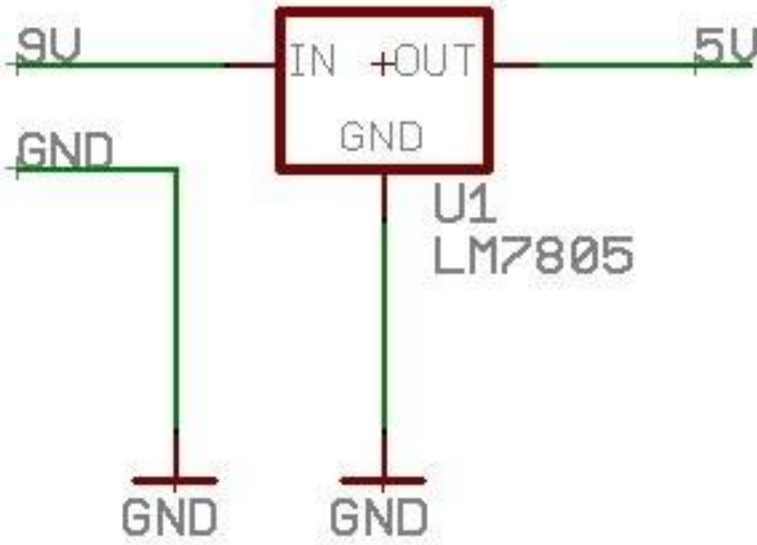
Mikrodnetleyiciler DC (Direct Current) gerilimle çalışan elemanlar olduğu için şehir şebekesi(AC) ile çalışmaya uygun değildir. Bunun için 220v AC şehir şebekesini 5v DC kaynağa dönüştürücü bir kaynağa ihtiyaç duymaktayız.

Not : Eğer mikrodnetleyiciniz Vcc ve GND pinlerine ters gerilim vererseniz yani Vcc pinine GND, GND pinine ise +5v vererseniz mikrodnetleyiciniz yanacak ve bir daha kullanılmaz hale gelecektir. Aynı şekilde +5v tan fazla bir gerilim verdiğiniz zaman da aynı sonuçla karşılaşrsınız. Bunun için denemelerinizi yaparken dikkatli olmalısınız.

8-9v'luk bir gerilimi 5v a indirmek için farklı regülatörler kullanılmaktadır. Fakat en yaygını LM7805'dir. Bu regülatörün girişine yaklaşık olarak 6v ve üzeri bir gerilim uyguladığınız takdirde çıkışınız sabit 5v olmaktadır. Fakat eğer çıkıştan çok fazla akım çekilecekse bu regülatör çok fazla ısındığı için uygun değildir. Tipik mikrodnetleyici uygulamaları için idealdir. Aşağıda LM7805 Voltaj Regülatörünün TO-220 kılıfı görülmektedir.



Breadboard üzerine bu regülatörü takıp aşağıdaki gibi bağlamalısınız.



Not : Şemada GND'ler ayrı ayrı bağlanmış gibi görünse de breadboard bağlantısında tüm GND'ler ortak olmalıdır.

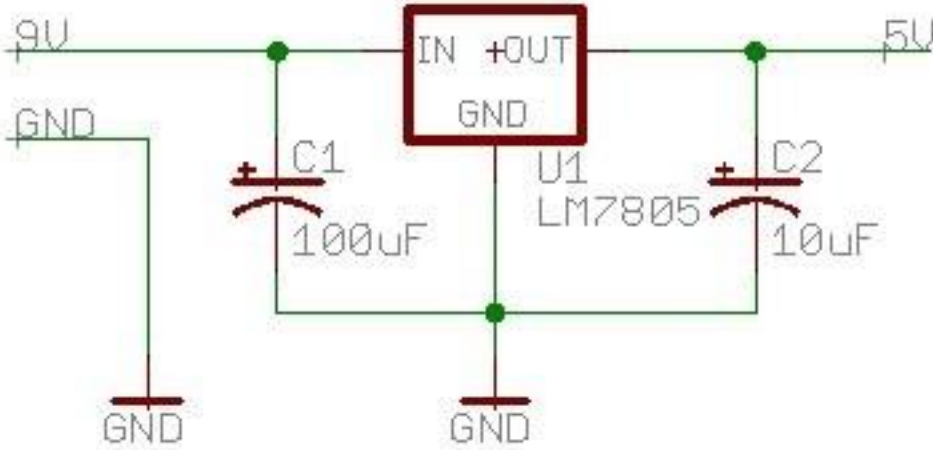
Bu devreyi bradboarda kurup AVO metre ile çıkış gerilimini ölçtüğümüz zaman tam olarak 5.00v gibi bir değer göremeyiz. Çünkü elektronikte herşey teorikteki gibi değildir. Her parçanın mutlaka toleranslar olacaktır. Bu regülatör için tolerans değeri +-%5 tir. Yani ölçtüğümüz değer ya 5.25v ya da 4.75v çıkacaktır. Pratikte biz bunu 5.1v veya 4.9v olarak görebiliriz.

Tipik olarak yukarıdaki gibi bir devre kurduğumuz zaman eğer sadece çıkış gerilimine bakıyorsak 5v a yakın bir değer alabiliriz. Fakat girişten gelen gürültü faktörünü ve dalgacıkları(ripple) da muhakkak düşünmeliyiz.

Bunu AVO metre ile görmeyiz fakat çıkışımızı bir osilaskop ile incelersek bu dalgacıkların ve gürültünün çıkış gerilimine etkisini görebiliriz. Bu faktörler, 100uF ya da 10uF kutuplu kondansatörlerle büyük ölçüde yok edilebilir.

Kutuplu kondansatörler, üzerlerindeki gerilimi depo edebilen devre elemanlarıdır. Ve üzerlerinde sadece DC gerilim depo ettiklerinden dolayı giriş geriliminde olan dalgacıklar bu kondansatör üzerinde oluşmayacaktır. Ve dolayısıyla da çıkışımız olabildiği kadar saf düz bir sinyal(DC) olacaktır. Burada bu saflığı giriş için kullanılan kondansatörün büyüklüğü belirlemektedir. Örneğin 330uF...

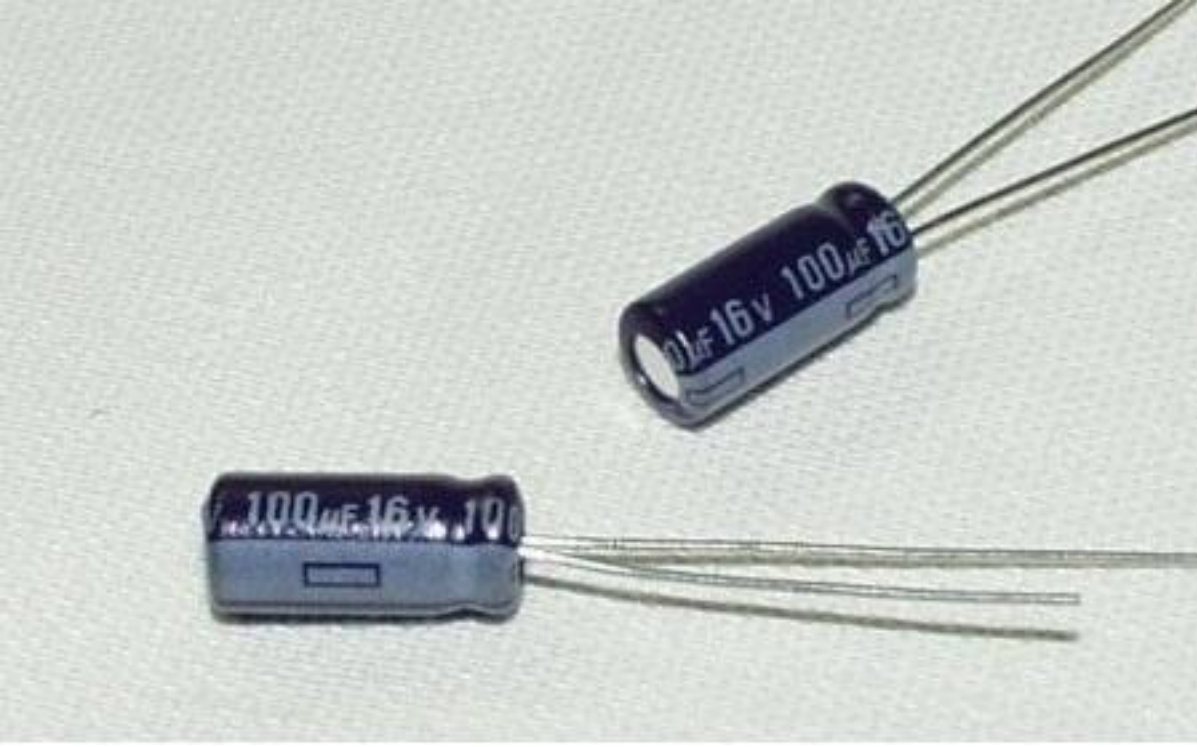
Filtre devremiz LM7805 regülatörüne ek olarak aşağıdaki gibi kurulabilir.



Not : Burada kullanılacak kondansatörlerin maksimum geriliminin 100uF için >9v, 10uF için >5v olmasına dikkat edilmelidir. Bunu iki katı olarak da hesaplayabiliriz. Mesela 9v için en az 18v'luk bir kondansatör, 5v için ise en az 10v'luk bir kondansatör kullanılabilir. Aksi halde kondansatörler arasındaki köprü delineceğinden kondansatörler bir daha kullanılmaz hale gelecektir.

Kondansatörleri biraz daha yakından inceleyecek olursak ;

- Elektrolitik Kondansatörler (Kutuplu Kondansatörler) : Çok büyük kapasite değerlerine ulaşabilen (10uF ... 1,000,000 F) kondansatörlerdir. Bir + ayağı bir de - ayağı olması sebebiyle "Kutuplu Kondansatör" olarak isimlendirilmiştir.



100UF - 16v Elektrolitik Kondansatör

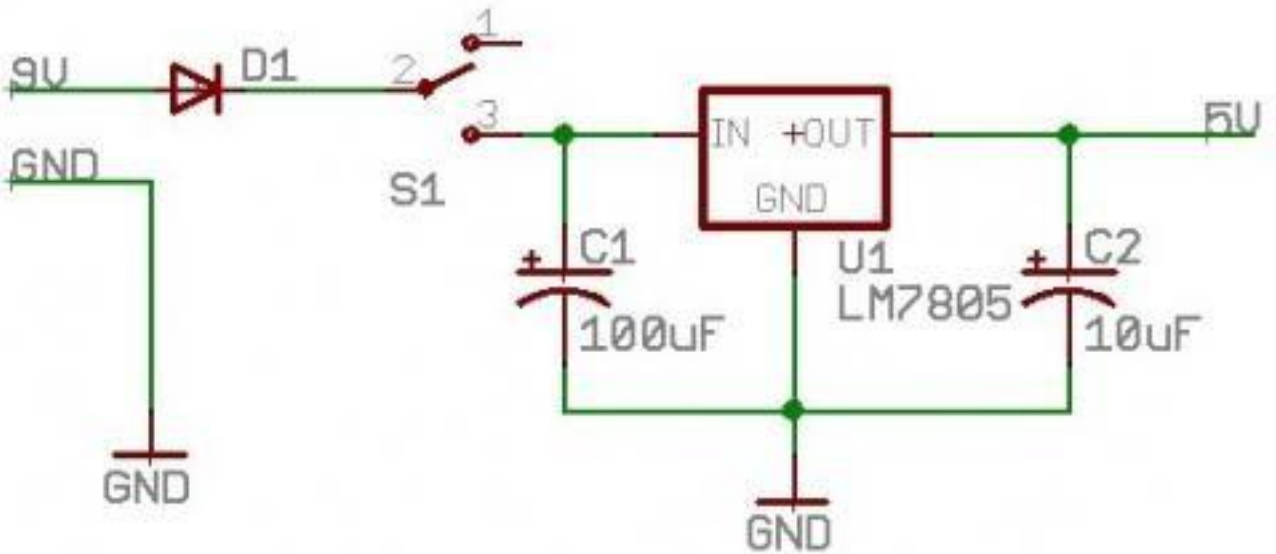
- Seramik Kondansatörler : En çok kullanılan fakat fazla ucuz olmayan kondansatör çeşididir. Elektrolitik kondansatörler gibi + ve - ayakları yoktur. Yani yönlü değildir. Ve elektrolitik kondansatörler kadar yüksek kapasitelere sahip değildir.
- Birçok farklı kondansatör çeşidi bulunduğu için hepsine burada değinilmeyecektir. Farklı çeşitleri internetten araştırabilirsiniz

Tekrar breadboarddaki devremize dönecek olursak; Bu kondansatörleri bağlarken dikkatli olmalıyız. Yani pozitif gerilim yönüne +, negatif gerilim yönüne - ayaklarını bağlamalıyız. Eğer tersini yaparsak kondansatörümüz delinecektir.



Yukarıdaki resimde en sağdaki kondansatör, üzerine uygulanan gerilimden dolayı delinmiş bir kondansatördür. Yani ya + - ayakları ters bağlandı veya kondansatörün taşıyabileceği maksimum gerilimden daha fazla gerilim uygulandı ise bu durum gerçekleşir. Böyle bir durum sonucunda ise kondansatör artık filtre görevi yapamayacaktır. Doğrudan toprak ile kısa devre olacaktır.

Herhangi bir terslik anında devrenin elektrikle ilişkisini kesmek için bir anahtar devresi şemamıza eklenebilir.



Bu şekilde girişle hiçbir bağlantımız olmadan devremizi açıp kapatabilmekteyiz. Bu, bize pratik olarak büyük ölçüde kolaylık sağlamaktadır. Anahtarlama elemanının 3 adet ayağı bulunmaktadır. Bunlardan 2. ayak ortak olarak kullanılmakta, diğer ikisi ise amacımıza göre bağlanabilmektedir. Biz bir ayağı devrenin elektrikle ilişkisini kesmek için boş bırakacağız. 2 ve 3. ayaklar sırası ile 9v ve GND olacaktır.

Güvenliği tam sağlamak amacıyla 9v'luk girişin + ve - uçlarını karıştırabileceğimizi de düşünerek devremize bir de koruma diyotu eklememiz gerekmektedir. Bu diyot sayesinde Vcc yerine GND, GND yerine ise Vcc geldiği zaman devremiz yanmayacaktır. Çünkü D1 diyotu sayesinde sadece tek yönlü iletim olmaktadır.

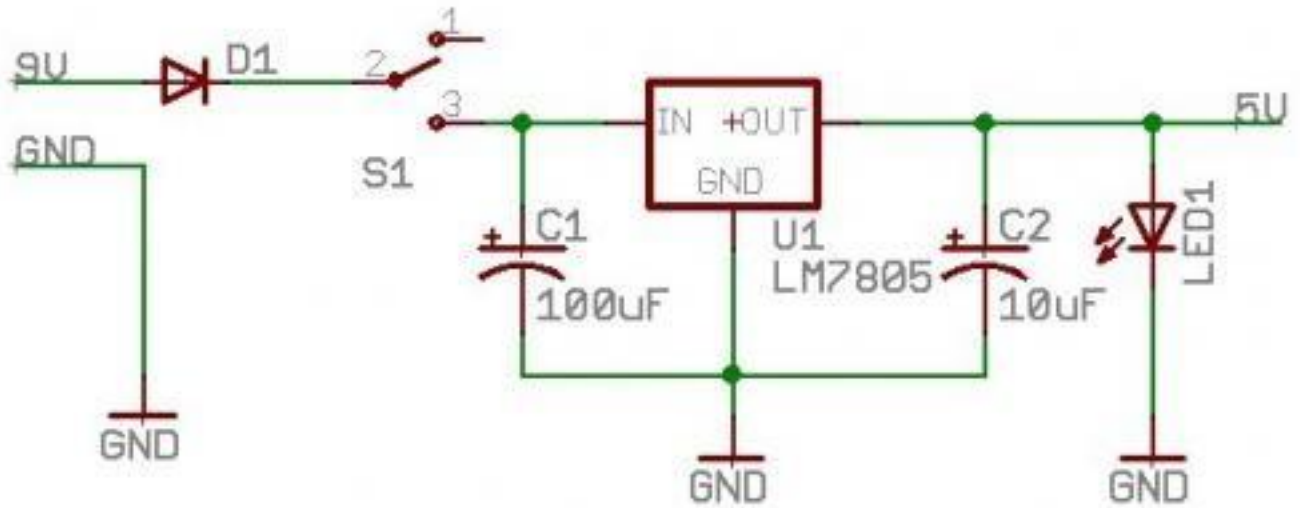
Burada diyot, üzerinden sadece pozitif yönlü gerilimin geçmesine izin vermektedir. Nasıl olduğunu incelersek ;

Bir diyotun Anot ve Katot uçları olmak üzere iki adet ucu vardır. Eğer biz Anot(A) ucuna +, Katot(K) ucuna ise - verirsek diyot iletme geçer ve kısa devre görevi görür. Bu işleme doğru polarma denilmektedir. Eğer tam tersi bir gerilim uygularsak diyot bu gerilime karşı koyacaktır. Diyotun bu tepkisine ise ters polarma denir. Burada doğru polarma durumunda diyotun iletme geçmesi için minimum 0.7v'luk bir gerilim uygulanmalıdır(silisyum diyotlarda).

Ters polarmada ise, diyotun kırılma gerilimi denilen bir durum vardır. Bu değer, her diyot için farklı olabilir. Eğer bu değer üzerinde bir gerilim diyota uygulanırsa diyot jonksiyonunda(birleşim yüzeyinde) oluşan ısı nedeniyle delinecek ve bulunduğu yeri tamamen kısa devre durumuna çevirecektir.

Bizim breadboarda kurduğumuz devrede 1 Amperlik 1N4001 silisyum diyodunu seçtik. Notmalde dijital bir devrede 1 Amperlik bir yüke ihtiyaç duyulmaz fakat her açıdan garanti olması için bu koruma diyotunu yüksek güçte seçtik.

Devremize bir de çıkışımızı görmek amacıyla LED ekleyebiliriz. LED'ler, genel olarak 20mA gibi bir akıma ihtiyaç duymaktadır.



Eğer bir LED üzerinden 20mA den daha fazla akım geçerse bu akım LED için tehlikeli olabilir. Bu sebeple devremize eklediğimiz LED için bir akım sınırlama devresi tasarlamamız gerekmektedir. Bu çok kolaydır. Devrede kullandığımız LED'e seri bir direnç eklememiz bu iş için yeterli olacaktır. Bu direncin ne değerinin ne olması gerektiğini hesaplırsak ;

$$V = I \times R \text{ (Ohm Kanunu)}$$

$V = 5v$ (Çıkış Gerilimi)

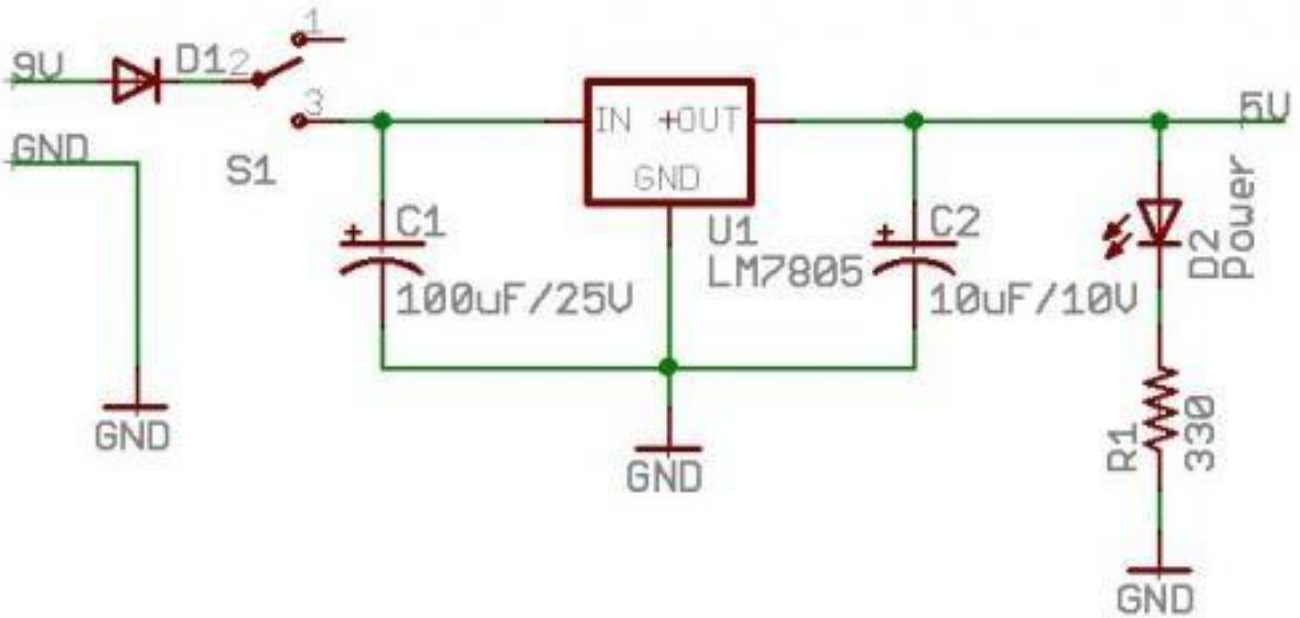
$I = 20mA$ (0.02 A)

$R = I$

$5 = 0.02 \times R$

$R = 250 \text{ Ohm}$

Bu hesap sonucunda kullanmamız gereken direnç değerinin 250 Ohm civarı bir değerde olması gerektiğini görüyoruz. Pratikte 250 Ohm gibi bir değerdeki direnci bulmak belki biraz zor olabilir. Bu sebeple eğer LED'inizin daha parlak yanmasını istiyorsanız 220 Ohm, daha sönük yanması ve az akım harcamasını istiyorsanız 330 Ohm kullanabilirsiniz. Yeni devre şemamız aşağıdaki gibi olacaktır.



Eagle schematic / PDF

Devremiz, bu haliyle çalışıyor gibi görünüyor. Devreyi breadboard a kurduktan sonra anahtarı açmadan önce her şeyin doğru olduğundan emin olun. Eğer anahtarı açtınız ve LED in yanmadığını gördüyseniz hemen anahtarı kapatın. Ve sonra tüm bağlantıların doğruluğunu tekrar kontrol edin. Eğer tüm bağlantıların doğru olduğundan eminseniz bir de LED in çalışıp çalışmadığını AVO metre ile test edin.

Şimdi regülatör devremiz için kullandığımız bu filtre devresinin ne işe yaradığına teknik olarak bakalım.

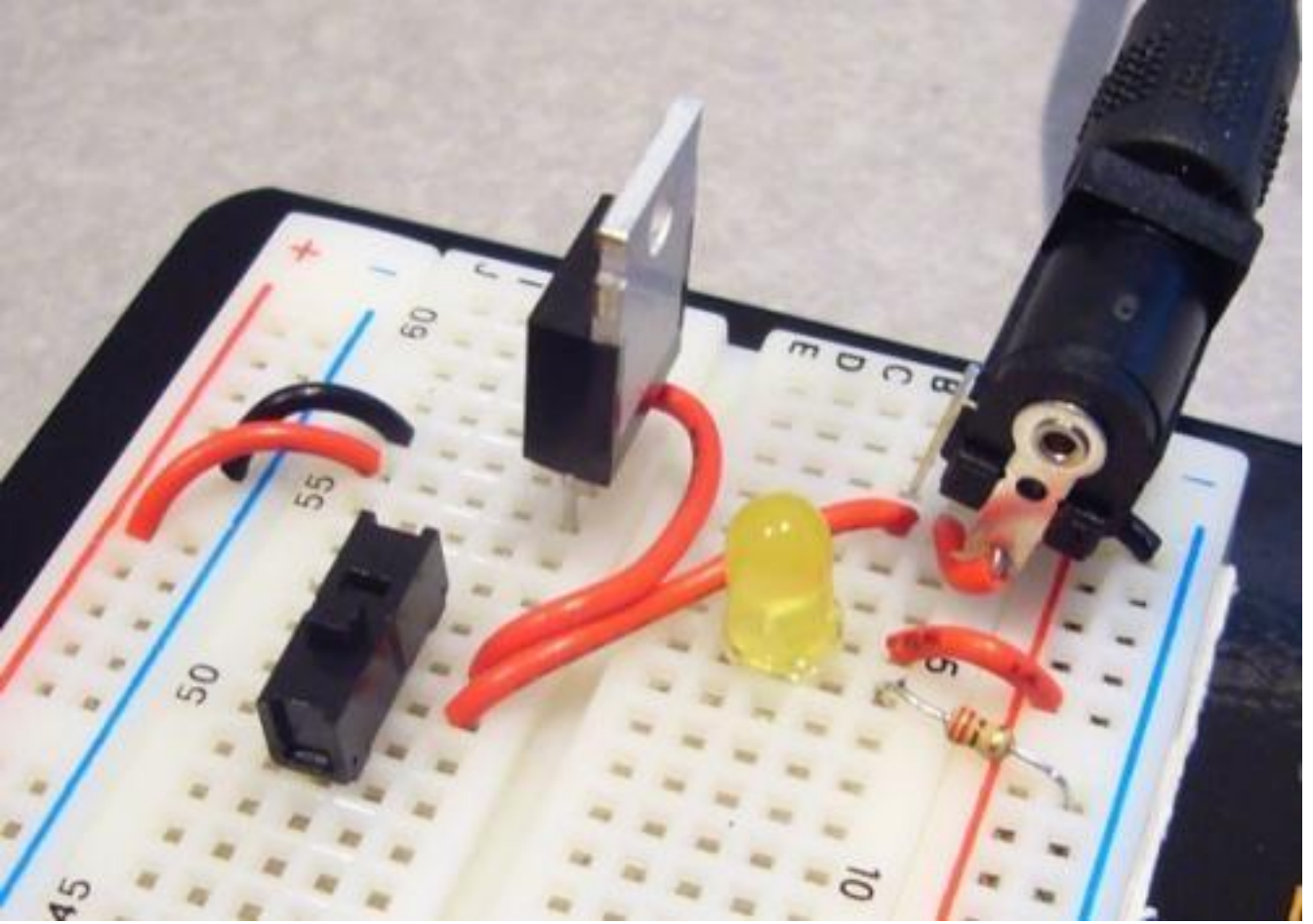
Eğer giriş geriliminizde 500mV gibi bir dalgacık(ripple) varsa bu dalgalanma mikrodenetleyicinizin kararsız çalışmasına sebep olabilir. Bu şu anlama gelmektedir : Eğer 4.8v gibi bir giriş geriliminiz varsa bu değer 4.3v a iniyor ve sonra tekrar 4.8v a çıkıyor demektir. Bu dalgalanma ise mikrodenetleyiciler, dijital entegreler için pek de iyi bir durum değildir. Hatta çoğu firma bunu engelleyip mikrodenetleyicilerinin yanmaması için çiplerinin içerisine koruma kısmı eklemiştir.

Örneğin giriş gerilimimizde 500mV gibi bir dalgalanma var ve Analog / Dijital dönüştürücü ile sıcaklık ölçümü yapıyorsak ve her 100mV un 1 santigrat C olduğunu düşünürsek 25 C için 2500mV(2.5V) gibi bir

değer okumamız gerekir. Fakat 500mV'luk bir dalgalanma neticesinde bu değer 2000mV(2.0v) ve 3000mV(3.0v) arasında sürekli oynayacaktır. Bu örnekte dalgalanmanın ölçüm sonuçlarına ne şekilde yansıdığı çok belirgin bir şekilde görülmektedir.

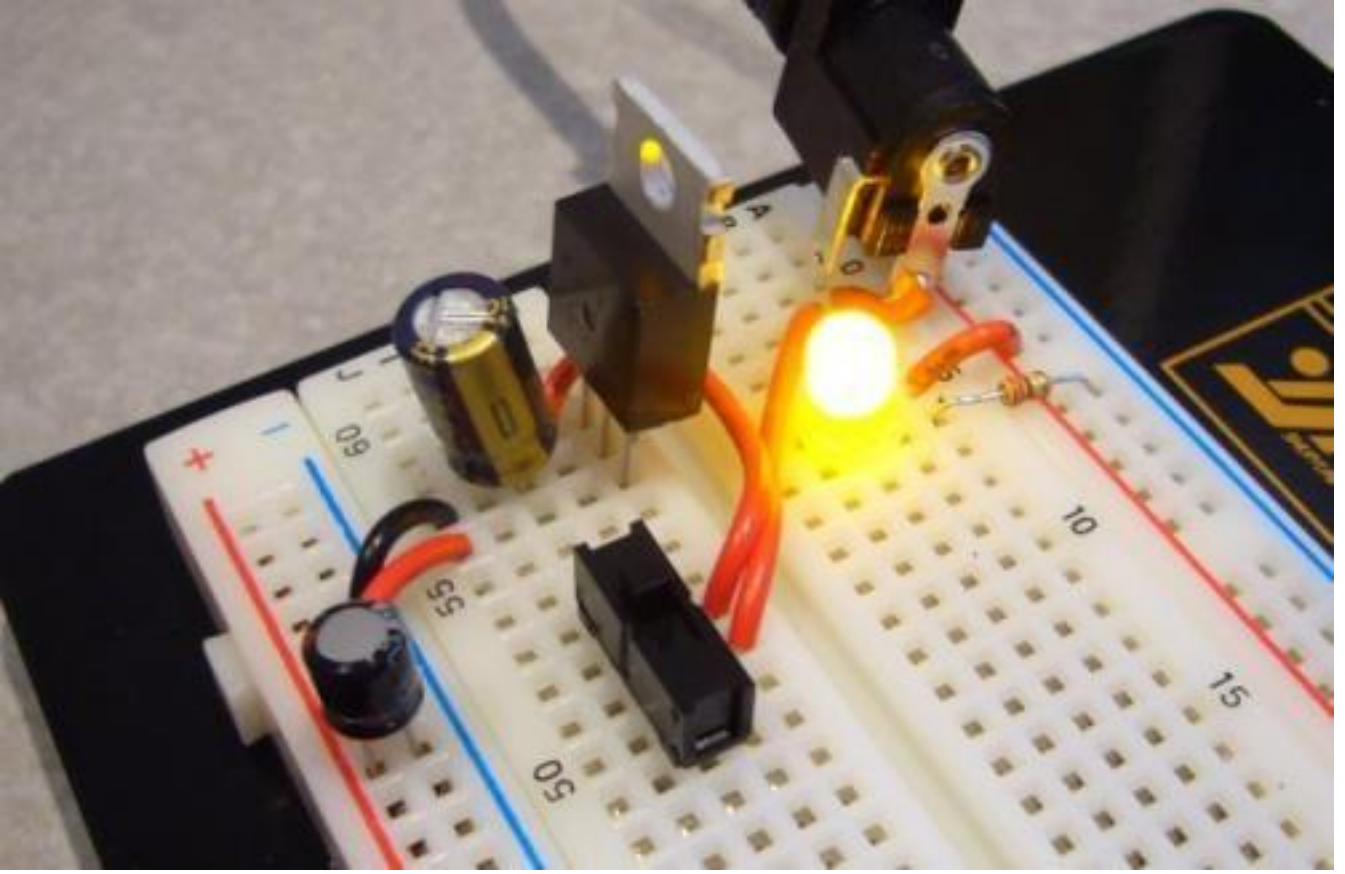
Breadboardlar

Breadboardlar hakkında daha geniş bir bilgi almak isterseniz Tom Igoe's breakdown un breadboard hakkındaki yazısını okuyabilirsiniz. Kısaca; sağ ve solda gördüğünüz kırmızı ve mavi sütunlar kendi arasında kısa devredir. Genellikle + ve - buradan verilerek breadboardun herhangi bir yerinden + ve - ye ulaşım kolayca sağlanmış olur. Diğer yerler ise aradaki boşluk hariç her sütun kendi arasında kısa devredir. Mesela aşağıdaki resimde görünen 10 numaralı sütundaki 5 delik kendi arasında tamamen kısa devredir. Bunun diğer yan sütunlarla hiçbir şekilde bağlantısı yoktur. Ve diğerleri de aynı şekilde.

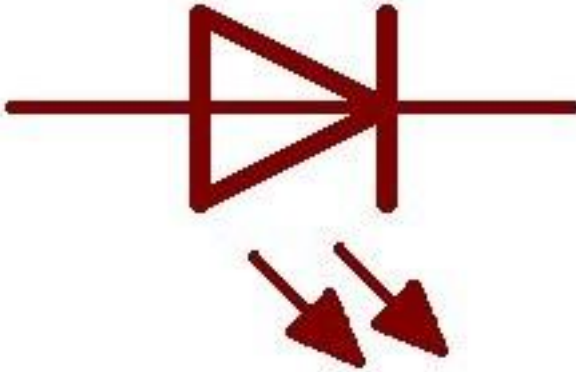


Güç girişi, anahtar, LM7805 ve LED
LED'ler

LED'ler (Light emitting diodes) tıpkı diyotlar gibi doğru polarmada çalışan devre elemanlarıdır. Zaten aslında LED'ler de bir çeşit diyottur. Türkçesi Işık yayan diyot olarak da bilinmektedir. LED'ler şematik çizimde aşağıdaki gibi gösterilirler.

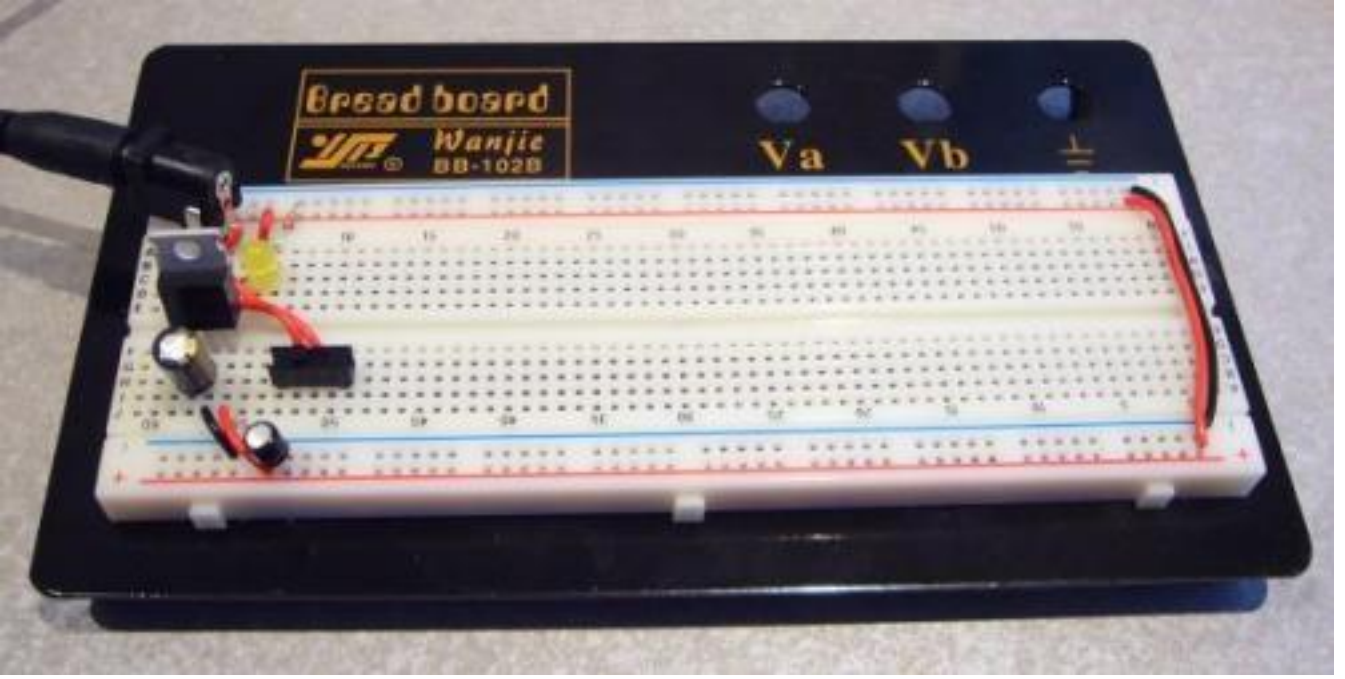


Tipik olarak LED'ler de normal diyotlar gibi anodunda 0.7v ve üzeri bir gerilim olduğu zaman iletme geçerler ve üzerlerindeki bu gerilimi ışık olarak etrafa yayarlar. LED'ler üzerlerinde kimyasal bir değişiklik olduğundan dolayı ışık yaydıkları için çalışma mantıkları ampullerden farklıdır. Çalışmak için mutlaka DC gerilime ihtiyaç duyarlar. Ve ömürleri ampullere göre çok daha uzundur. Ayrıca çok az akım çektikleri için pilli devrelerde kullanılabilirler.



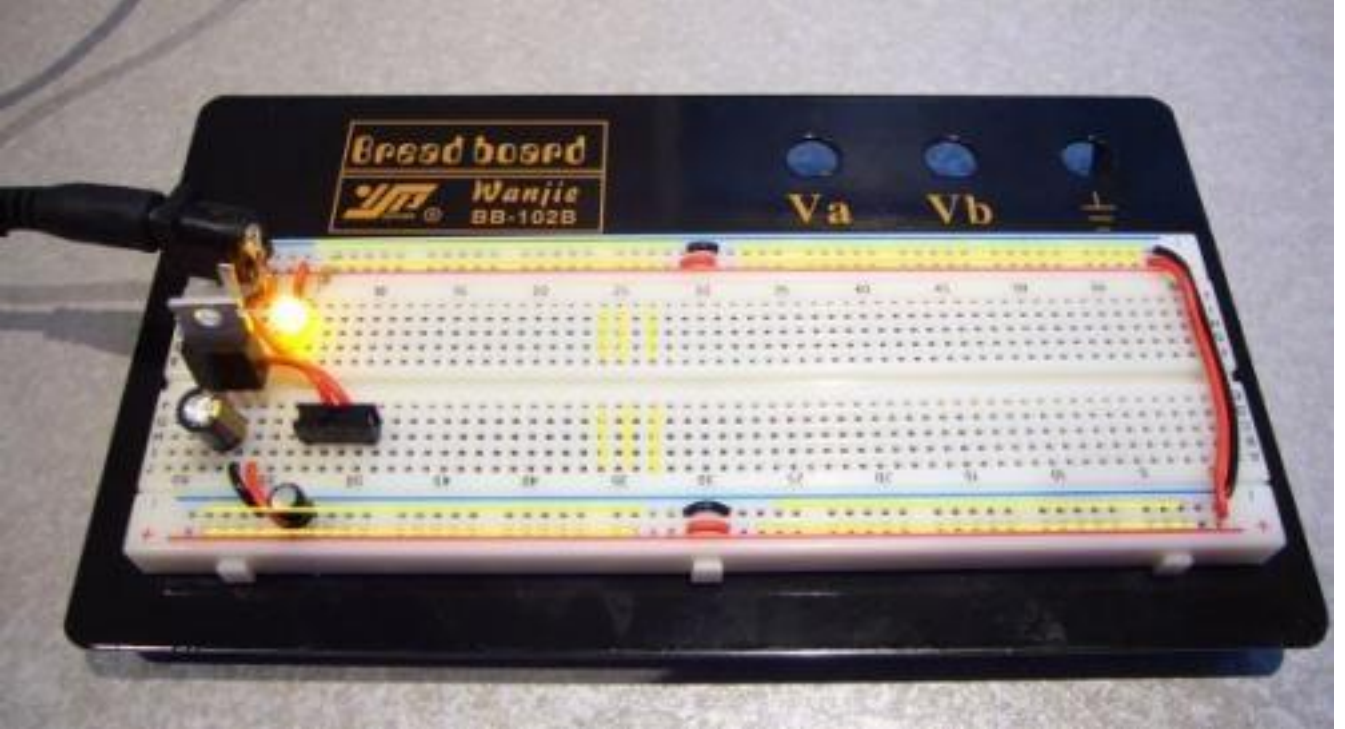
Devrede kullanılan anahtarın toplam üç adet ayağı vardır. Bunlardan ortadaki ayak sol ayak ve sağ ayak arasında iletişimi sağlamaktadır. Eğer anahtarı sol tarafa çevirirseniz sol ve orta ayak kısa devre olmakta, sağ tarafa çevirirseniz sağ ve orta ayak kısa devre olmaktadır.

Fakat siz aşağıdaki gibi bir devre kurup anahtarı açsanız bile LED'in yanmadığını göreceksiniz.



Anahtar Açık İken Devrenin Durumu

Çünkü breadboard için ufak bir düzenleme gerekmektedir. Breadboardlar, üzerinde iki ayrı devrenin kurulabilmesi için orta kısımdan elektriksel olarak ayrılmışlardır. Yani breadboardun tam ortadan solda kalan kısmının + ve - girişleri, sağda kalan kısmının + ve - girişleri ile bağlantılı değildir. Eğer sağ taraftaki gibi + ve - leri birleştirmek istiyorsanız aşağıdaki ufak düzeni breadboard üzerine eklemeniz gerekmektedir.



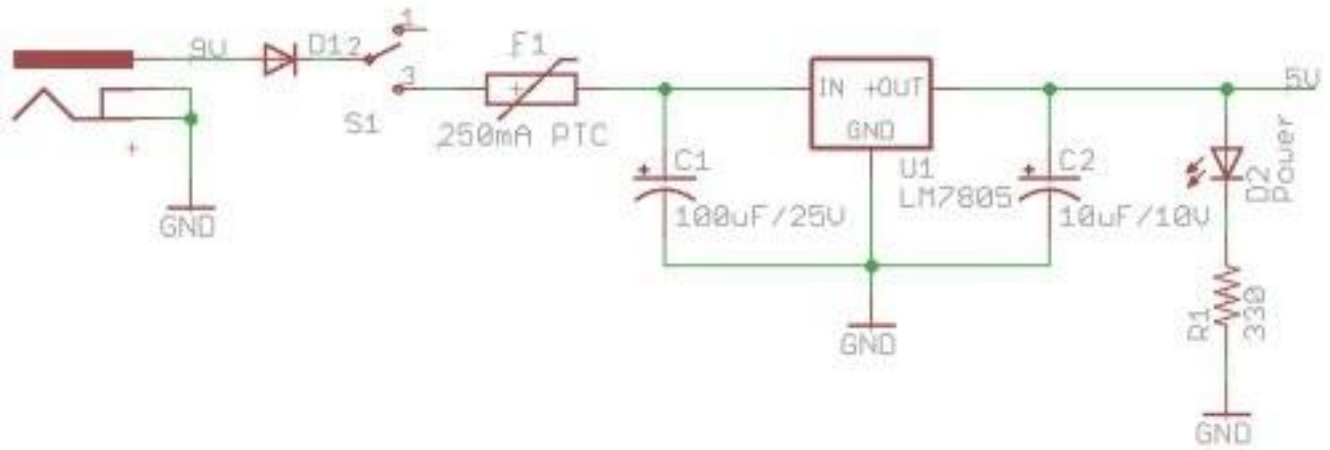
Bağlantılar Yapıldıktan Sonra Devre Çalışmaktadır

Breadboard üzerine voltaj regüle devremizi kurduktan sonra çıkış voltajımızın ne kadar doğru olduğunu AVO metre ile gözlemleyebiliriz.



Resimde de görüldüğü gibi 5v'luk regülatör kullanmamıza rağmen ölçtüğümüz gerilim 4.98v çıkmıştır. Bu, tüm pratik devrelerde böyledir. Bir direncin, kondansatörün vb... gibi devre elemanlarının değeri hiçbir zaman bire bir doğru çıkmaz. Çünkü pratikte toleranslar vardır. Bu toleranslar sonucunda devredeki değerler bizim hesapladığımızdan ger zaman farklı çıkmaktadır. Fakat 4.8volt gerilim ile Atmega328(veya 168) çalışabileceğinden dolayı bu durum bizim için bir problem oluşturmamaktadır.

Son olarak devremize güvenlik açısından bir PTC ekleyebiliriz. Bu ekleyeceğimiz PTC sayesinde eğer breadboard üzerinde Vcc ve GND (+ ve -) yi kısa devre yapsak bile PTC kısa devre oluşmasını önleyecektir. Fakat maksimum 250mA kadar dayanabilecektir. Böyle bir kısa devre olursa PTC ısınacak ve devrenizdeki malzemeleri yanmaktan kurtaracaktır.

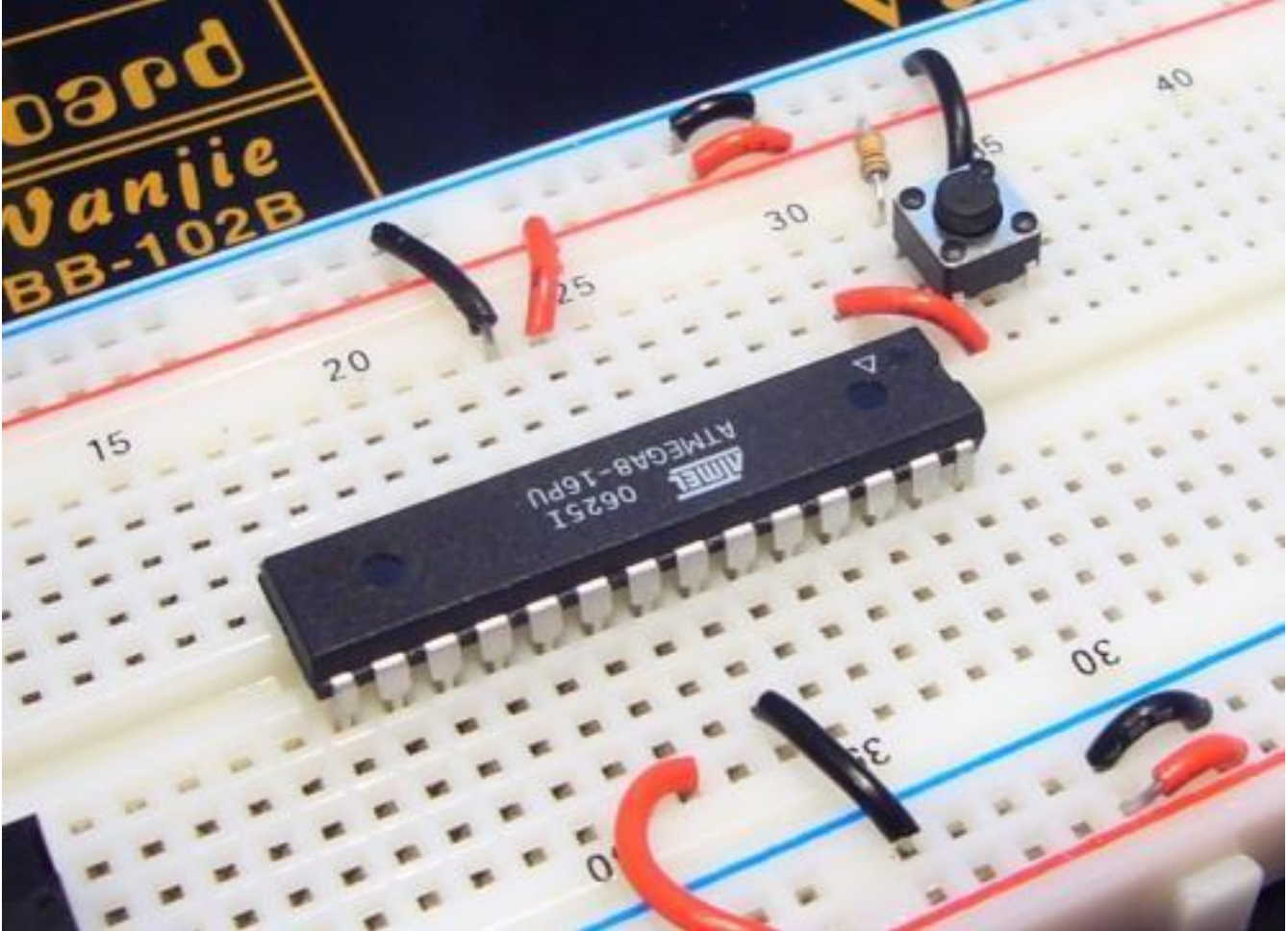


PTC'nin Devreye Seri Bağlanması

Ders 2 - Bir Mikrodenetleyiciye Nasıl Kod Yüklenir?

Bu dersimizde Atmel serisi mikrodenetleyicilere bilgisayarda yazılan kod nasıl yüklenir onu inceleyeceğiz. Bunun için buradaki tüm parçaları edinmiş olmanız gerekmektedir. 60\$ dan 300\$ a kadar farklı AVO metre çeşitleri vardır ben yaklaşık 60\$ civarında olanlarından öneriyorum.

Bir önceki yazıda verdiğim 5v devresini kurup, çalıştırdığınızı ve bu breadboarda bir Atmega168 yerleştirip, Vcc ve GND bağlantılarını yaptığınızı varsayıyorum.

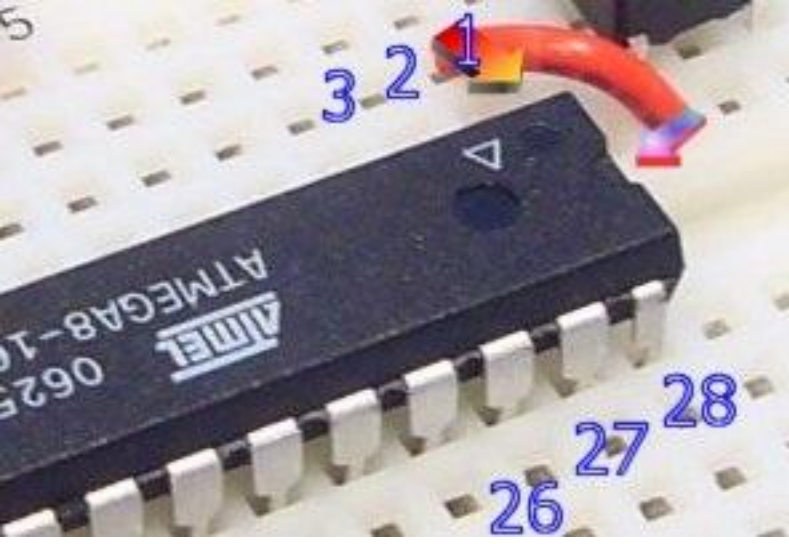


Atmega8(ATmega168 veya Atmega328 ile benzer şekilde çalışırlar) Breadboard Üzerinde

Atmega8 mikrodenetleyiciyi resimde görüldüğü gibi breadboarda yerleştirmek için bacaklarını hafifçe bükmeniz gerekmektedir. Bu işlemi, mikrodenetleyicinin tek tarafını masa gibi sert bir cisim üzerine hafifçe bastırarak gerçekleştirebilirsiniz. Ve diğer tarafı için de aynı şekilde. Böylece pinleriniz breadboarda girecek şekilde düzgünleşmiş olur.

Not : Voltaj regülatöründen oluşturduğunuz 5v kaynağı breadboardun kırmızı ve mavi sütunlarına bağlayınız. Böylece pratikte 5v ve GND ye her yerden erişebilirsiniz. Karışıklık olmaması için kırmızı sütuna +5v, mavi sütuna ise GND yi bağlamalısınız.

Aşağıdaki resimde de görüldüğü gibi ATMEL firması çiplerine 1.pini belirten bir ok işareti koymaktadır. Bazı üreticilerin çiplerinde böyle bir ok işareti bulunmamaktadır. Fakat bunu da tespit etmenin kolay bir yolu var. Aşağıdaki resimde mavi okun gösterdiği çentik entegrenin baş kısmını göstermektedir. Bu çentiğin soldaki ilk pini 1.pin sağdaki ilk pini ise son pin olan 28.pindir. Aşağıdaki resimde pin numaraları ve oklar görülmektedir.

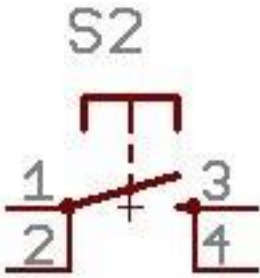


28 Pinlik DIP Kılıf Entegrelerin Pin Dizilimi

Resimlerde Atmega8 görülüyor ama sizin breadboardunuzda Atmega168 olmalı. Atmega168'in 7.pini Vcc, 20. pini Avcc(breadboard üzerinde 5v a bağlayın), 22.pini ise GND dir. Eğer anahtarı açıp devrenizi çalıştırırsanız. Atmega168 çalışacaktır fakat hiçbir iş yapmayacaktır.

Atmega238'in RESET pini(1.pin), mikrodnetleyicinin çalışması için sürekli olarak Vcc konumunda tutulmalıdır. Bu işlemi doğrudan bir 10K lık pull-up direnci ile yapabilirsiniz. Eğer çipi istediğiniz zaman resetlemek isterseniz bunun için bir butona ihtiyacınız var. Mikrodnetleyicileri resetlemek tıpkı bilgisayarları resetlemek gibidir. Mikrodnetleyiciler de resetlendiği zaman içerisindeki yazılımı tekrar baştan başlatırlar. Ve çoğu zaman mikrodnetleyicileri resetlemek gerekir. Bu sebeple breadboard üzerindeki devrenize aşağıdaki gibi bir buton bağlantısı ekleyebilirsiniz.

Bunun için önce buton bağlantısının nasıl yapıldığına değinelim. Butonlar farklı çeşitlerde olabilir. 2 pinli, 4 pinli hatta 5 pinli... Aslında temel olarak hepsinin yaptığı iş aynıdır. Butonlar sadece bulunduğu konumdaki yolları kısa devre veya açık devre yaparlar. Yani ya basılı tutulduğu an bağlantıya izin verirler, basılmadığı anda ise aradaki bağlantı olmadığı için devre tamamlanamaz. Aşağıda 4 pinli bir butonun şematik çizimi görülmektedir.

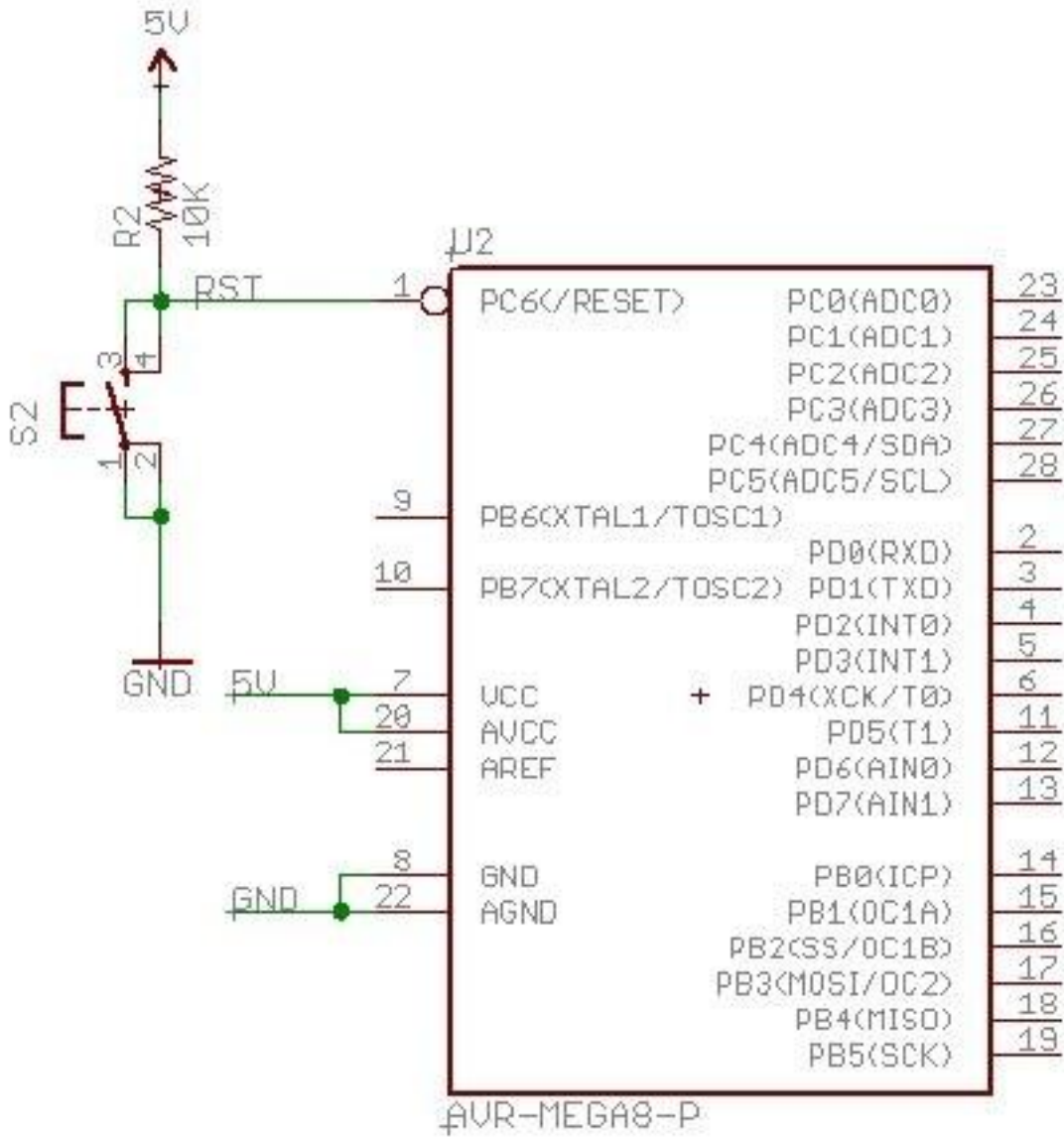


4 pinli gibi görünen bu butonun aslında yukarıda da görüldüğü gibi 2 adet pini vardır. Yani 1 ve 2. pin kendi arasında kısa devre, 3 ve 4. pin de kendi arasında kısa devre edildiği için pratikte 2 pinli bir buton kullanmış olmaktadır.



Yukarıdaki resimde görülen butonlarda 5 adet ayak bulunmaktadır. Fakat siz ortada görülen 5. ayağı dikkate almayınız. Çünkü o ayağın herhangi bir ayakla bağlantısı yoktur. Eğer hangi ayakların kendi aralarında kısa devre olduğunu bulmak isterseniz (ki bu bağlantıların doğruluğu için muhakkak yapılması gereken bir işlemdir) AVO metreniz ile kısa devre kontrolü yaparak bu işlemi doğrulayabilirsiniz. AVO metreniz bip sesi çıkardığı anda ölçtüğünüz iki ayak kısa devredir.

Aşağıda, bu butonun Atmega8'e bağlantı şeması verilmiştir.

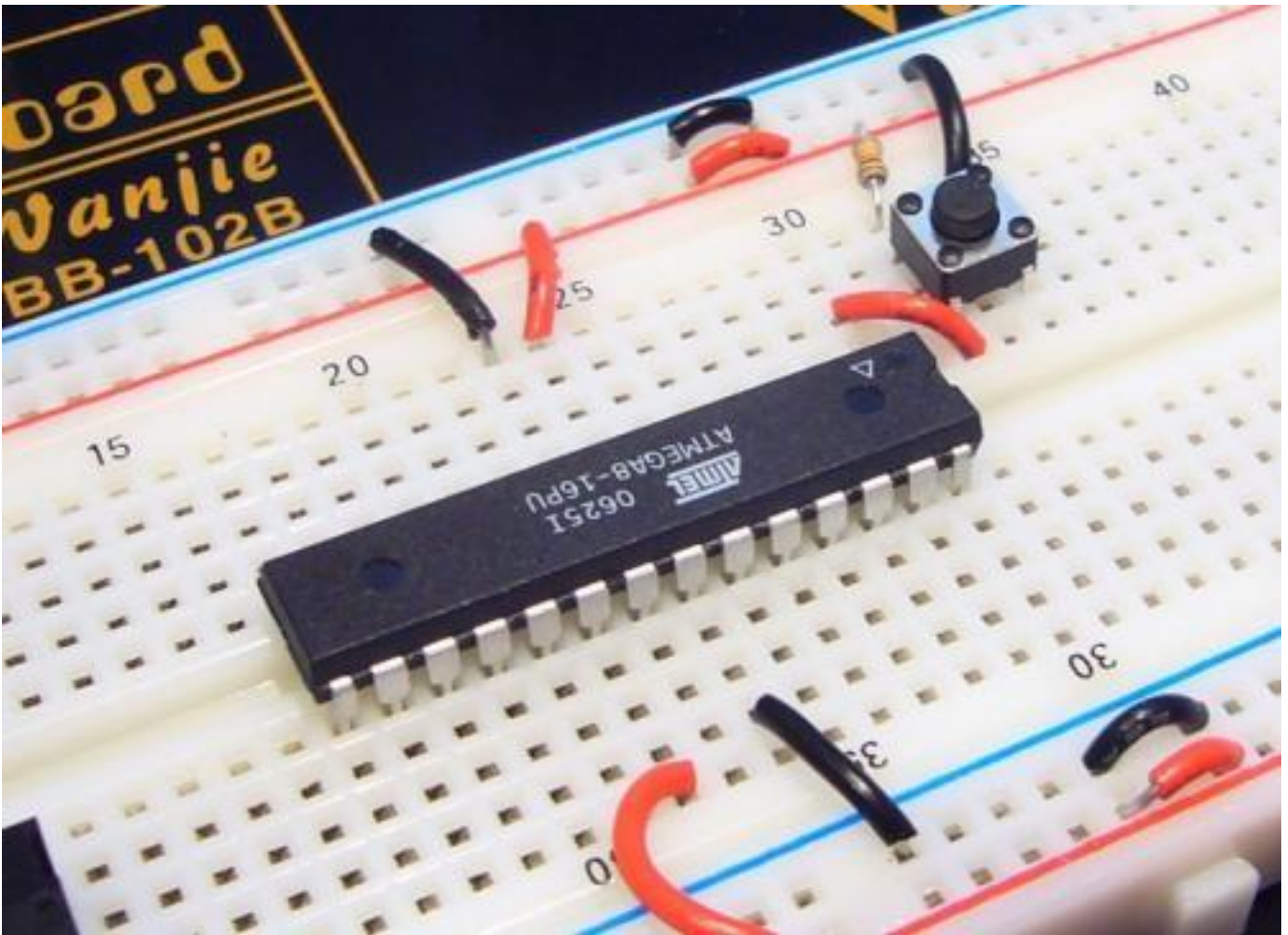


Yukarıdaki şemada da görüldüğü gibi RESET pininin sürekli olarak 5v konumunda tutulması, mikrodenetleyicinin çalışmasını sağlıyor. Eğer S2 butonuna basarsanız bu pin GND konumunda olacağından mikrodenetleyici RESET durumunda olacaktır ve çalışmayacaktır. Butonu bıraktığınız anda içerisinde yüklü olan yazılım, en başından başlayacaktır.

Aşağıda Atmega168'in pin diyagramı yer almaktadır. Atmega8 ile aradaki farklılıkları görebilirsiniz. Pin dizilimi aynı ama bazı pinlerde farklı özellikler bulunuyor.

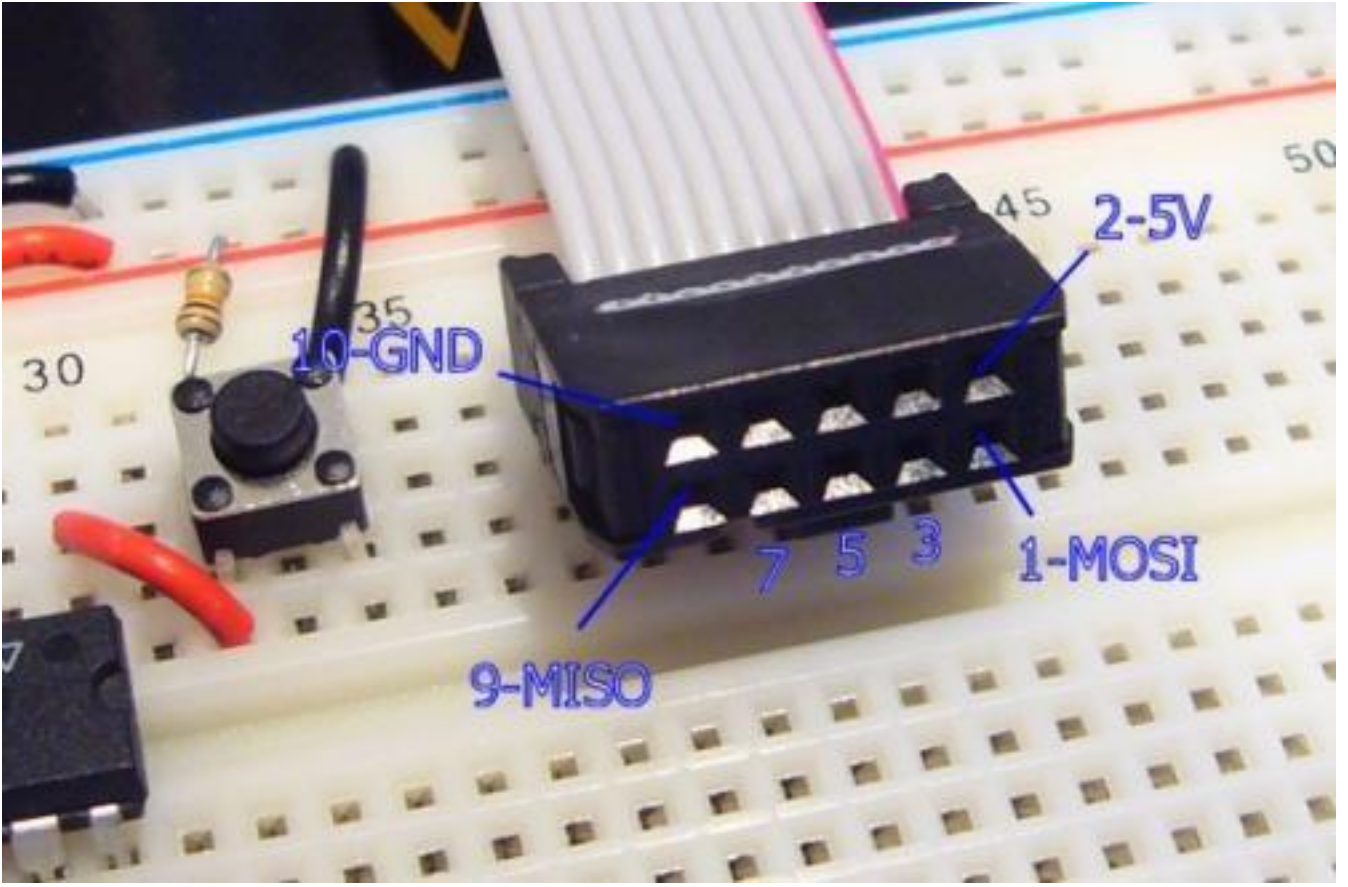
(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

Atmega168 Pin Dizilimi



ATmega8 RESET Butonu Bağlanmış Hali(Aynı şema ATmega168 ve ATmega328 için de geçerlidir)

Mikrodenetleyici üreten her firma ürettiği mikrodenetleyicileri programlamak için farklı sistemler geliştirmişlerdir. ATMEL, ISP(In System Programming) adında bir protokolü, mikrodenetleyicileri için geliştirmiştir. Bu işlem 6 pin üzerinden harici bir donanım kullanılarak yapılmaktadır. Bu donanımın(programlayıcı) kullanımı çok kolay ve yapılışı da çok ucuzdur.



Kırmızı şerit 1.pini Göstermektedir

Atmega168'i programlamak için aşağıdaki adımlar izlenmelidir.

1. AVR-GCC derleyicisi ile C kodları yazılmalı ve mikrodenetleyiciye yüklenecek HEX dosyası oluşturulmalı.
2. AVR-GCC derleyicisi Windows altında çalışan WinAVR programına entegre edilebilir.
3. Oluşan HEX kodu ISP pinleri aracılığıyla Atmega168'e yüklenmeli.
4. Hem seri hem de paralel port programlayıcılar AVR serisi işlemcilere program yüklemek için kullanılabilir
5. Mikrodenetleyici, içerisine yüklenen HEX kodunu bir kez resetlendikten sonra çalıştırmaya başlayacaktır.

C Derleyicisi Nedir ?

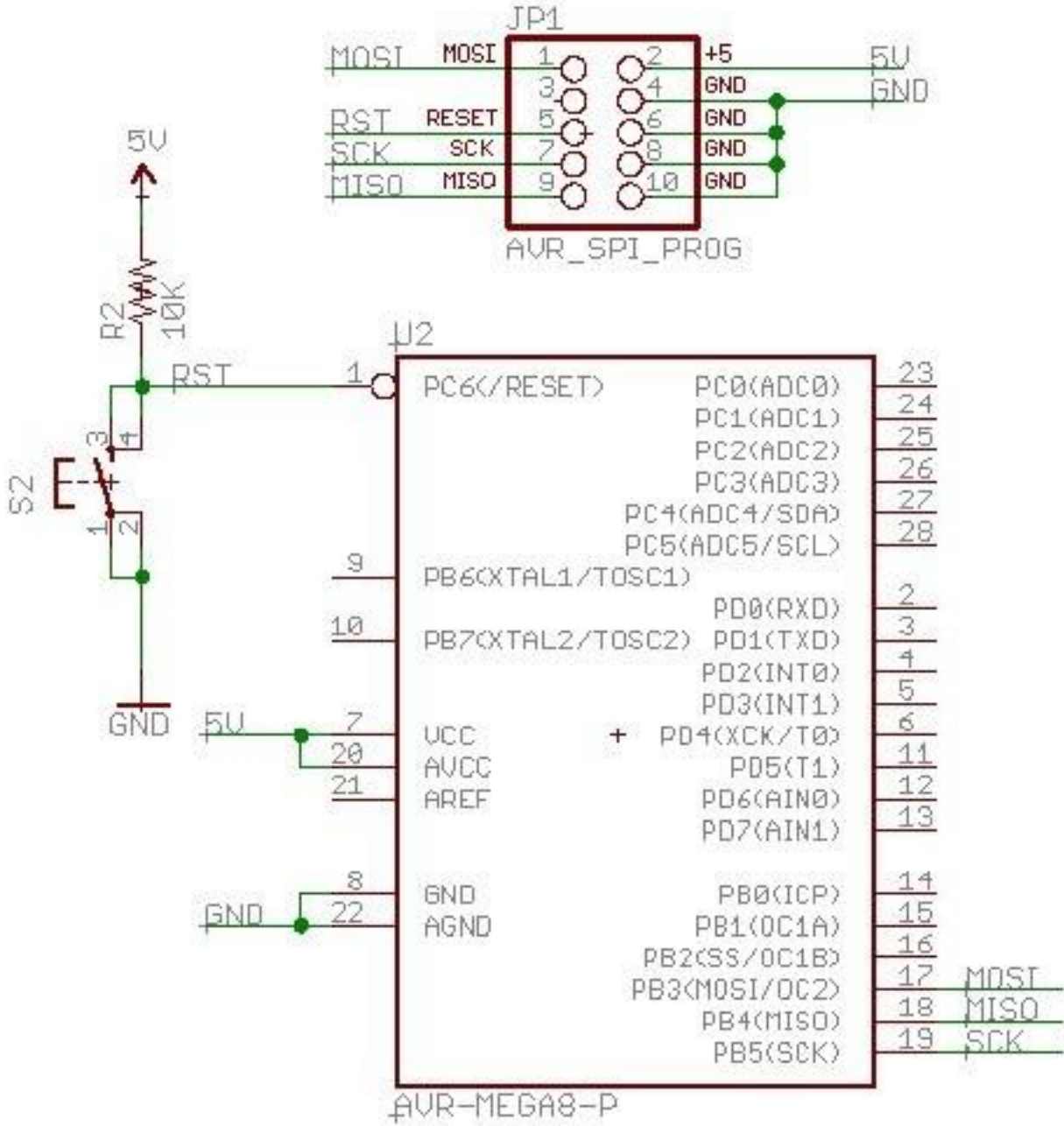
C derleyicisi, C dilinde yazılan programları mikrodenetleyicinin anlayabileceği kod olan HEX formatına dönüştüren bir araçtır. C dilini BASIC ten daha esnek ve assemblyden daha anlaşılır olduğu için tercih ediyoruz.

HEX Dosyası Nedir ?

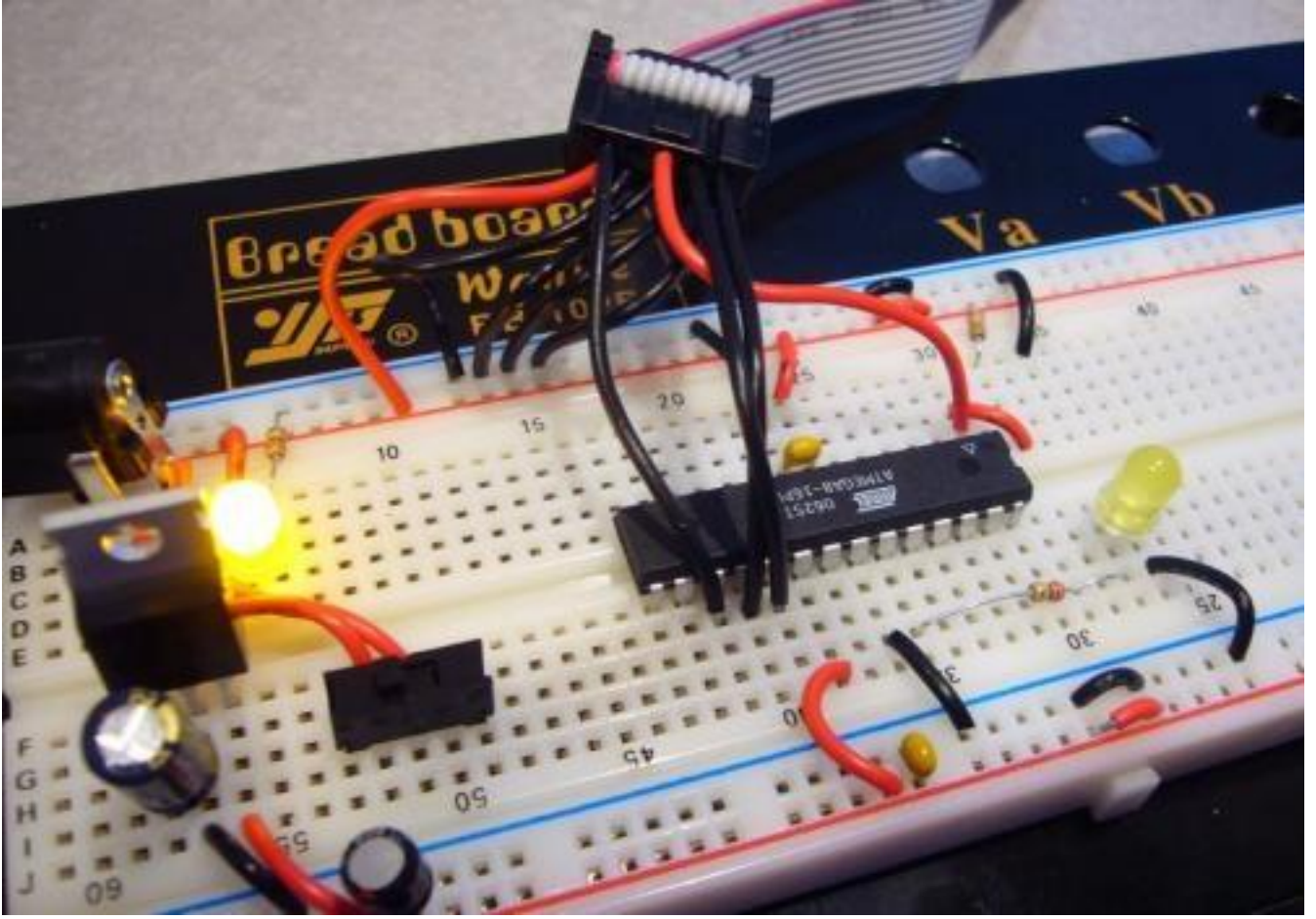
HEX dosyası, tamamen hexadesimal sayıları içeren bir dosyadır. Bu dosya, mikrodenetleyicinin anlayacağı hexadesimal sayıları temsil etmektedir. Bu dosya programlayıcı üzerinden Atmega168 içerisine yüklenecek olan dosyadır.

Bilgisayarda C kodlarınızı yazabilmek için WinAVR programını yüklemeniz gerekmektedir. Ben bu yazıda size basit bir LED yak-söndür uygulaması vereceğim. Eğer bu kodu mikrodenetleyicinize yükleyip çalıştığını görürseniz her şey eksiksiz ve doğru çalışıyor demektir.

blink_1MHZ.hex dosyasını Atmega168'e yüklemek için aşağıda verilen bağlantıyı breadboardunuza kurmanız gerekmektedir.



Aşağıda, yukarıda verilen şemanın breadboard üzerine uyarlanmış halini görmektesiniz. Breadboard üzerindeki gibi kısa kablolar kesmek uzun süreli bir çözüm değildir fakat çabuk ve pratik olması açısından bu yöntem tercih edildi. Siz kendinize pin headerlı daha kullanışlı bir programlama adaptörü tasarlayabilirsiniz.



AVR-PG2 Paralel Port Programlayıcı. Vcc ve GND arasındaki dekaplaj kapasitörleri 5v ve GND arasındaki oluşan paraziti bastırıp mikrodenetleyiciye gitmesini önlemektedir.

Şimdi tekrar uygulamamıza dönecek olursak; Bir LED yak-söndür uygulaması yapacağız ve bunun için herhangi bir GPIO pinini kullanmamız gerekiyor.

C kodunu derlemek için Programmer's Notepad'i açıp blink_1MHZ.c dosyasını editör içerisine dahil ettikten sonra Tools -> Make All a tıklayıp .c uzantılı dosyayı derlemelisiniz. Bu işlem komut isteminde şu kodu yazmakla aynıdır.

```
C:\Code\Blink>make all
```

Windows Komut İstemini kullanmaktansa WinAVR Programmer's Notepad'i kullanmak daha kolaydır. Çalışmalarınızı Programmer's Notepad üzerinden yürütmeniz size büyük ölçüde hız kazandıracaktır.

Eğer .c uzantılı dosyayı derledi ve .hex uzantılı dosyayı oluşturduysanız sıra bu dosyayı AVR'nin içerisine yüklemeye geldi demektir. Bu işlem, breadboard üzerindeki devrenizin çalışması için son adımdır. Bunun için AVR-PG1 veya PG2 denilen ucuz bir programlayıcı kullanacağız. Bu programlayıcılar çalışabilmek için gereken gücü devre üzerinden alırlar. Bunun için breadboard üzerindeki devrenizin anahtarını açıp sarı LEDin yandığından emin olmalısınız.

Eğer AVR-PG1(Seri Port Programlayıcı) kullanıyorsanız Makefile dosyasını aşağıdaki gibi değiştirmeniz gerekmektedir.

```
#AVRDUDE_PROGRAMMER = stk200
```

```
AVRDUDE_PROGRAMMER = ponyser
```

```
#com1 = serial port. Use lpt1 to connect to parallel port.
```

```
#AVRDUDE_PORT = lpt1
```

```
AVRDUDE_PORT = COM1
```

Eğer AVR-PG2(Paralel Port Programlayıcı) kullanıyorsanız Makefile dosyasını aşağıdaki gibi değiştirmeniz gerekmektedir.

```
#AVRDUDE_PROGRAMMER = stk200
```

```
AVRDUDE_PROGRAMMER = ponyser
```

```
#com1 = serial port. Use lpt1 to connect to parallel port.
```

```
AVRDUDE_PORT = lpt1
```

```
#AVRDUDE_PORT = COM1
```

Not : Burada '#' işareti o satırın işlenmeyeceği anlamına gelmektedir.

Eğer aşağıdaki gibi bir hata alırsanız

```
can't open device "giveio"
```

Bu sayfayı okuyun. Giveio.sys dosyasını C:\WinAVR\bin dizininden C:\Windows dizinine kopyalamanız ve sonra da install_giveio.bat dosyasını Komut İsteminden çalıştırmanız gerekmektedir.

Karşınıza Çıkabilecek Problemler

1. ISP Programlama konnektörünün tüm pinlerinin doğru olduğundan emin olun.

2. Breadboard üzerinde regüle ettiğiniz 5v un sağlamlığından ve her yere ulaştığından emin olun.
3. Atmega168'in Vcc ve GND pinlerini bağladığınızdan emin olun.
4. Doğru COM veya LPT potunu kullandığınızdan emin olun.

Bu ders için gereken malzemelerin tamamını [buradan](#) edinebilirsiniz.

Not:Buradaki karışıklık için özür diliyorum. Çünkü programlayacağımız mikrodnetleyici Atmega168. Fakat resimlerde Atmega8 yer alıyor.

Kaynaklar

- <http://piconomic.co.za/fwlib/index.html>
- http://www.salvitti.it/geo/sequencer/dev_tools/tutorial/GNU_C_Tutorial.html
- <http://palmavr.sourceforge.net/cgi-bin/fc.cgi>

Ders 3 – Osilatörler

Bir önceki derste Atmega328 ile basit bir LED yak-söndür uygulaması yapmıştık. Şimdi bu LED'i daha hızlı bir şekilde yakıp söndürelim.

İlk uygulamada Atmega328 içerisine çalışan, LED yakıp söndüren bir kod yüklemiştik. Fakat bu kodun nasıl ve ne kadar hızda çalıştığı hakkında pek detaya değinmemiştik.

Herhangi bir mikrodnetleyici veya işlemci içerisine yüklenen kodların işletilip çalıştırılabilmesi için muhakkak bir işaret kaynağına ihtiyaç duymaktadır. Eğer bu kaynak olmazsa, mikrodnetleyici, içerisindeki hiçbir kodu işletemez. Ve kaynak düzensiz bir dalga olsa bile işlemci çalışır fakat hızı hakkında kesin bir şey söylenemez. Genellikle kodların hatasız çalışabilmesi için kesin bir frekans üreten bir işaret kaynağına ihtiyaç duyarız.

Atmega168'de bu işaret kaynağı, birkaç farklı şekilde sağlanabilir.

1. Harici RC Osilatör : Basit ve ucuz bir yöntemdir. Bir direnç ve bir kondansatörle yapılabilir. Kondansatörün şarj/deşarj oranı bu kaynağın frekansını belirler.
2. Dahili RC Osilatör : Mikrodnetleyici içerisinde bulunan yine bir direnç ve bir kondansatörle yapılan işaret kaynağıdır. Ttek eksik yanı ise : istenilen frekansın tam ve kesin olarak üretilmemesidir.
3. Harici Osilatör : Harici bir quartz kristalinin mikrodnetleyicinin OSC pinlerine bağlanması ile elde edilir.
4. Harici Resonatör : Harici osilatör gibidir. Fakat toleransı fazladır.
5. Harici Clock Sinyali : Bir osilatör kullanmak yerine birkaç farklı cihazın aynı frekansta çalışması istenen durumlarda kullanılır.

Dahili RC Osilatör

Atmel firmasının ürettiği AVR serisi mikrodenetleyicilerde 1MHz ? 8MHz arası frekans üretebilen dahili RC osilatör bulunmaktadır. AVR serisi mikrodenetleyiciler, 1 komutu bir clock palsinde işleyebilirler. Bu da demek oluyor ki; 1MHz lik bir osilatörde 1MIPS(Million Instruction Per Second) hıza 12MHz lik bir osilatörde 12MIPS hıza çıkılabilmektedir. Fakat bu dahili RC osilatörün bir toleransı vardır. Dahili osilatör frekansı 8MHz iken tolerans +/- %5 olmaktadır. Yani $1.000.000 * 1.05 = 1.050.000$ IPS veya $1.000.000 * 0.95 = 950.000$ IPS bu tolerans değeri aslında küçük gibi görünse de işlemci hızı arttığı zaman istenilen dengenin bozulması kaçınılmaz olacaktır. Bu nedenle dahili RC osilatörünün genellikle hassas zamanlama gerektiren uygulamalarda kullanılmaması tavsiye edilir.

Harici Osilatör

Bu osilatör çeşidi aşağıda görüldüğü gibi yaygın olan tiptedir.



Quartz kristal osilatörler farklı çeşitlerde ve frekanslarda olabilirler. En yaygın; 20MHz, 16MHz, 10MHz, 4MHz gibi değerlerde kullanılırlar. Kristal değerleri küsuratlı da olabilir. Mesela 14.7456MHz, 9.216MHz, 32.768kHz. Bu küsuratlı değerler seri iletişim, zamanlama gibi uygulamalarda olmazsa olmazlardandır. Örnek vermek gerekirse ; 9600 baud hızında bir seri iletişim yapmamız gerekiyor ve 9.216MHz'lik bir kristal kullanıyoruz. $9.216.000 / 960 = 9600$. Bu değer 16MHz gibi tam değerli bir kristal kullanıldığında tam sayı çıkamayacağı için seri iletişimde küçük de olsa bazı sorunlar oluşabilmektedir.

Metal kılıfın içerisinde yer alan quartz kristal parçası kesin bir frekans değeri belirlenip ona göre farklı boyutlarda kesilmektedir. Bu büyüklüğün sebebi quartz kristalinin büyüklüğü ile titreşim frekansının arasındaki ilişkiden kaynaklanmaktadır. Quartz kristalleri genellikle +/- %20ppm toleransta üretilirler. Yani +/- %20 parts per million anlamına gelmektedir. Bunu bir örnekle açıklayacak olursak; 16MHz lik bir kristalde hata payı sonucu frekans değeri 16.000.020 MHz veya 15.999.980 MHz olmaktadır. Bu, +/- %0.00000125 gibi bir tolerans değerine denk gelmektedir. Sonuç olarak bir quartz kristali, bir RC osilatörden 4 milyon kat daha kesin bir frekans üretmektedir.

Teknik Detay : Bir kristal ile bir osilatör aynı şey değildir. Kristaller ucuzdur fakat osilatörler biraz daha pahalıdır(\$2-\$4 arası). Aradaki fark ise; osilatörler istenilen frekansı doğrudan üretirler. Kristal ise pasif bir devre elemanıdır ve bir osilatör gibi davranabilmesi için kapasitörlere ihtiyaç duyarlar. Genellikle kristalin

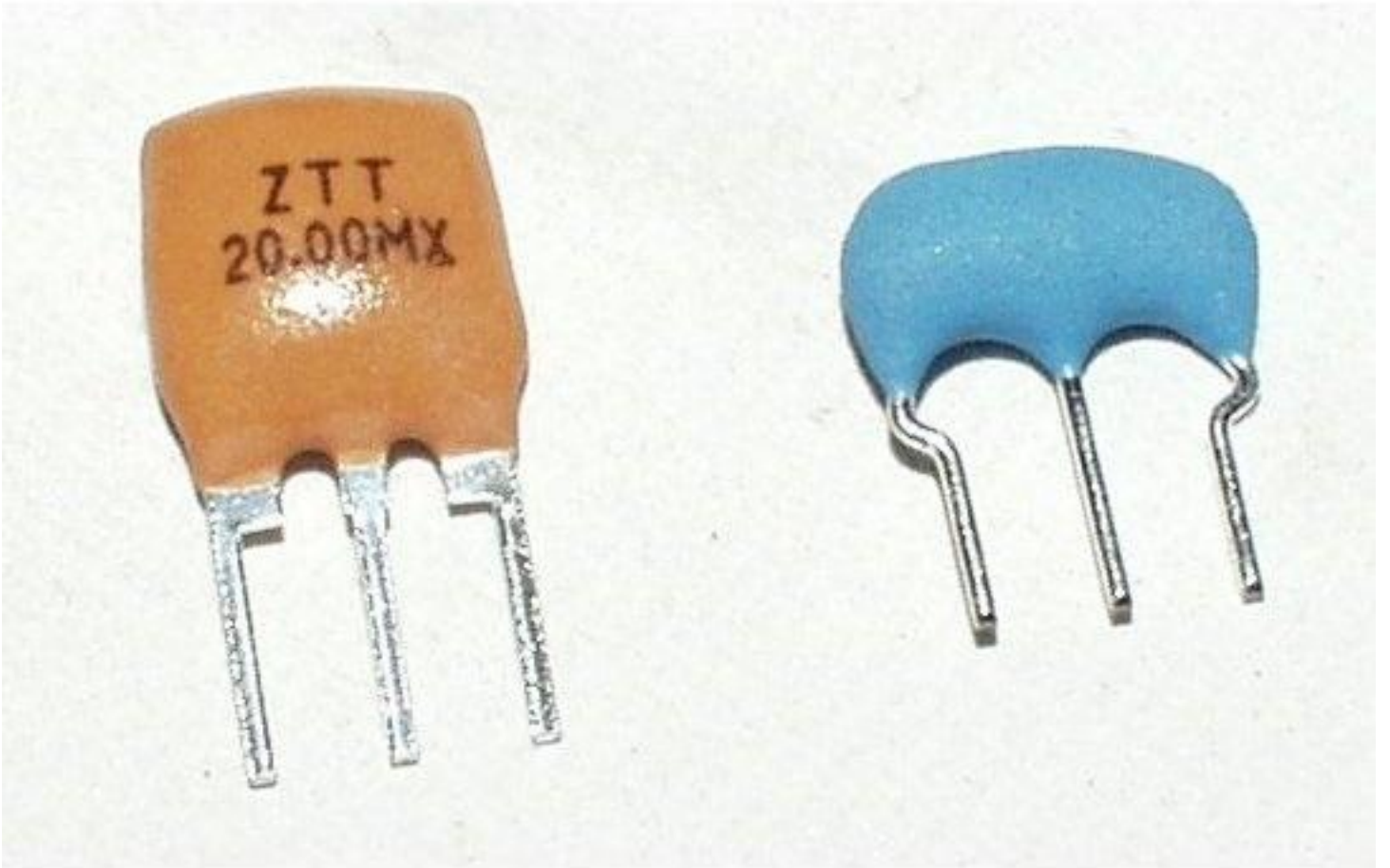
iki ucuna bağlanmış kapasitörler bir bütün halinde mikrodenetleyiciye bağlandığı zaman gereken osilatör devresi sağlanmış olur.

Kristallerin Bazı Dezavantajları

1. PCB üzerinde resonatörlerden daha fazla yer kaplarlar.
2. Kristallerin çalışması için gereken ekstra kapasitörler PCB üzerinde resonatörlerden daha fazla yer kaplarlar.
3. Kristaller, resonatörlerden biraz daha pahalıdır.

Harici Resonatör

Harici resonatörler aşağıdaki şekilde görüldüğü gibidir.

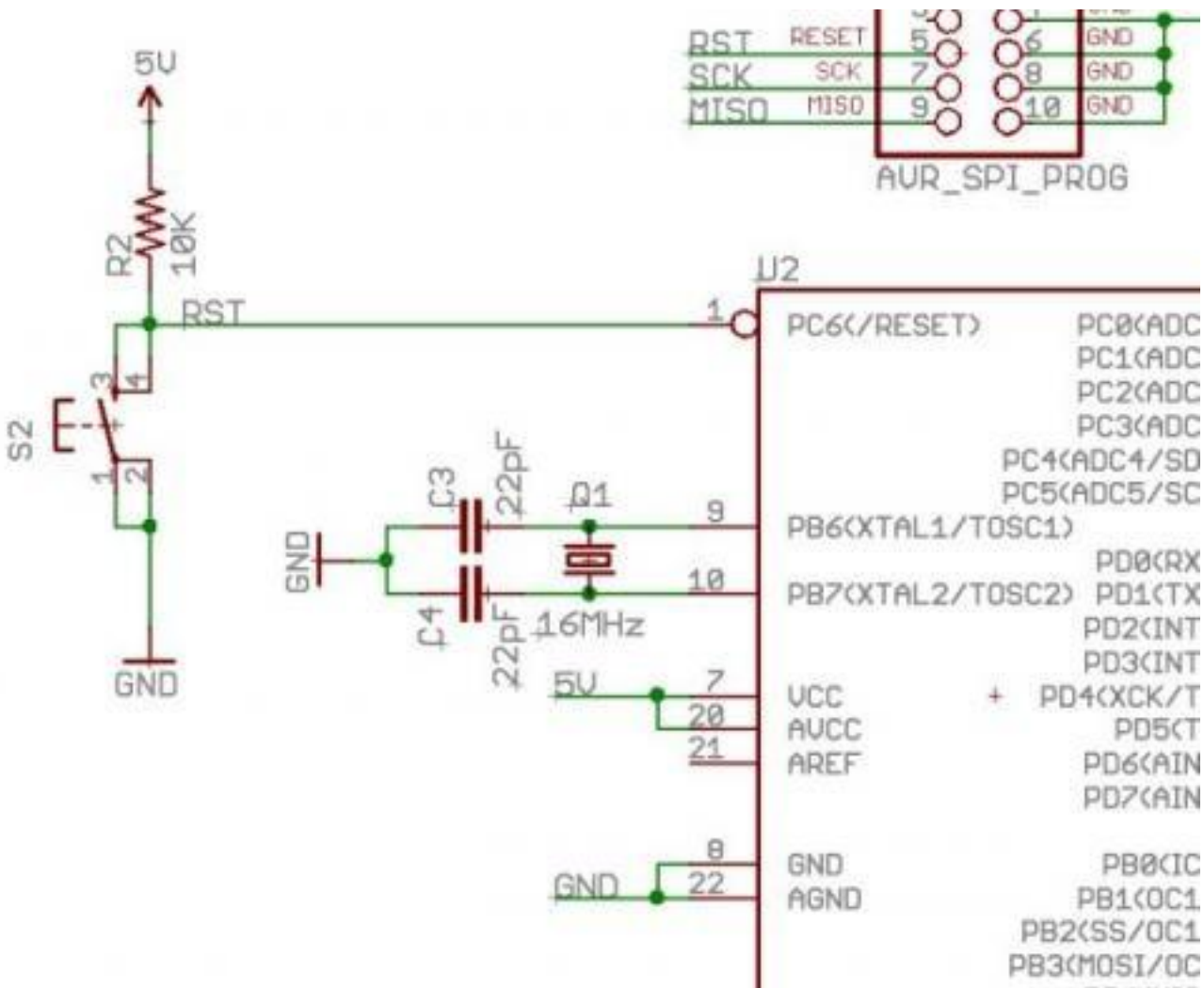


Resonatör, belirlenen frekansı üretecek şekilde boyutlandırılmış seramik parçasıdır. Toleransları +/-%0.5 olduğundan dahili RC osilatörlerden 10 kat daha kesin frekans değeri üretirler. Resonatörler içerisinde kapasitör de bulundurdıkları için PCB'de daha az yer kaplarlar ve bu sebeple daha kullanışlıdır. Ve kristallerin sıcaklıktan olumsuz yönde etkilendikleri düşünülürse bazı durumlarda resonatör kullanmak daha uygun olmaktadır.

Daha önceden de bahsedildiği gibi Atmega328 içerisinde 1MHz lik dahili osilatör bulunmaktadır. Fakat bunu kolayca ayarlayıp kullanabilmekteyiz. Şimdi Atmega168'e 16MHz'lik harici kristal osilatörün nasıl bağlandığını inceleyelim.

Atmega168'in dahili RC osilatörünü yüksek frekansta çalıştırmanın yan etkilerinden biri de Vcc gerilim 2.8v veya 3.3v gibi bir değer olursa işlemcinin bu Vcc geriliminde istenen frekansta çalışmayacağıdır. Fakat bizim Vcc kaynağımız 5v olduğu için bu durum sıkıntı teşkil etmeyecektir.

Atmega328'e 16MHz lik harici bir kristal PB6 ve PB7 pinlerinden bağlanmaktadır.



Kristalin iki ayağını da 22pF lık kondansatörler üzerinden GND ye bağladığımızda Atmega168 için harici kristal işlemini tamamlamış oluyoruz. Şimdi bu kristali yazılımsal olarak nasıl seçeceğimizi öğrenelim.

Fuse Bitleri

Atmel AVR'lerin başlangıç için en kafa karıştırıcı kısmı şüphesi Fuse bitleridir. Bu işlem PIC gibi bir mikrodenetleyici ile kıyaslandığında başlangıç için oldukça zor bir işlemdir. Atmega328 için 2 bytelık Fuse saklayıcıları bulunmaktadır. Eğer Atmega168 datasheet ini indirmediyseniz buradan indirebilirsiniz.

Atmega168'in datasheeti 376 sayfa fakat Fuse bitlerini ayarlamak için bu datasheetin tamamını bilmek zorunda değilsiniz.

Blink_1MHz.hex programında dahili RC osilatör 1MHz olarak ayarlanmıştır. Fuse bitlerini değiştirerek dahili RC osilatör frekansını 8MHz yaparsak LED daha hızlı yanıp sönmeye başlayacaktır. Aşağıda RC osilatörün nasıl seçildiği görülmektedir.

Table 8-1. Device Clocking Options Select⁽¹⁾

Device Clocking Option	CKSEL3..0
Low Power Crystal Oscillator	1111 - 1000
Full Swing Crystal Oscillator	0111 - 0110
Low Frequency Crystal Oscillator	0101 - 0100
Internal 128 kHz RC Oscillator	0011
Calibrated Internal RC Oscillator	0010
External Clock	0000
Reserved	0001

Note: 1. For all fuses "1" means unprogrammed while "0" means programmed.

Dahili RC osilatörünün frekansı ön tanımlı olarak 8Mhz dir. Fakat blink_1MHz.HEX programında bu ayar değiştirildiği için biz tekrar 8MHZ olarak ayarlayacağız. Bunun için CKDIV bitini set etmemiz gerekir.

Table 27-7. Fuse Low Byte

Low Fuse Byte	Bit No	Description	Default Value
CKDIV8 ⁽⁴⁾	7	Divide clock by 8	0 (programmed)
CKOUT ⁽³⁾	6	Clock output	1 (unprogrammed)
SUT1	5	Select start-up time	1 (unprogrammed) ⁽¹⁾
SUT0	4	Select start-up time	0 (programmed) ⁽¹⁾
CKSEL3	3	Select Clock source	0 (programmed) ⁽²⁾
CKSEL2	2	Select Clock source	0 (programmed) ⁽²⁾
CKSEL1	1	Select Clock source	1 (unprogrammed) ⁽²⁾
CKSEL0	0	Select Clock source	0 (programmed) ⁽²⁾

Default Fuse Byte Değeri(Low) : 0b 0110 0010

Yeni Fuse Byte Değeri(Low) : 0b 1110 0010

Bu ayarı yaptıktan sonra Atmega168, 8 MHz hızda çalışmaktadır. Bu da PC0 pinine bağlı LED'in daha hızlı yanıp sönmelerini sağlayacaktır.

Bu ayarlar birkaç farklı şekilde yapılabilir. Diğer bir yol ise avrdu programını kullanarak bu işlemi komut istemi'nde yapmaktır. Bunun için komut istemi'ne aşağıdaki satırları yazmanız gerekmektedir.

PG1(Seri Programlayıcı)

- avrdu -p m168 -P COM1 -c ponyser -U lfuse:r:-:h -U hfuse:r:-:h

PG2(Paralel Programlayıcı)

- `avrdude -p m168 -P lpt1 -c stk200 -U lfuse:r:-:h -U hfuse:r:-:h`

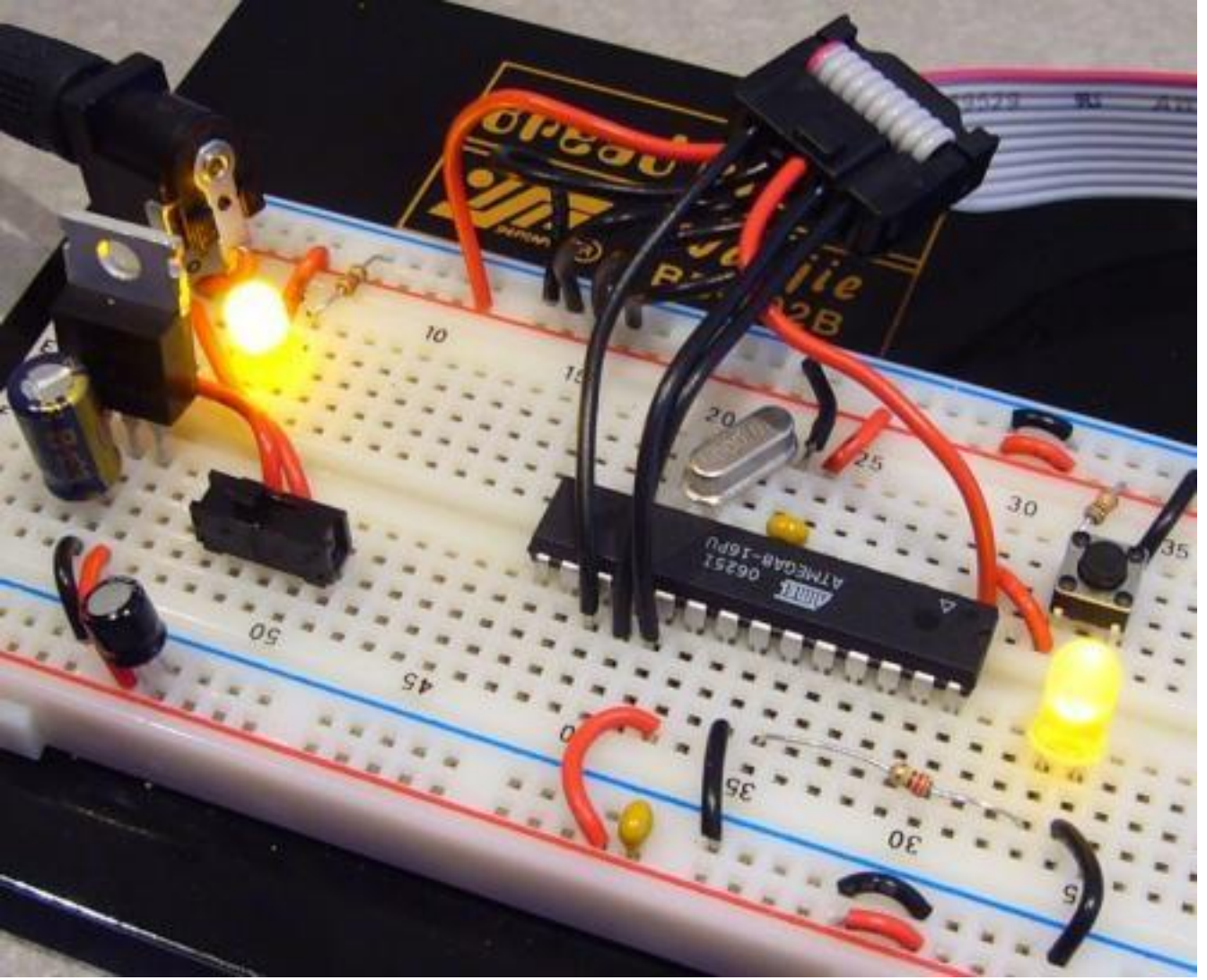
PG2 programlayıcı kullanıp, dahili RC osilatörü 8MHz de çalıştırmak için;

- `avrdude -p m168 -P lpt1 -c stk200 -U lfuse:w:0xE2:m`

Komut İstemi'nde bu satırları yazıp Atmega168'i programladığınızda işlemci 8MHz'lik osilatörle çalışacaktır.

Atmega168'e önceden de bahsedildiği gibi harici kristal de bağlanabilmektedir. Bunun için Fuse bitlerini aşağıdaki gibi ayarlamak gerekir.

- `CKDIV8 = 1`
- `CKOUT = 1`
- `SUT1 = 1`
- `SUT0 = 0`
- `CKSEL3 = 0`
- `CKSEL2 = 1`
- `CKSEL1 = 1`
- `CKSEL0 = 0`



Avrude programında doğrudan aşağıdaki komut satırı kullanılabilir.

16MHz lik harici osilatör için;

- `avrdude -p m168 -P lpt1 -c stk200 -U lfuse :w:0xE6:m`

Atmega168'i bu ayarla programlayıp, breadboard üzerindeki devrenize 16MHz kristal ve 22pF kapasitörleri eklerseniz LED'in çok hızlı yanıp söndüğünü görürsünüz. Atmega168'in maksimum hızı 20Mhz(20 MIPS) tir. İşlemciyi overclock edip bunu 30MIPS ya da 40MIPS de yapabilirsiniz. Fakat endüstriyel uygulamalarda önerilen hızın üzerine çıkmamanız önerilir.

Konuyu daha iyi kavramak açısından aşağıdaki ölçümleri yapıp sonuçlarını gözlemlemeniz önerilir.

Bu ölçümlerin hepsinde Atmega168'in ne kadar akım harcadığını gözlemleyiniz.

- pAtmega168'in reset anında = ?
- p1MHz'lik dahili RC osiltaör kullanılırken = ?
- p8MHz'lik dahili RC osilatör kullanılırken = ?
- p16MHz'lik harici osilatör kullanılırken = ?

Fuse bitlerini ayarlamak için aşağıdaki linkteki programı kullanabilirsiniz.

<http://palmavr.sourceforge.net/cgi-bin/fc.cgi>

Yukarıdaki ölçümleri yaparken, özellikle mikrodnetleyiciyi yüksek frekansta çalıştırdığınız zaman voltaj regülatörünüzün biraz ısındığını göreceksiniz. Çünkü 5v'luk kaynaktan normalden daha fazla bir güç ihtiyacı olmaktadır. Bu gücün büyük kısmı da LM7805 üzerinde ısıya dönüşmektedir. Örnek verecek olursak;

50mA akım ihtiyacımız olsun ve bize ne kadar güç gerekiyor hesaplayalım;

$$(9-5) \times 0.050 = 0.2W = 200mW$$

Bu değer LM7805 i birazcık zorlamakta ve ısıtmaktadır.

Bir de 1A için hesaplama yapalım.

$$(9-5) \times 1.000 = 4W$$

Görüldüğü gibi 1Amperlik bir akım ihtiyacı sonucunda ne kadar fazla güç harcanmaktadır. Ve dolayısıyla sıcaklık da buna bağlı olarak artacaktır. Bu işlem sonucunda voltaj regülatörünüz muhtemelen thermal shut-down moduna geçecektir(eğer destekliyorsa). Yani fazla sıcaklıktan dolayı kendini kapatacaktır. Ve breadboard üzerindeki LED de sönecektir. Eğer böyle bir özellik yoksa voltaj regülatörünüz büyük ölçüde hasar görecektir belki de yanacaktır.

Bu sebepten dolayı eğer bağlantılarınızda herhangi bir kısa devre olmuşsa hemen anahtarı kapatmanız gerekmektedir. Çünkü voltaj regülatörünün yanması ve gerilimi regüle edememesi sonucunda diğer tüm dijital devreleriniz zarar görecektir.

Bu dersin tüm materyallerini buradan temin edebilirsiniz.

Not:Buradaki karışıklık için özür diliyorum. Çünkü programlayacağımız mikrodnetleyici Atmega168. Fakat resimlerde Atmega8 yer alıyor.

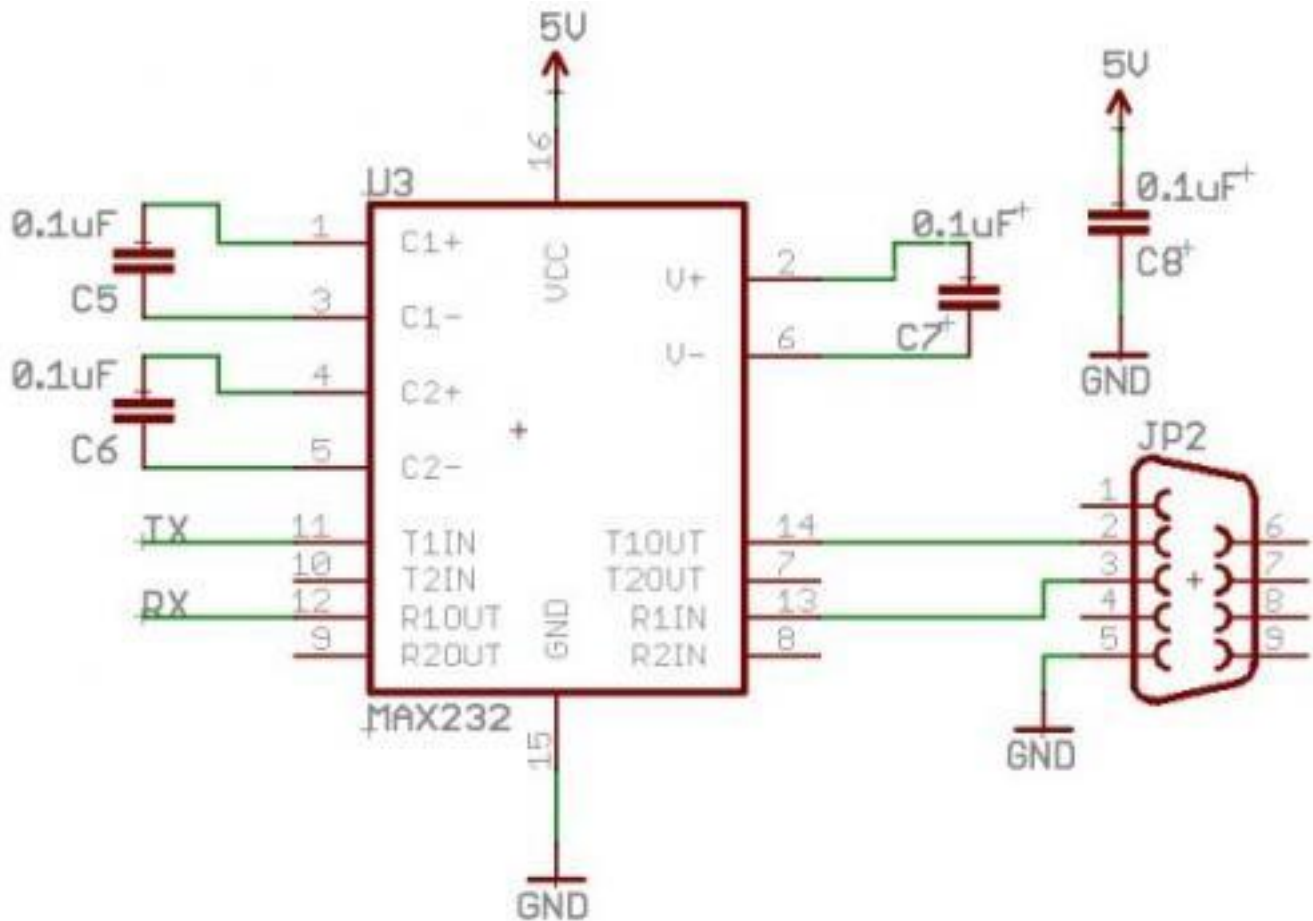
Ders 4 - UART ve Seri Haberleşme

Atmega168, içerisinde dahili olarak birçok çevrebirim barındırmaktadır. Bunlardan biri de UART modülüdür. Bu yazıda Atmega168'in UART modülü kullanılarak bilgisayar ile nasıl seri haberleşme yapıldığını inceleyeceğiz.

Öncelikle RS232'nin tarihine kısaca değinelim;

RS232 ile iletişim, ilk başlarda bilgisayarların kendi aralarında haberleşmesi için kullanılan bir yöntemdi. Bugün ise kullanımı yaygınlaşmış olan bu yöntem, yazıcılar, tarayıcılar, jostickler gibi bilgisayara bağlanabilen donanımlar tarafından tercih edilmektedir. Fakat mikrodenetleyiciler gibi düşük gerilimle çalışan devre elemanlarına RS232 portunu doğrudan bağlamak uygun değildir. Çünkü RS232 portu +/-12v arası gerilim üretmektedir. Ve Atmega168 ise TTL(5v) bir devre elemanı olduğundan RS232 portu ile mikrodenetleyici arasında +/-12v u TTL seviyeye indirgeyen bir sistem kullanmak gerekmektedir. Bu iş için, kullanımı çok yaygın olan MAX232 entegresini kullanacağız.

MAX232 entegresi Maxim firması tarafından üretilmiş olan RS232 portundaki +/-12v gerilimi lojik devre elemanları için 5v seviyesine indirgeyen bir entegredir. Ve bu sayede mikrodenetleyici, yüksek gerilimden dolayı zarar görmeden seri haberleşme sağlanmış olur. Aşağıda bu entegre için breadboardunuza eklemeniz gereken bağlantılar verilmiştir.

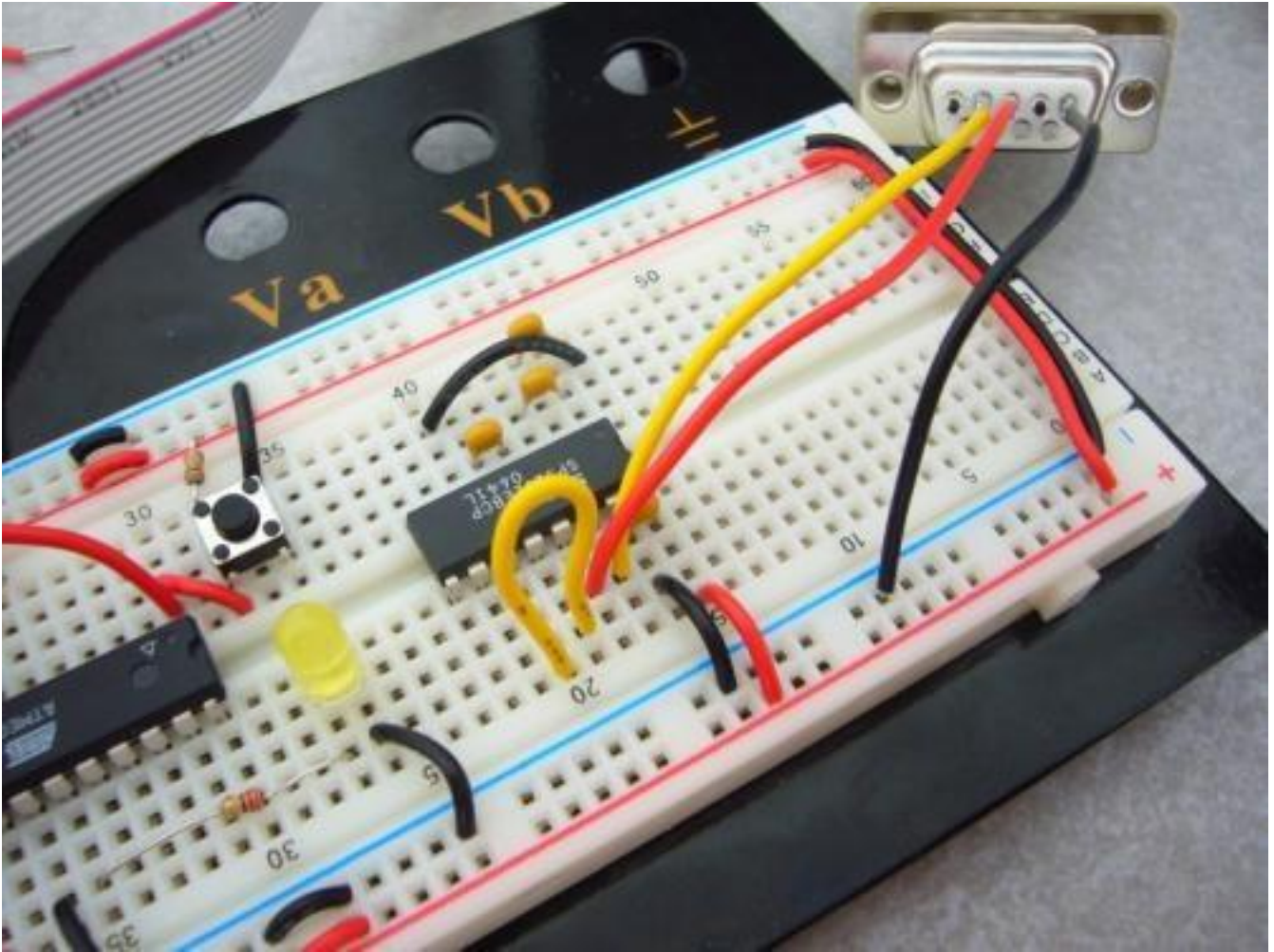


MAX232 Devre Şeması - Eagle schematic / PDF

MAX232 entegresinin sağlıklı çalışması için 0.1uF(c5,c6,c7) kondansatörlere ihtiyaç vardır(charge_pumps). İlerideki c8 kondansatörü ise dekuplaj kondansatörüdür. Bu kondansatör çıkışı filtre edip, daha temiz bir 5v gerilim elde etmemizi sağlar. Bu kondansatör, Vcc ve GND ye ne kadar yakın olursa filtre etme vazifesini

o kadar iyi yapacaktır. Bu sebeple breadboard üzerinde bu kısmı devrenize eklerken, bu şartı da göz önünde bulundurunuz.

JP2 ise DB9 konektörüdür. Bu konektörü bilgisayarınızdan devrenize bağlamak, kabloları lehimleyebilmek için aşağıdaki gibi bir çevirici kullanmanız gerekmektedir.



MAX232 Devresinin Breadboard Üzerine Kurulmuş Hali. Tx ve Rx Uçları Birbirine Bağlı.

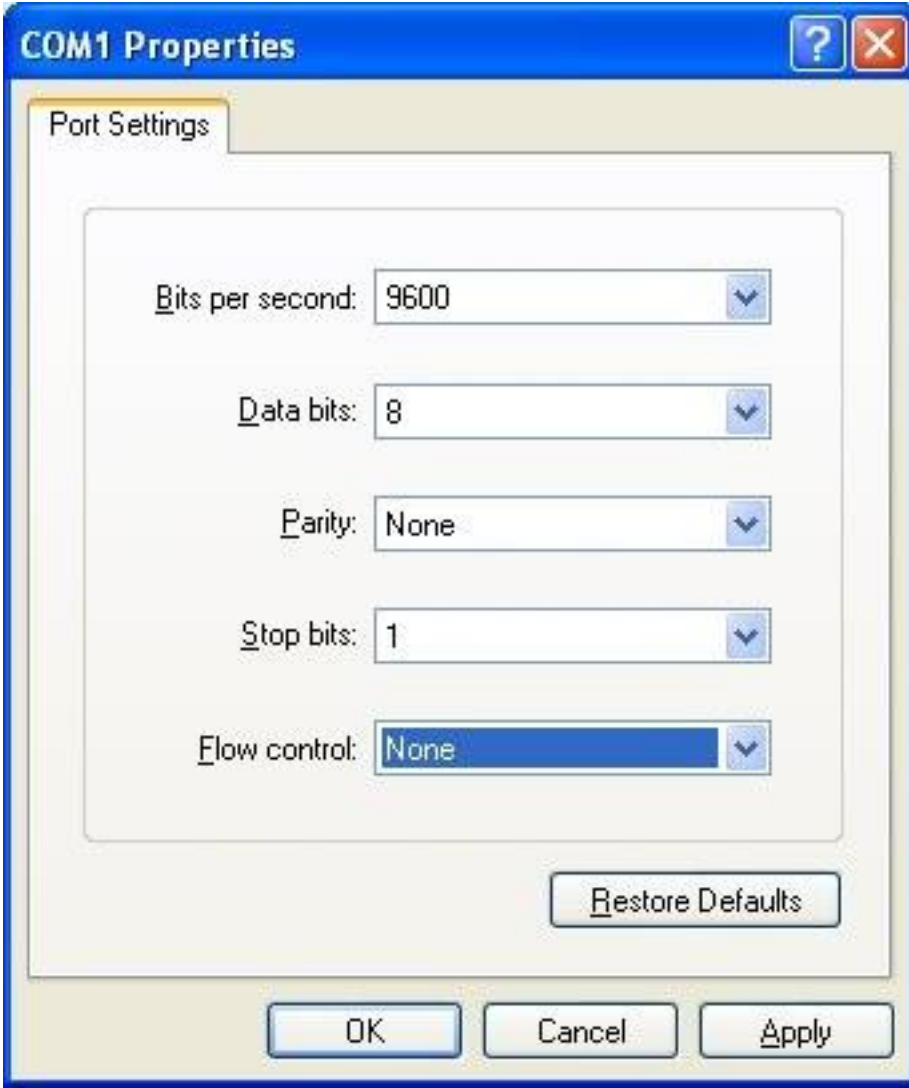
Devreyi breadboard üzerine kurduktan ve bağlantıların doğru yapıldığından emin olduktan sonra bir terminal programı aracılığı ile seri haberleşmemizi yapabiliriz. Bunun için Windows altında çalışan HyperTerminal programını kullanacağız. Eğer HyperTerminal programını kullanmak istemiyorsanız kullanıcıların da önerdiği TeraTerm programını kullanabilirsiniz.



Bağlantı ismini Girin. Herhangi bir isim girebilirsiniz.



Port Olarak COM1 Seçiliyor.



9600bps, 8-N-1, Flow Control devre dışı

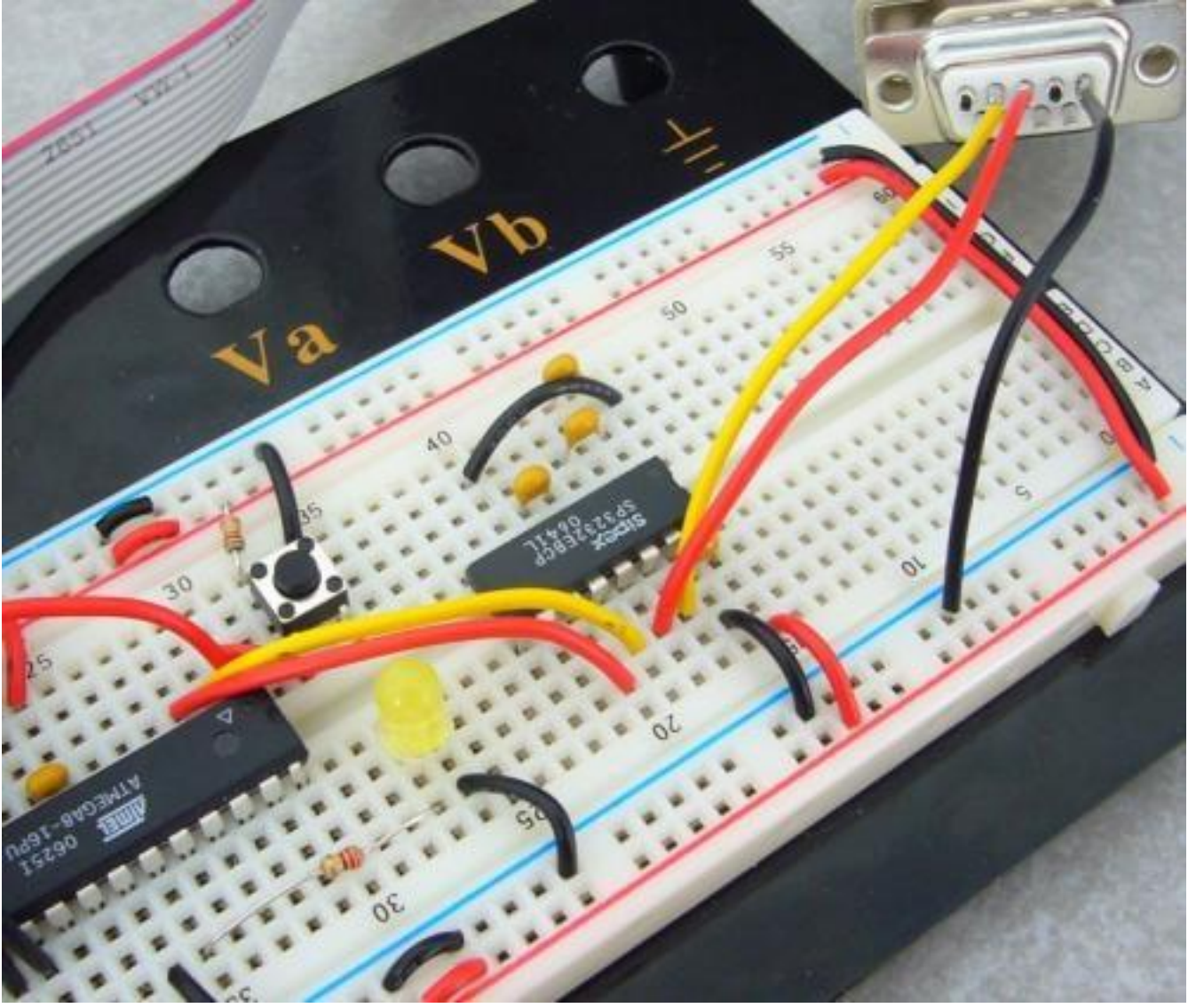
9600bps ve 8-N-1 şu anlama gelmektedir; Bilgisayar ve mikrodenetleyici arasındaki iletişim hızı, saniyede 9600 bit olacaktır ve Parity(eşlik) bitleri kullanılmadan sadece 8 bitin bittiğini belirten bir dur(stop) biti kullanılacaktır. Parity biti hakkında daha detaylı bilgiyi buradan edinebilirsiniz.

Şimdi tüm bağlantıların doğru olduğundan tekrar emin olup ilk denemelerimize başlayalım . Bilgisayarda herhangi bir karaktere bastığınız anda HyperTerminal programına bunun ASCII karşılığı gidecektir. ASCII, her karaktere özgü bir karşılık anlamına gelir ve 16 lık tabanda gösterilir. Örneğin ; Bilgisayar klavyesinden 'A' tuşuna bastınız, 'A' karakterinin ASCII karşılığı ikilik tabanda = 0b 01000001, onluk tabanda = 65, onaltılık tabanda = 0x41 dir. Yani siz klavyeden 'A' tuşuna bastığınız anda seri port çıkışından 0x41 sayısı gönderilecektir(Tx). Ve Tx ve Rx uçları birbirine bağlı olduğundan bu bilgi tekrar seri potunuzdan geri gelip(Rx) HyperTerminal programında görünecektir. Bu şekilde iletişiminizin sağlıklı çalışıp çalışmadığını kontrol edebilirsiniz. Eğer iletişim başarılı olduysa şimdi bu işlemi Atmega168 ile de yapabiliriz demektir. Bunun için breadboard üzerine aşağıdaki şemada verilen bağlantıları yapmanız gerekmektedir.

The schematic diagram illustrates the internal circuitry of an RS232C module. It includes a power regulation section, a signal processing section with a transformer, and a detailed pinout for a 25-pin connector. The components are labeled with their respective values and types, and the layout is organized into distinct functional blocks.

Legend:

- RES: Resistor
- CAP: Capacitor
- IC: IC
- SW: Switch
- BT: Button
- LED: LED
- DIODE: Diode
- TRANSFORMER: Transformer
- RELAY: Relay
- CON: Connector
- WIRE: Wire
- GROUND: Ground
- POWER: Power
- SIGNAL: Signal
- CONTROL: Control
- STATUS: Status
- ERROR: Error
- RESET: Reset
- ENABLE: Enable
- DISABLE: Disable
- STOP: Stop
- START: Start
- PAUSE: Pause
- RESUME: Resume
- ABORT: Abort
- RETRY: Retry
- OK: OK
- NO: NO
- YES: YES
- UNKNOWN: UNKNOWN
- UNDEF: UNDEF
- NULL: NULL
- EMPTY: EMPTY
- FALSE: FALSE
- TRUE: TRUE
- 0: 0
- 1: 1
- 2: 2
- 3: 3
- 4: 4
- 5: 5
- 6: 6
- 7: 7
- 8: 8
- 9: 9
- 10: 10
- 11: 11
- 12: 12
- 13: 13
- 14: 14
- 15: 15
- 16: 16
- 17: 17
- 18: 18
- 19: 19
- 20: 20
- 21: 21
- 22: 22
- 23: 23
- 24: 24
- 25: 25



Tx ve Rx Bağlantıları Atmega168 ile Yapılıyor

Not : Bu devrede MAX232 kullanılmadı. Çünkü MAX232 çıkış olarak sadece 5v çıkış vermektedir. Fakat bazı durumlarda bazı mikrodenetleyiciler 3.3v ile çalışıp portlarının maksimum 3.5v gibi bir gerilime izin verdikleri için SP232 kullanıldı. SP232, Sipex firmasının ürettiği çıkışı 3v olan bir seri iletişim entegresidir. MAX232'den tek farkı budur. Ve 5v sistemlerle sorunsuz olarak çalışabilmektedir.

Bu dersin tüm parçalarını buradan edinebilirsiniz.

Ders 5 - AVR GCC Kullanımı ve Derleme İşlemi

Atmel serisi için ilk yazdığımız kod, basit bir şekilde bir LED'i yakıp söndürmekteydi. Bu C dosyasını Programmer's Notepad'de açıp Make All tuşuna bastığımız zaman 'Procces Exit Code: 0' yazısını derleme işleminin sonunda görmeliyiz. Eğer bu satır, derleme işlemi sonucunda çıkmıyorsa muhtemelen programın bir yerinde hata vardır ve programınızı tekrar gözden geçirmeniz gerekmektedir.

İkinci örnek ise, bu yazıda verilecek olan seri haberleşme örneğidir.

Atmega168 için : basic-out-atmega168.c

Atmega8 için : basic-out.c

Programı başından itibaren satır satır inceleyecek olursak;

```
#define FOSC 16000000
```

```
#define BAUD 9600
```

```
#define MYUBRR FOSC/16/BAUD-1
```

Bu satırlarda ilk olarak osilatör frekansı belirlenmiş, daha sonra seri haberleşme için baud hızı belirlenmiş ve son olarak ise seri haberleşme için UBRR saklayıcısına yüklenmesi gereken değer hesaplanmıştır. Bunun için daha hassas bir hesaplama da yapılabilir.

```
#define MYUBRR (((((FOSC * 10) / (16L * BAUD)) + 5) / 10) - 1)
```

UBRR hesaplaması yapan kullanışlı bir program

<http://www.wormfood.net/avrbaudcalc.php>

```
static FILE mystdout = FDEV_SETUP_STREAM(uart_putchar, NULL, _FDEV_SETUP_WRITE);
```

Bu satır ise printf() fonksiyonu için hafızada bir tampon oluşturmaktadır. Daha detaylı bilgi için ioinit() fonksiyonunu inceleyebilirsiniz.

```
UCSROB = (1<<RXEN0)|(1<<TXEN0);
```

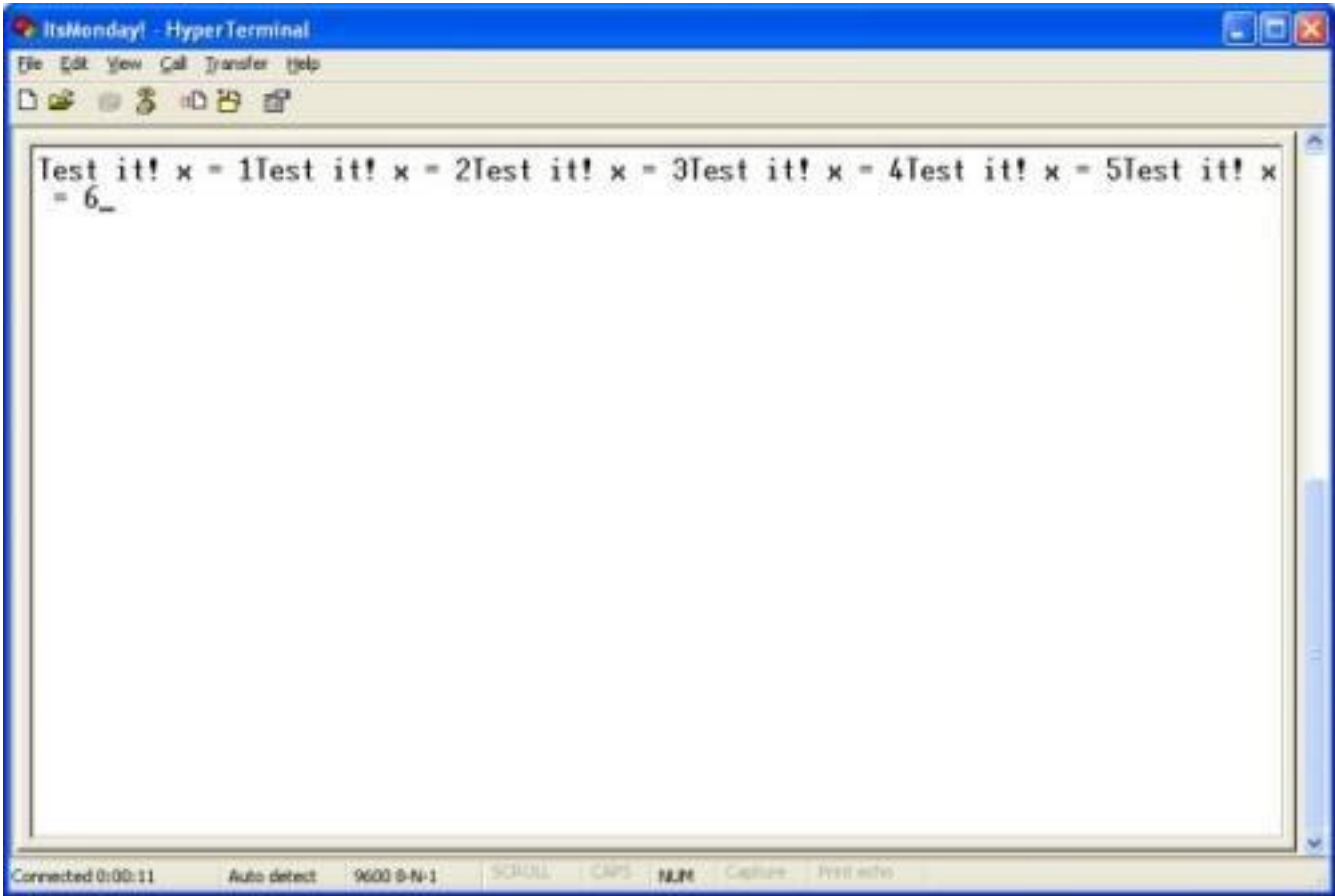
Bu satırda ise Tx ve Rx pinleri aktif edilmektedir. RXEN0 saklayıcısındaki değer kütüphane dosyalarında 5 olarak tanımlanmıştır. 1<<RXEN0 demek ise binary 0b00000001 sayısını 5 bit sola kaydır demektir. Yani yeni sayımız 0b00100000 olmaktadır. Burada sabit olarak 5 de kullanılabilirdi fakat RXEN0 saklayıcısının ön tanımlı değeri farklı modellere göre değişebileceğinden sadece bu saklayıcının ismini bilmemiz yeterlidir. İçerisinde hangi değer olduğunu bilmek zorunda değiliz.

Hemen yanındaki 1<<TXEN0 kısmı da ilk anlatılanla aynı işi yapmaktadır. Aradaki | işlemi ise bu iki kısmı mantıksal olarak toplamaktadır.

```
printf("Test it! x = %d", x);
```

Bu satır ise C'nin standart fonksiyonu olan çıktı sağlama fonksiyonu printf() tir. Tırnak içerisindeki kısımlar olduğu gibi ekrana yazılır. Sadece %d yazan kısım yerine hemen virgülden sonra verilen x in değeri yazılır. Burada %d demek oraya integer(tam sayı) bir değişken yazılacak demektir. Aşağıda HyperTerminal programının ekran çıktısında bunu daha da iyi anlayacaksınız.

Şimdi, basic-out-atmega168.c programını derleyip breadboard üzerindeki devremizde deneyelim. C dosyasını Programmer's Notepad editöründe açıp derledikten sonra, Tools->[WinAVR] seçeneğinden breadboard üzerindeki Atmega168'i programlayabilirsiniz. Sonra HyperTerminal'de 9600 bps'lik bir yeni bir haberleşme başlatmanız gerekmektedir.



Atmega168 ve Seri Haberleşme

```
sbi(PORTC, STATUS_LED);
```

Bu fonksiyon parametreleri içerisinde verilen PORT bilgilerine göre istenilen pini lojik 1 yapmaya yarar. Bunun için başta aşağıdaki tanımlamayı yapmanız gerekmektedir.

```
#define sbi(var, mask) ((var) |= (uint8_t)(1 << mask))
```

sbi fonksiyonu istenilen biti lojik 1, cbi fonksiyonu ise istenilen biti lojik 0 yapar.

```
#define STATUS_LED 0
```

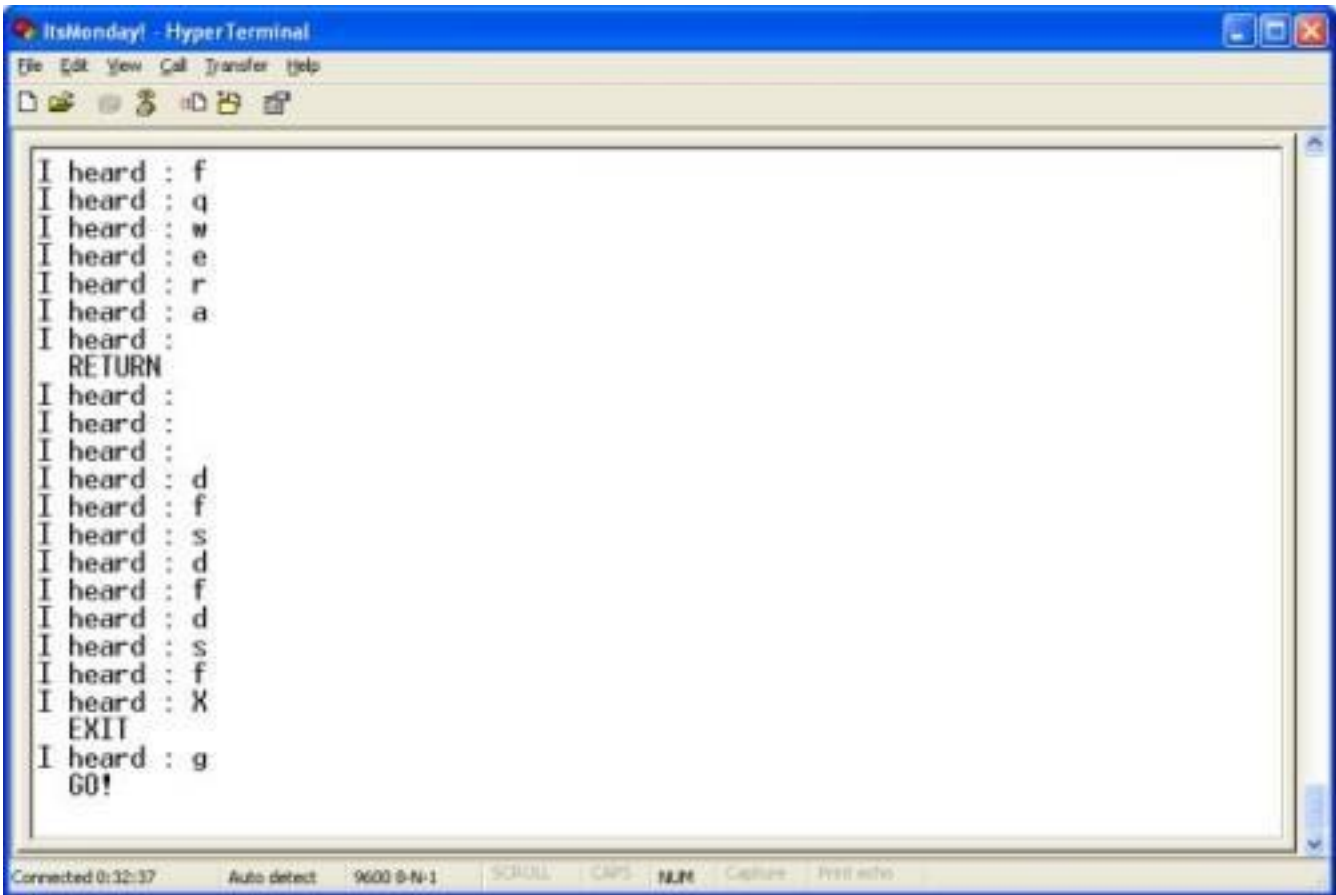
Bu tanımlamayı da yaptıktan sonra sbi ve cbi fonksiyonlarını kullanarak STATUS_LED'i yakıp söndürebiliriz.

```
sbi(PORTC,STATUS_LED); //STATUS_LED yandı
```

```
cbi(PORTC,STATUS_LED); //STATUS_LED söndü
```

PORTC'nin STATUS_LED bağıladığımız pini ise Atmega168'in 23.pinidir.

Şimdi basic-in-atmega168.c (ATmega8 için basic-in.c) programını derleyip breadboard üzerindeki devrenize yükleyin.



```
I heard : f
I heard : q
I heard : w
I heard : e
I heard : r
I heard : a
I heard :
RETURN
I heard :
I heard :
I heard :
I heard : d
I heard : f
I heard : s
I heard : d
I heard : f
I heard : d
I heard : s
I heard : f
I heard : X
EXIT
I heard : g
GO!
```

Bu program klavyeden girilen herhangi bir karakteri HyperTerminal ekranında göstermektedir. Eğer ENTER, X veya g tuşlarından herhangi birine basarsanız normalden daha farklı bir çıktı görürsünüz. Bunun için aşağıdaki sorgular kullanılmıştır.

```
key_press = uart_getchar();
```

```
printf("I heard : %c\n", key_press);
```

```
if(key_press == 'g') printf(" GO!\n");
```

```
if(key_press == 'X') printf(" EXIT\n");
```

```
if(key_press == 13) printf(" RETURN\n");
```

uart_getchar() fonksiyonu klavyeden basılan tuşun ASCII karşılığını key_press değişkenine atamaktadır. Ve if sorguları ise bunun g,X veya ENTER tuşlarından biri olup olmadığını sorgulamak içindir. Printf() fonksiyonundaki %c değeri ise ekrana yazılacak değerin bir karakter olduğunu belirtmektedir. Eğer burada, yukarıdaki gibi %d kullanılsaydı ekrana, basılan tuşun ASCII kodu yazılacaktı. Ve son olarak \n ile bir alt satıra geçilmektedir.

Ders 6 – Lehimleme

Öncelikle, lehim yaparken bilinmesi gereken temel kurallara değinelim.

- Havyalar lehimi eriten bir madde olduğu için çok sıcaktırlar. Bu nedenle kullanırken dikkatli olunuz.
- Lehim yapacağınız zaman havyanın süngerini mutlaka ıslatmalısınız.
- Havya lehimlemek için değil lehimlenecek yüzeyi ısıtmak için kullanılır. Eğer bu kurala uyararsanız düzgün ve parlak lehimler yapabilirsiniz.
- Lehim yaptıktan sonra eğer lehimi beğenmediyseniz lehimi pompa ile çekip tekrar lehimlemeye çalışmayın. Çünkü bu işlem sonucunda PCB'deki yollar ve delikler büyük ölçüde zarar görecektir.
- Lehimleme yapmadan önce lehimlenecek yeri havya ile biraz ısıtmanız lehimin daha kolay eriyip, yayılmasını sağlayacaktır.

Havya ucunun nasıl temizlendiğini ve bir devre elemanının nasıl lehimlendiğini aşağıdaki videoda görebilirsiniz.

http://www.youtube.com/playlist?list=PLfcIZXsDLA1_6d_50JFv19ZdBzhj4AIB1

abone 27.327 görüntüleme Video Yöneticisi Herkese açık olarak görüntüle

CİZGİ TAGEM Bilgi Paylaştıkça Çoğalır

Niyazi Saral Ayarlar

Videolar Tartışma Hakkında

Oynatma listeleri Eklenme tarihi (en yeniden en eskiye)

Favori videolar 9 video

Soldering (Through Hole) 24 video | 4 hafta önce

Soldering (Surface Mount Technology) 103 video | 4 hafta önce

Soldering (BGA) 105 video | 4 hafta önce

Bilgisayar Donanım Elemanı Eğitimi 44 video | 4 hafta önce

Çizgi RMA 20 video | 1 yıl önce

İleri Bilgisayar Donanım Eğitimi (Sieguntur-Türkçe) 20 video | 3 hafta önce

NoteBook Servis Videoları 15 video | 3 hafta önce

Teknik Servis Eğitimleri (Türkçe) 22 video | 3 hafta önce

CompTIA Network+ (Türkçe) 45 video | 3 hafta önce



Isı Ayarlı Havya ve Temizleme Süngeri

Normal kullanım için standart havya ucu yeterlidir. Resimde görülen havya Sunko marka ucuz bir havyadır. Sıcaklığı 350C ye kadar çıkabilmektedir.



Soğuk Havya Ucu



Lehim Teli

Yukarıda görülen lehim telinin çapı oldukça küçüktür. İnce olmasının iyi tarafı ; lehim yapılacak yere fazla miktarda lehim akmasını sağlamak ve daha hassas kontrol sağlamak içindir. Kötü tarafı ise DB9 konektörlerinin ayakları gibi büyük deliklerde lehimin yeterince akması için fazla beklenilmesidir.

SN63PB37 ise %63 Kalay, %37 Kurşun karışımı olduğunu belirtmektedir. Oranlardan da görüldüğü gibi lehim telinde fazla miktarda kimyasal bulunmaktadır. Bu kimyasallar insan sağlığına zararlıdır. Bu sebeple lehim yaparken oluşan dumanı mümkün olduğunca koklamamaya çalışmalısınız. Ve lehimleme işlemi bittikten sonra ellerinizi yıkamalısınız.



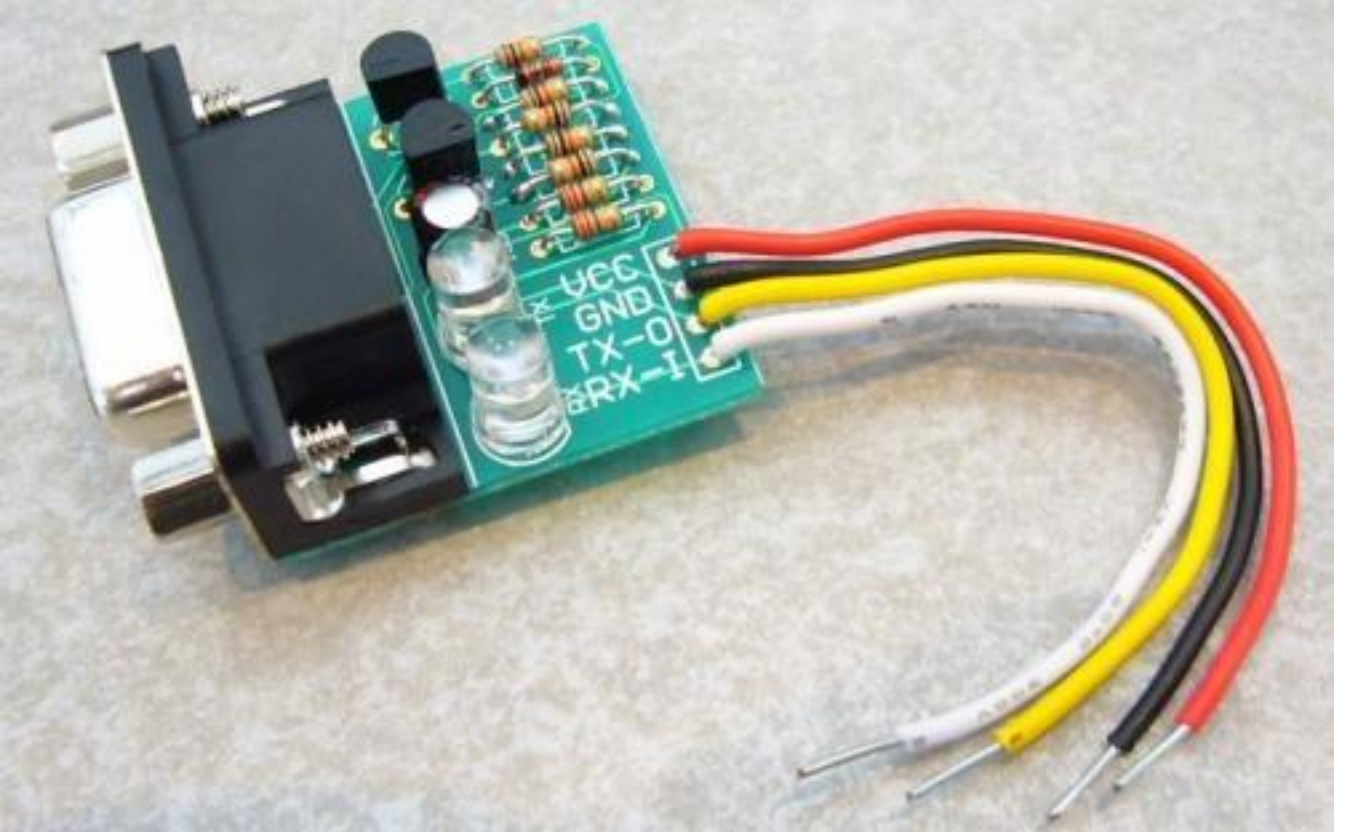
Sıcak Havya Ucu ve Bir Miktar Oksitlenmiş Lehim



Havya Ucu Temizlendikten Sonra Havya Ucuna Alınan Lehim

Eğer havyanızın ucunu ıslak sünger ile temizleyip sonra lehimleme işlemine geçerseniz daha temiz bir lehimleme işlemi yapmış olursunuz.

Bu derste aşağıdaki kartın malzemelerini lehimleyeceğiz.



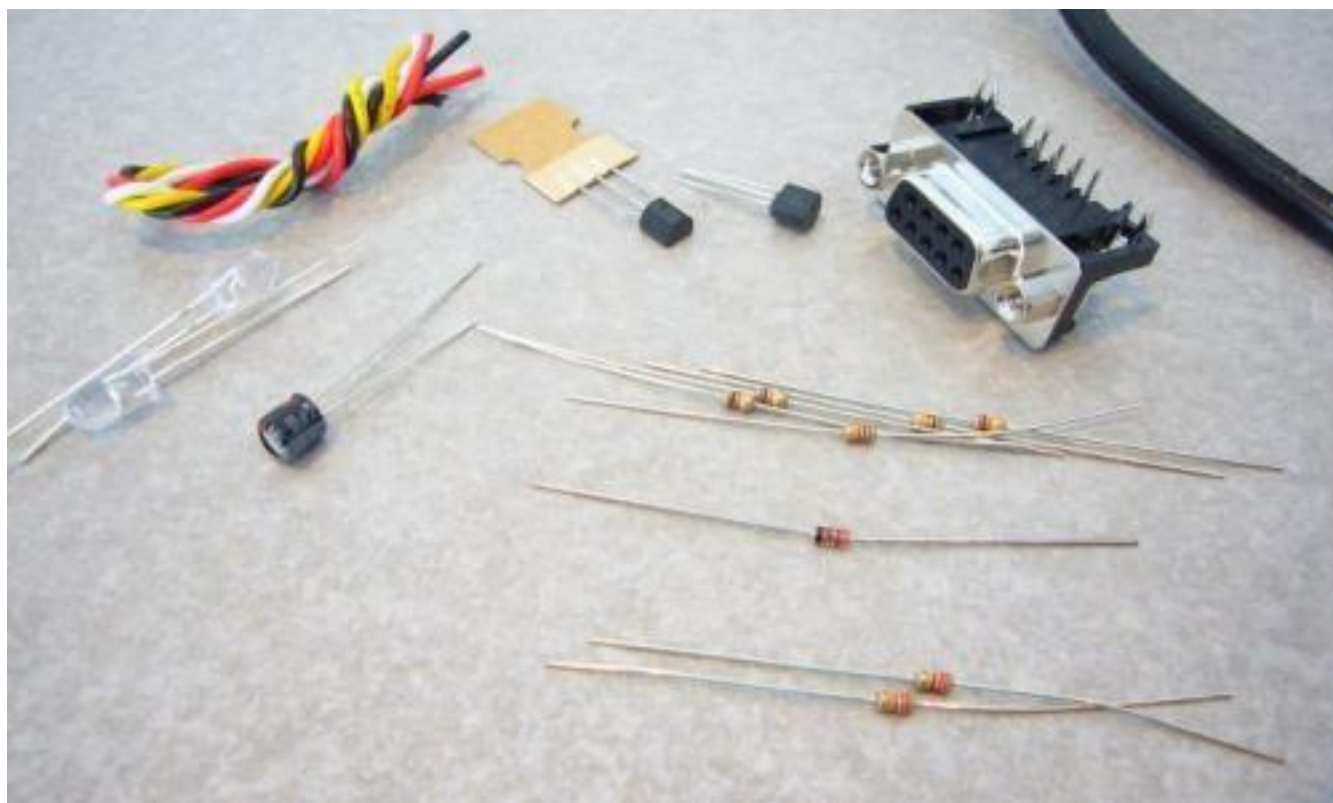
Yukarıda görülen devre Level Shifter diye adlandırılan temelde MAX232 ile aynı işi yapan bir devredir. Fakat MAX232 entegresi kullanılmadan doğrudan transistörler ile bu iş yapıldığından bazı problemler oluşmaktadır.

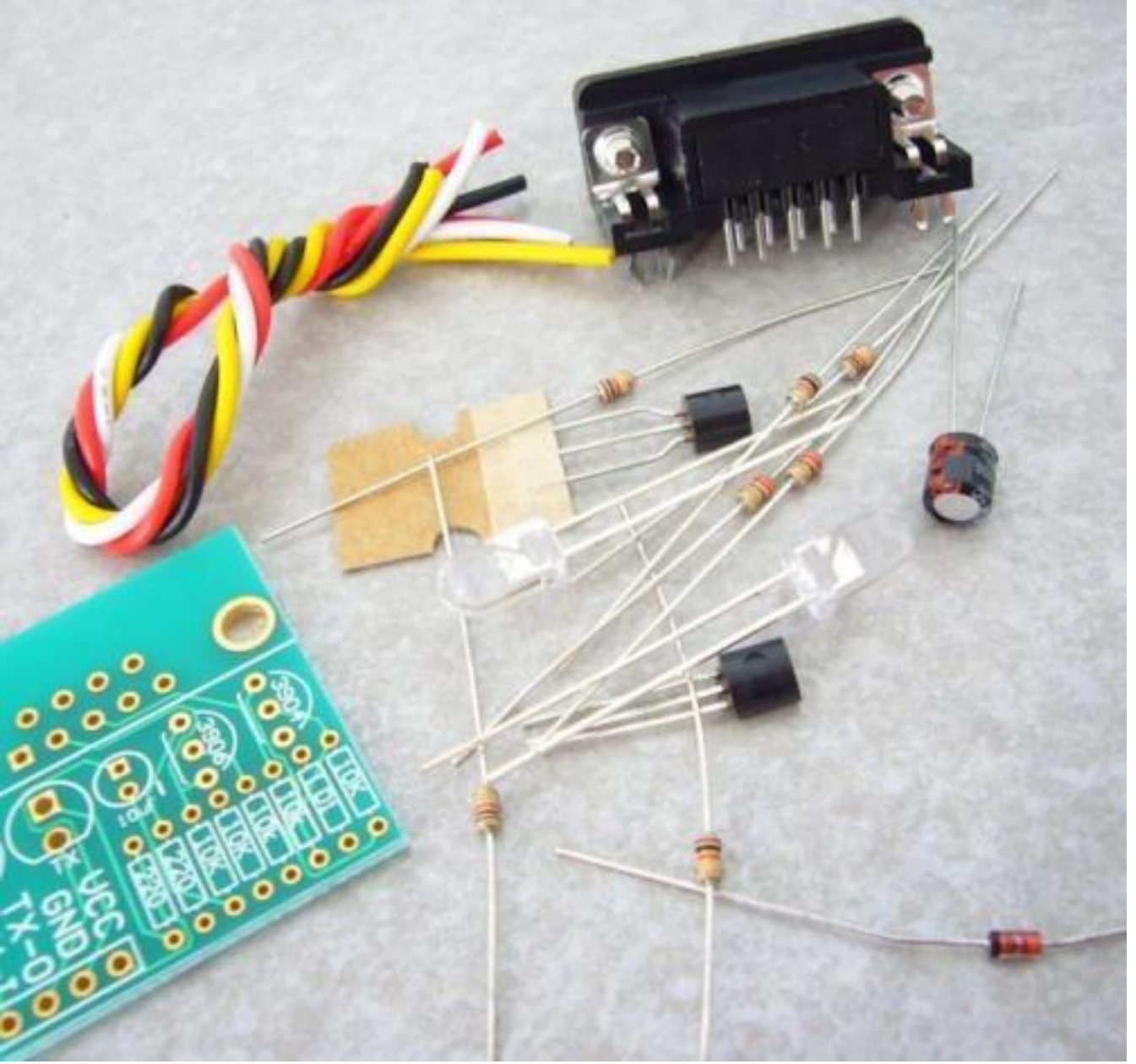
Artıları;

1. Shifter Board, MAX232 devresinden daha az yer kaplamaktadır.
2. Daha ucuzdur.
3. 2.5v ile dahi çalışabilmektedir.

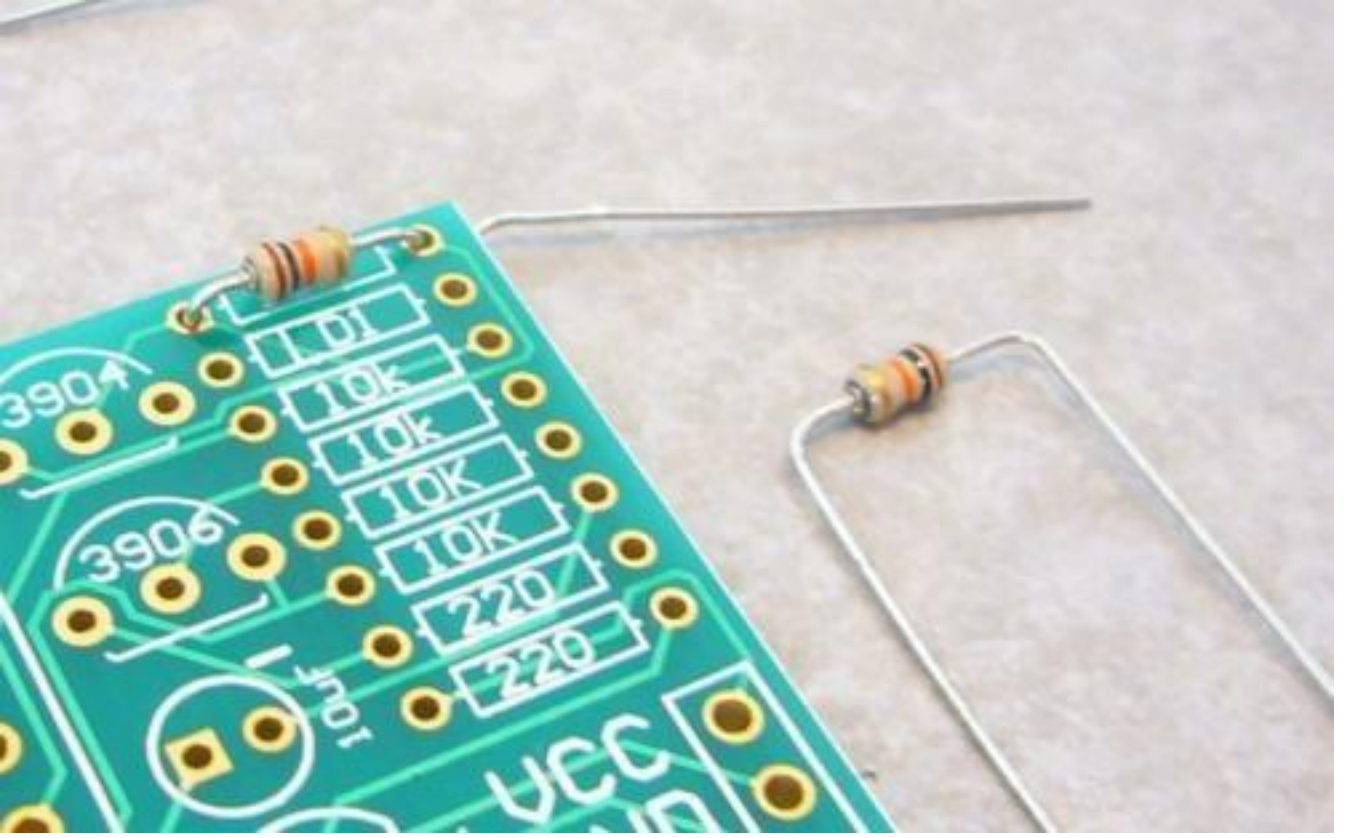
Eksileri;

1. MAX232 gibi +/-12V değil de sadece -3v ve +5v girişi uygunlaştırmaktadır.
2. MAX232 gibi yüksek hızlara çıkamamaktadır(Örn : 115200bps)
3. Sadece TX/RX çıkışları vardır. MAX232'de ise TX/RX/CTS/RTS çıkışları da vardır.

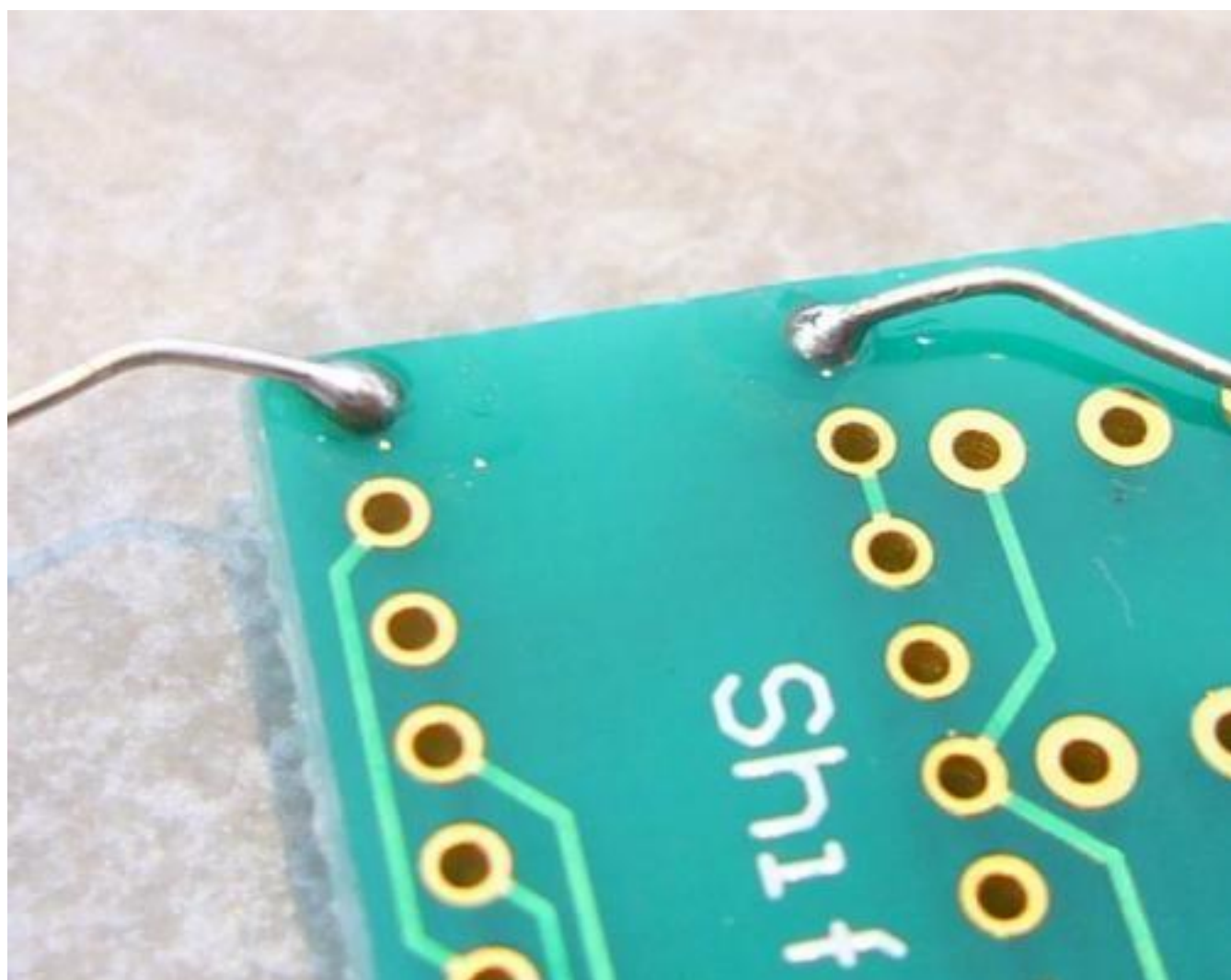


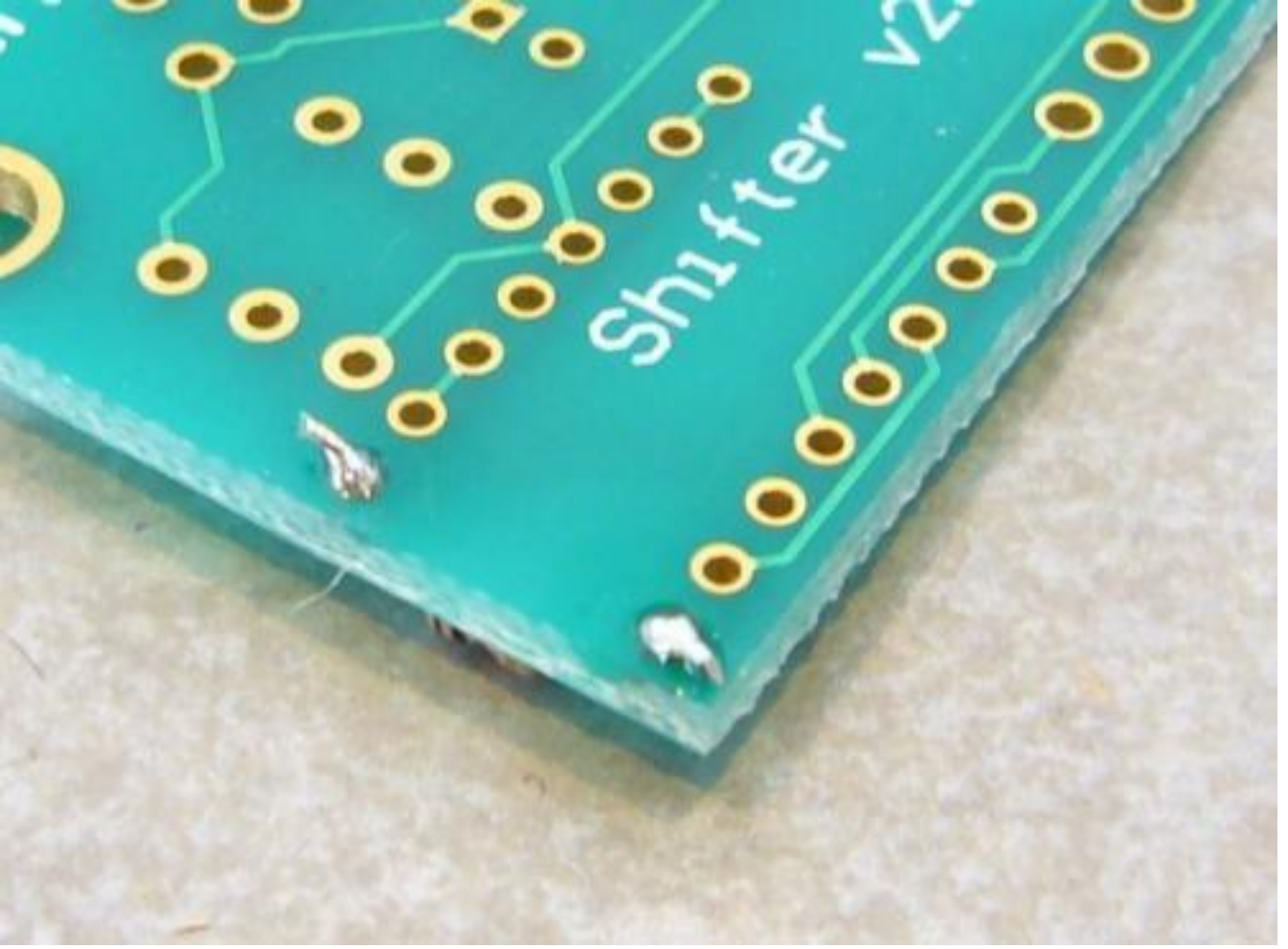


Eğer direnç değerlerini okumakta güçlük çekiyorsanız AVO metreniz ile dirençleri ölçüp ona göre uygun yere takabilirsiniz. Shifter Board'da sadece 5 adet 10k ve 2 adet 220Ohm direnç bulunmaktadır.

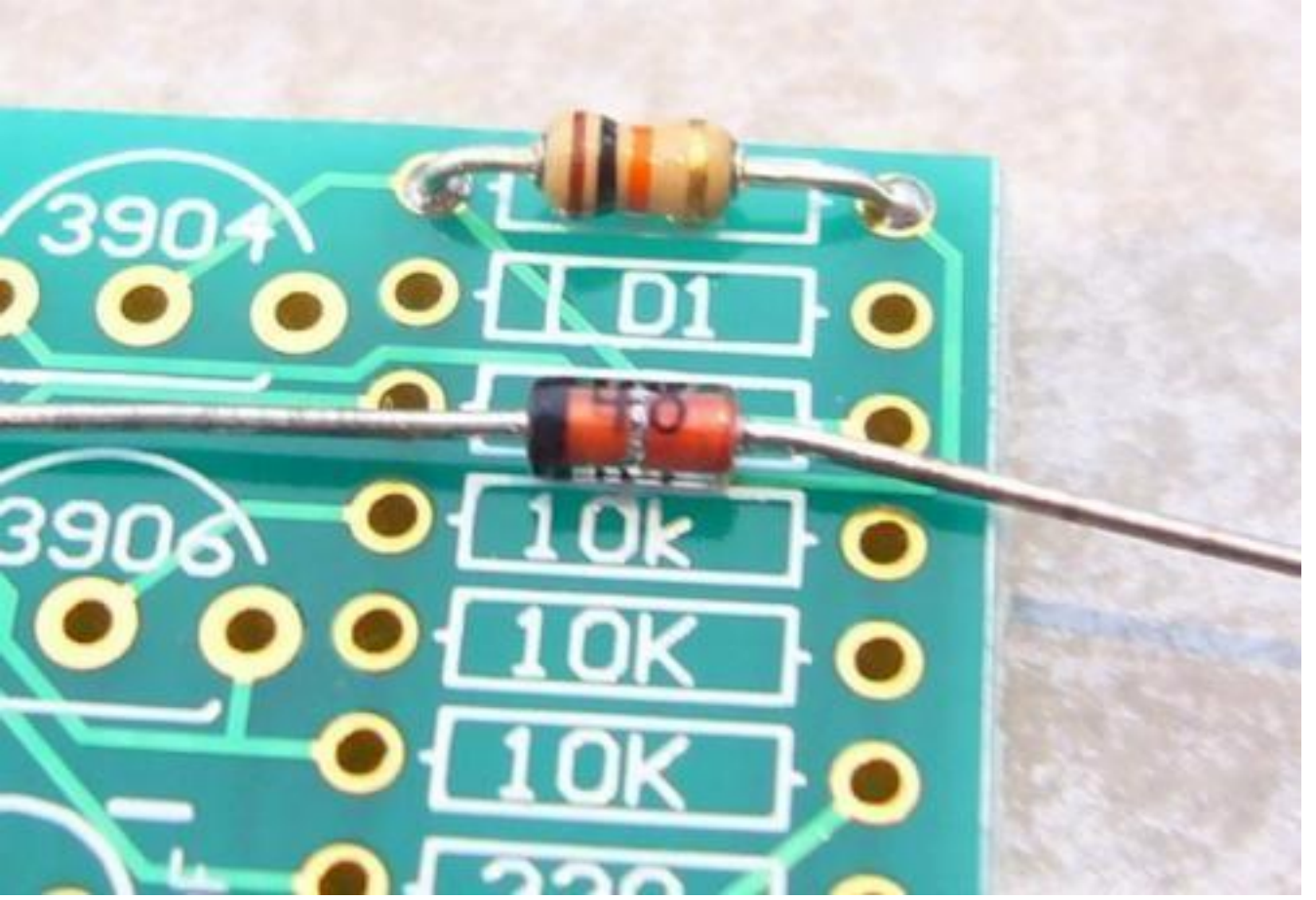


Dirençler ynsz devre elemanları olduęu iin herhangi bir Őekilde baęlayıp lehimleyebilirsiniz. Direnci devre zerine taktıktan sonra dŖmemesi iin ayaklarını bkmeniz gerekmektedir.





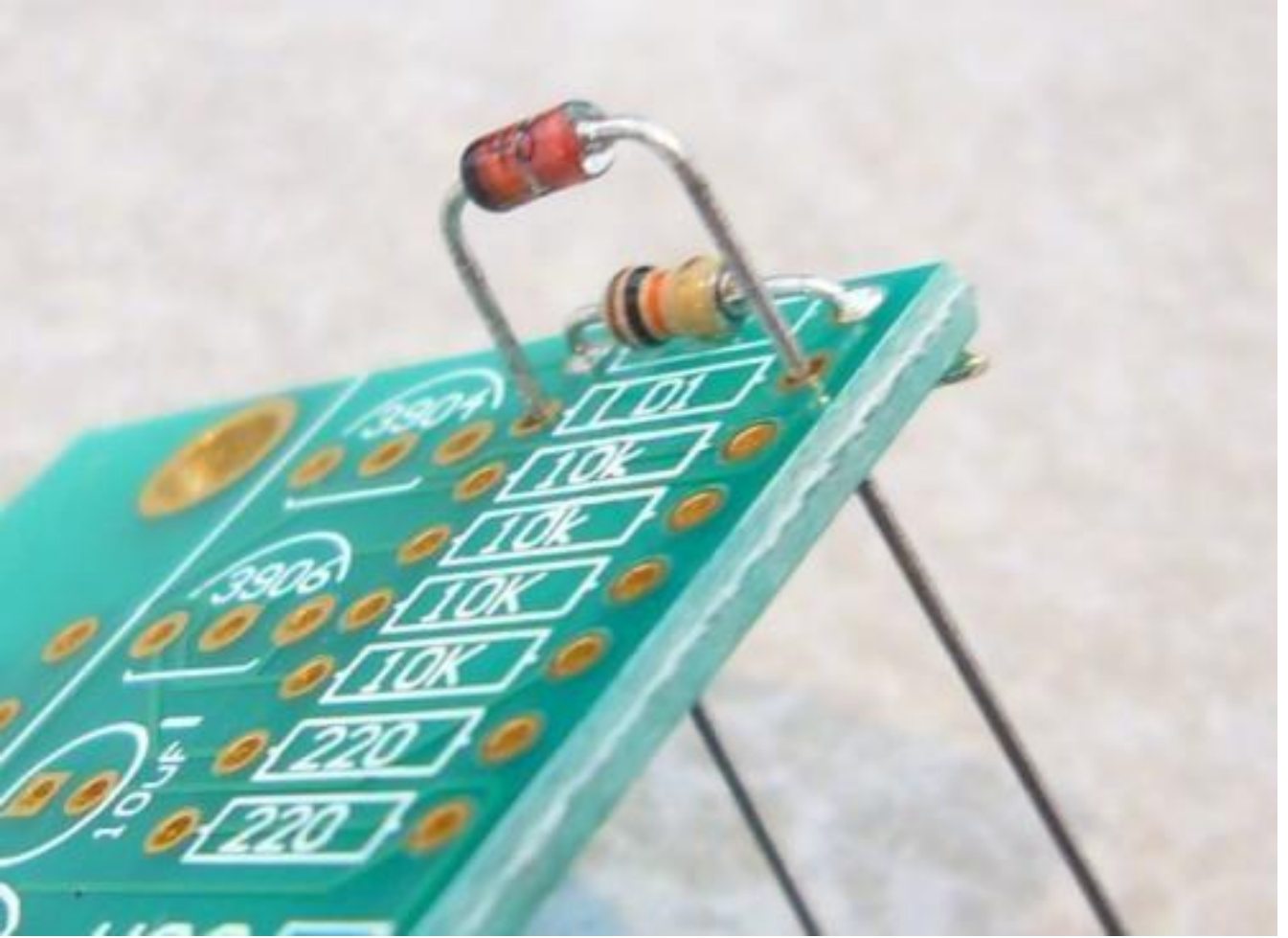
Lehimleme işlemini tamamladıktan sonra fazla kısımları yan keski ile kesebilirsiniz.



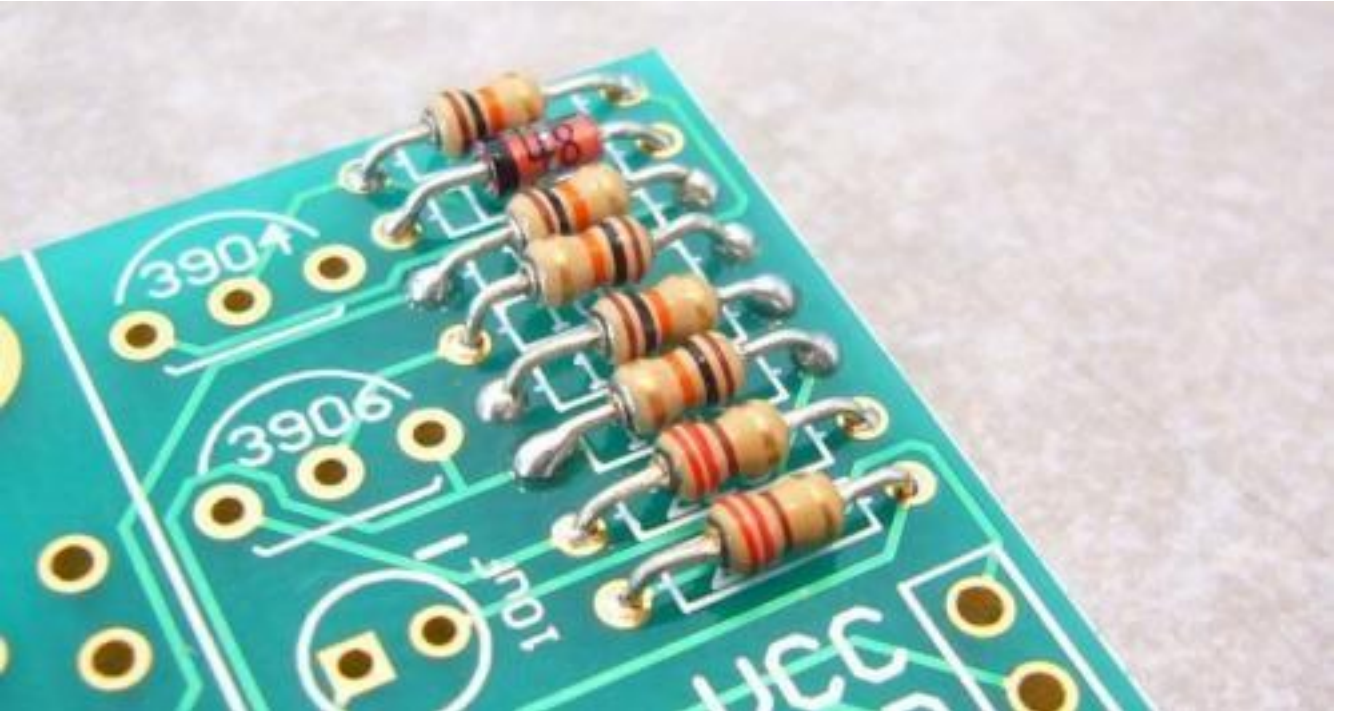
Shifter Board PCB'sinde, hangi devre elemanının nereye lehimleneceğini üzerindeki malzeme yerleşim planından daha kolay anlayabilirsiniz.

Eğer resimdeki gibi alt taraftan akıttığınız lehim, üst tarafta da görünüyorsa lehimleme işlemi başarılı olmuş demektir.

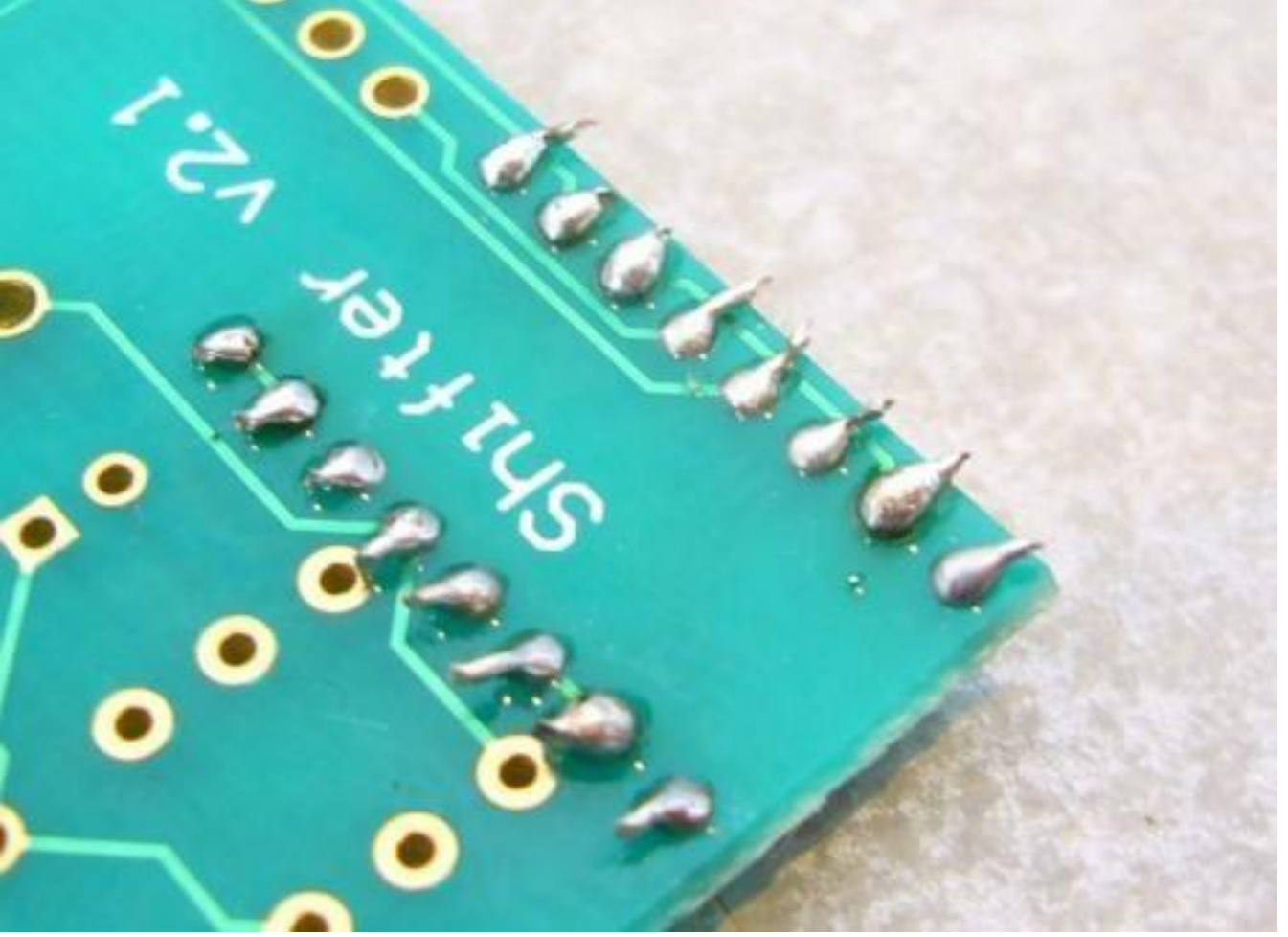
D1 Diyotunu resimde görüldüğü gibi lehimlemeniz gerekmektedir. Çünkü diyotlar yönlü yani artısı ve eksisi olan devre elemanlarıdır.



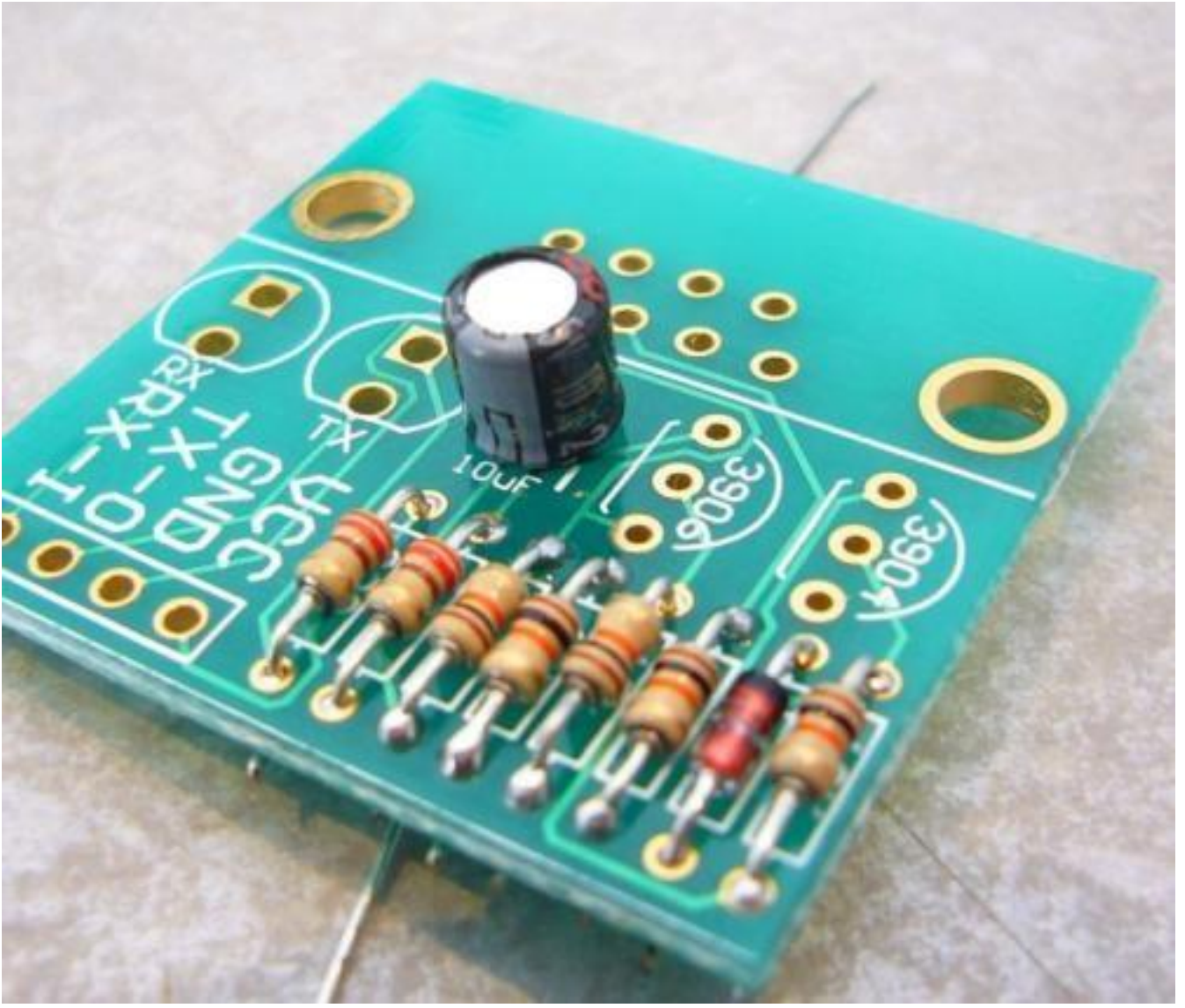
Diyotlar, hassas devre elemanlarıdır. Bu yüzden lehimleme işlemi sırasında havya ile bu devre elemanını fazla ısıtmayın. Yüksek sıcaklığa fazla maruz kalma sonucunda diyotların kimyasal yapısı zarar göreceğinden diyot, sağlıklı çalışmayabilir.



D1 diyotdu ve tüm dirençlerin lehimlenmiş hali...



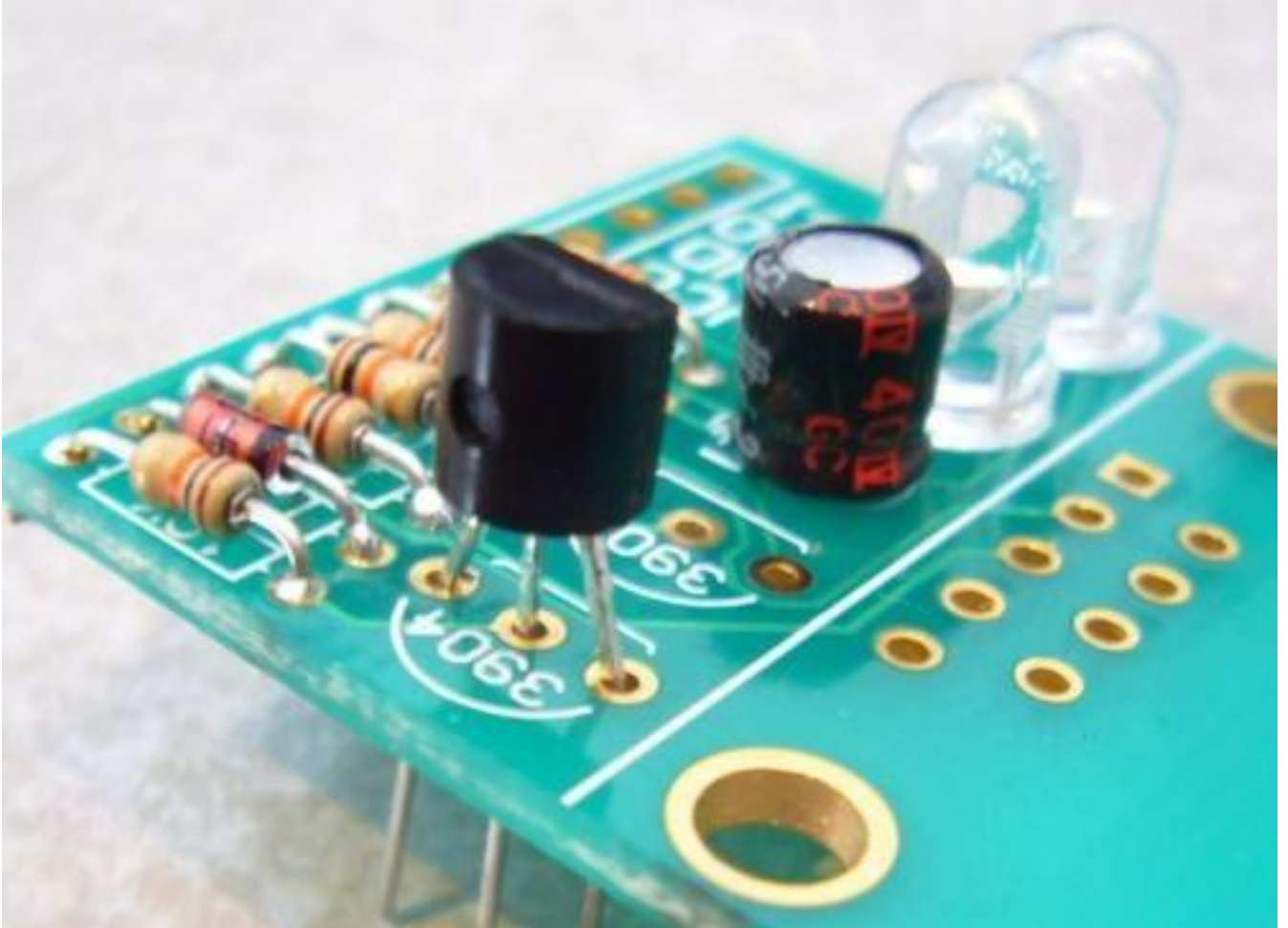
Lehimleme işlemi bittikten sonra devre üzerinde kalan katı lehim pastasını flux ile temizleyebilirsiniz. Bu şekilde lehimleriniz daha parlak ve PCB daha temiz görünecektir.



Elektrolitik kondansatörün lehimlenmesi. Lehimlerken + ve - yönlerine dikkat ediniz.



Tx ve Rx LED'lerinin lehimlenmesi. Lehimlerken + ve - yönlerine dikkat ediniz.



2N3904 ve 2N2906 transistörlerinin lehimlenmesi. Lehimlerken PCB üzerindeki şekle dikkat edip aynı şekilde deliklerden geçiriniz.

Havya ucunun nasıl temizlendiğini ve bir devre elemanının nasıl lehimlendiğini aşağıdaki videoda görebilirsiniz.

http://www.youtube.com/playlist?list=PLfcIZXsDLA1_6d_50JFv19ZdBzhj4AIB1

abone 27.327 görüntüleme Video Yöneticisi Herkese açık olarak görüntüle

ÇİZGİ TAGEM Bilgi Paylaştıkça Çoğalır

Niyazi Saral Ayarlar

Videolar Tartışma Hakkında

Oynatma listeleri Ekleme tarihi (en yeniden en eskiye)

Favori videolar 9 video

Soldering (Through Hole) 24 video | 4 hafta önce

Soldering (Surface Mount Technology) 103 video | 4 hafta önce

Soldering (BGA) 105 video | 4 hafta önce

Bilgisayar Donanım Elemanı Eğitimi 44 video | 4 hafta önce

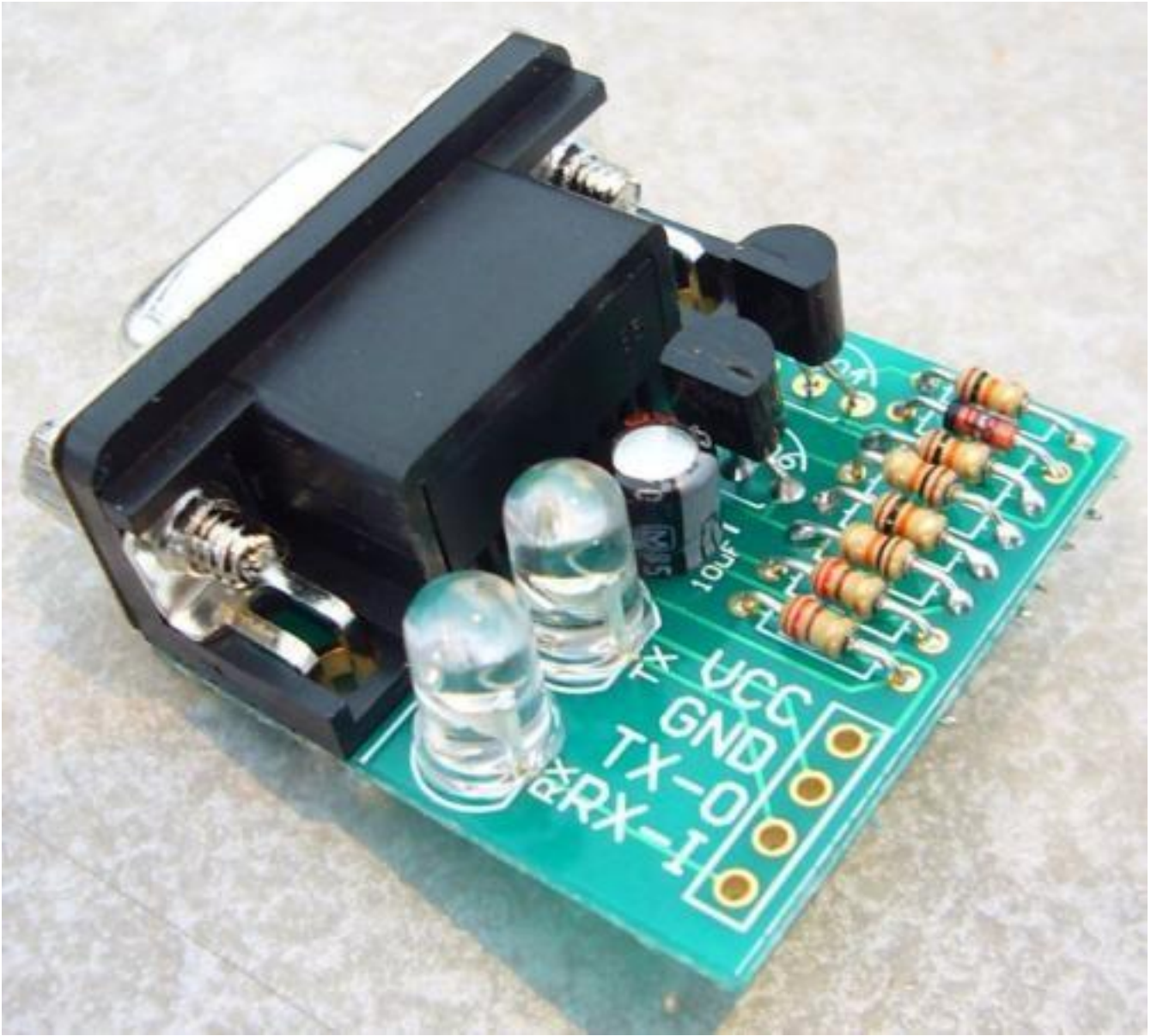
Çizgi RMA 20 video | 1 yıl önce

İleri Bilgisayar Donanım Eğitimi (Sieguntur-Türkçe) 20 video | 3 hafta önce

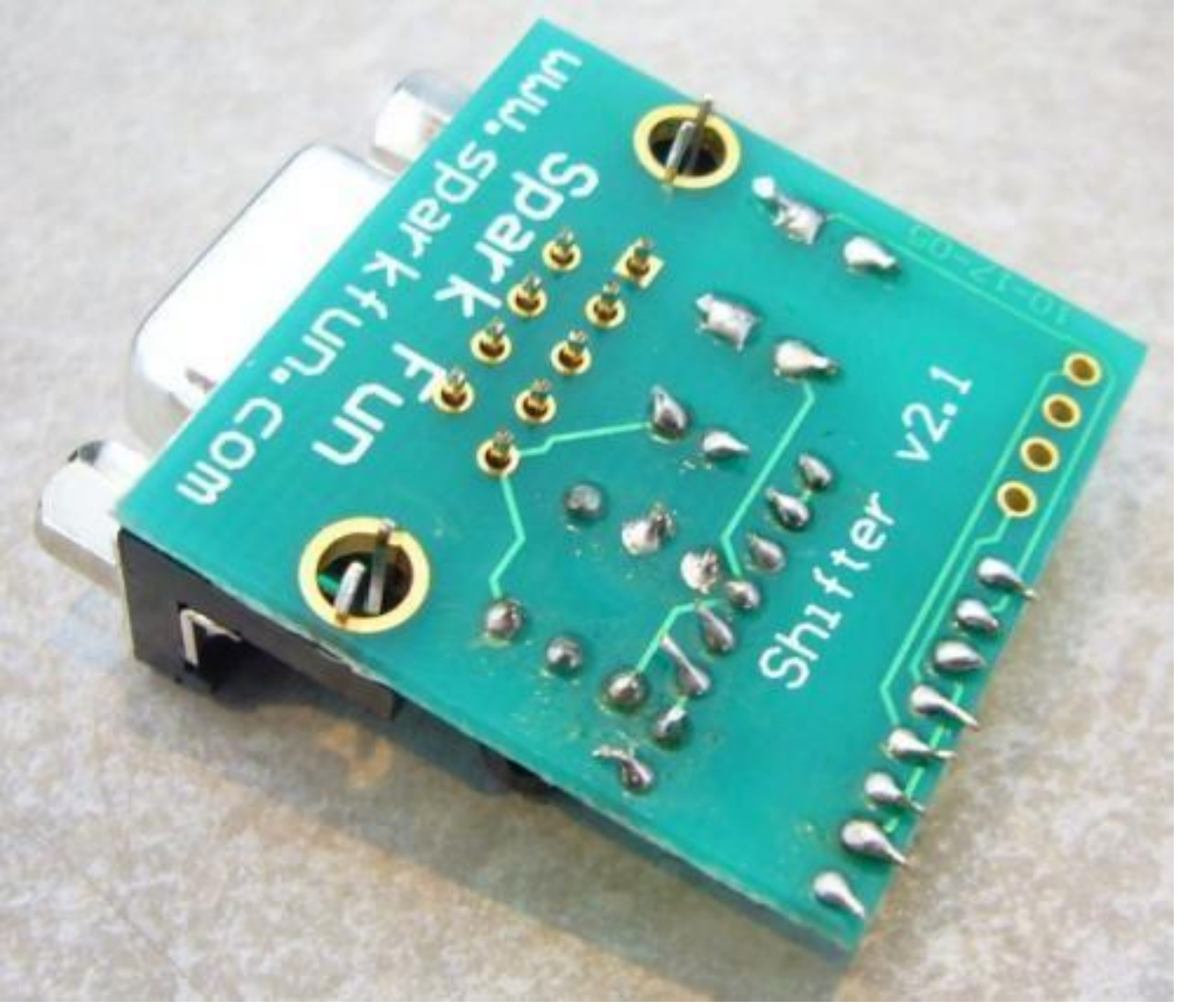
NoteBook Sevis Videoları 15 video | 3 hafta önce

Teknik Servis Eğitimleri (Türkçe) 22 video | 3 hafta önce

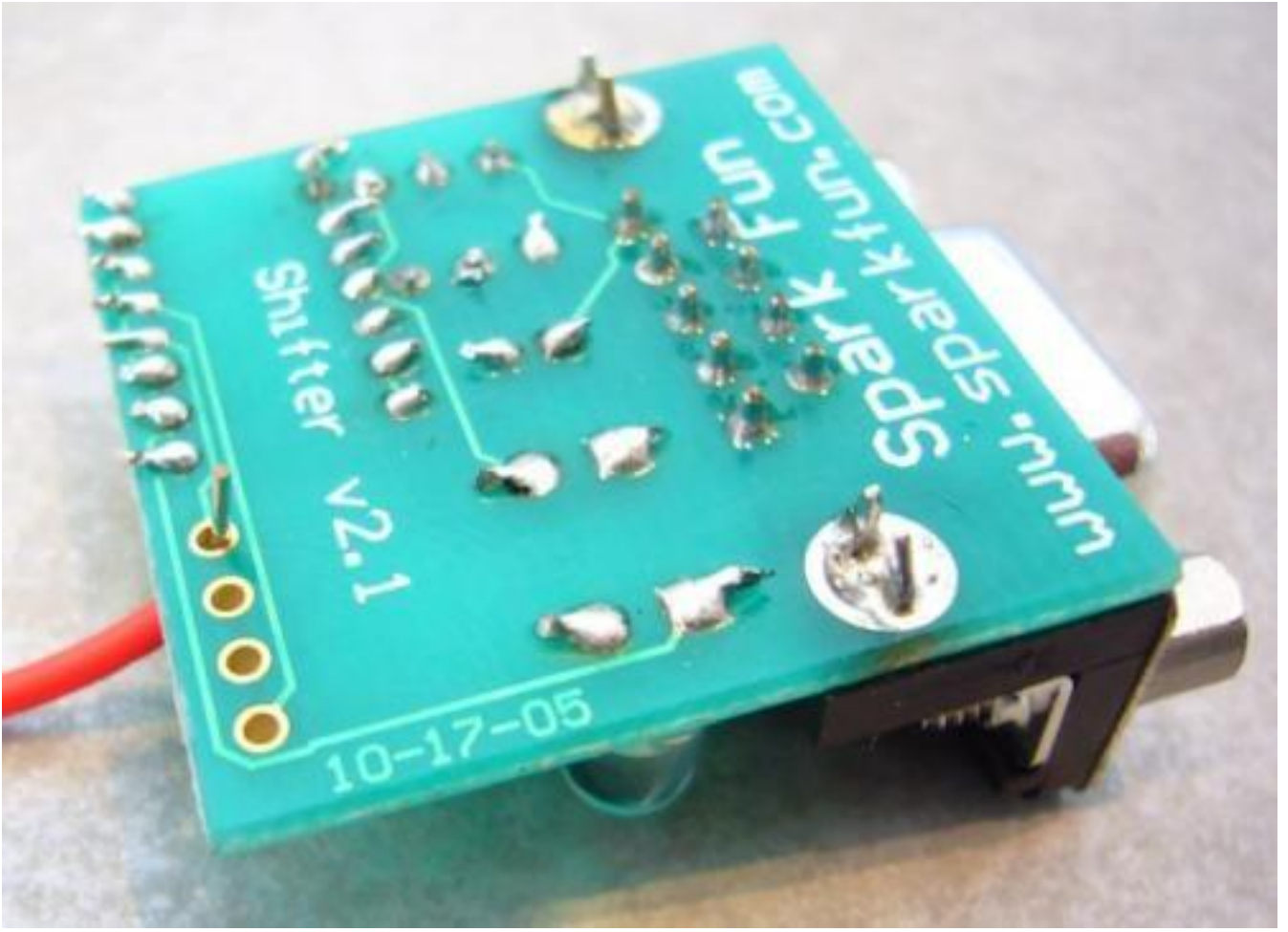
ComptIA Network+ (Türkçe) 45 video | 3 hafta önce



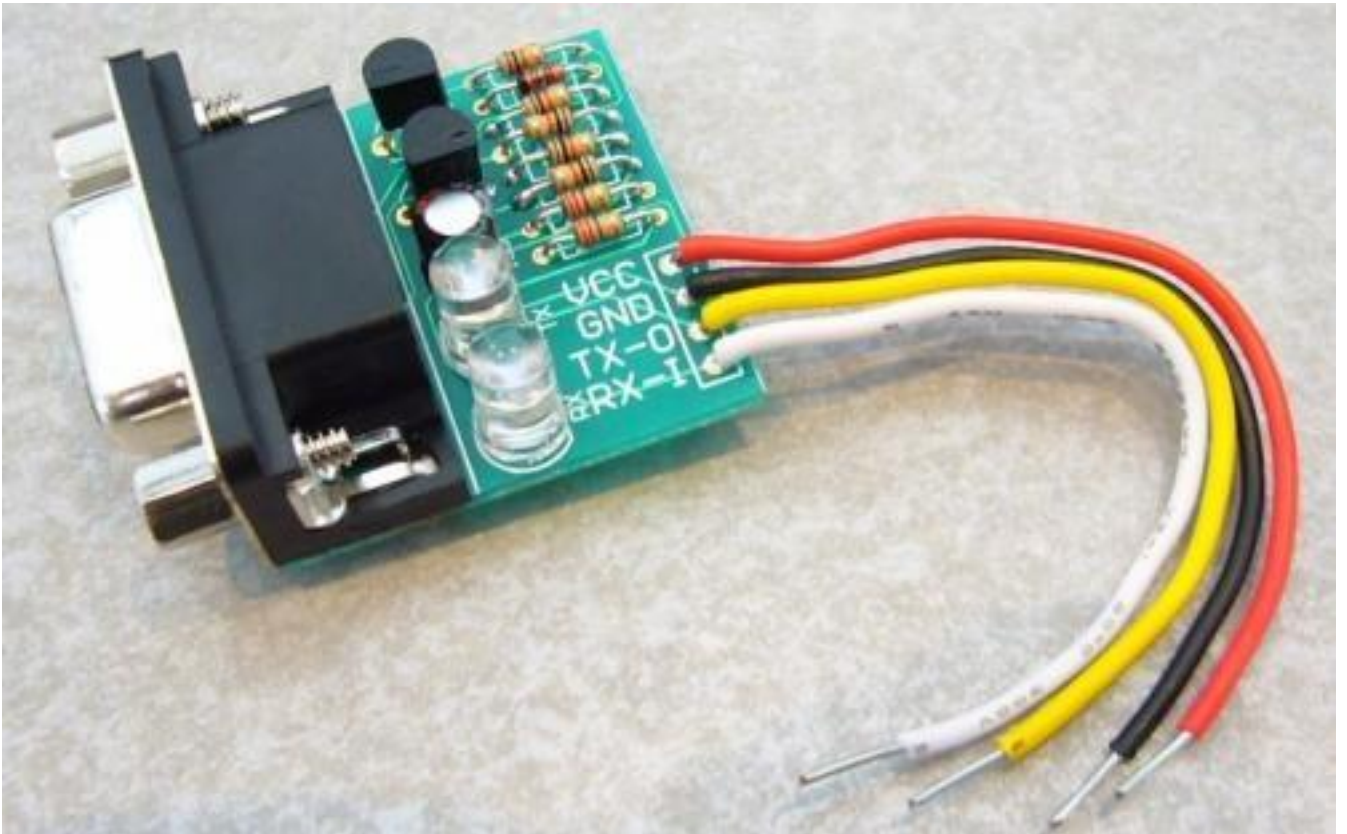
DB9 Konektörünün lehimlenmesi.



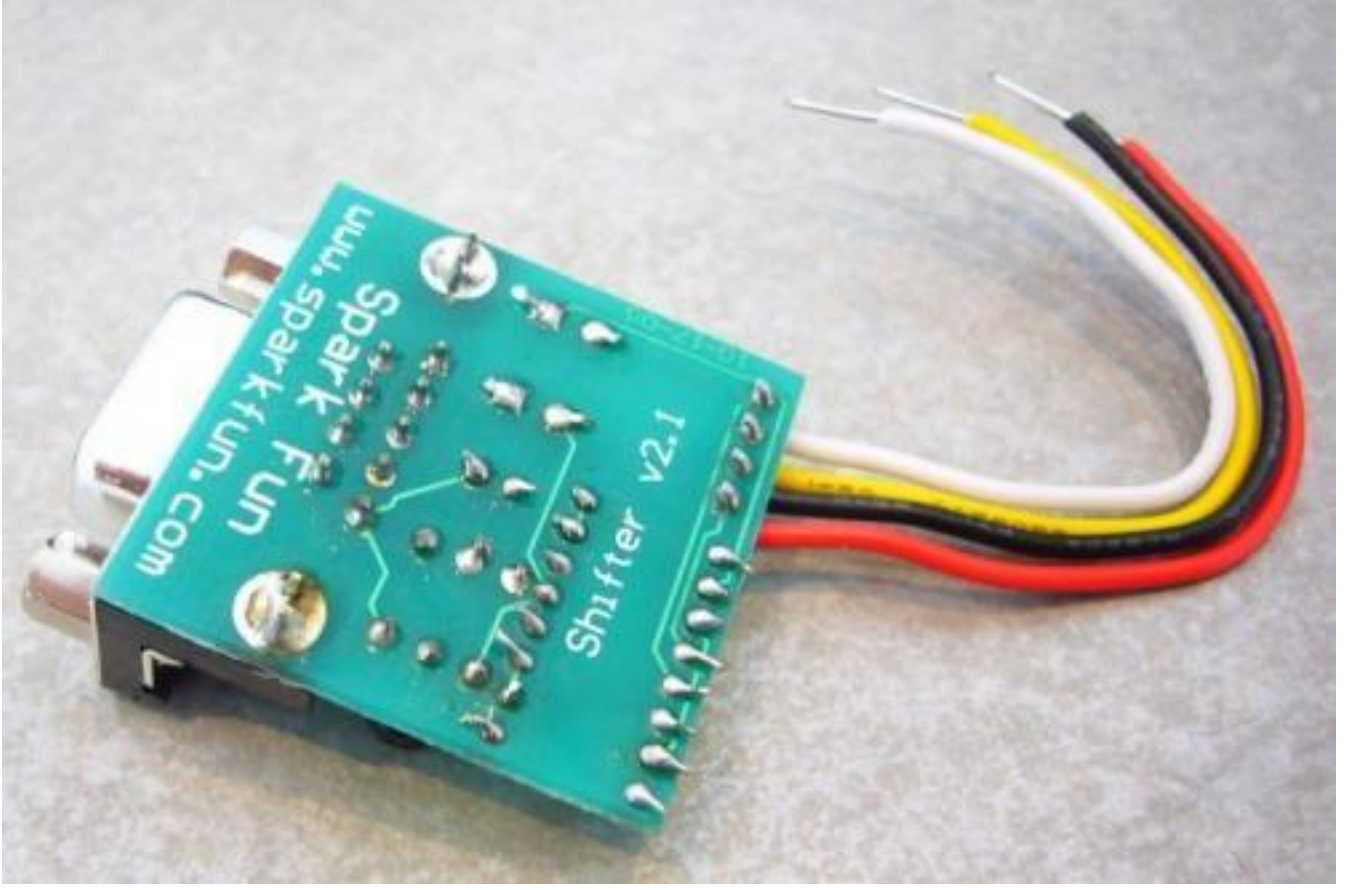
Resimde görüldüğü gibi büyük bir komponent takıldığında daha küçük komponentleri lehimlemek zor olacağı için bu tarz büyük malzemeleri en son lehimliyoruz.



Vcc, GND, Tx ve RX çıkışlarına breadboard bağlantısı için kabloların lehimlenmesi.



Shifter Board'un tüm parçaları lehimlendi ve kullanıma hazır hale geldi.



Son olarak ise Shifter Board çıkışlarını breadboard üzerindeki devremizde uygun yerlere bağlamamız gerekmektedir. Vcc ucu 5volta, GND ucu GND'ye ve Tx ve Rx uçları ise tam tersi olarak yani Tx ucu Atmega168'de Rx ucuna, Rx ucu ise Atmega168'de Tx ucuna gelecek şekilde bağlantılarımızı yapmamız gerekmektedir.

Breadboard'a bağlama işlemini tamamladıktan sonra breadboard üzerindeki anahtarı açıp, devreye enerji verdiğimizde ve HyperTerminal'den bir şeyler yazıp gönderdiğimizde Tx ve Rx LED'lerinin yanıp söndüğünü görüyorsak lehimleme işlemi başarıyla gerçekleştirilmiş demektir.

Ders 7 - SMD Komponent Lehimleme

Bu dersimizde SMD komponentlerin nasıl lehimlendiğini inceleyeceğiz. Bunun için aşağıdaki dosyalara ihtiyacınız olabilir.

- Simon Assembly Procedure (5MB)
- Simon Board (Malzeme Yerleşimi)
- Simon Eagle Files
- Simon ATmega168 Firmware
- Simon (Old) ATmega8 Firmware
- Simon Schematic

Burada ise SMD komponent lehimleme hakkında faydalı bilgiler var.

- SMD Soldering Basics (big 4MB)
- SMD Soldering Workshop

LED'leri kontrol eden program dosyası

- Basic Simon Control



Simon SMD Kit



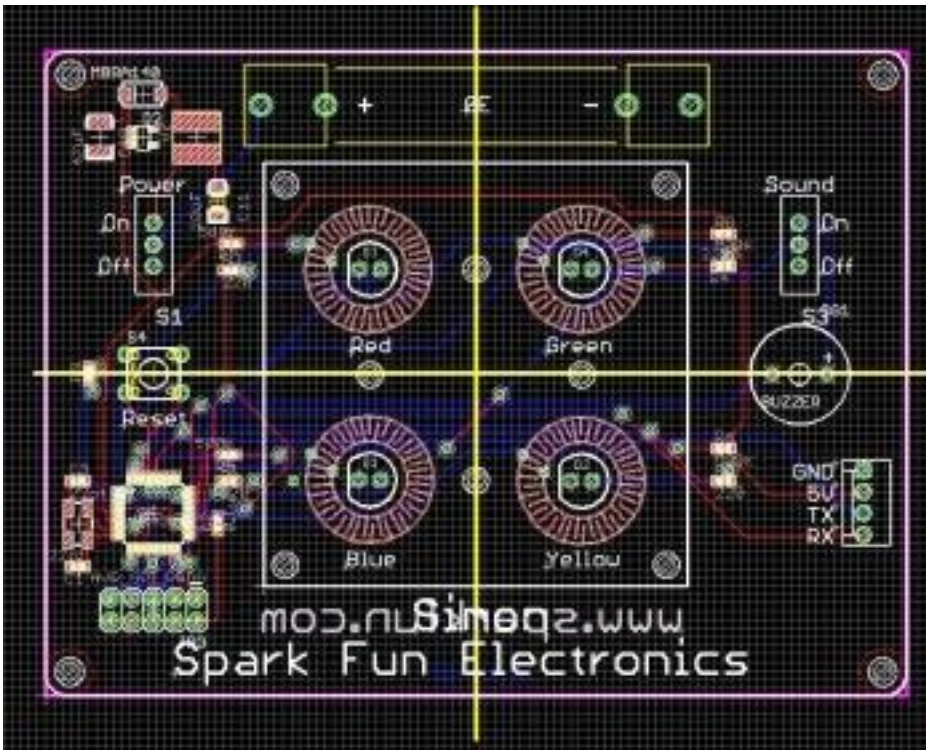
Devrenin Tamamlanmış Hali

Bu devre için aşağıdaki DIP malzemeler gerekmektedir;

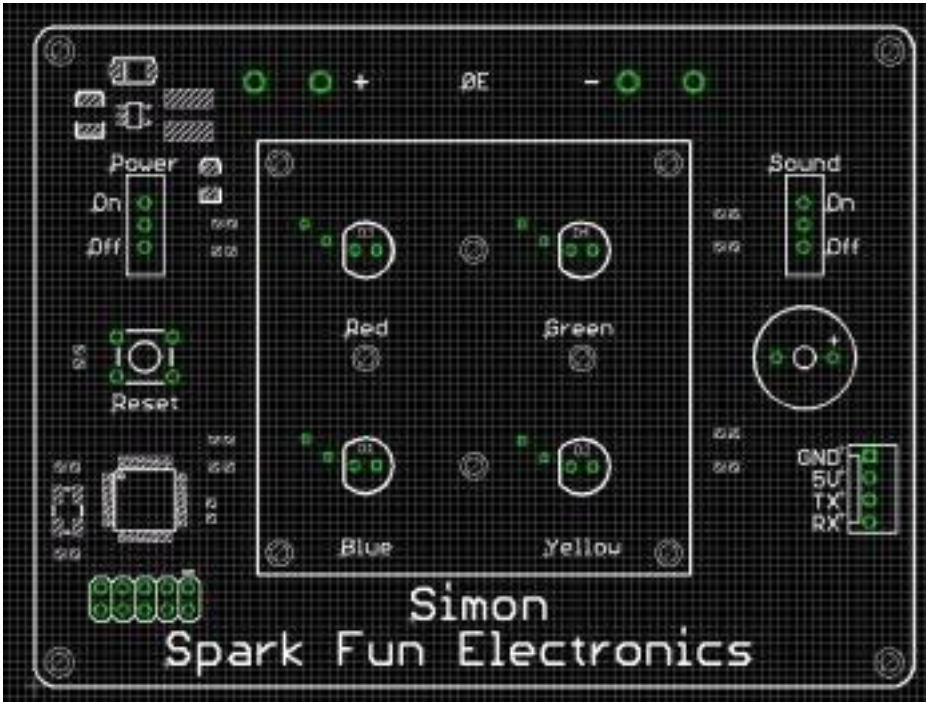
- 1x Simon PCB
- 2x AA Pil
- 1x Buzzer
- 2x Açma-Kapama Anahtarı
- 1x ISP Header
- 4x LED (Sarı, Mavi Kırmızı, Yeşil)
- 6x Vida
- 6x Plastik Ayaklar
- 1x Reset Butonu

Gerekli Olan SMD Malzemeler;

- 5x 10k Direnç
- 4x 220 Ohm Direnç



Tüm katmanlar ve Yollar

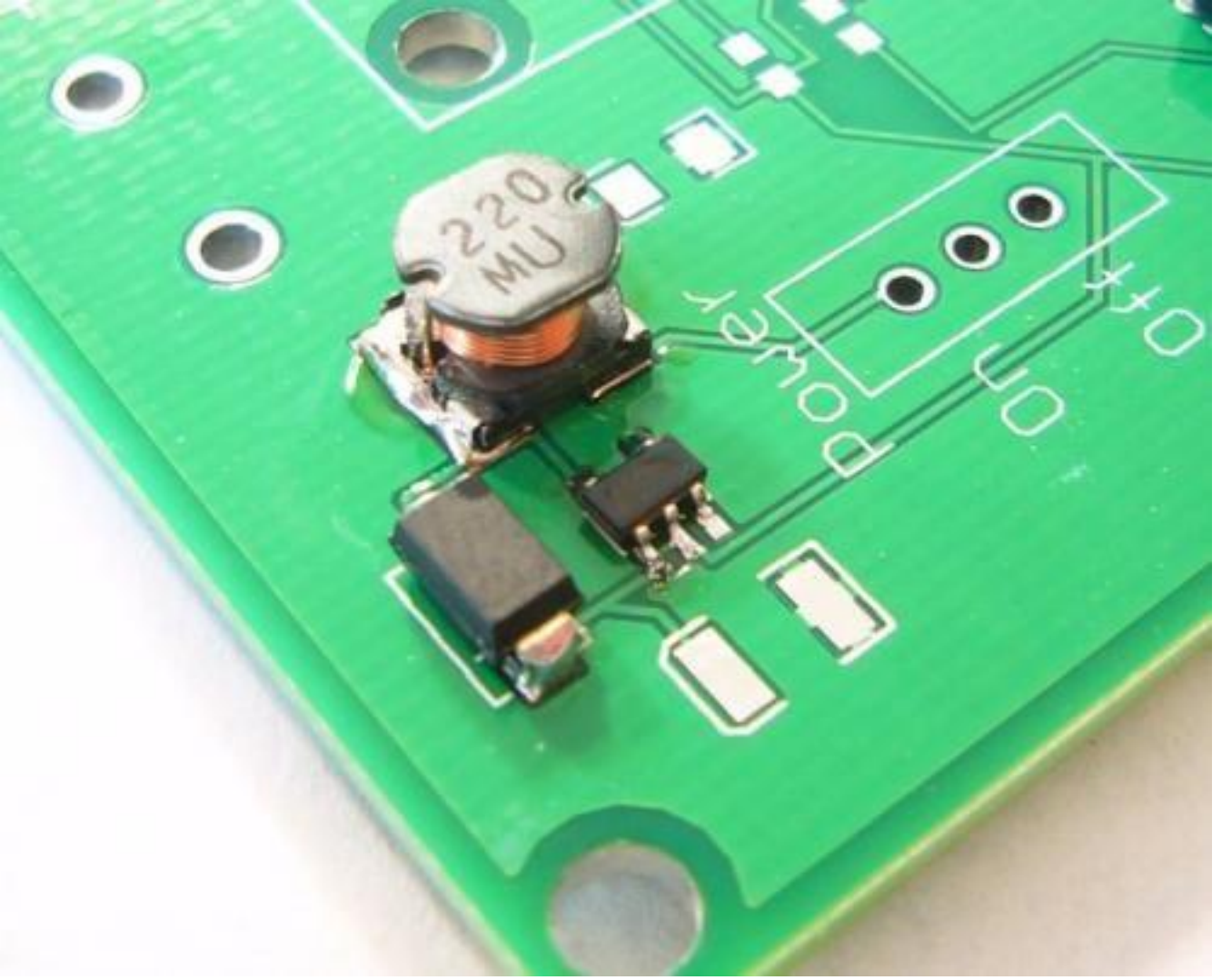


Sadece Lehimlenecek Malzemelerin Yerleşim Planı

Lehimleme işlemine başlamadan önce sadece lehimlenecek malzemelerin olduğu yerleşim planını, 3 kat daha fazla büyüterek çıktı almanız yararınıza olacaktır. Bu şekilde malzemeler hem daha büyük ve anlaşılır şekilde görünmektedir hem de bir karışıklık olması büyük ölçüde önlenmektedir.

DIP ve SMD malzemelerin beraber olduğu bir PCB de öncelik her zaman daha küçük boyutlu malzemelerin olmalıdır. Biz ilk başta boost devresinden başlıyoruz.

Boost devresi : Gerilim kazancı sağlayıp akımdan kayıp sağlayan bir yöntemdir. Bu şekilde 1.5v luk pili 5volta çıkarıyoruz. Bu gerilim ise Atmega168'in çalışması için yeterlidir. Fakat bu yöntem sonucunda 1.5voltluk pilin ömrü büyük ölçüde azalmaktadır.



NCP1400, Bobin ve Diyot Üçlüsü

Bu malzemelerin nasıl lehimlendiği hakkında bir video;

http://www.youtube.com/playlist?list=PLfcIZXsDLA1_6d_50JFv19ZdBzhj4AIB1

abone 27.327 görüntüleme Video Yöneticisi Herkese açık olarak görüntüle

ÇİZGİ TAGEM Bilgi Paylaştıkça Çoğalır

Niyazi Saral

Videolar Tartışma Hakkında

Oynatma listeleri Eklenme tarihi (en yenden en eskiye)

Favori videolar 9 video

Soldering (Through Hole) 24 video | 4 hafta önce

Soldering (Surface Mount Technology) 103 video | 4 hafta önce

Sodlering (BGA) 105 video | 4 hafta önce

Bilgisayar Donanım Elemanı Eğitimi 44 video | 4 hafta önce

Çizgi RMA 20 video | 1 yıl önce

İleri Bilgisayar Donanım Eğitimi (Sieguntur-Türkçe) 20 video | 3 hafta önce

NoteBook Sevis Videoları 15 video | 3 hafta önce

Teknik Servis Eğitimleri (Türkçe) 22 video | 3 hafta önce

CompTIA Network+ (Türkçe) 45 video | 3 hafta önce

Havyanızın ucunu süngerle temizleyin ve ucuna bir miktar lehim alın. Sonra ucundaki lehim devre üzerinde lehimlenecek olan padlere sürün. Sonra cımbız ile bu malzemenizi lehimlenecek yere doğru bir şekilde yerleştirdikten sonra havya ile önceden sürdürdüğünüz lehim ısıtın ve malzemenizi lehimleyin.

http://www.youtube.com/playlist?list=PLfcIZXsDLA1_6d_50JFv19ZdBzhj4AIB1

abone 27.327 görüntüleme Video Yöneticisi Herkese açık olarak görüntüle

ÇİZGİ TAGEM Bilgi Paylaştıkça Çoğalır

Niyazi Saral Ayarlar

Videolar Tartışma Hakkında

Oynatma listeleri Ekleme tarihi (en yeniden en eskiye)

Favori videolar 9 video

Soldering (Through Hole) 24 video | 4 hafta önce

Soldering (Surface Mount Technology) 103 video | 4 hafta önce

Soldering (BGA) 105 video | 4 hafta önce

Bilgisayar Donanım Elemanı Eğitimi 44 video | 4 hafta önce

Çizgi RMA 20 video | 1 yıl önce

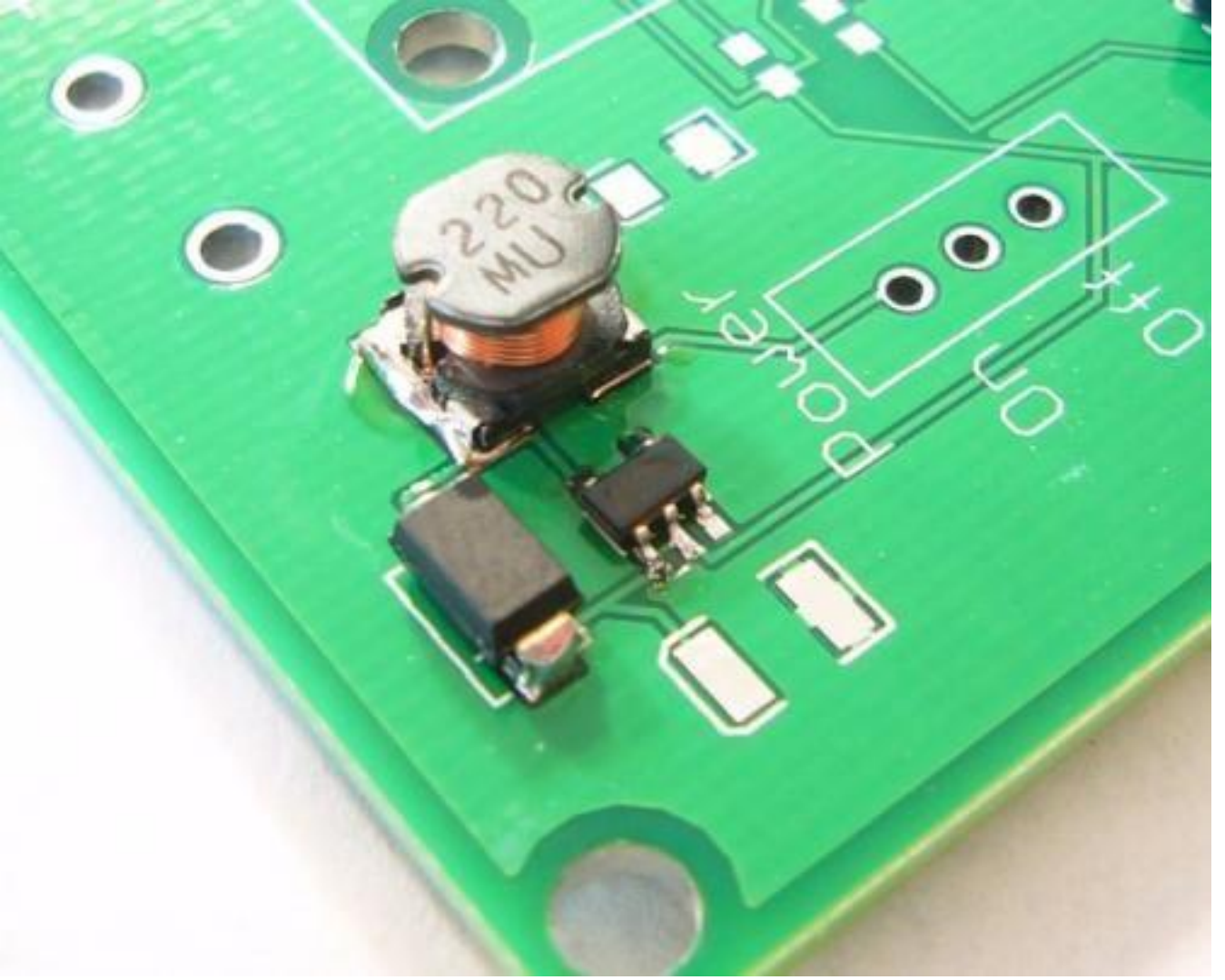
İleri Bilgisayar Donanım Eğitimi (Sieguntur-Türkçe) 20 video | 3 hafta önce

NoteBook Sevis Videoları 15 video | 3 hafta önce

Teknik Servis Eğitimleri (Türkçe) 22 video | 3 hafta önce

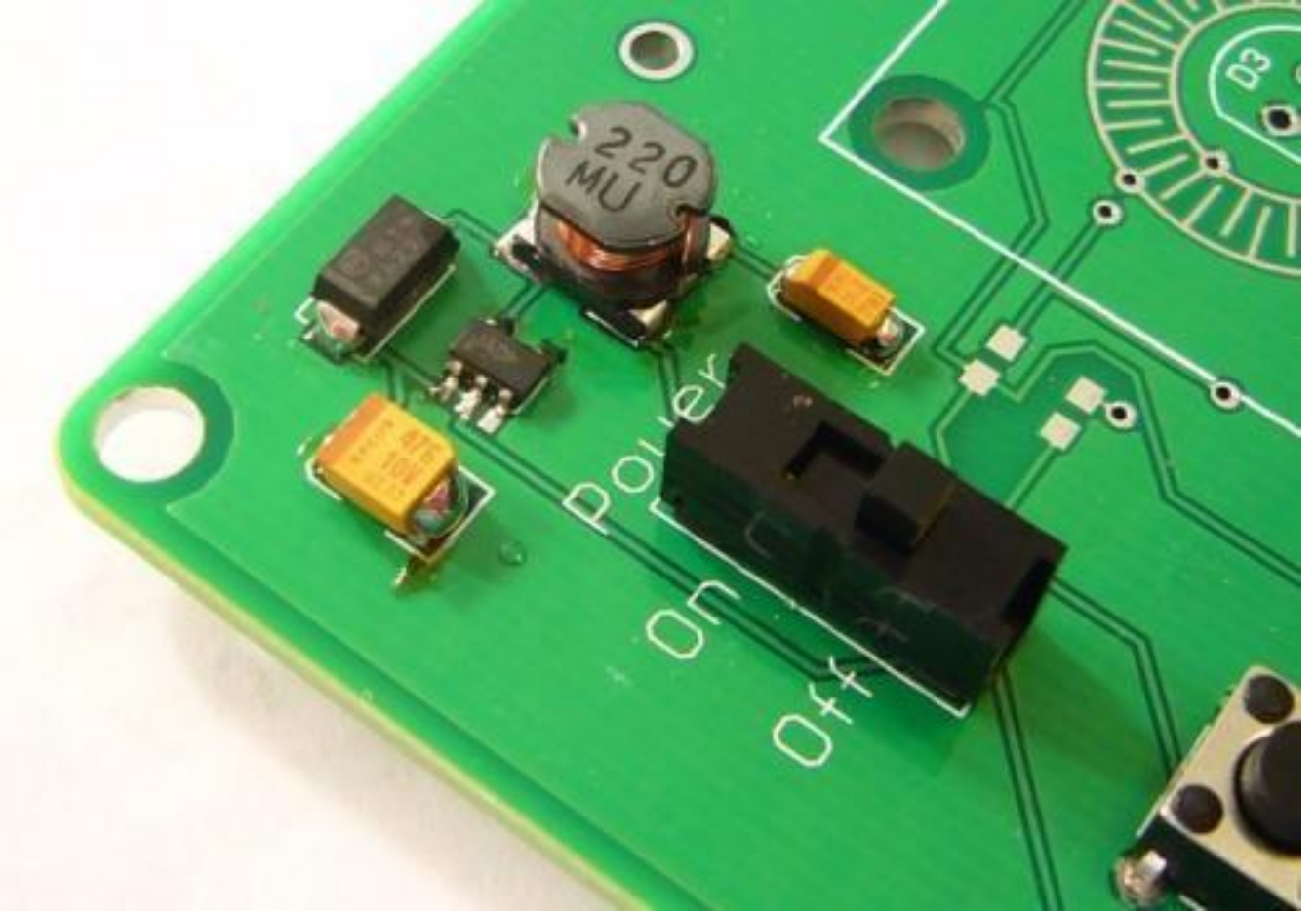
ComptIA Network+ (Türkçe) 45 video | 3 hafta önce

Diyotu lehimledikten sonra yerleşim planına tam oturduğundan emin olunuz. Aksi halde düzgün olmadığı için göze pek de hoş gelmeyecektir. Eğer düzgün değilse havaya ile diyotun bacaklarındaki lehimini eriterek tekrar düzeltebilirsiniz. Bu işlemi de tamamladıktan sonra diğer malzeme olan bobini lehimleme işlemine geçebiliriz.

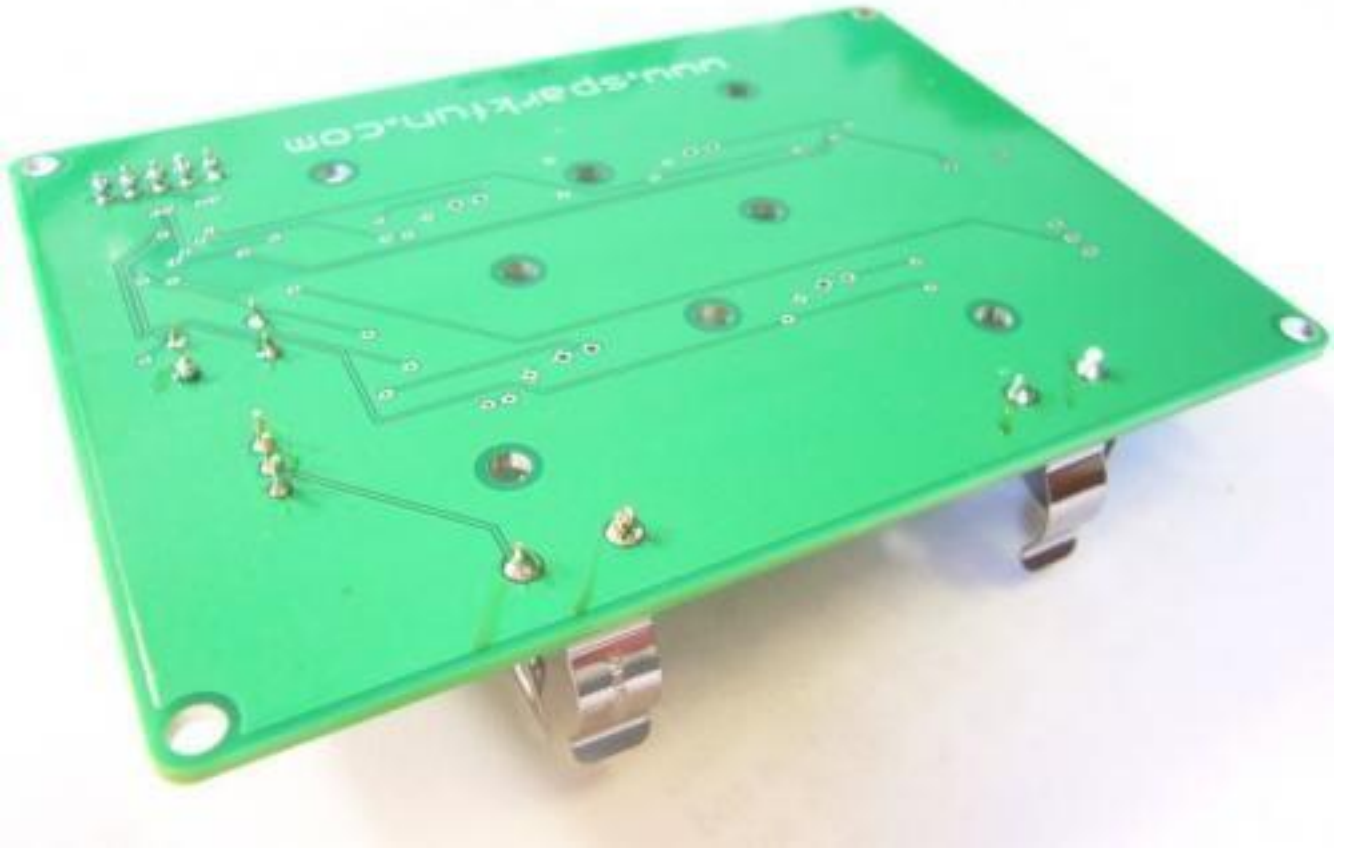


NCP1400, Bobin ve Diyot Üçlüsü

SMD bobini lehimlemek biraz zor bir iştir. Çünkü lehimlenecek yüzey çok dardır ve pad sayısı fazladır. İlk padin doğru bir şekilde lehimlendiğinden eminseniz diğer padlere geçebilirsiniz.



**47uF ve 10uF Kondansatörlerin lehimlenmesi. Lehimlerken + ve - uçlara dikkat ediniz.
Ve Açma-Kapama Anahtarının Lehimlenmesi.**



AA Pil Yatağının Lehimlenmesi. Lehimlerken + ve - uçlara dikkat ediniz.

Pil yatağını da lehimledikten sonra içerisine bir adet AA pil takıp boost devremizin doğruluğunu test edebilirsiniz. Bunun için boost devresinin çıkışını AVO metre ile ölçmeniz gerekmektedir. Eğer çıkışta 4.8v - 5.2v arası bir gerilim görüyorsanız boost devreniz doğru çalışmaktadır. Eğer 5volta yakın bir değer göremiyorsanız boost devrenizi tekrar kontrol edin ve malzemeleri doğru yerleştirdiğinizden ve kısa devre olmadığından emin olun.

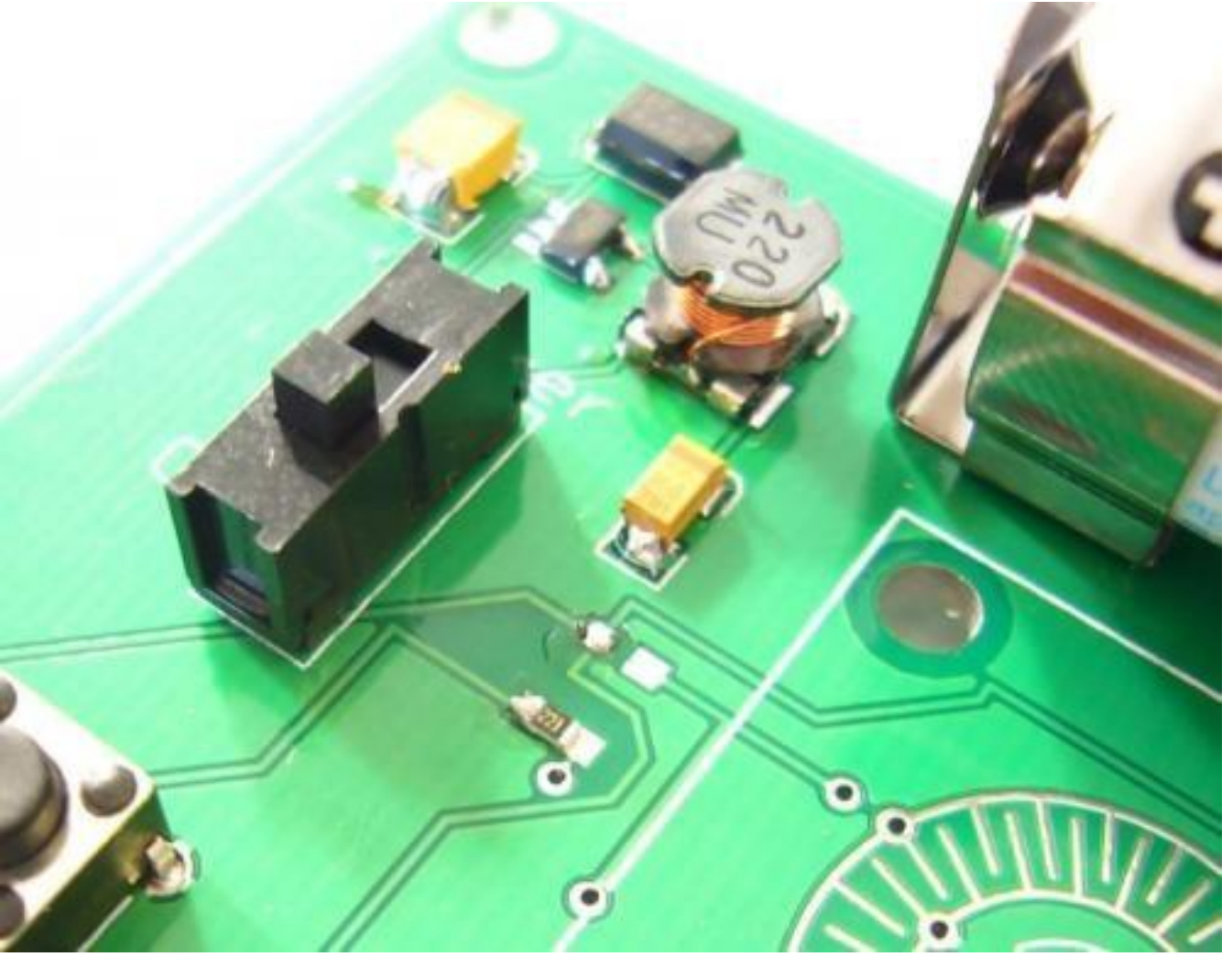
TQFP Kılıftaki Atmega168'in lehimlenmesi;

http://www.youtube.com/playlist?list=PLfcIZXsDLA1_6d_50JFv19ZdBzhj4AIB1

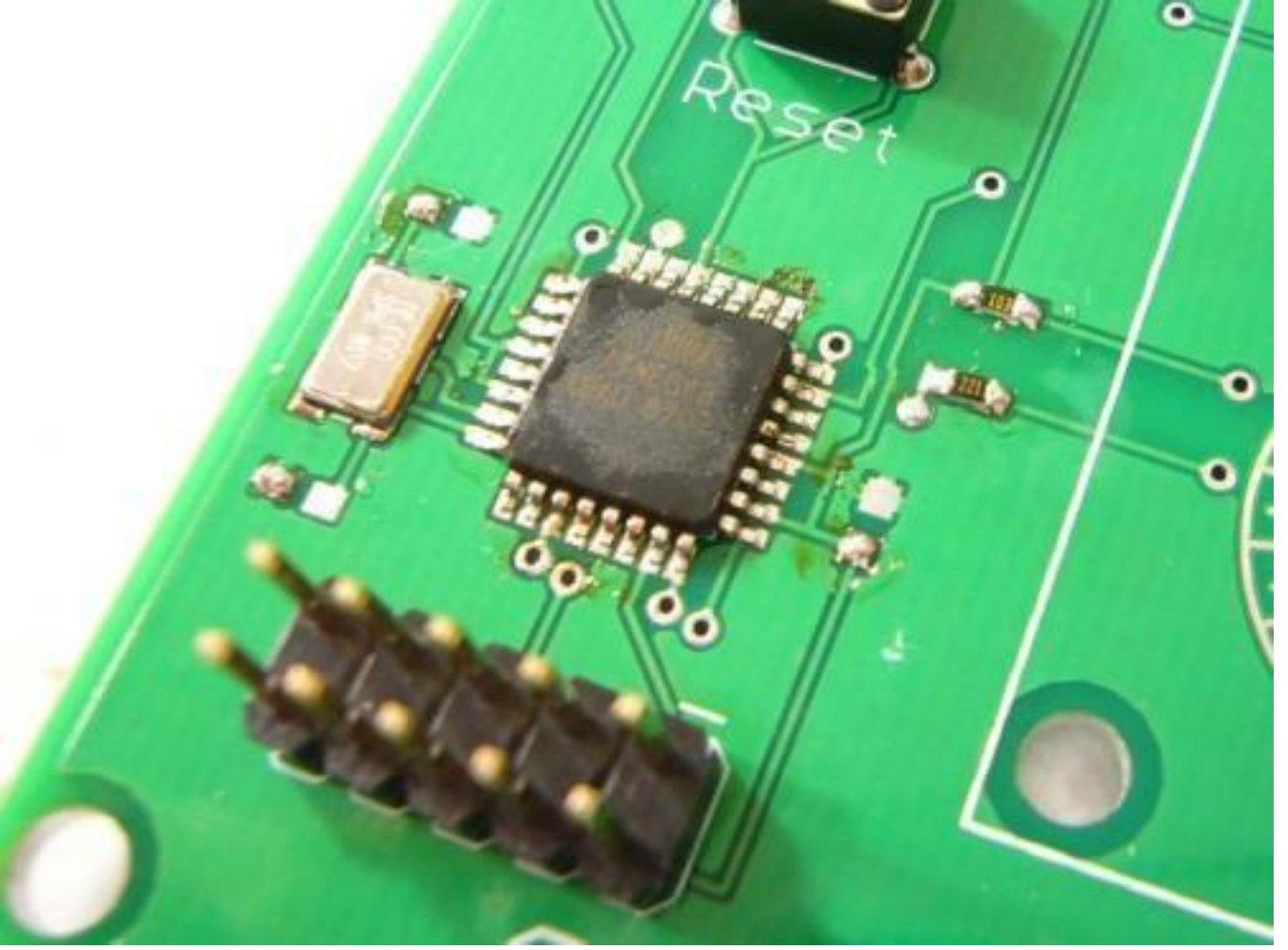
Malzemeyi lehimlemeden önce 2-3 pade bir miktar lehim akıtın ve malzemeyi doğru bir şekilde yerleştirdikten sonra lehimli padler ile pinleri lehimleyin. Ve entegre tam olarak yüzeye oturunca diğer padleri lehimleyin.

Atmega168'i lehimleme işlemi tamamlandıktan sonra 16MHz lik kristali lehimleyebiliriz.

Kristali lehimlerken dikkatli olmalısınız. Çünkü kristaller hassas olduğundan dolayı fazla ısıya maruz kalırlarsa kimyasal yapısının bozulması sonucu doğru çalışmayacaklardır.



220 Ohm Direncin Lehimlenmesi



Atmega168, 220 Ohm ve 0603 Dirençlerin Lehimlenmiş Hali

Lehimleme işlemi sırasında PCB üzerinde lehimin içerisindeki lehim pastasından dolayı kahverengi lekeler oluşacaktır. Bunun için endişelenmeyin. Bütün malzemeleri lehimledikten sonra PCB yi flux ile temizlerseniz bu lekeler ortadan kalkacaktır.

SMD malzemeler bittikten sonra DIP malzemeleri lehimleyebiliriz.

ISP Konektörü lehimlerken PCB yi ters çevirip ağırlığından faydalanabiliriz. PCB'nin ağırlığından dolayı ISP konektörünün pinleri, lehimlemek için uygun pozisyona gelecektir.

Sırada reset butonunu lehimleme işlemi var.

Reset butonunu da lehimledikten sonra diğer malzemeleri(Buzeer, Ses Açma-Kapama Anahtarı...) bir önceki derste anlatıldığı gibi lehimleyiniz.



Son olarak LED'leri lehimleyeceğiz. Eğer 4 tane LED'in 4'ü de şeffaksa ve hangi renk olduğunu bilmiyorsanız AVO metrenizi diyot ölçüm konumuna getirip uzun bacağa kırmızı, kısa bacağa siyah probu takarak hem LED'in çalışıp çalışmadığını hem de çalışıyor ise hangi renk olduğunu öğrenebilirsiniz.

Bu işlemi de tamamladıktan sonra sıra Atmega168'i programlaya geliyor. Atmega168'i programlamadan önce fuse bitlerinden 16MHz harici kristal ayarının seçildiğinden emin olmalısınız. Bunun için avrdude programına aşağıdaki komutu yazabilirsiniz.

- `avrdude -p atmega168 -P lpt1 -c stk200 -U lfuse:w:0xff:m -U hfuse:w:0xc9:m`

Bu linkte Atmega168 için Simon'un yazdığı bir firmware bulunmaktadır. Simon.c dosyasını Programmer's Notepad ile açıp Board üzerine yükleyebilirsiniz. Firmware yükleme işlemi tamamlandıktan sonra devreyi kapatıp tekrar açarsanız LED'lerin yandığını görürsünüz. Eğer LED'ler yanmıyorsa;

1. LED'leri Anot ve Katot olmak üzere doğru bağladığınızdan emin olun.
2. Firmware'in doğru bir şekilde board üzerine yüklendiğinden emin olun.



PCB'nin Tüm Malzemeleri Lehimlendikten Sonra Çalışır Haldeki Resmi

Devreyi çalıştırdıktan sonra sağlamlık açısından son birkaç test daha yapmanız gerekli.

- Vcc ve GND arasında kısa devre olup olmadığını kontrol edin.
- Vcc geriliminin her zaman 5volta yakın değerde olup olmadığını kontrol edin.
- ISP pinleri arasında kısa devre olup olmadığını kontrol edin.

Eğer bu koşullardan herhangi biri devrenizde sağlanmamışsa ve devreniz çalışıyorsa muhtemelen lehimleme sırasında yollardan birine zarar verip aradaki bağlantının kopmasını sağlamışsınızdır.

Bu dersin tüm malzemelerini buradan edinebilirsiniz.

Ders 8 - Eagle Şematik Editörü

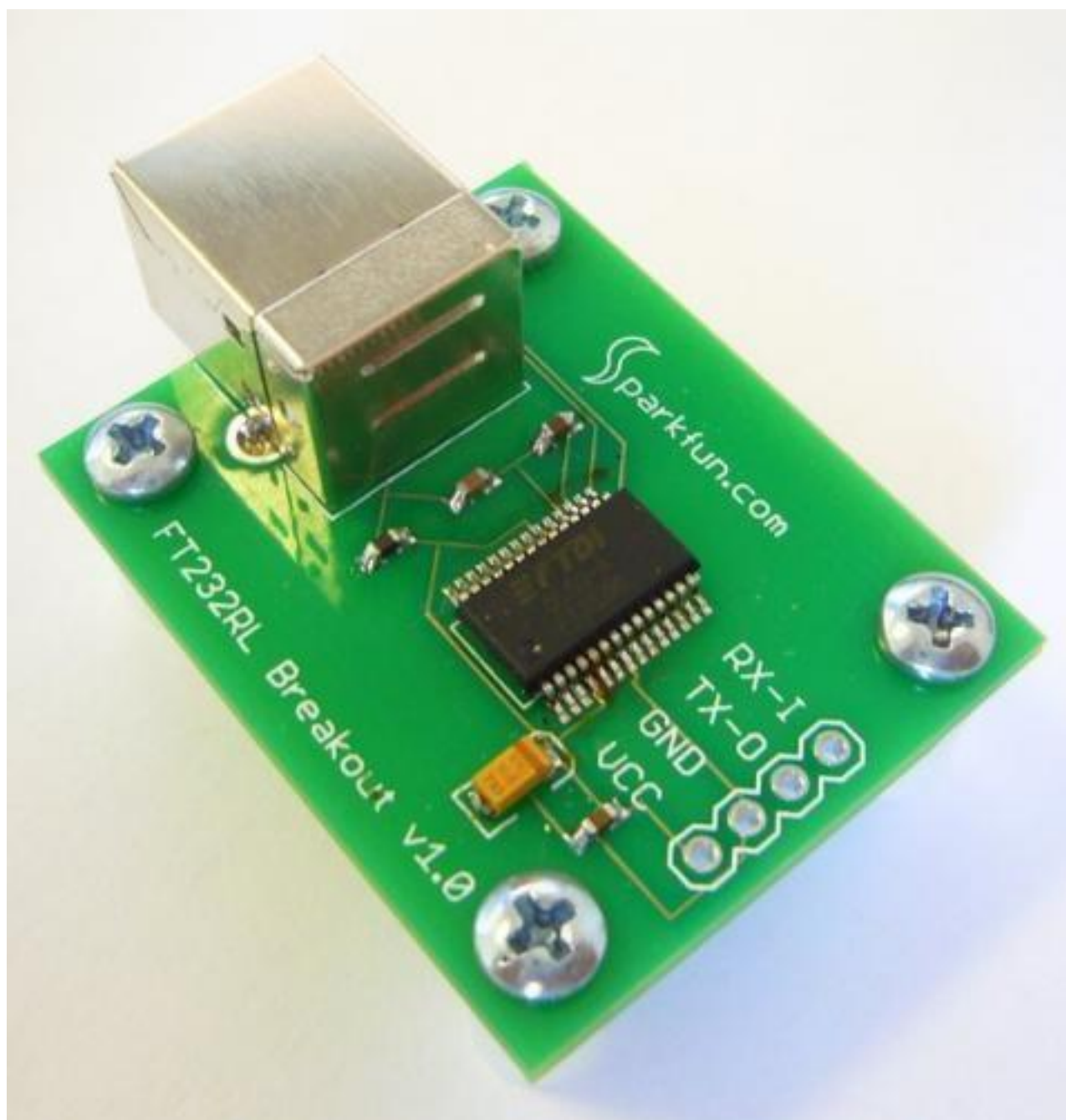
Bu dersimizde Eagle Şematik Editörü'nün nasıl kullanıldığını inceleyeceğiz. Eagle, başlangıç için ucuz ve hızlı bir seçimdir. Ve oldukça yeteneklidir de.

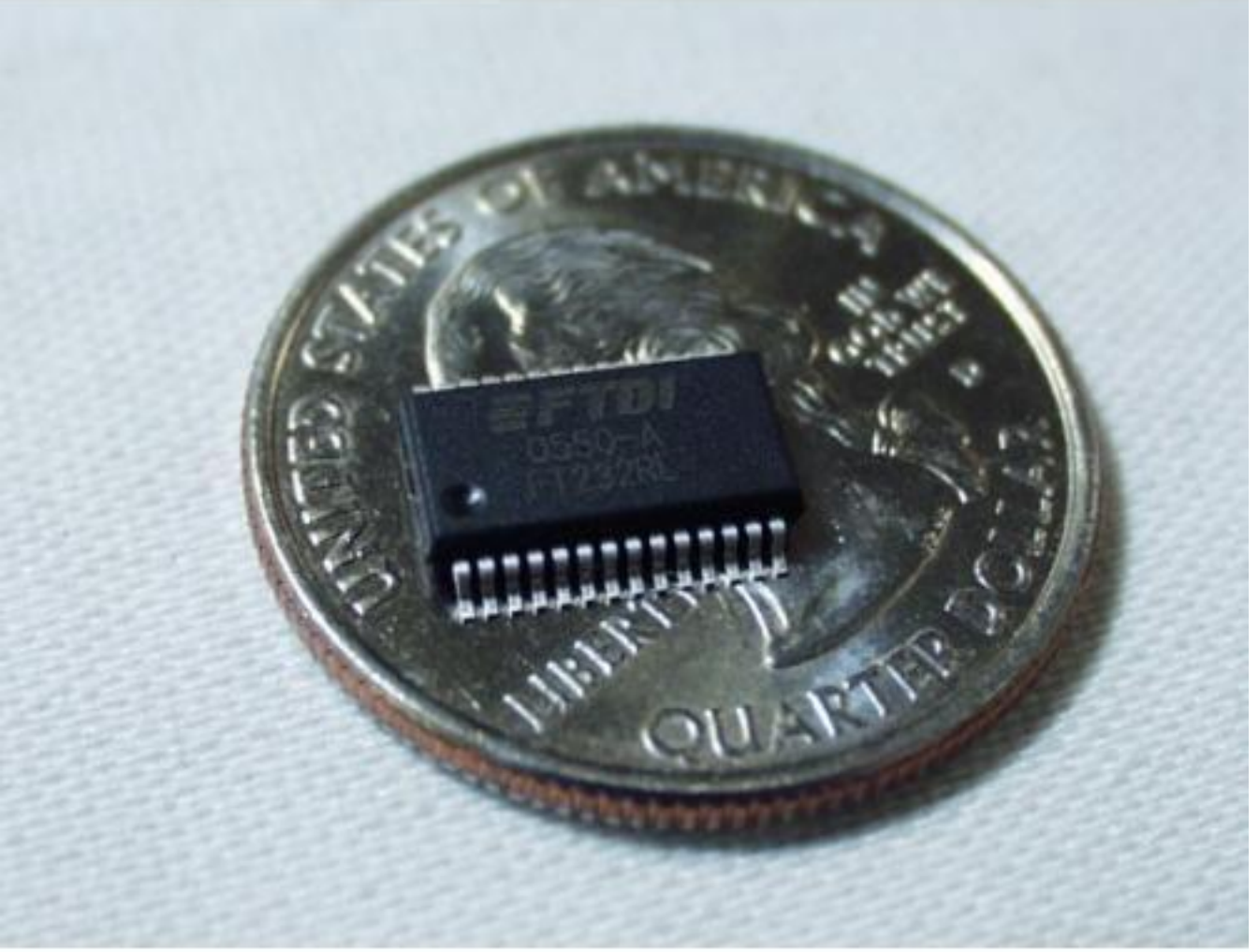
Cadsoft firması kişisel kullanım için sadece bir adet şema ve PCB dosyası açabileceğiniz kısıtlı bir versiyon sunmaktadır. Diğer tüm özellikler FULL sürümdeki ile aynıdır.

Bu ders için aşağıdaki dosyaları bilgisayarınıza yüklemeniz gerekmektedir.

1. Eagle programını indirin ve kurun. Windows, Linux ve Mac için farklı sürümler mevcuttur.
2. SparkFun Eagle Kütüphanesini indirin ve .lbr uzantılı dosyayı Eagle/lbr dizinine kopyalayın. Bu kütüphanede SparkFun'ın hazırlamış olduğu boardların footprintleri de bulunmaktadır.
3. Eagle Klavye Kısayollarını indirin ve .scr uzantılı dosyayı Eagle/scr dizinine kopyalayın.
4. Sparkfun CAM dosyasını indirin ve ..cam uzantılı dosyayı Eagle/cam dizinine kopyalayın.

Bu işlemleri yaptıktan sonra Eagle programının kullanımı basit bir USB-TTL(FT232RL) dönüştürücü tasarlayarak inceleyelim.





USB-TTL Dönüştürücü Nedir ?

FT232RL'yi bilgisayarınıza bağladığınızda bu donanımın tanınması için FTDI firmasının, sitesinde vermiş olduğu Virtual COM Port sürücülerini indirmek ve bu donanımı bilgisayara tanıtmak gerekiyor. Eğer bu donanımı bilgisayara başarılı bir şekilde tanıttıysanız bilgisayar bunu bir sanal seri port olarak görüyor ve HyperTerminali açıp uygun portu seçtiğinizde gönderdiğiniz veriler Tx pininden seri olarak gönderilmeye başlıyor ve gelen bilgiler ise Rx pininden yine aynı şekilde seri olarak alınabiliyor. Bu entegrenin en güzel yanı ve şüphesiz onu popüler kılan en önemli şey ise USB hakkında hiçbir şey bilmeseniz bile üzerinde USB portu olan ve USB'den çalışabilen cihazlar tasarlayabilmenizdir.

- [FT232R Sayfası](#)
- [FT232R Datasheet](#)
- [FT232R VCP\(Virtual COM Port\) Sürücüler](#)

Aşağıda standart uygulamalar için FTDI firmasının önerdiği devre şeması bulunmaktadır.

7. Device Configurations

Please note that pin numbers on the FT232R chip in this section have deliberately been left out as they vary between the FT232RL and FT232RQ versions of the device. All of these configurations apply to both package options for the FT232R device. Please refer to [Section 4](#) for the package option pin-out and signal descriptions.

7.1 Bus Powered Configuration

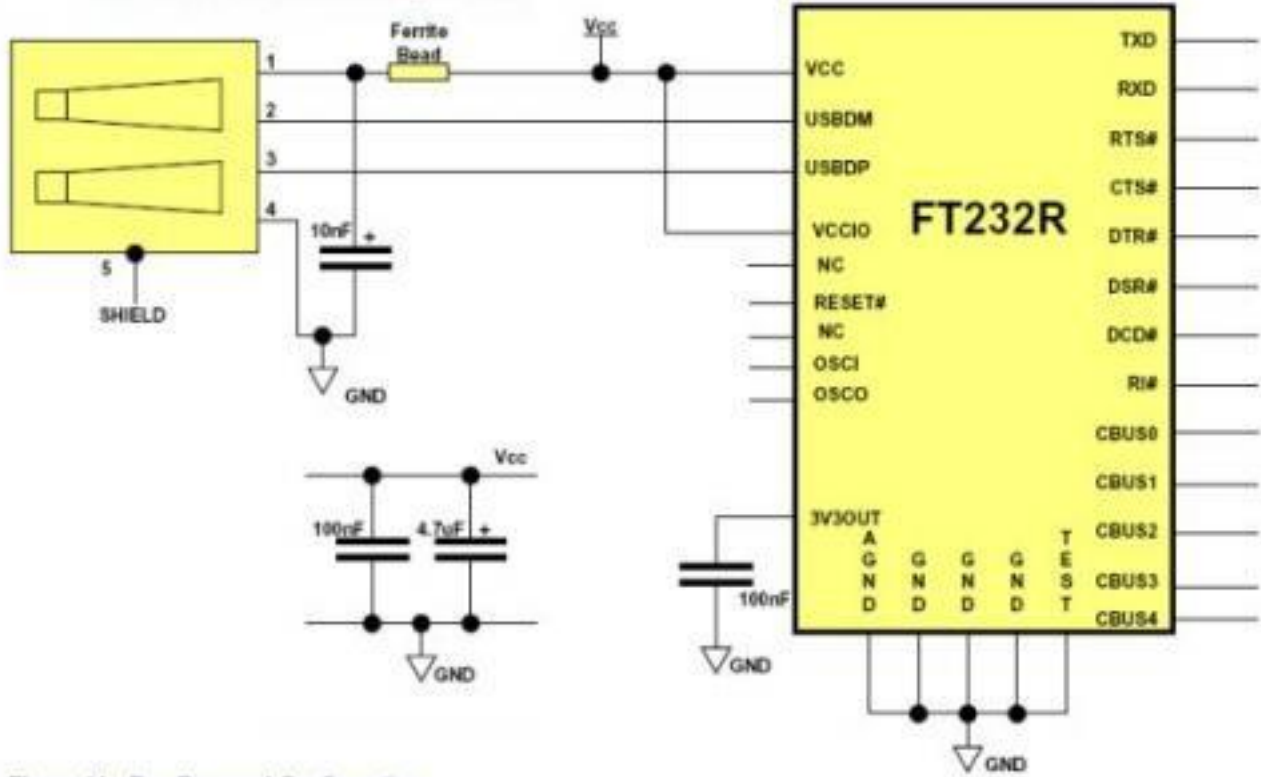
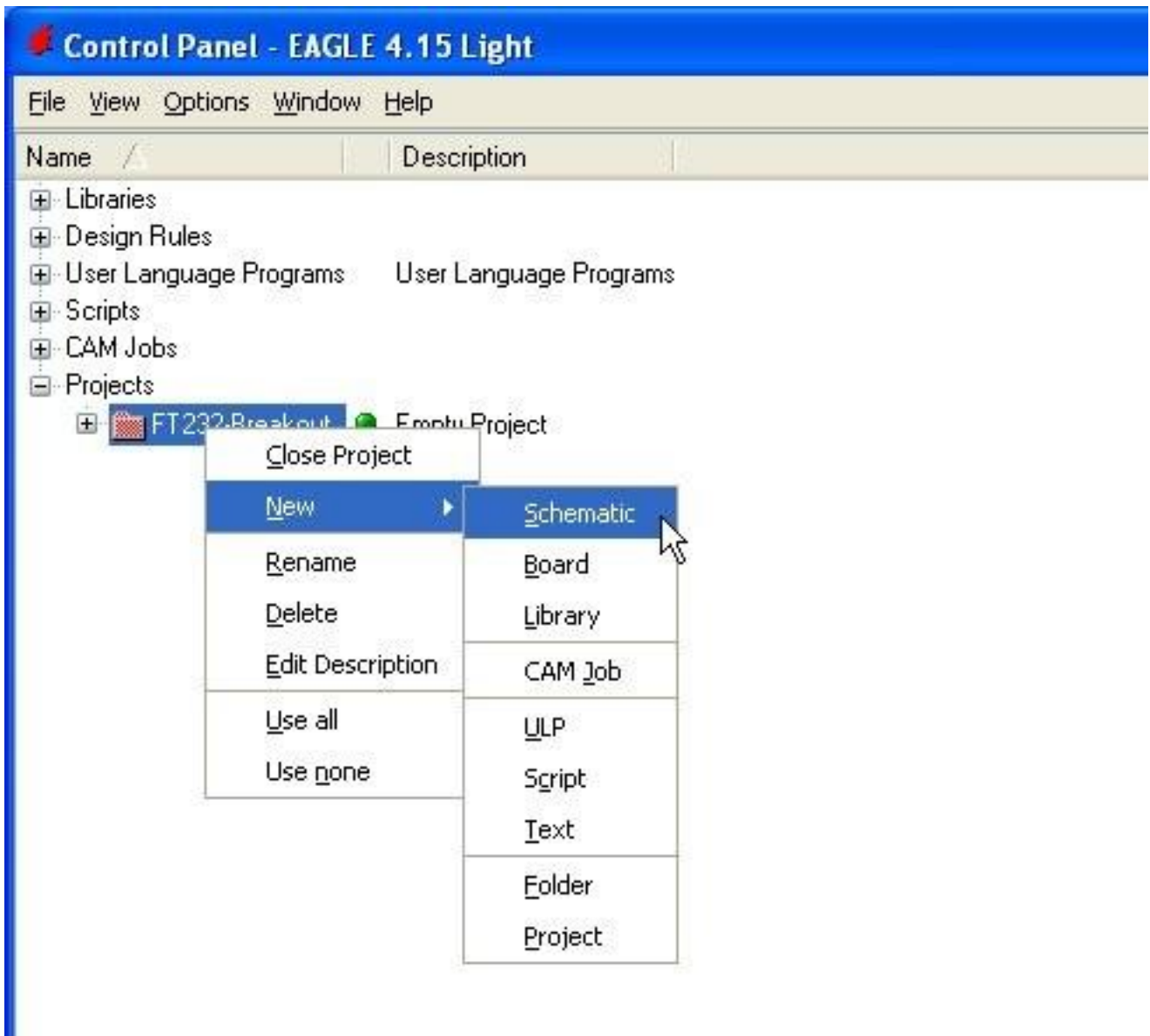
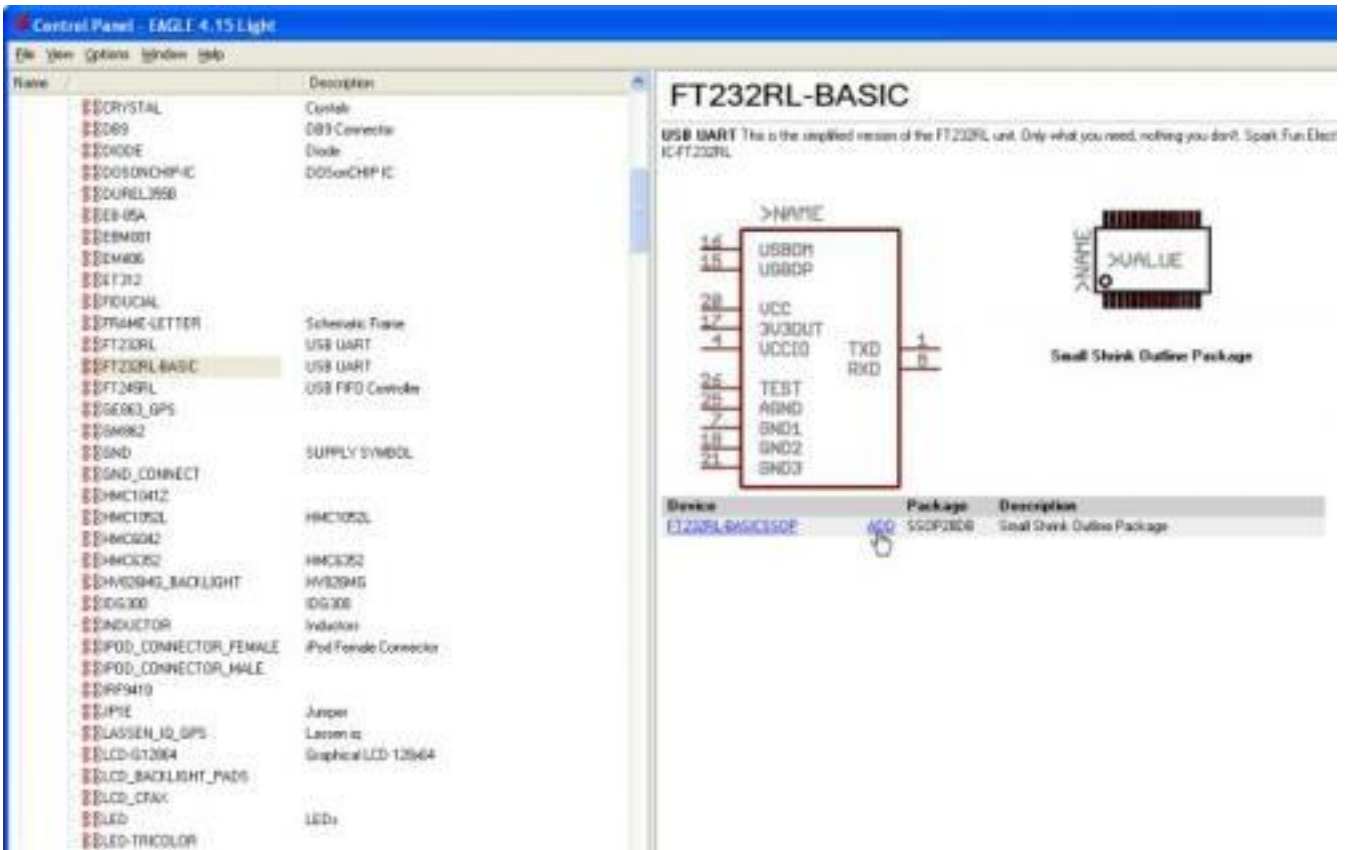


Figure 11 - Bus Powered Configuration

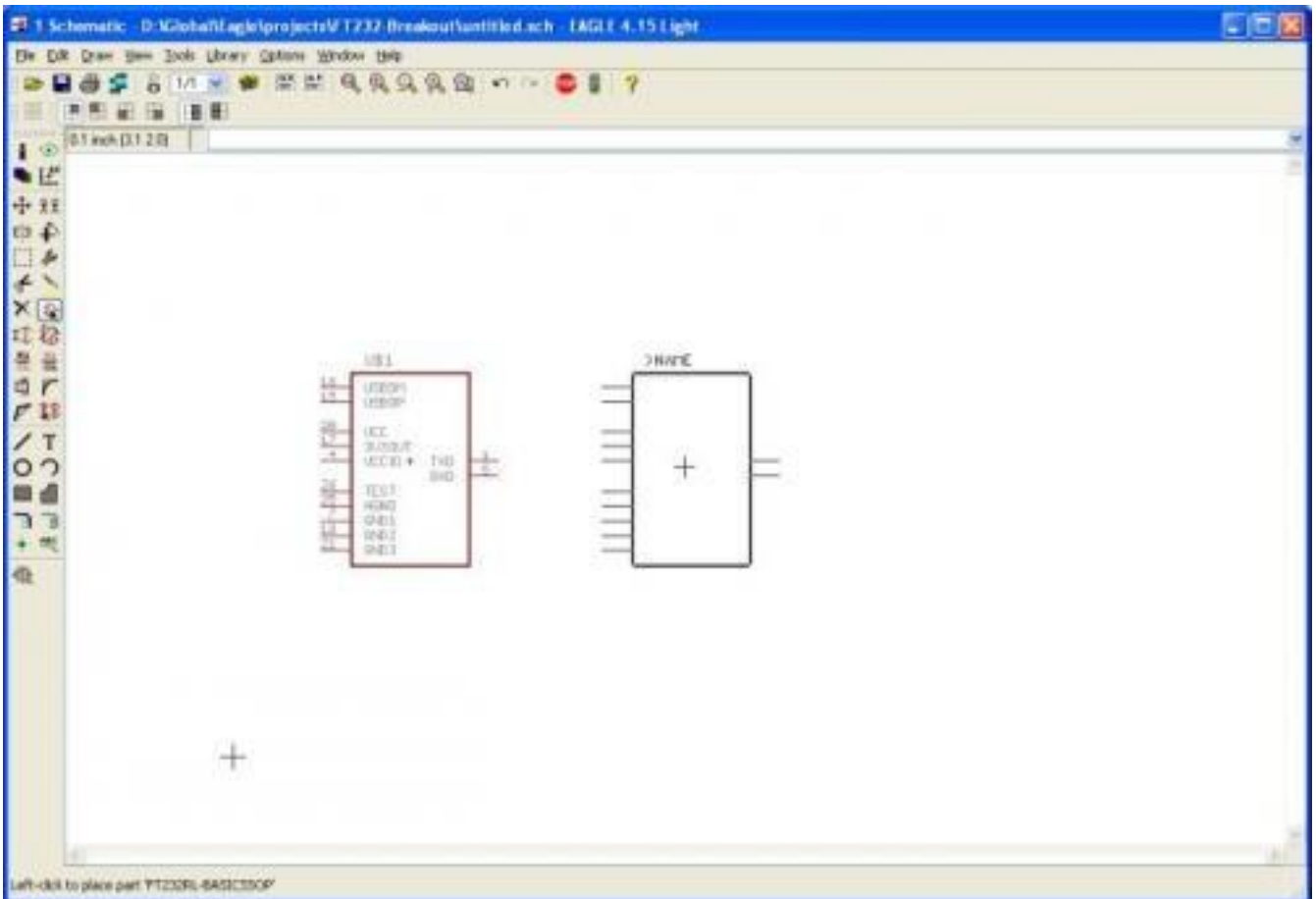
Şimdi FT232RL entegresini şemamıza ekleyelim. Bu entegreyi önceden yüklediğiniz SparkFun kütüphanesinde bulabilirsiniz. İlk olarak ise her şeyi kapatıp File->New->Project sekmesinden yeni proje oluşturup 'FT232-Breakout' olarak adlandıralım. Sonra aşağıdaki resimde görüldüğü gibi bu projeye sağ tıklayıp New->Schematic diyerek yeni bir şema oluşturalım.



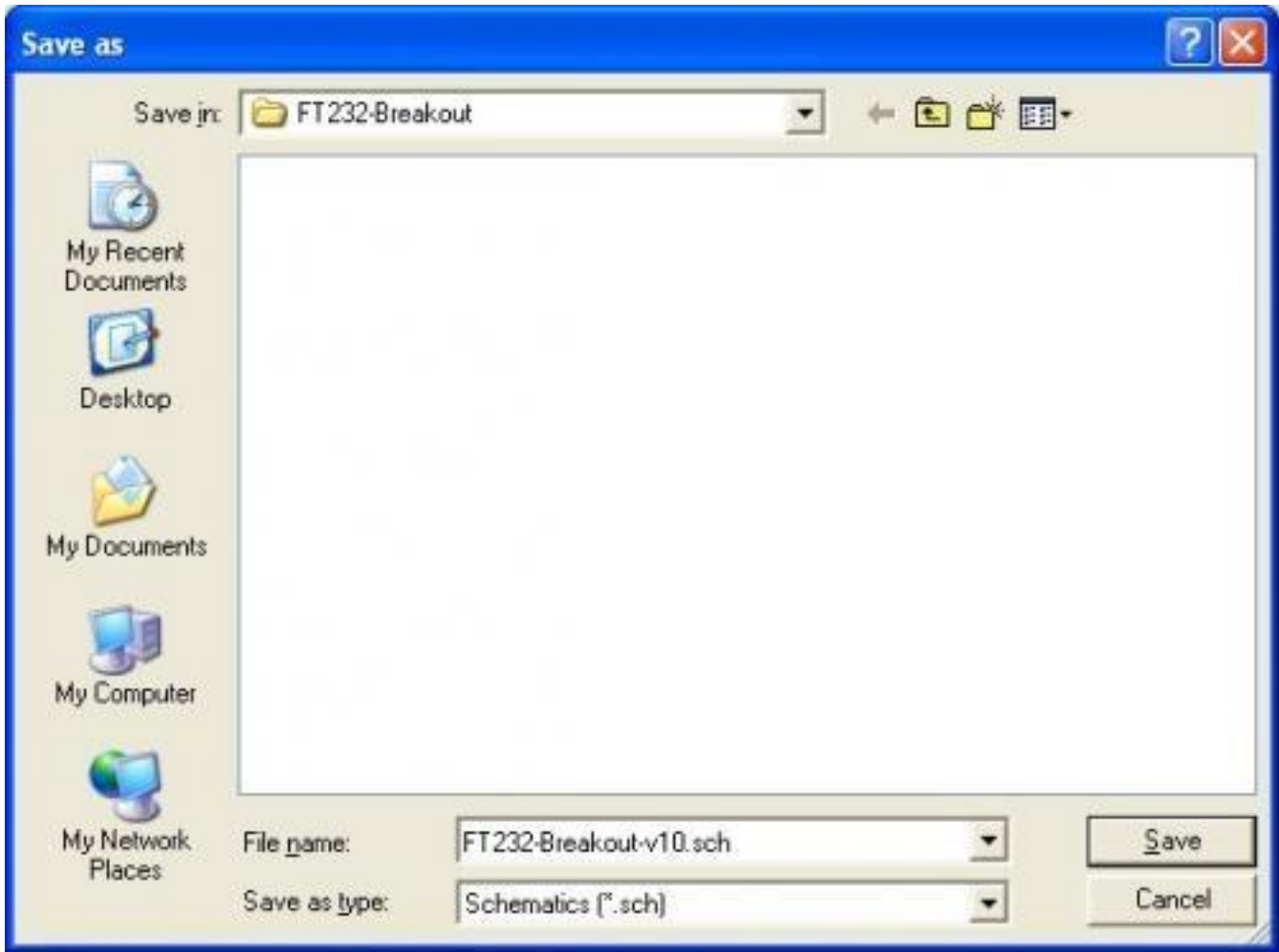
Şema Editörünü açtıktan sonra tekrar Library kısmına gelerek SparkFun kütüphanesini açın.



Buradan FT232RL-Basic parçasını bularak ADD tuşuna basın ve şemanıza ekleyin.



Sonra şemanızı kaydedin.



Sonra aşağıdaki parçaları da şemanıza ekleyin.

- 3x CAP (Parça adı : CAP0603) : 0.1uF/0.01uF 0603 Kondansatörler
- 1x CAP_POL (Parça adı : CAP_POL1206) : 10uF tantalyum kondansatör
- 1x INDUCTOR (Parça adı : INDUCTOR0603) : Ferrite Bead
- 4x STAND_OFF (Parça adı : STAND-OFF) : Vidalar için delik ölçüsü
- USB (Parça adı : USBPTH) : USB B Tipi konektör
- M04 (Parça adı : M04PTH) : 4 Pin 0.1" konektör
- GND
- VCC



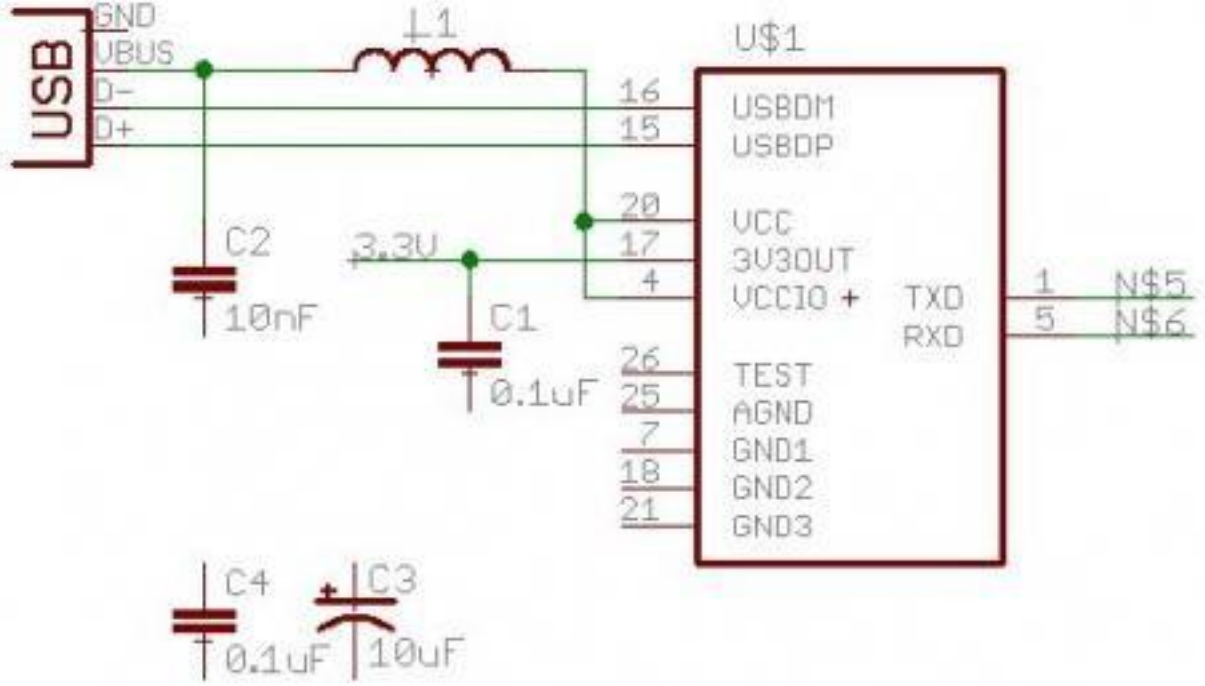
Info			Show
Display			Mark
Move			Copy
Mirror			Rotate
Group			Change
Cut			Paste
Delete			Add
Pinswap			Gateswap
Name			Value
Smash			Miter
Split			Invoke
Wire			Text
Circle			Arc
Rectangle			Polygon
Bus			Net
Junction			Label
ERC			

Soldaki menüden amacınıza göre açıklaması verilen butonları kullanabilirsiniz. Veya kısayol tuşlarını kullanarak da aynı işlemleri gerçekleştirebilirsiniz. Aşağıda, şema oluşturmak için gerekli olabilecek kısayol tuşları verilmiştir.

- Escape tuşu herhangi bir yerden çıkmak için kullanılır
- F7 bir parçayı taşımak için kullanılır
- Alt+F7 bir grup parçayı taşımak için kullanılır
- F3 bir parçayı siler
- F4 bir parçanın ismini değiştirir
- F5 parçanın değerini değiştirir (0.1uF kondansatörü 0.01uF yapmak gibi)
- F6 malzemeyi daha küçük parçalara ayırır

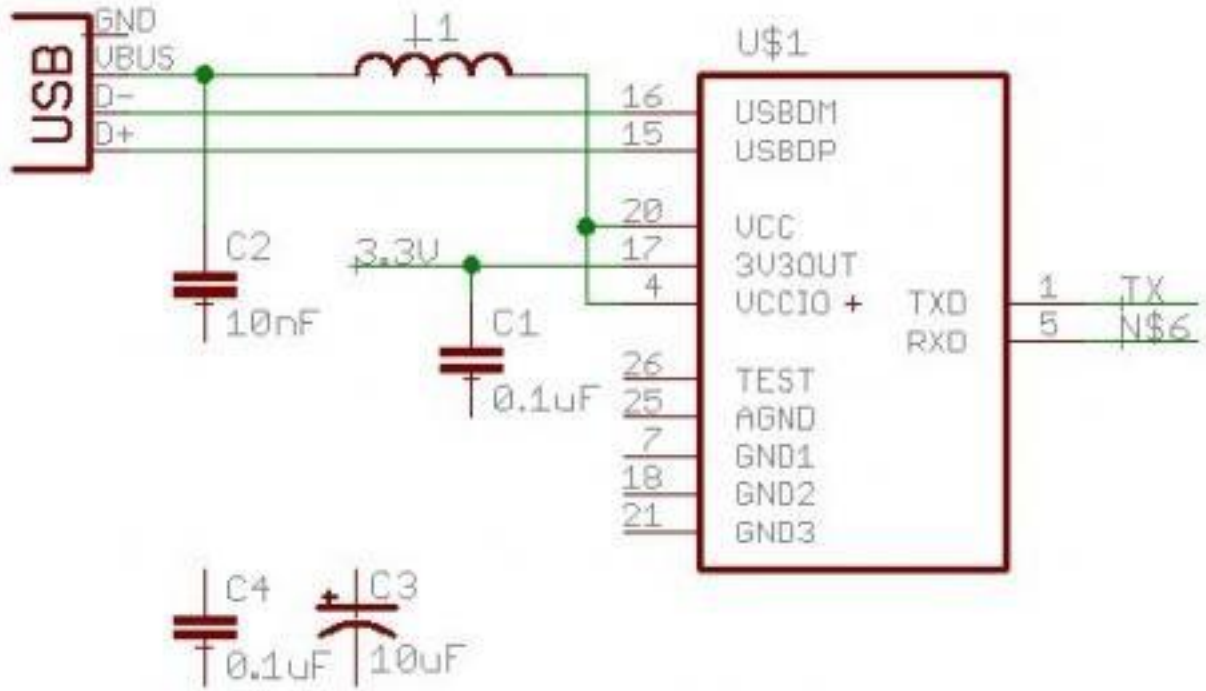
- F9 bağlantı(kablo) oluşturur
- Alt+F9 bu bağlantıya bir isim verir

Klavye kısayollarını öğrendikten sonra oluşturmanız gereken şema için aşağıdaki bağlantıları yapmanız gerekmektedir.



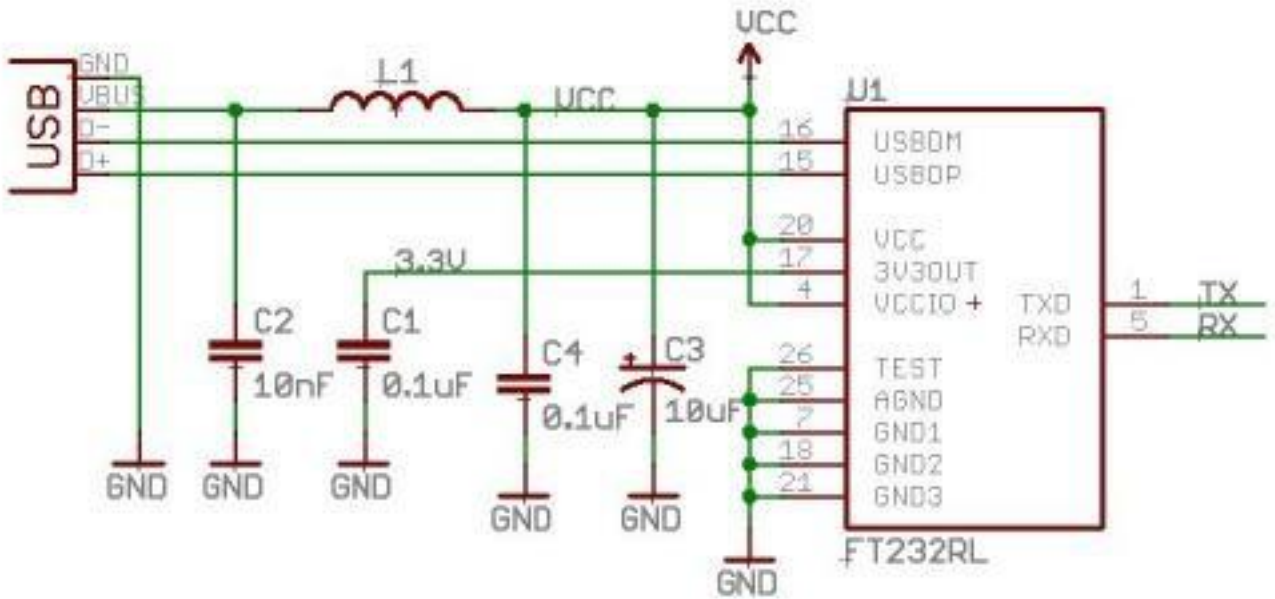
FT232RL Breakout Board

Sonra FT232RL Entegresinin 1. ve 5. pinlerine sırasıyla TX ve RX ismini verin. Bunun için F4 kısayol tuşunu kullanabilirsiniz.



Sonra ismi U\$1 olan FT232RL entegresinin ismini karışıklık olmaması için U1 olarak değiştirin. (F4).

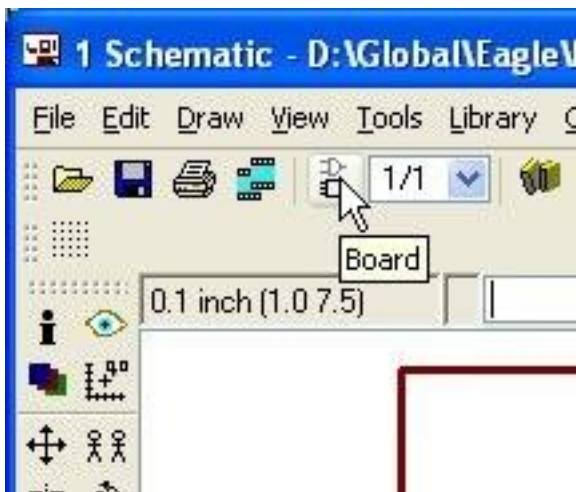
Sonra F7 kısayol tuşunu kullanarak alttaki iki konsatörü yukarı taşıyın.

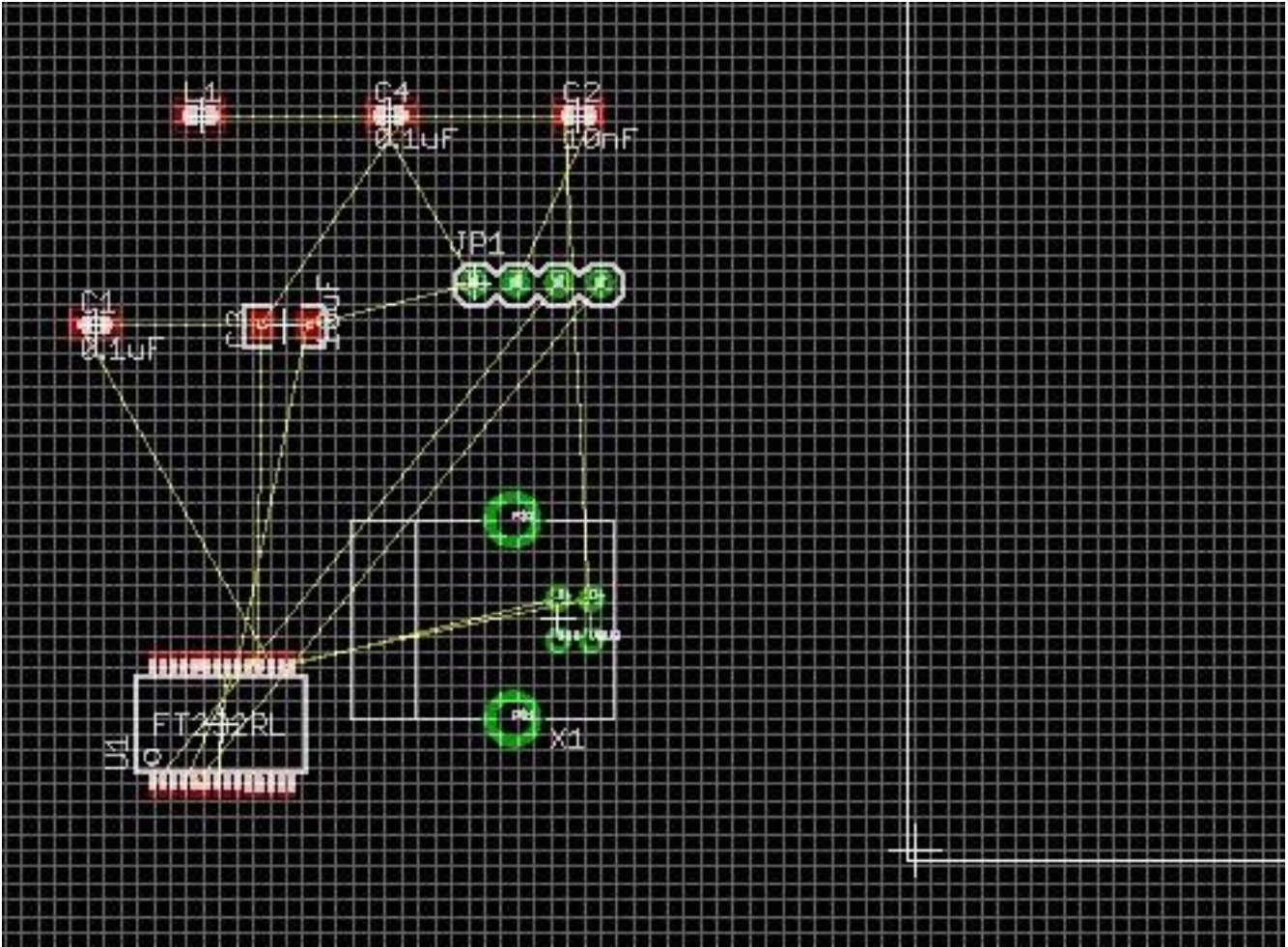


Ve son olarak ise 4 Pinlik konektörün pin adlarını aşağıdaki gibi değiştirin.

Eagle files / PDF

Eğer tüm işlemleri eksiksiz olarak tamamladıysanız Board butonuna basarak PCB editörüne geçebilirsiniz. Böylece şematik editörde oluşturduğunuz şema, bağlantı referansları ile beraber PCB editörüne aktarılmaktadır.



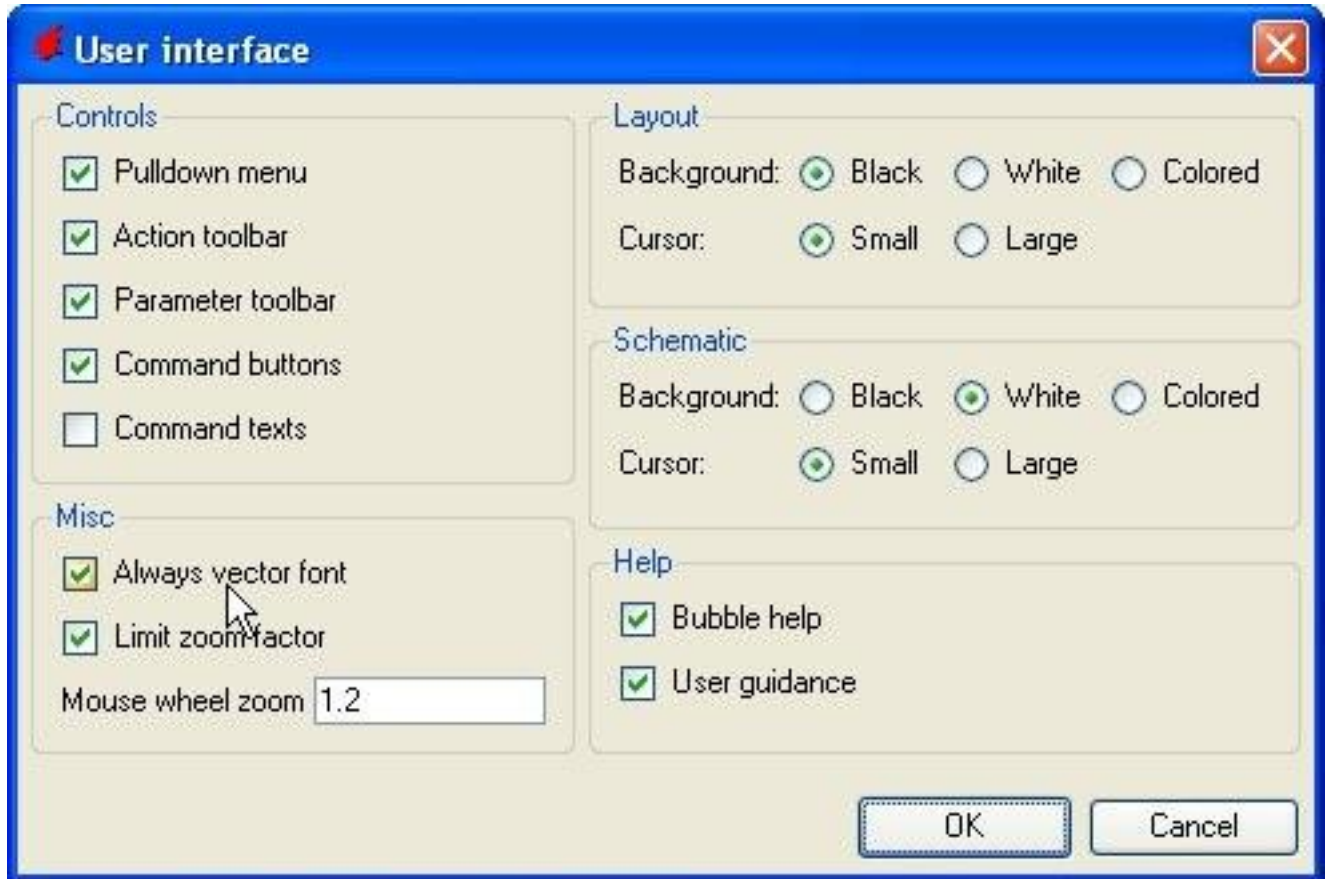


Eagle Eğitimi için gerekli dosyalara [buradan](#) ulaşabilirsiniz.

Ders 9 - Eagle PCB Editörü

Eagle Şematik Editörünü öğrendikten sonra bu yazımızda yine Eagle programının PCB editörünü öğreneceğiz. Bir önceki derste Eagle Şematik Editörü ile basit bir USB-TTL dönüştürücünün şemasını tasarlamıştık. Bu derste ise oluşturulan şemayı PCB ye aktarmayı ve düzenlemeyi öğreneceğiz.

Öncelikle PCB düzenleme işlemine başlamadan ufak bir ayarlama yapmamız gerekiyor. Bu seçenek seçili olduğunda PCB içerisine yazılan her türlü yazının vektörel bir font olması sağlanmaktadır. Bunun için Options->User Interface sekmesine tıklayınız.



Buradan Always Vector Font u işaretleyiniz.

Info		Show
Display		Mark
Move		Mirror
Mirror		Rotate
Group		Change
Cut		Paste
Delete		Add
Pinswap		Replace
Name		Value
Smash		Miter
Split		Optimize
Route		Ripup
Wire		Text
Circle		Arc
Rectangle		Polygon
Via		Signal
Hole		
Ratsnest		Auto
ERC		DRC
Errors		

Araç çubuğu ve her butonun fonksiyonu

Bu butonların haricinde SparkFun Script dosyasında verilen kısayollar ile de bu işlemleri hızlıca yapabilirsiniz.

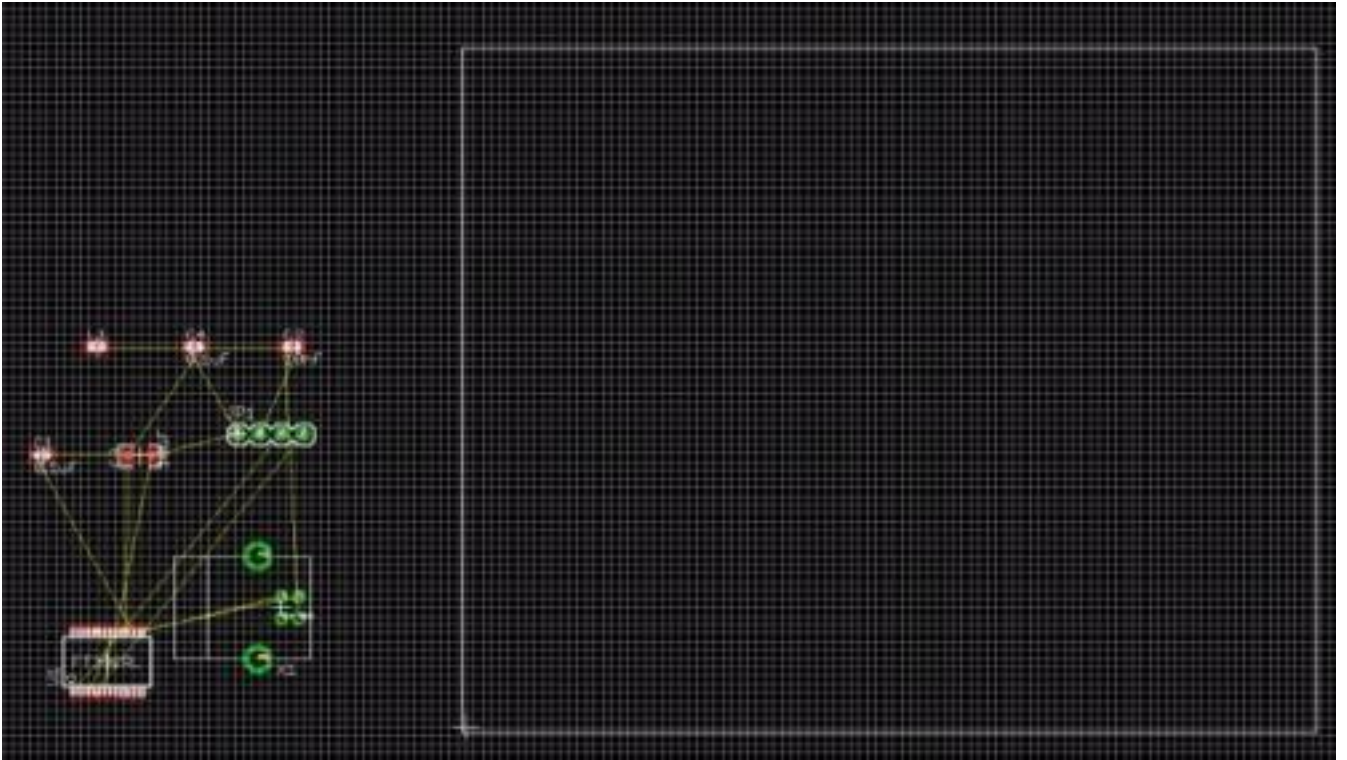
- Escape tuşu herhangi bir yerden çıkmak için kullanılır
- F7 bir parçayı taşımak için kullanılır
- Alt+F7 bir grup parçayı taşımak için kullanılır
- F3 bir parçayı siler
- F4 bir parçanın ismini değiştirir
- F5 parçanın değerini değiştirir (0.1uF kondansatörü 0.01uF yapmak gibi)
- F6 malzemeyi daha küçük parçalara ayırır

- F9 bağlantı(kablo) oluşturur
- Alt+F9 bu bağlantıya bir isim verir

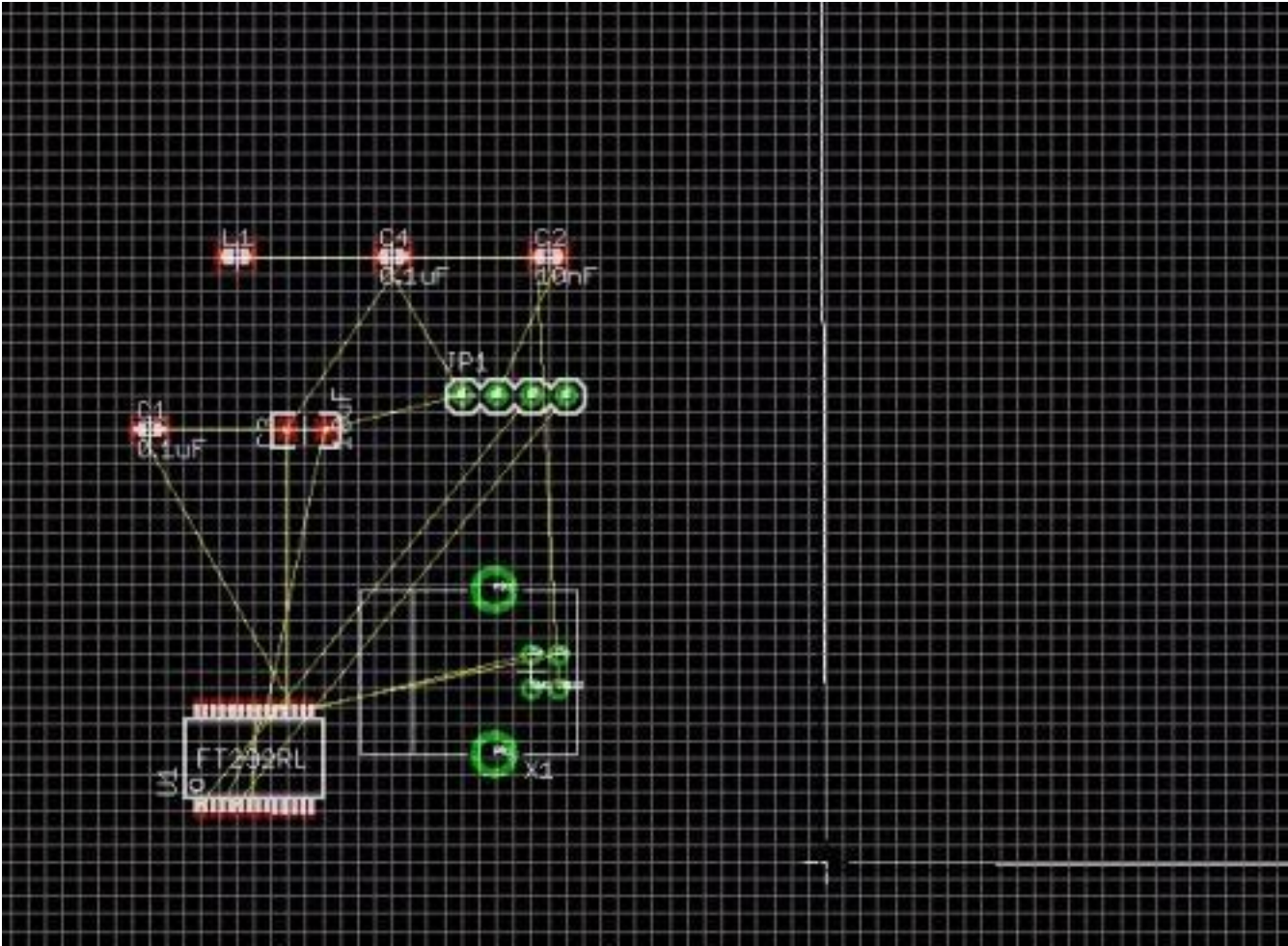


Parçaları editör ekranında bir yerden başka bir yere taşıırken yukarıdaki uyarıyı alabilirsiniz. Bu uyarı, Eagle programının ücretsiz sürümünün kısıtlamalarındandır. Ücretsiz sürümde sadece PCB'yi oluşturacağınız bölge içerisinde işlemlerinizi gerçekleştirebiliyorsunuz.

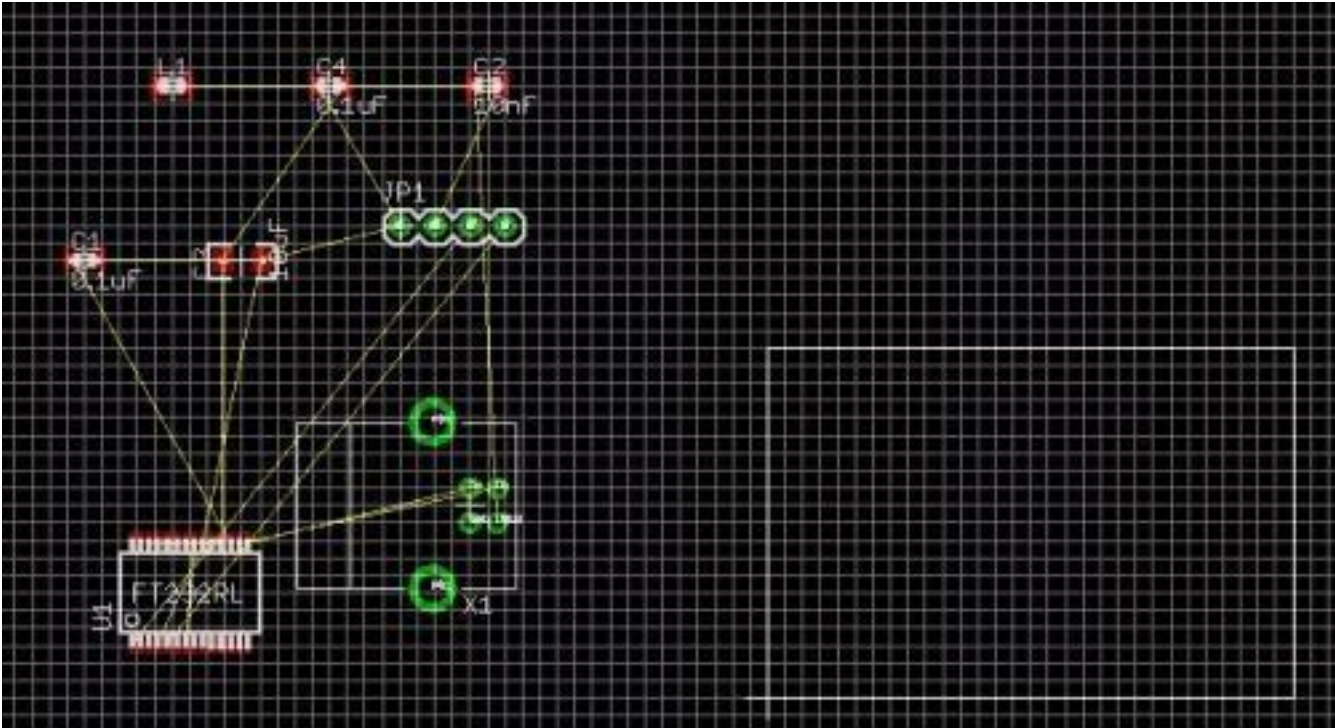
Eagle'da şemayı oluşturup, PCB editörüne geçmek için board butonuna bastıktan sonra standart baskı devre sayfası ekrana gelmektedir.



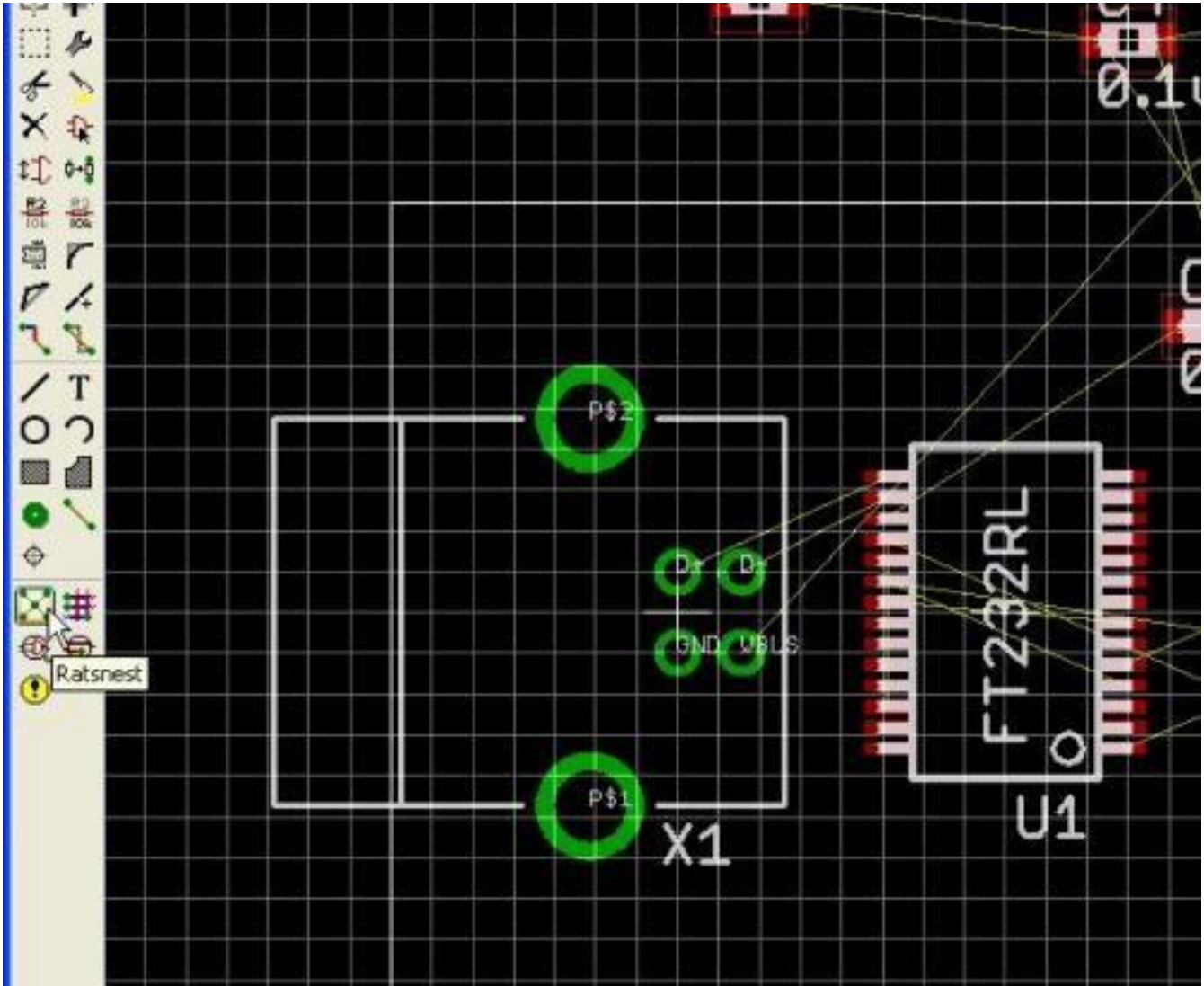
F7 tuşuna basarak grid 0.1"(inch) olduğundan emin olunuz. Grid : Editördeki kareler arasındaki sıklığı belirtmektedir.



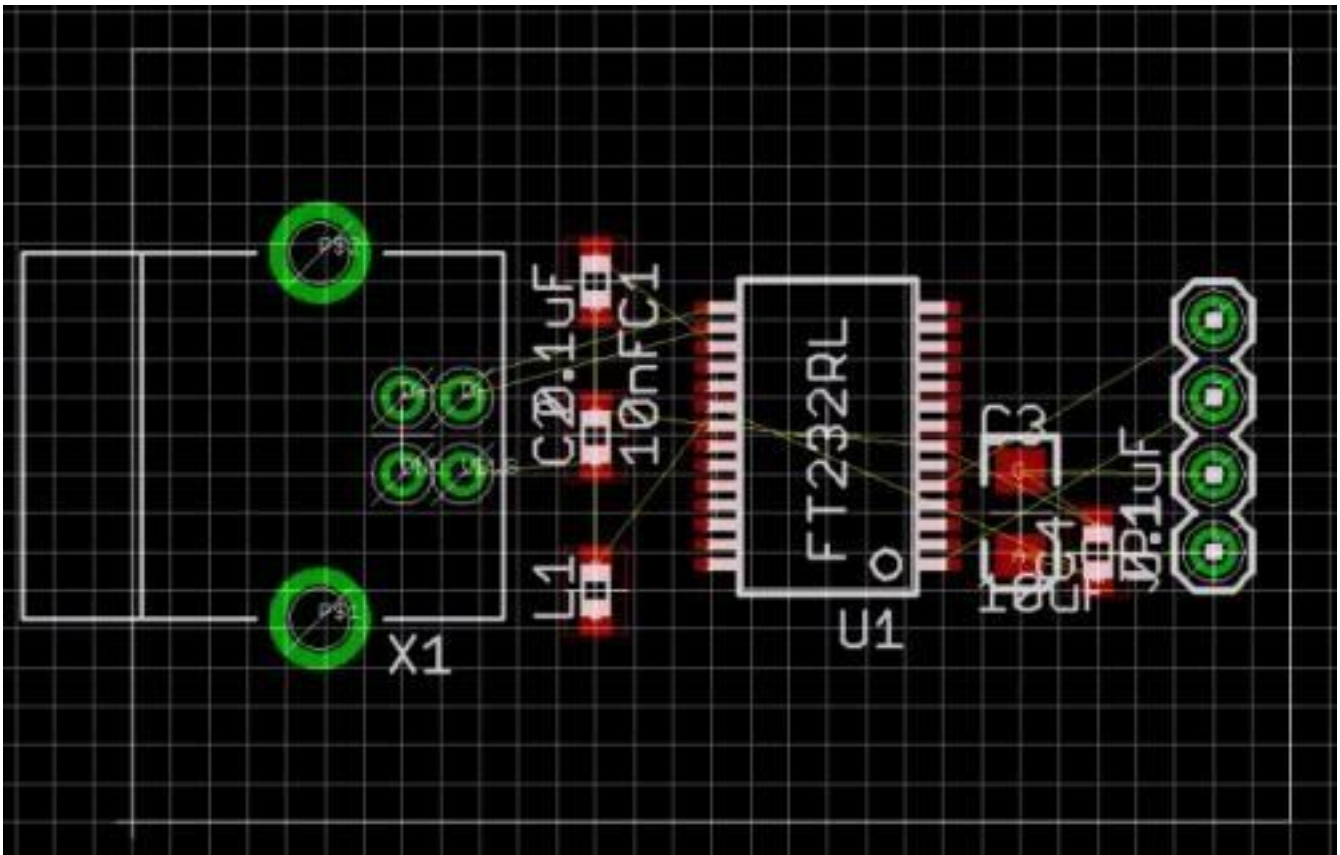
Sonra PCB çerçevesini biraz daha küçülterek 1.5 x 1.0"(inch) boyutunda yapabilirsiniz.



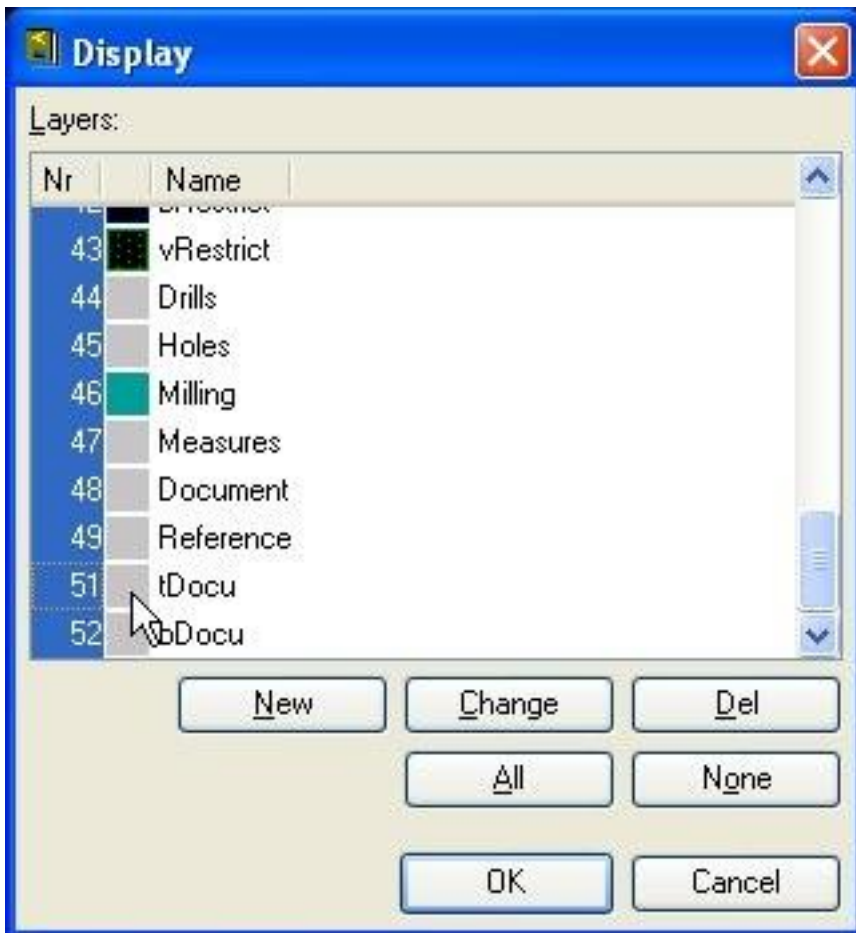
Hata mesajı almamak için komponentleri verilen beyaz çerçeve içersine taşımak gerekiyor. Bunun için parça üzerine bir kere tıklamak ve taşımak istediğiniz yere bırakmak yeterli. Veya kısayol olarak F7 tuşunu da kullanabilirsiniz.

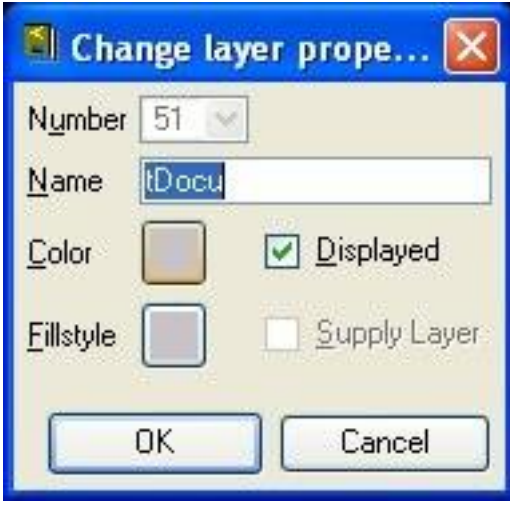


Soldaki Ratnest butonuna basarak yeni yerleşim için bağlantıların tekrar düzenlenmesini sağlayabilirsiniz.

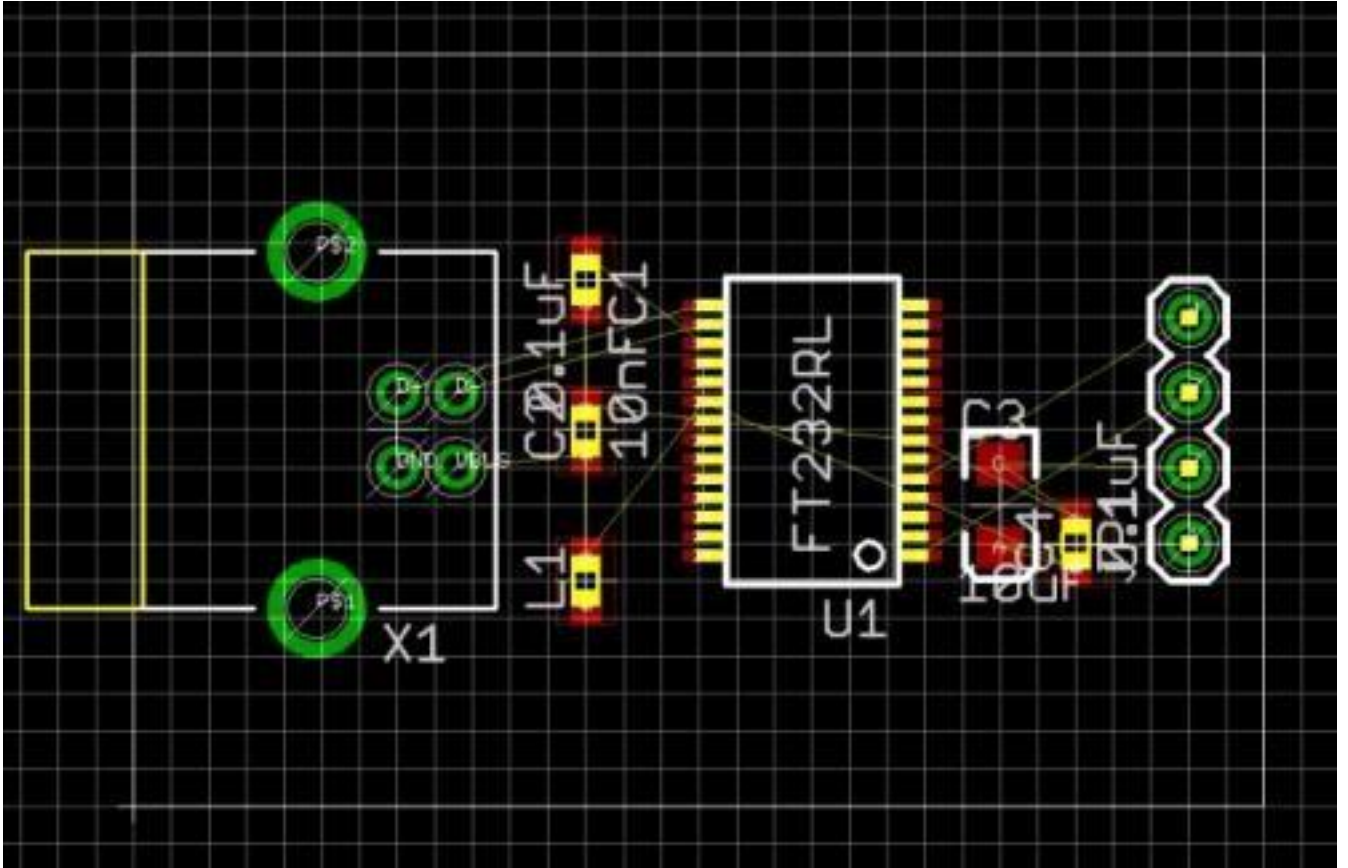


Malzemelerin biraz daha belirgin olması açısından ufak bir ayarlama yapabiliriz. Bunun için soldaki araç çubuğundan Display butonuna basıp tDocu katmanını seçiyoruz.

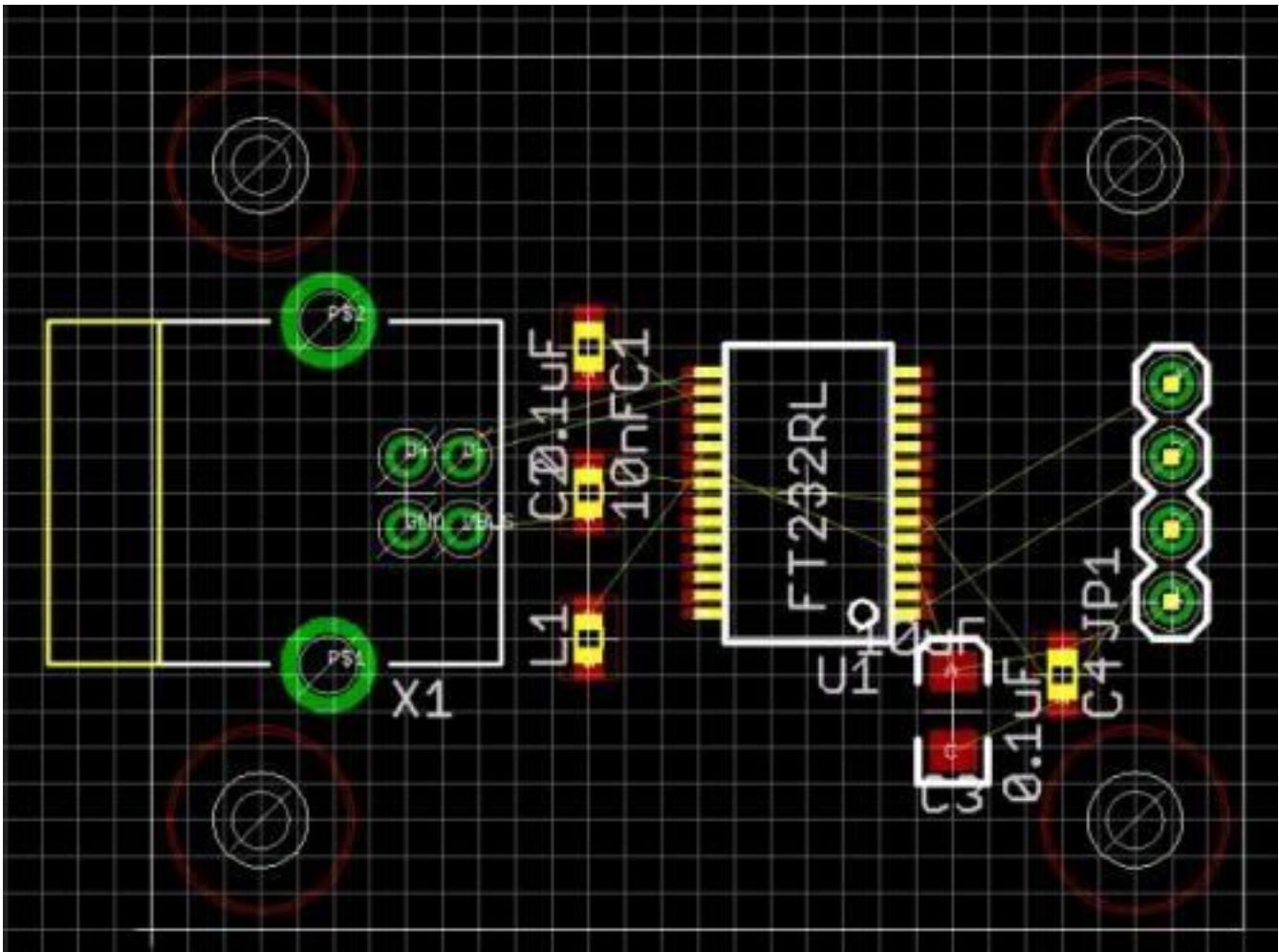




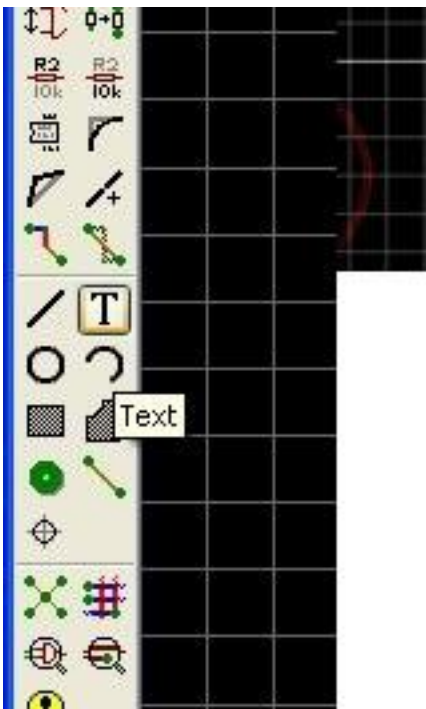
Color butonundan istediğiniz rengi seçebilirsiniz. Bu uygulamada padlerin daha belirgin olması için sarı renk seçildi.



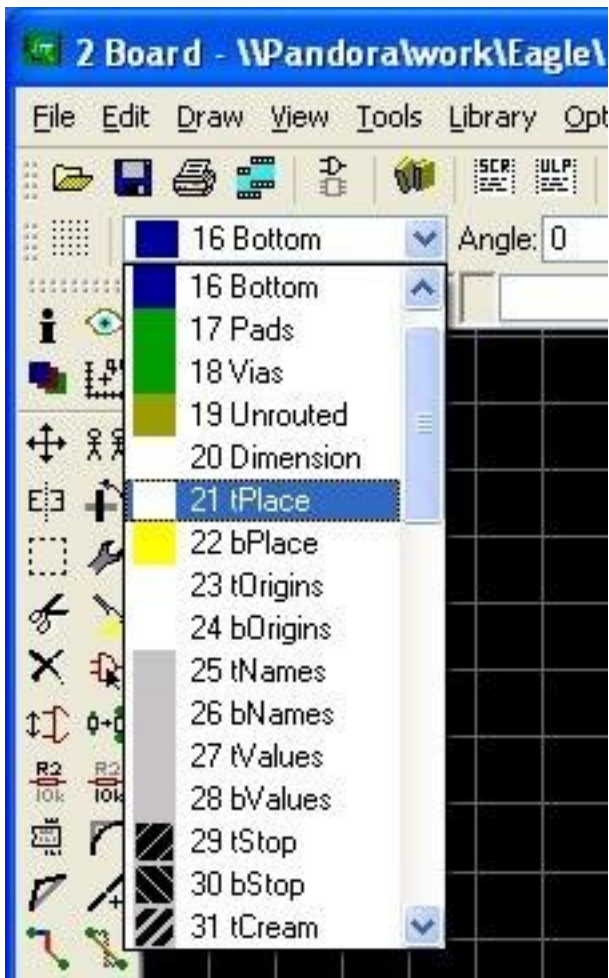
Devremize daha kullanışlı olması için 4-40 vida delikleri ekliyoruz. Bu vidalar üzerine 0.25" çaplı plastik ayaklardan geçirebilirsiniz. Bu ayaklar, vidaların çalıştığımız yüzeyi çizmesini engellemektedir. Vida deliklerini de PCB'ye ekledikten sonra tasarım aşağıdaki gibi olmaktadır.



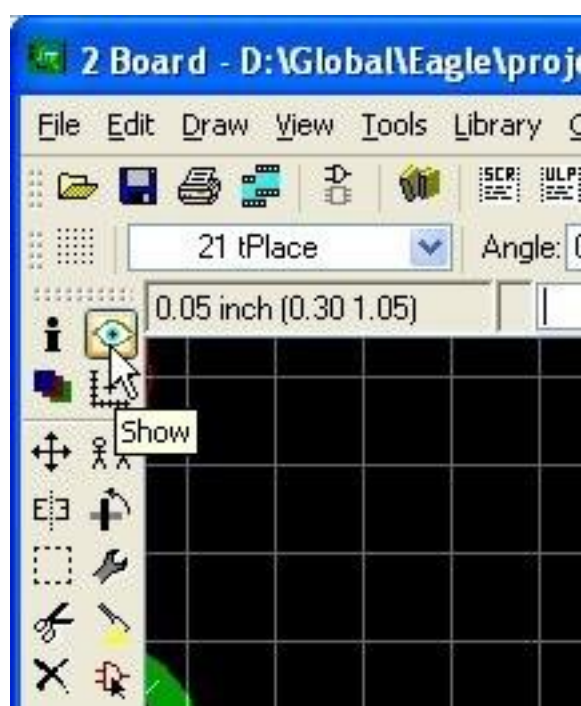
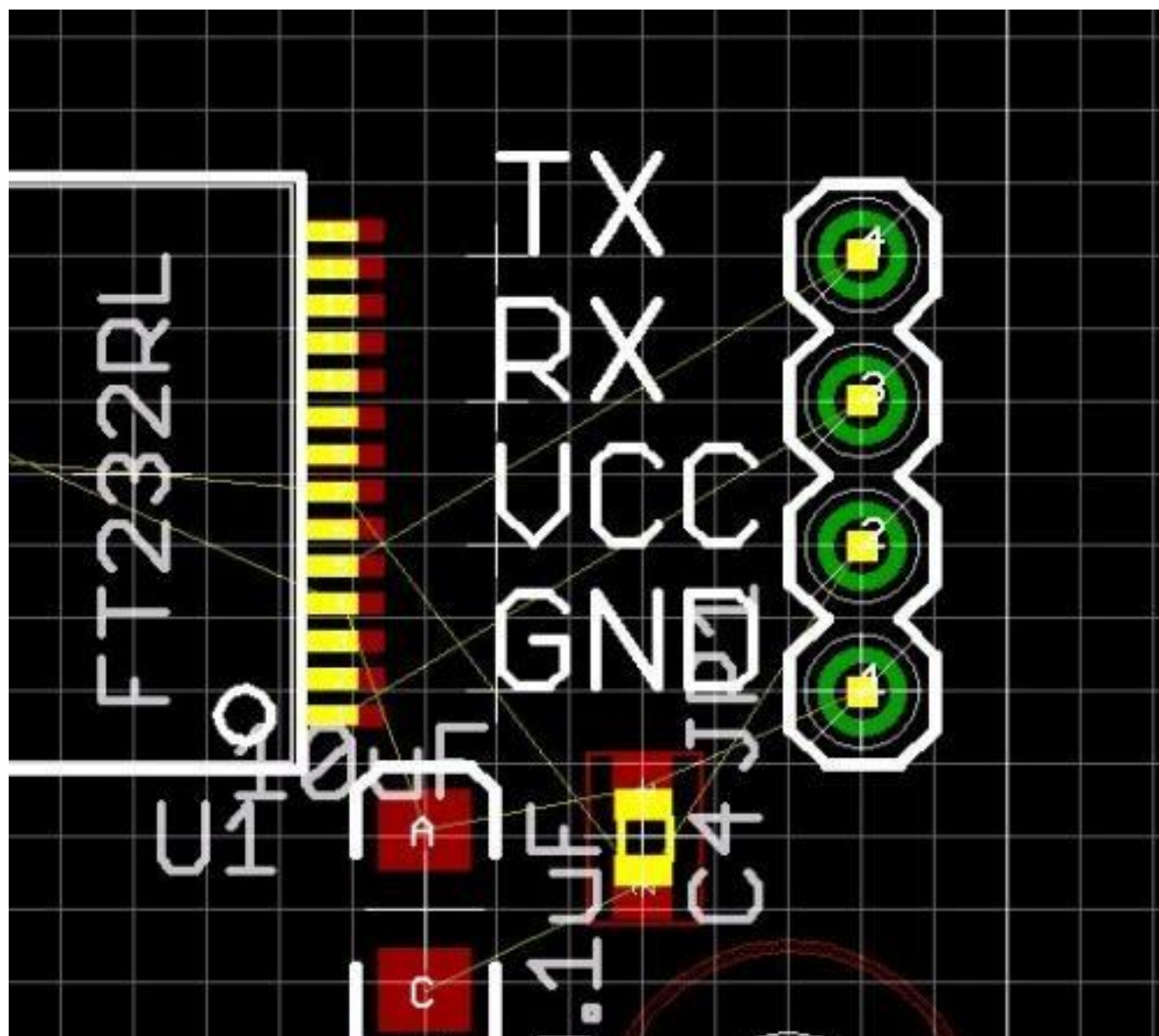
Vida deliklerini de ekledikten sonra soldaki araç çubuğundan "Text" butonunu kullanarak çıkış pinlerini isimlendirebilirsiniz.



Yazılar için beyaz rengi(tPlace) seçiyoruz.

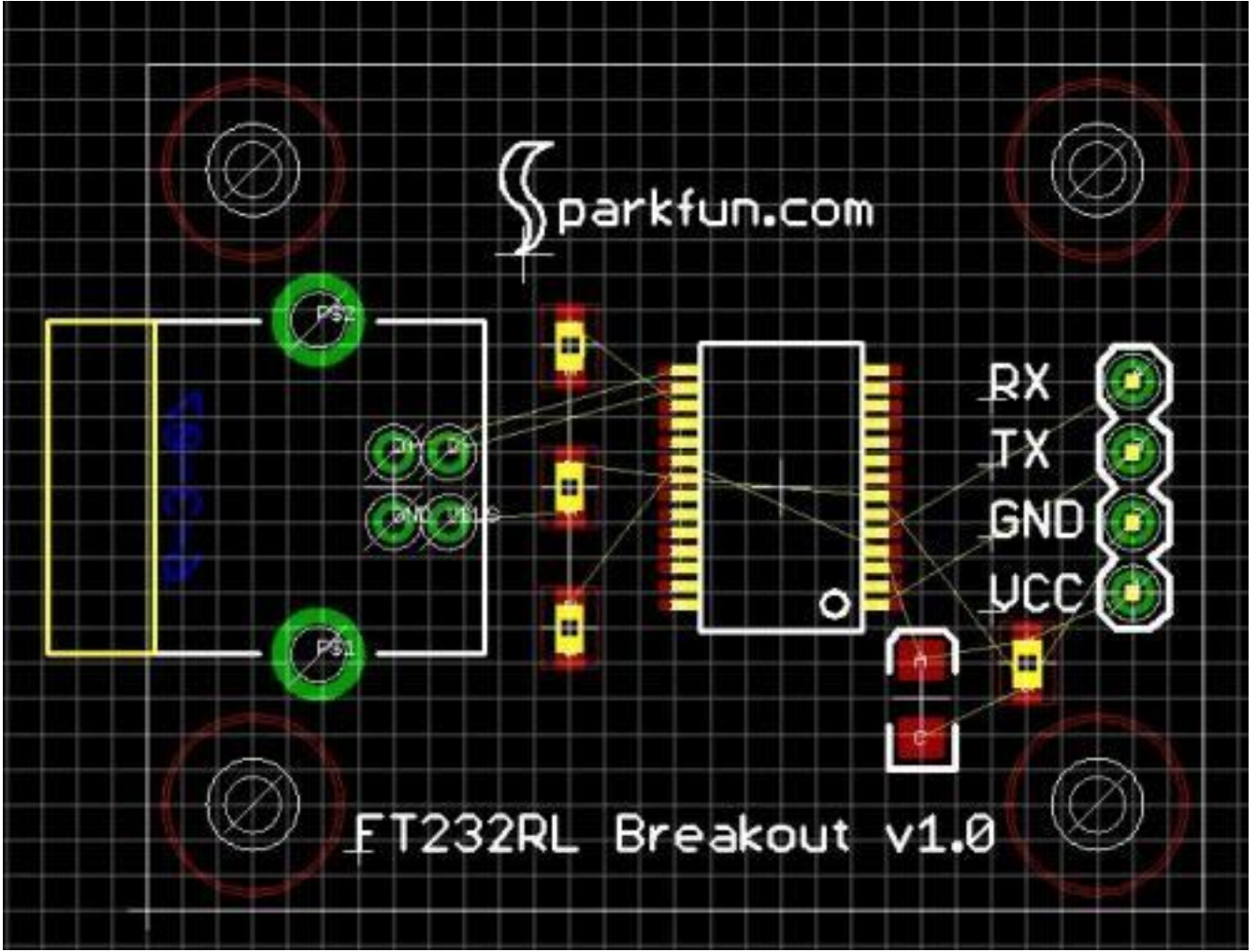


Text butonuna basıp önce TX yazdıktan sonra Esc ile işlemi tamamlayın. Sonra diğer pinleri de sırası ile RX, VCC ve GND olarak isimlendirin.



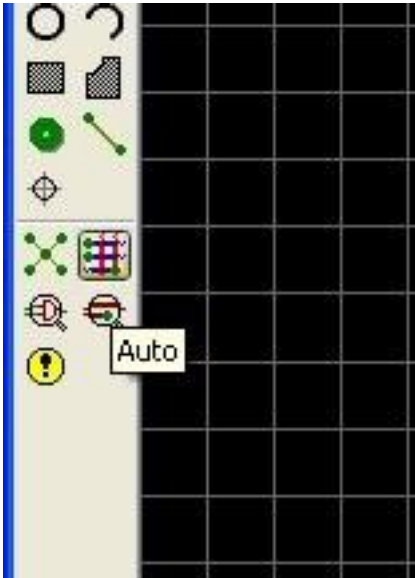
Show butonu ile tüm yazıların tPlace katmanında olduğundan emin olun.

Son olarak baskı devremize birkaç açıklama yazısı ekleyebiliriz.

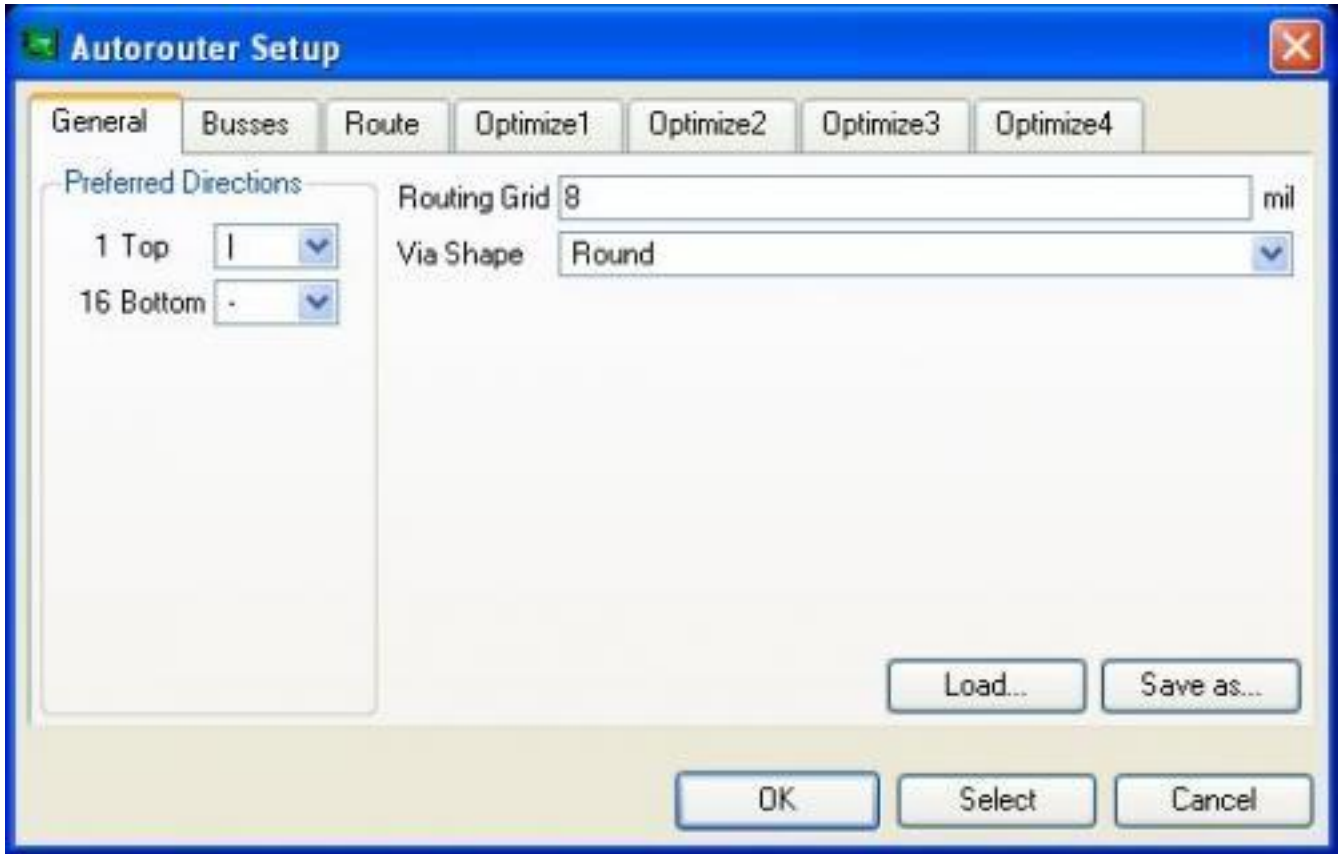


Bu işlemi de tamamladıktan sonra artık baskı devrenin yollarını oluşturabiliriz.

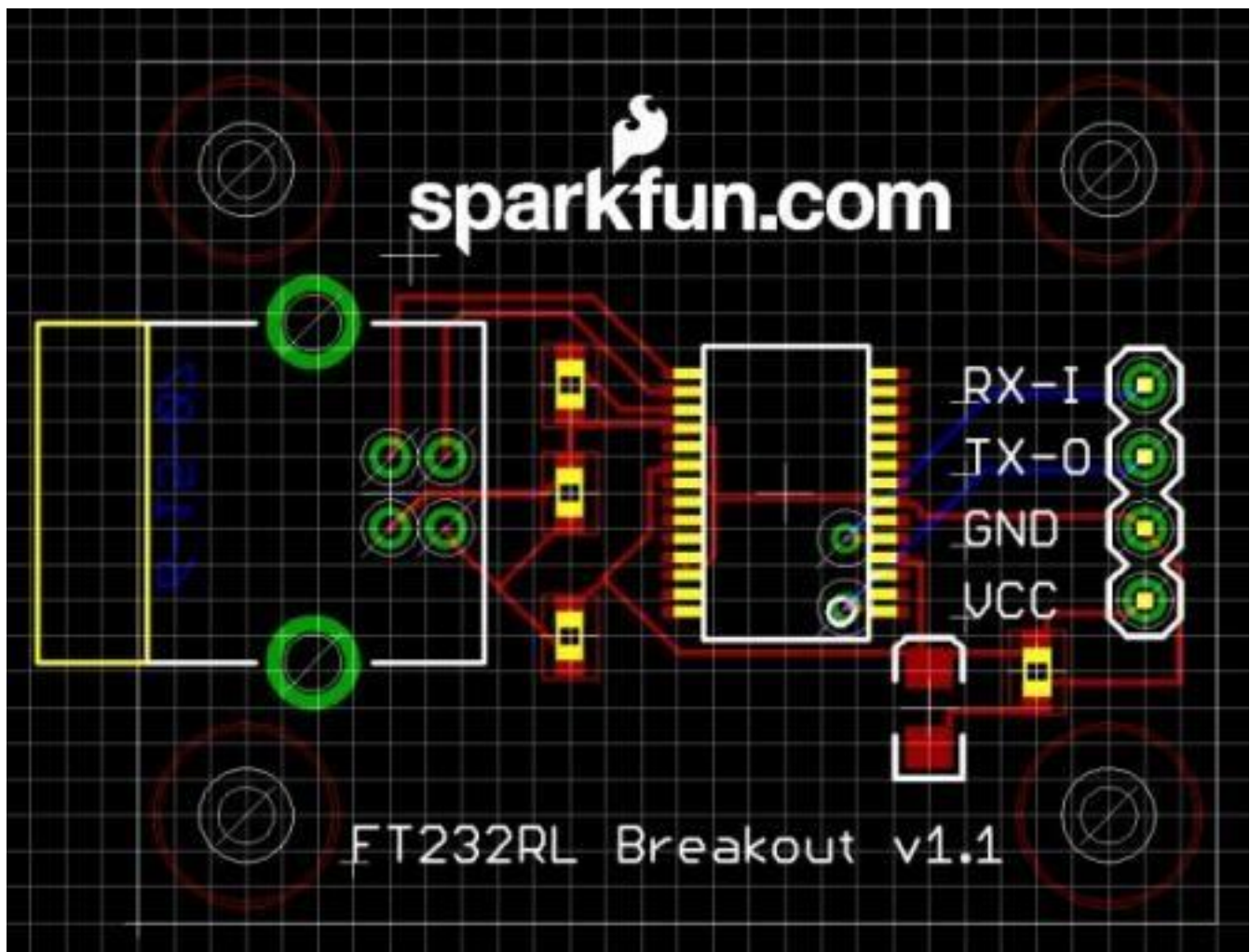
Bu işlemi Eagle, şemada oluşturulan bağlantıları referans alarak otomatik olarak yapabilmektedir. Bunun için araç çubuğundaki "Auto" butonunu kullanacağız.



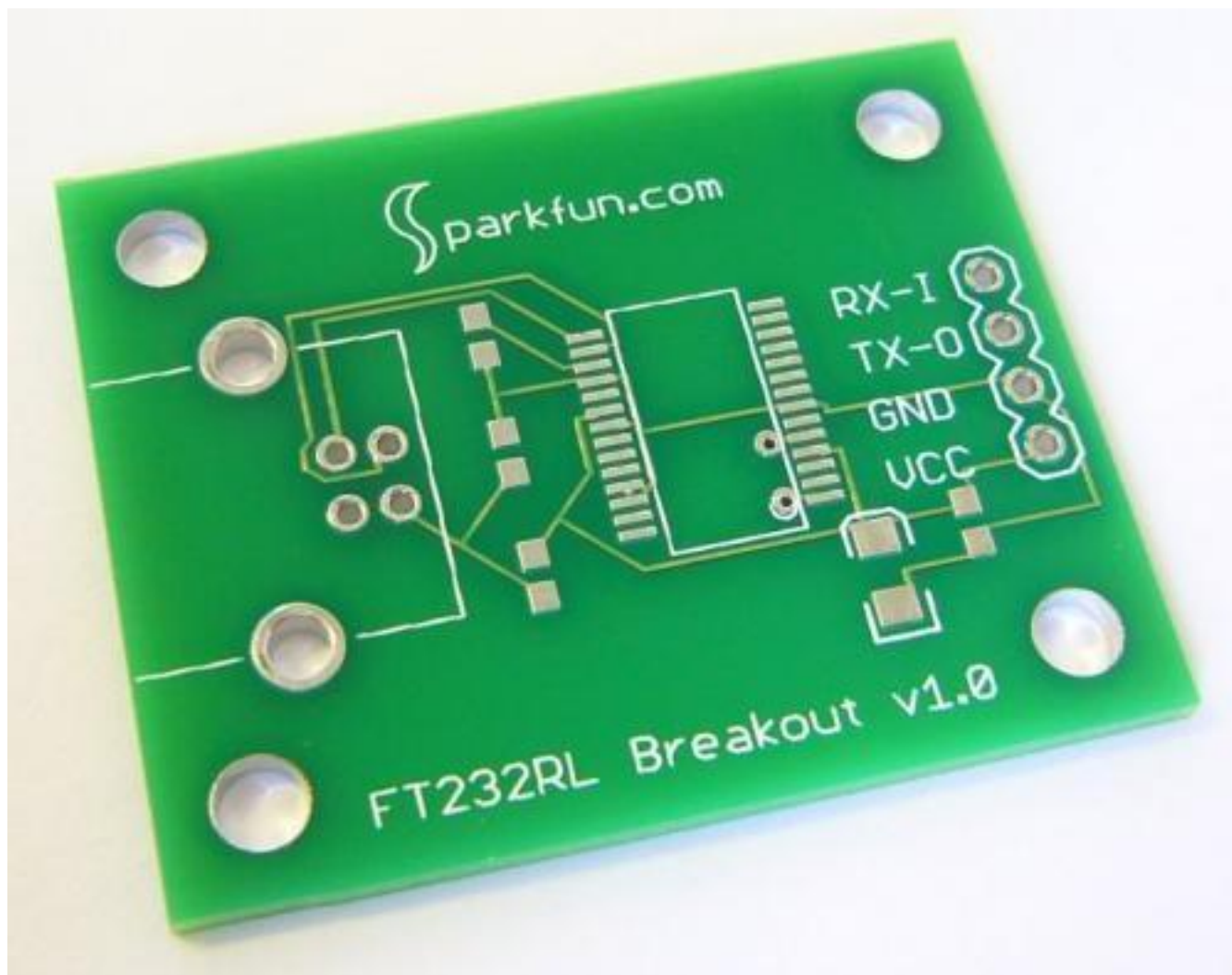
"Routing Grid" ön tanımlı olarak 50mil seçilidir. Bunu 8mil olarak değiştiriyoruz.

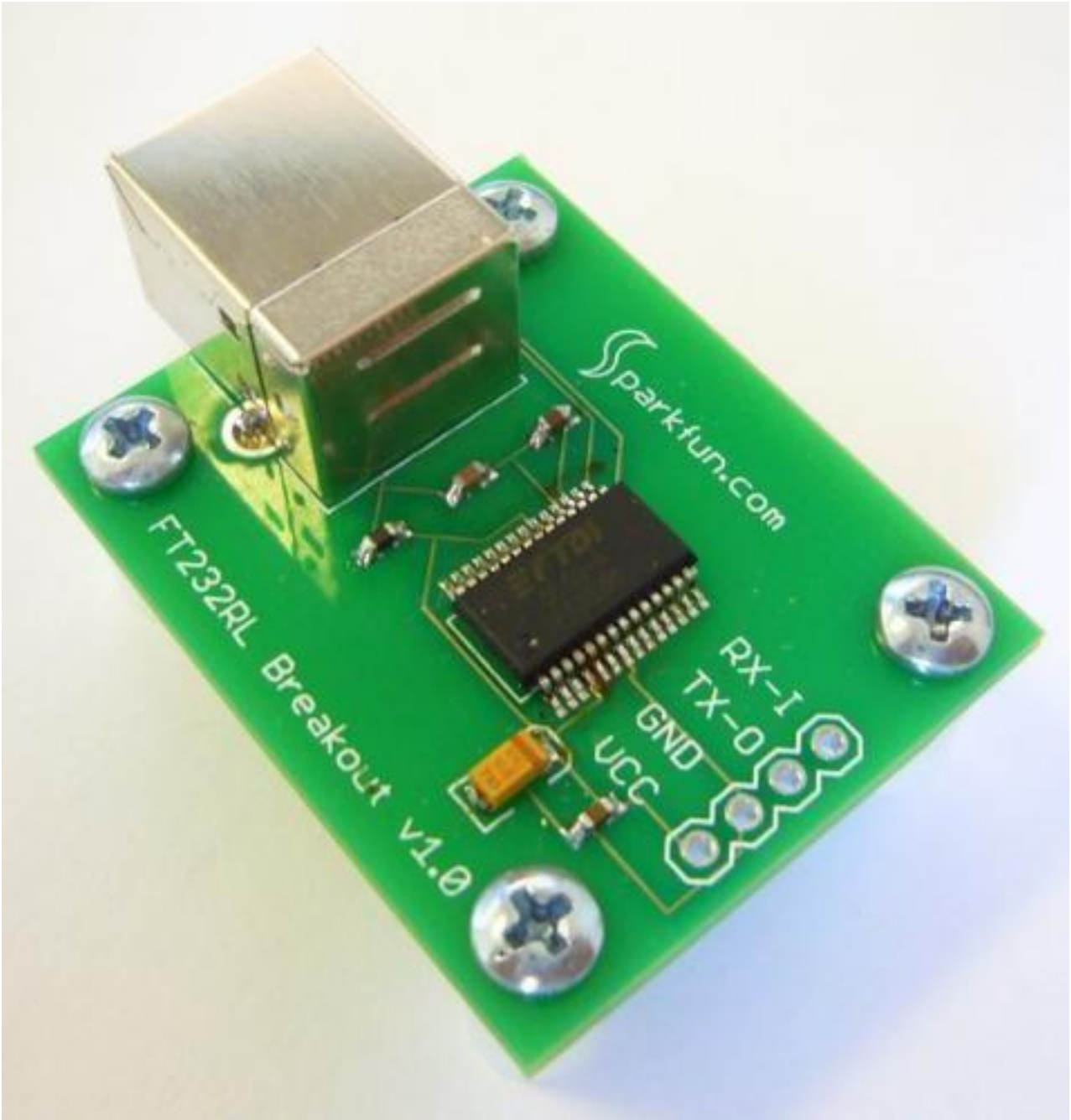


"Ok" tuşuna bastıktan sonra PCB'nin yolları otomatik olarak çizilmektedir.



FT232RL Breakout Board. Eagle [files](#) / [PDF](#)





Eagle Eğitimi için gerekli dosyalara [buradan](#) ulaşabilirsiniz.

Ders 10 - Eagle'da Yeni Bir Para Oluřturma

Bu derste ise Eagle kütüphanelerinde olmayan bir paranın nasıl oluřturulduėunu öğreneceėiz. Bunun için 1.5volt gerilimi 5 volta ıkaran bir step-up devresinin en önemli elemanı olan NCP1400 parasını oluřturacaėız.(datasheet)

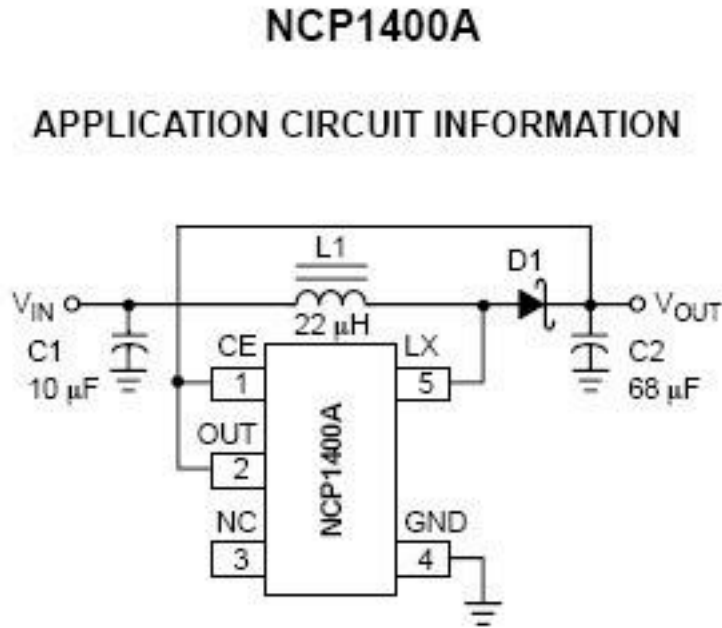
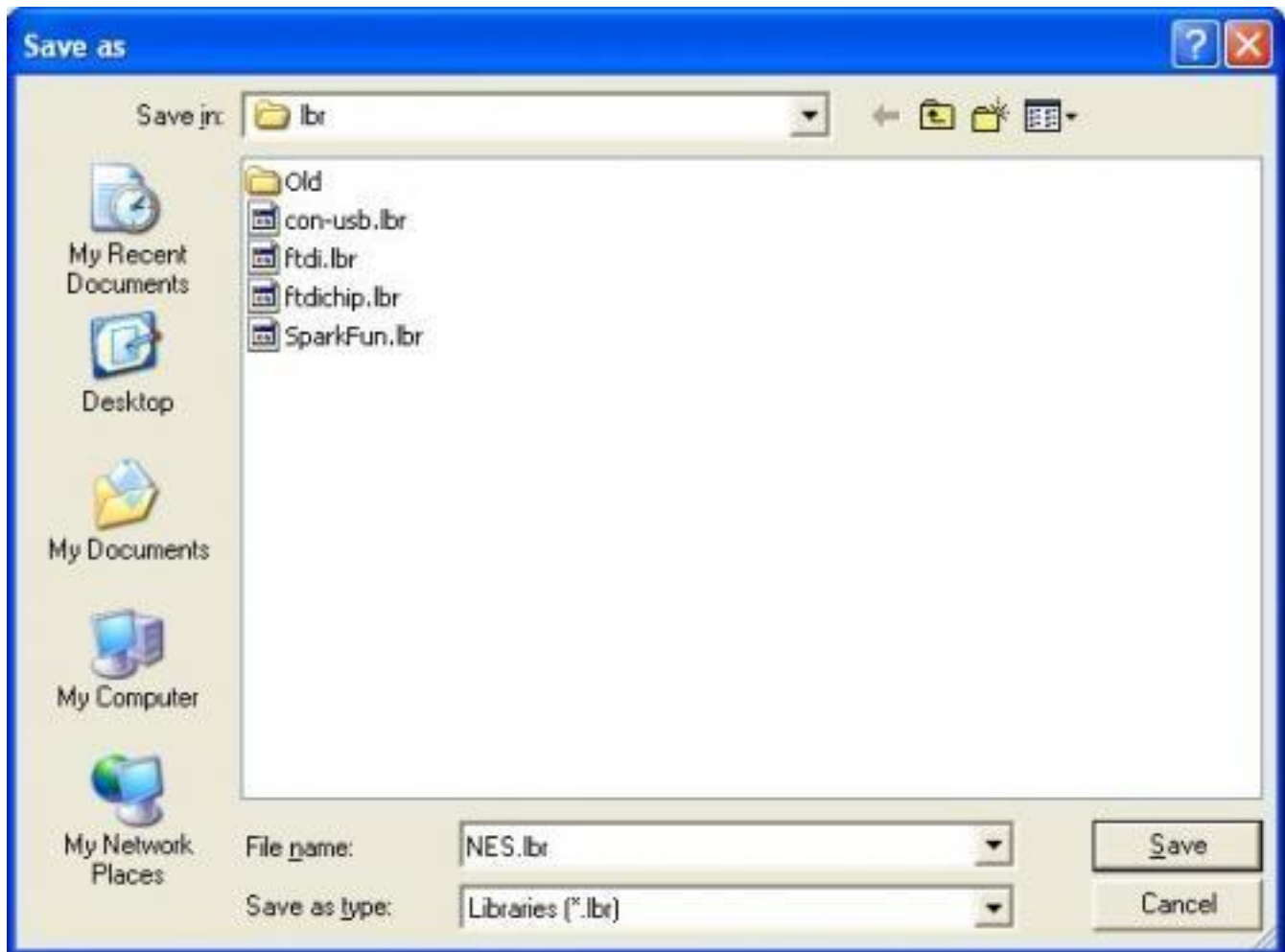


Figure 41. Typical Step-Up Converter Application

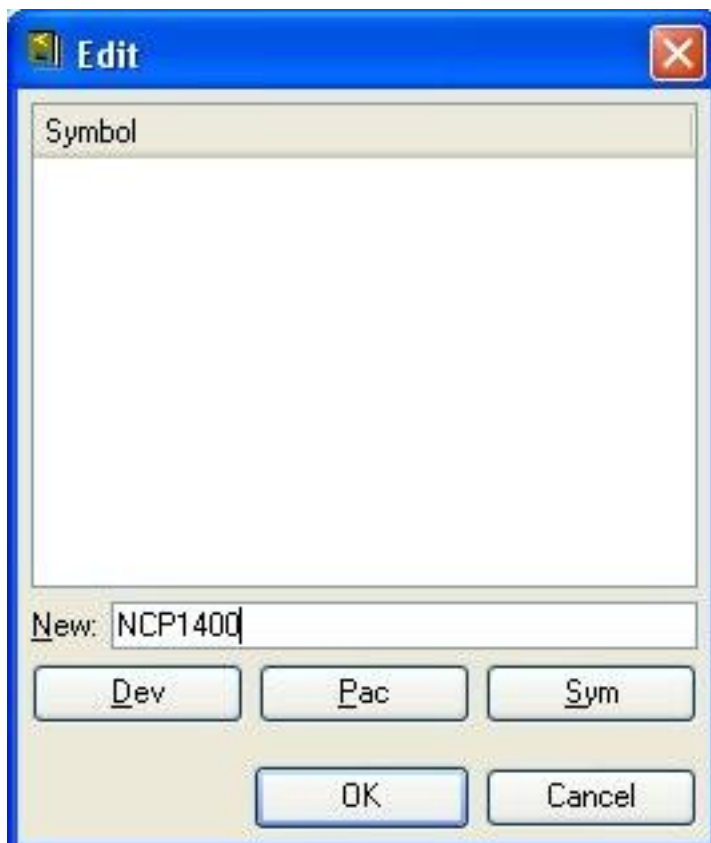
Öncelikle File->New->Library sekmelerini takip edip yeni bir kütüphane oluřturuyoruz.



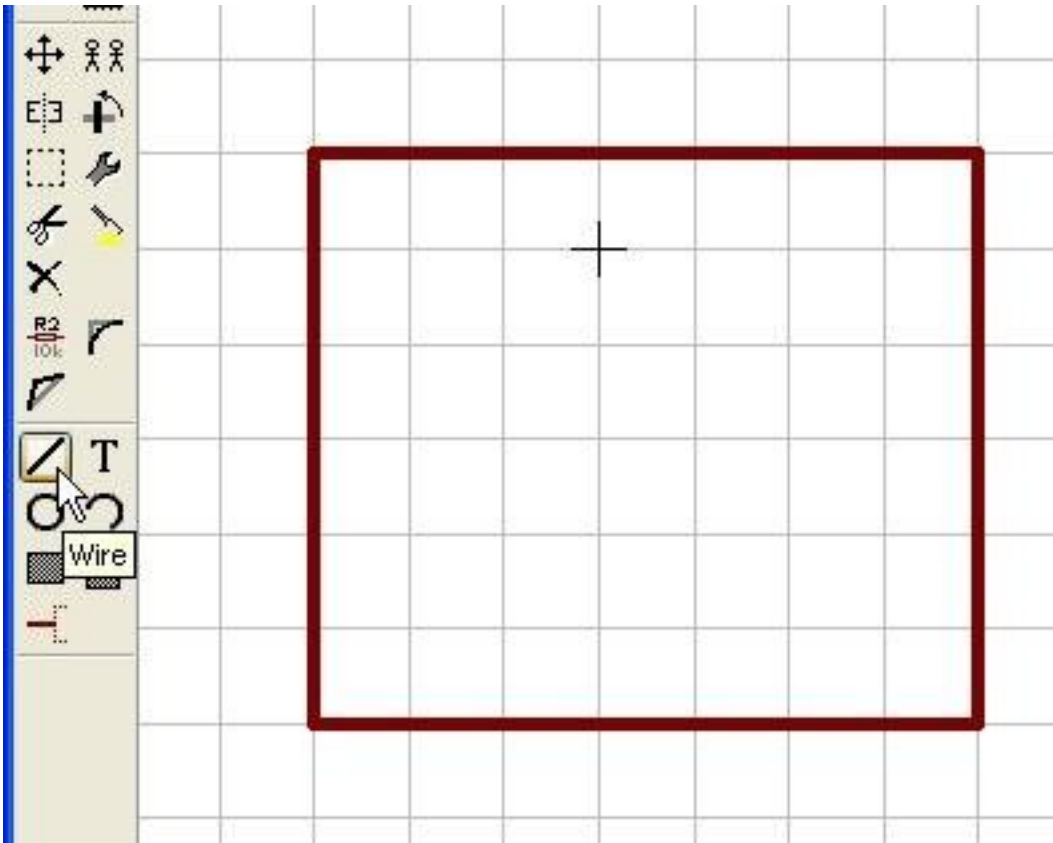
Ve oluřturacaėımız kütüphaneye ismini veriyoruz.



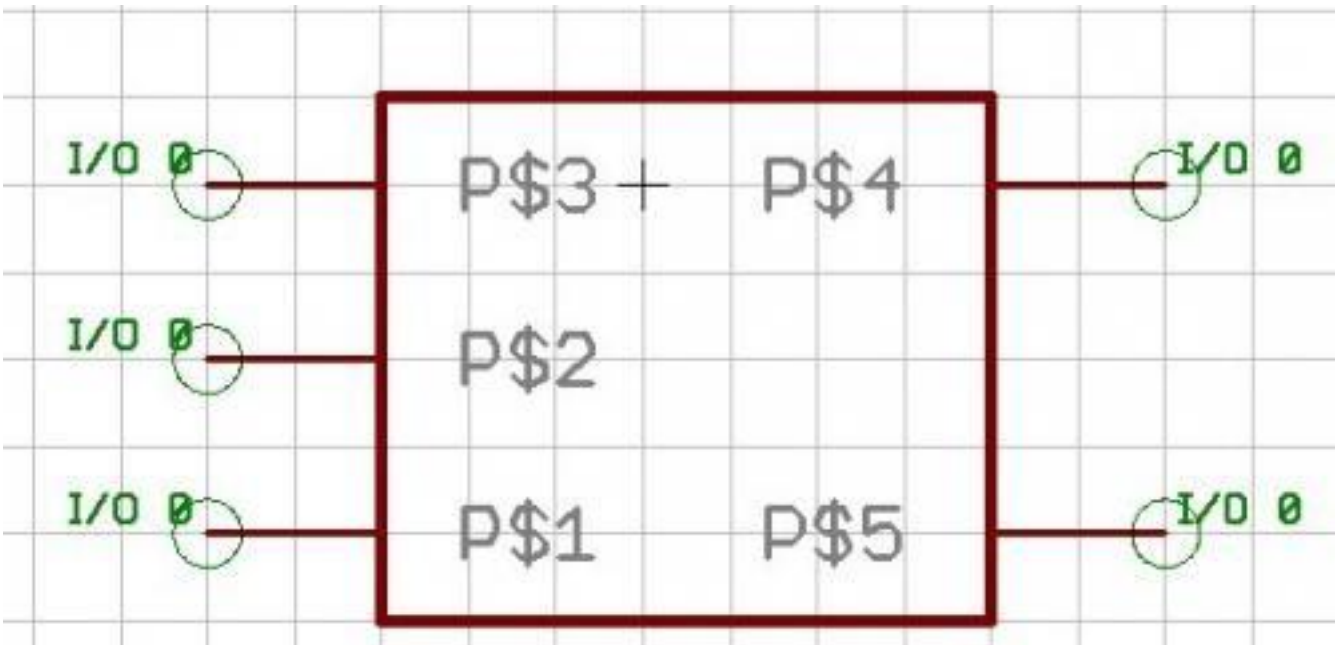
NCP1400 adında yeni bir "symbol" oluşturunuz.



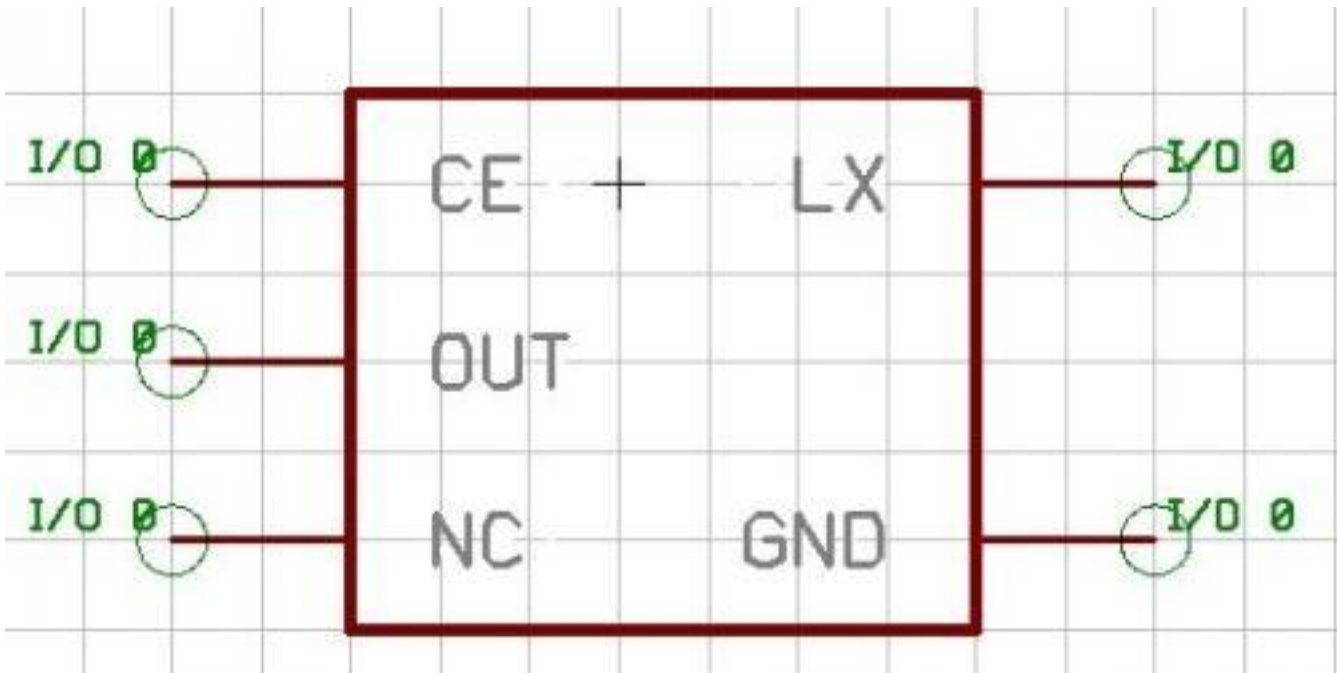
Wire butonuna tıklayarak şekildeki gibi bir kutu oluşturuyoruz.



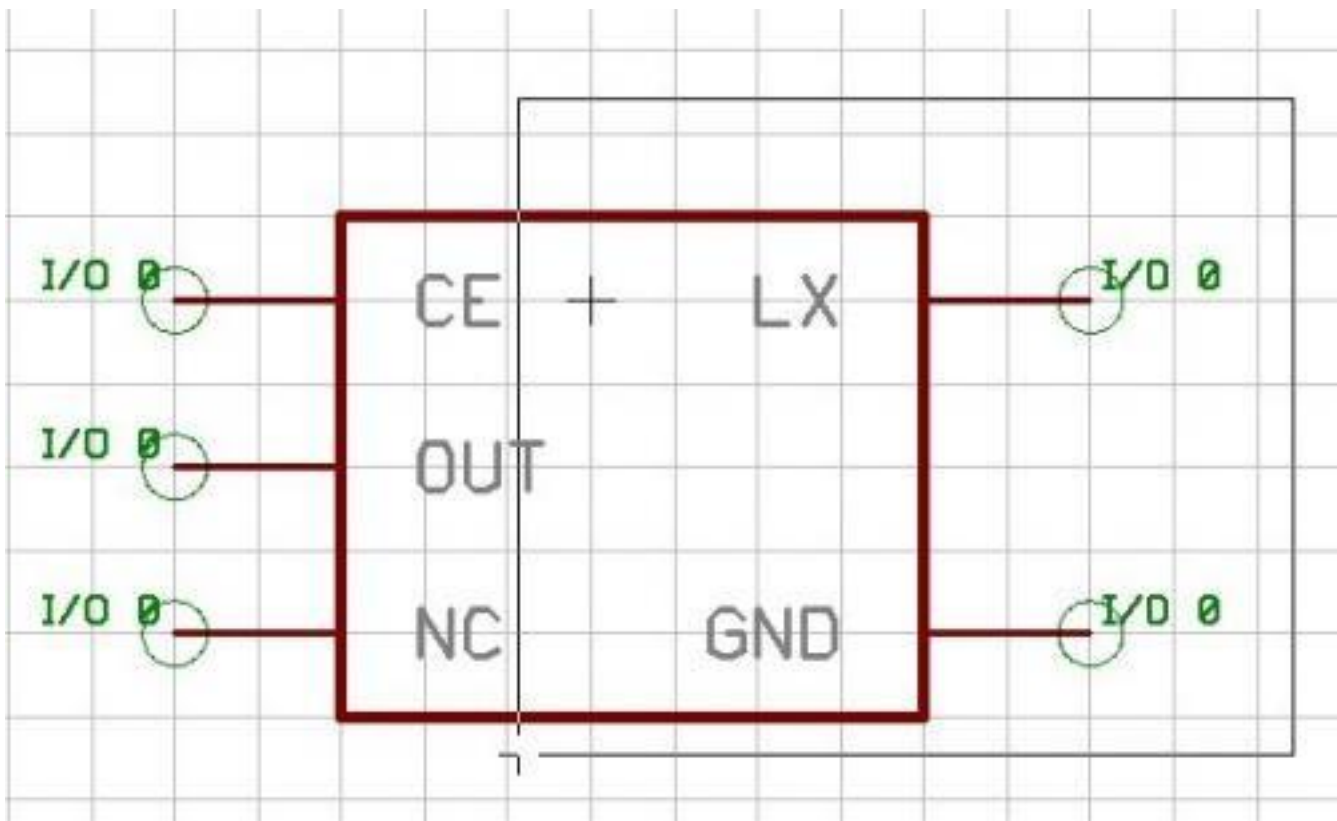
Sonra bu kutuya 5 adet pin ekliyoruz.



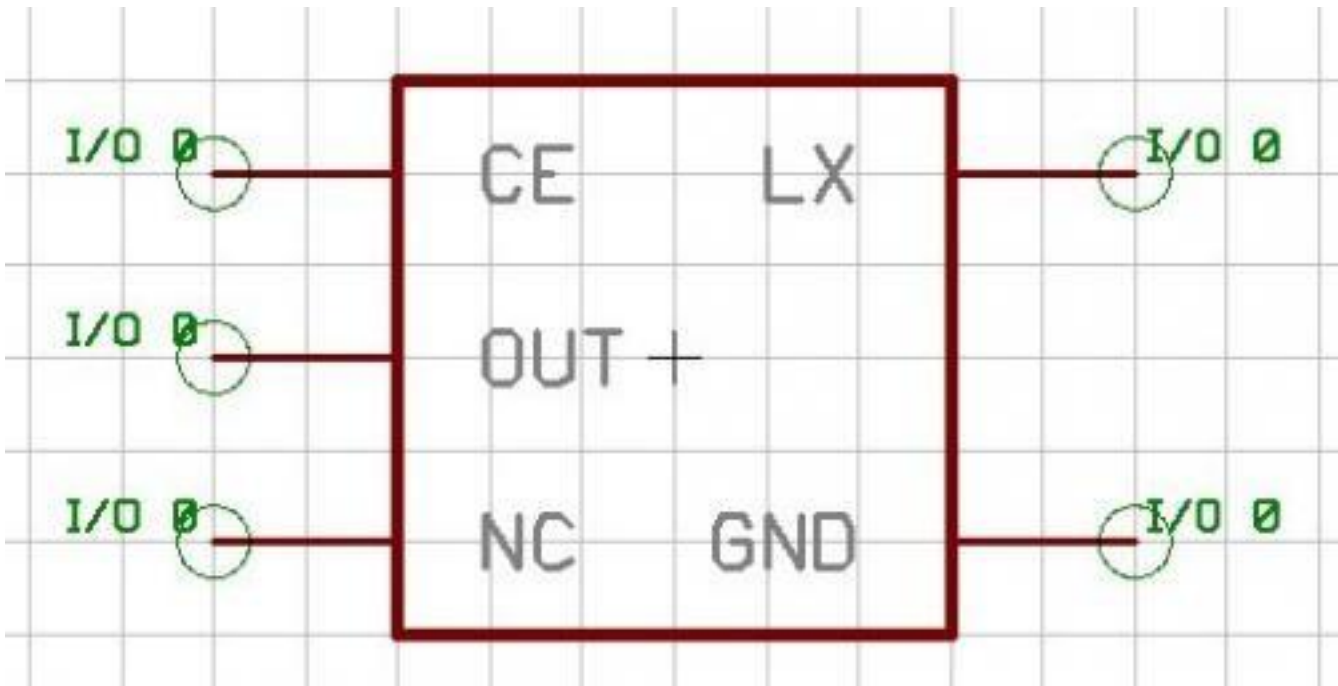
Ve datasheetdeki gibi isimlendiriyoruz.



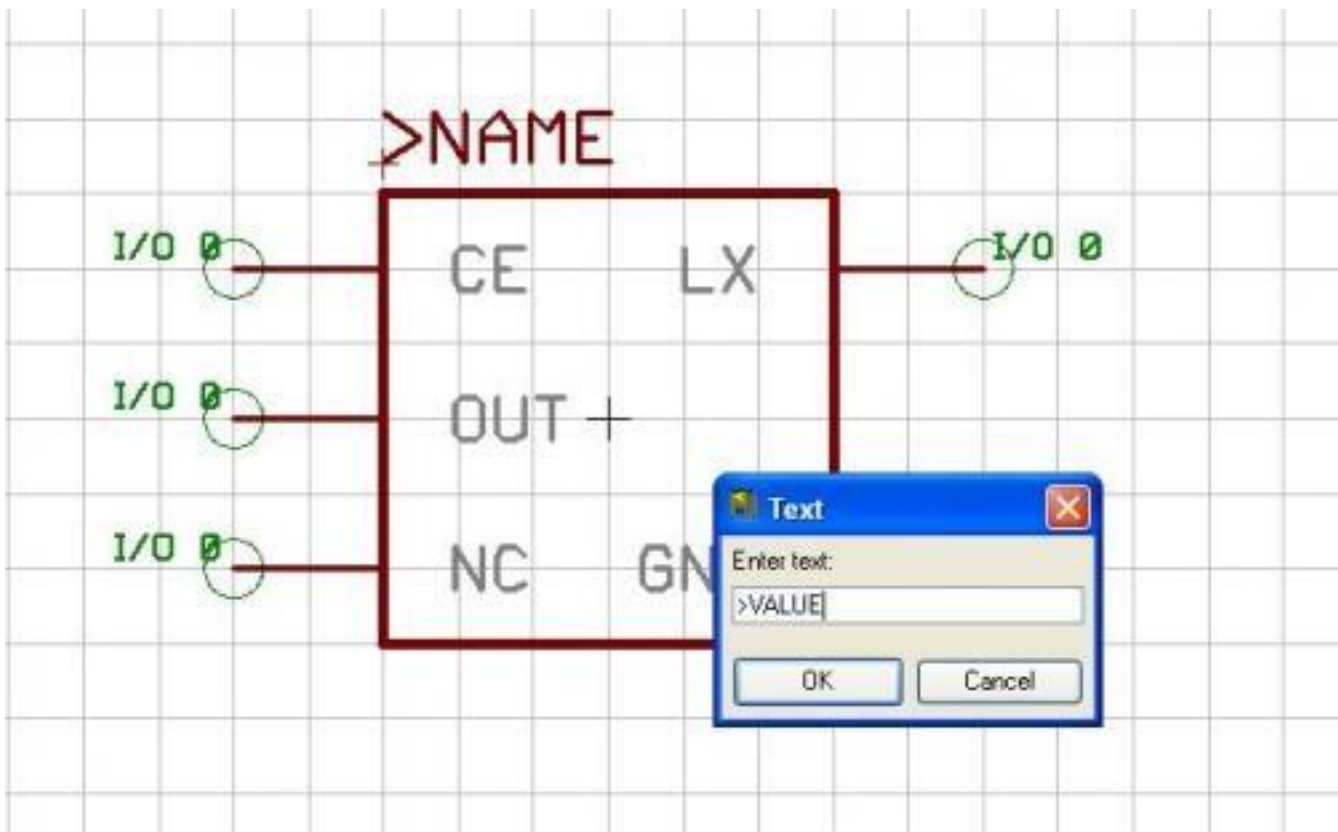
Pin isimlerini girdikten sonra Alt+F7 tuşları ile oluşturulan bu grubu merkeze taşıyoruz.



Yeni görünüm aşağıdaki gibi olmalıdır(Ortadaki "+" ya dikkat edin)

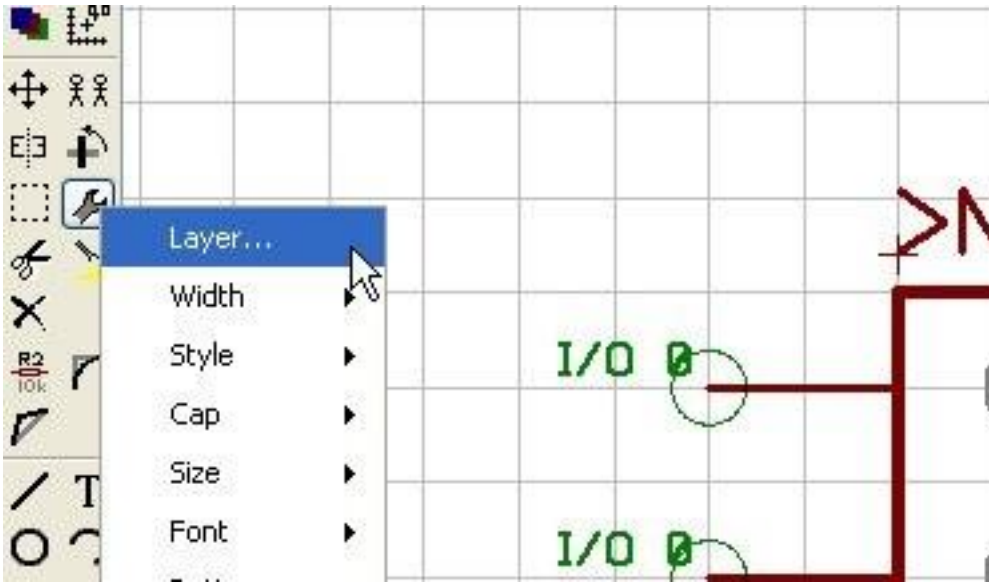


Soldaki araç çubuğundan "Text" butonuna tıklayarak ">VALUE" adında bir yazı oluşturuyoruz.

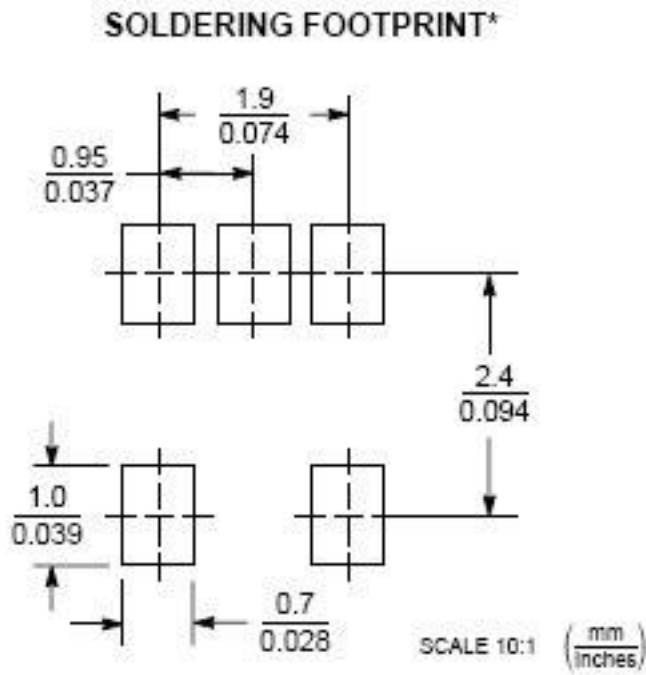


Sonra ">NAME" ve ">VALUE" yazılarını yeni bir katman(layer) olarak tanımlıyoruz.

Bunun için soldaki araç çubuğundan "Layer" ı seçmelisiniz.



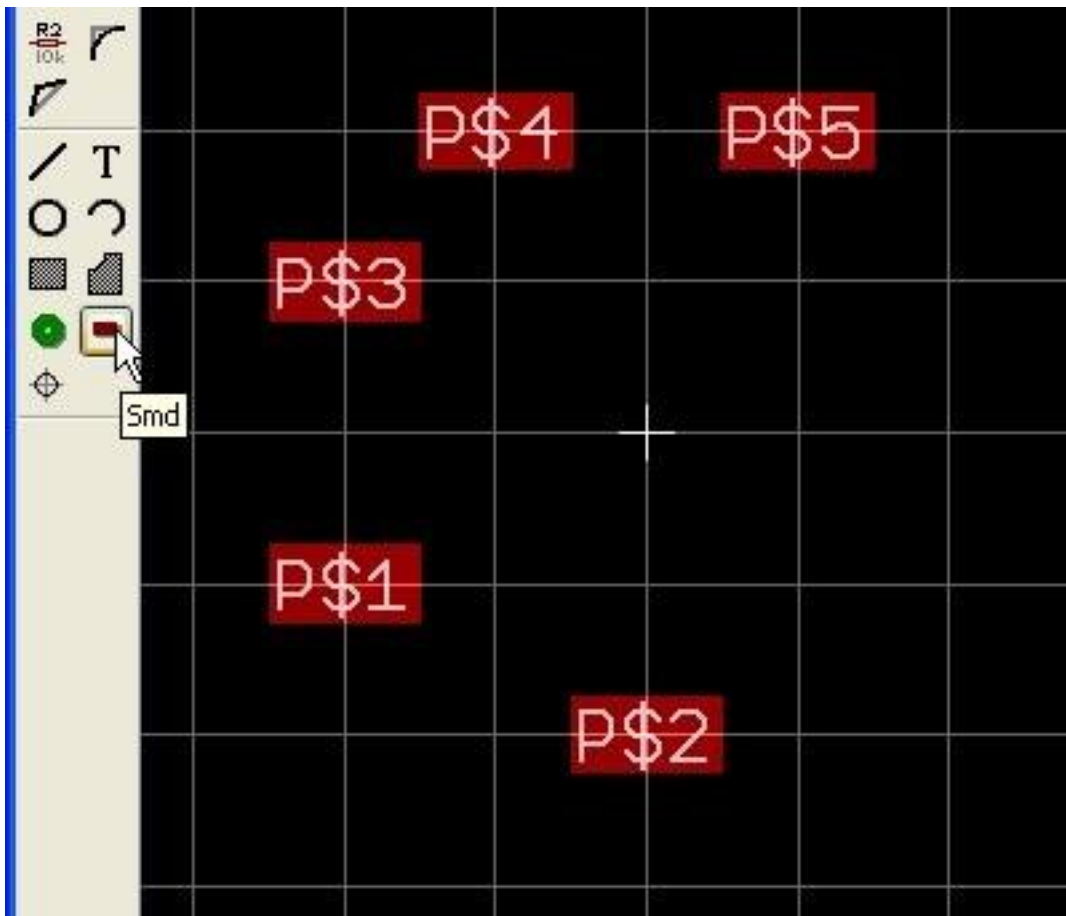
Bu aşamadan sonra şematik kısım tamamlanmaktadır. Şimdi oluşturduğumuz bu parçanın baskı devrelerde kullanılmak üzere gereken kılıfını tasarlayacağız. Bunun için referans olarak datasheeti kullanabilirsiniz.



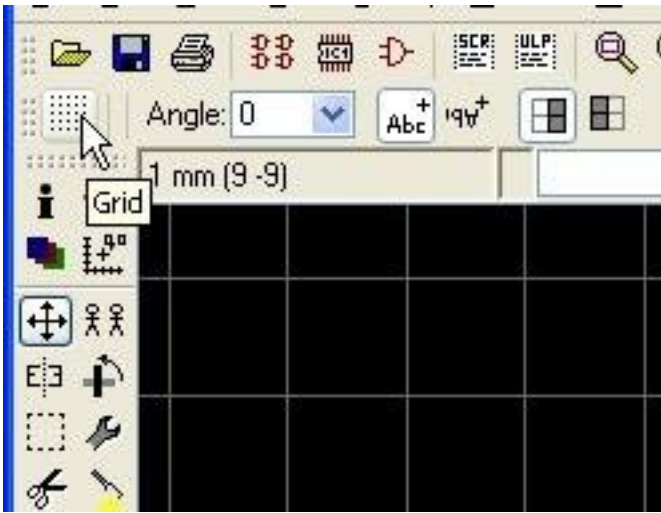
Library sekmesinden Packagei seçerek yeni bir kılıf tipi(SOT23-5) oluşturuyoruz.



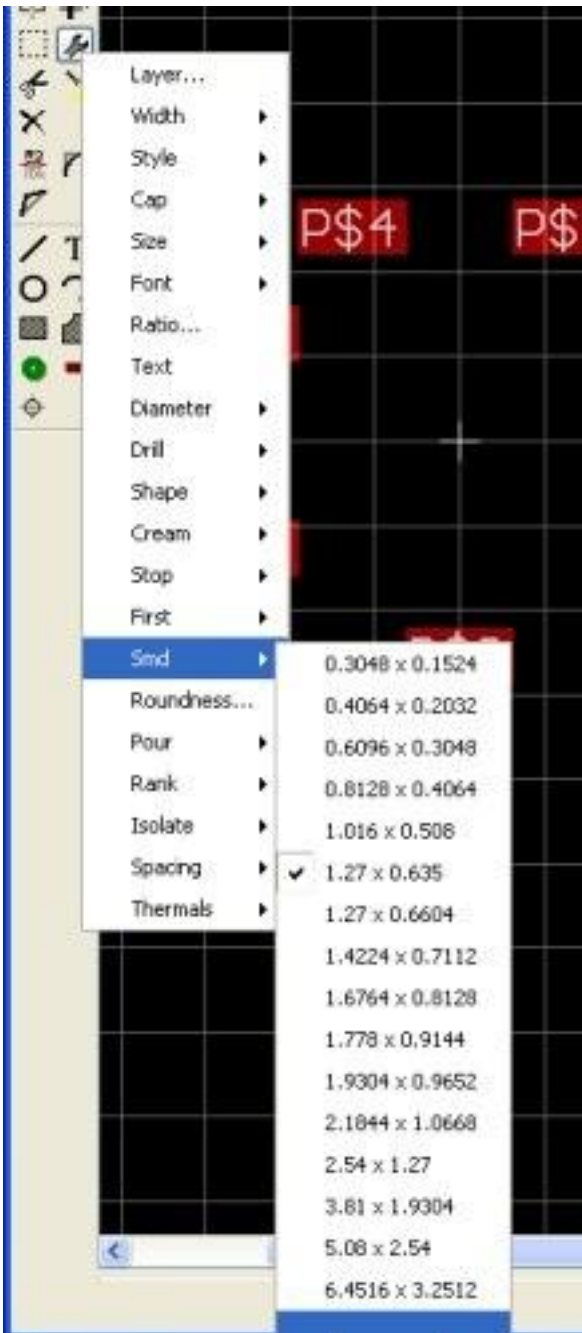
Ve 5 adet pad ekliyoruz.



Bazı Layer lar görünmeyebilir. Bunun için Display->All yapabilirsiniz.

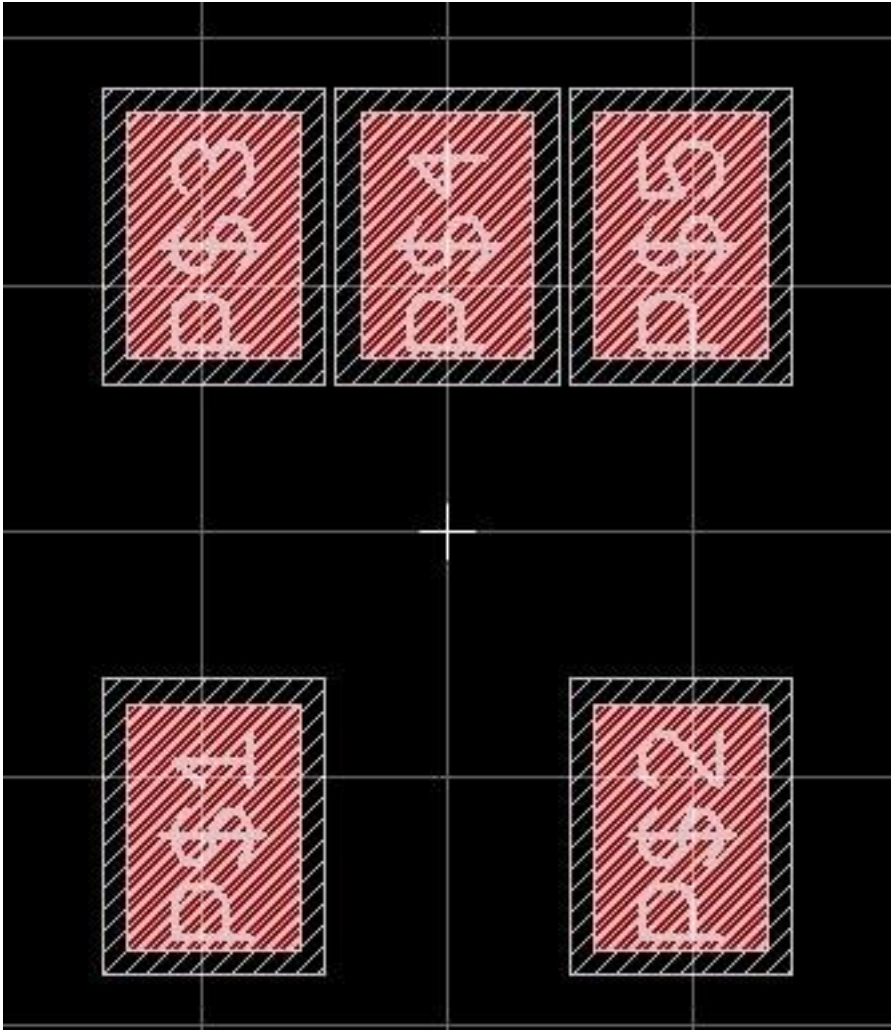


SOT23-5 kılıfı için datasheetdeki bilgileri referans alırsak, her bir padin genişliği 0.7mm ve yüksekliği 1.0mm'dir. Birebir orantı için bu ölçüleri de girmemiz gerekmektedir.

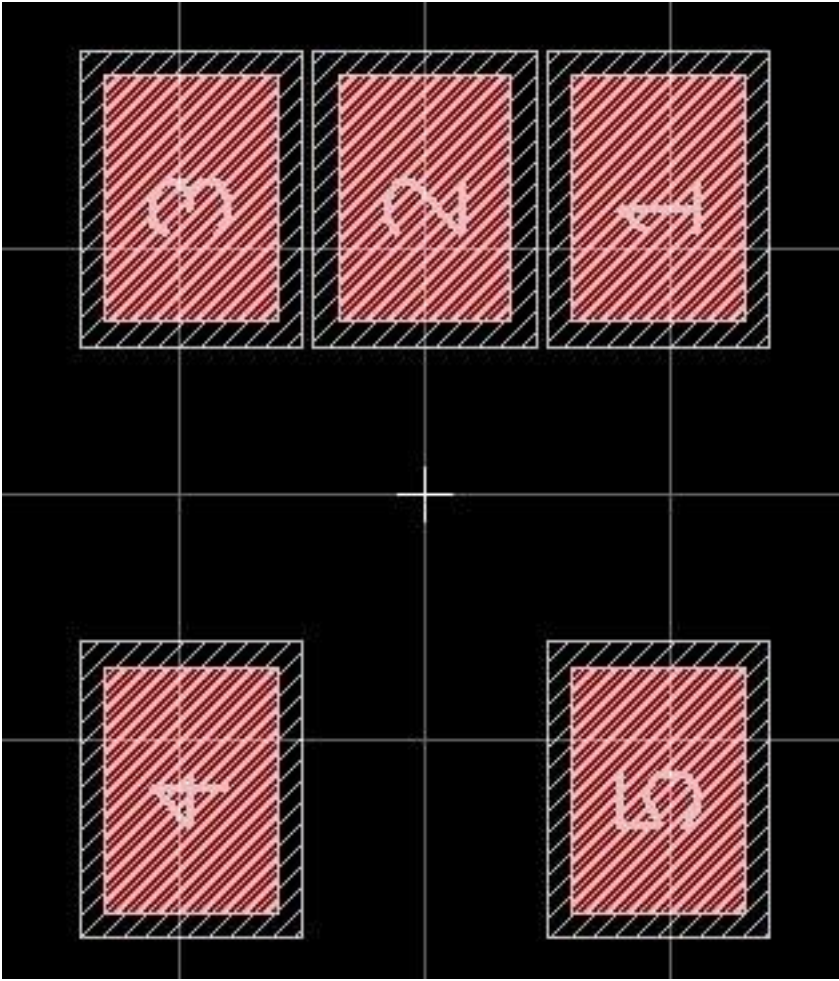




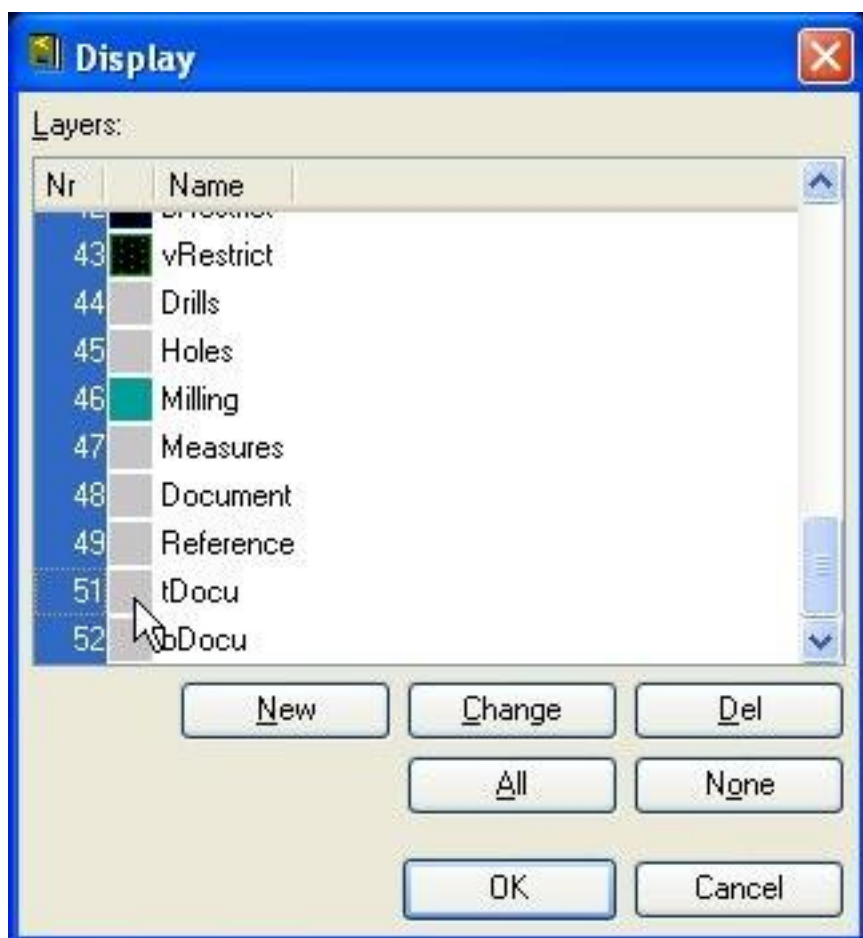
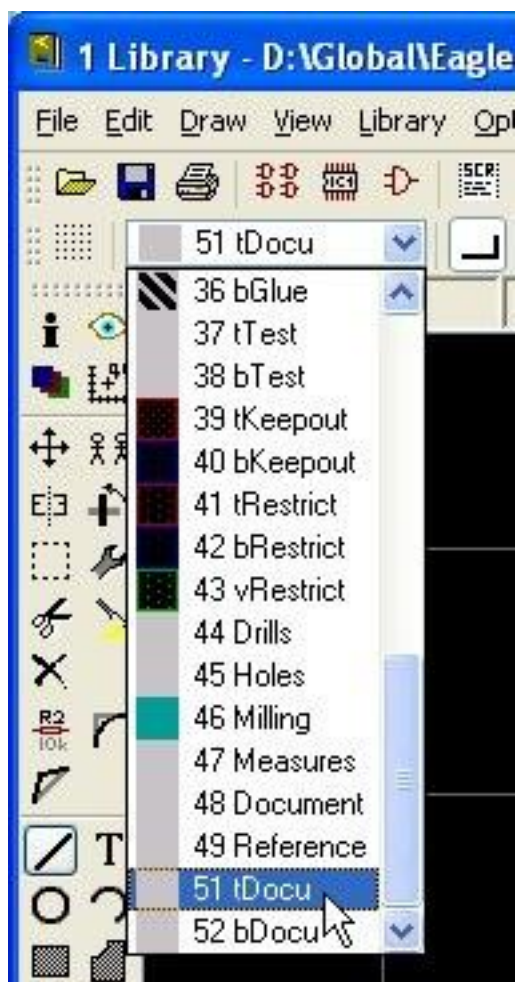
Sonra aşağıdaki gibi SOT23-5 kılıfına uygun şekilde düzenliyoruz.

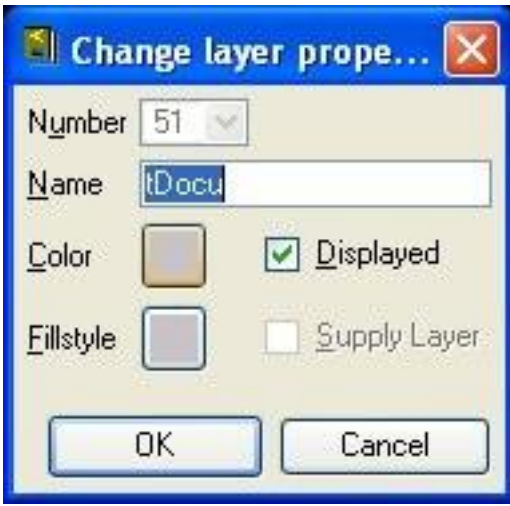


Ve isimleri datasheeti referans alarak değiştiriyoruz. Bunun için F4 kısayolunu kullanabilirsiniz.

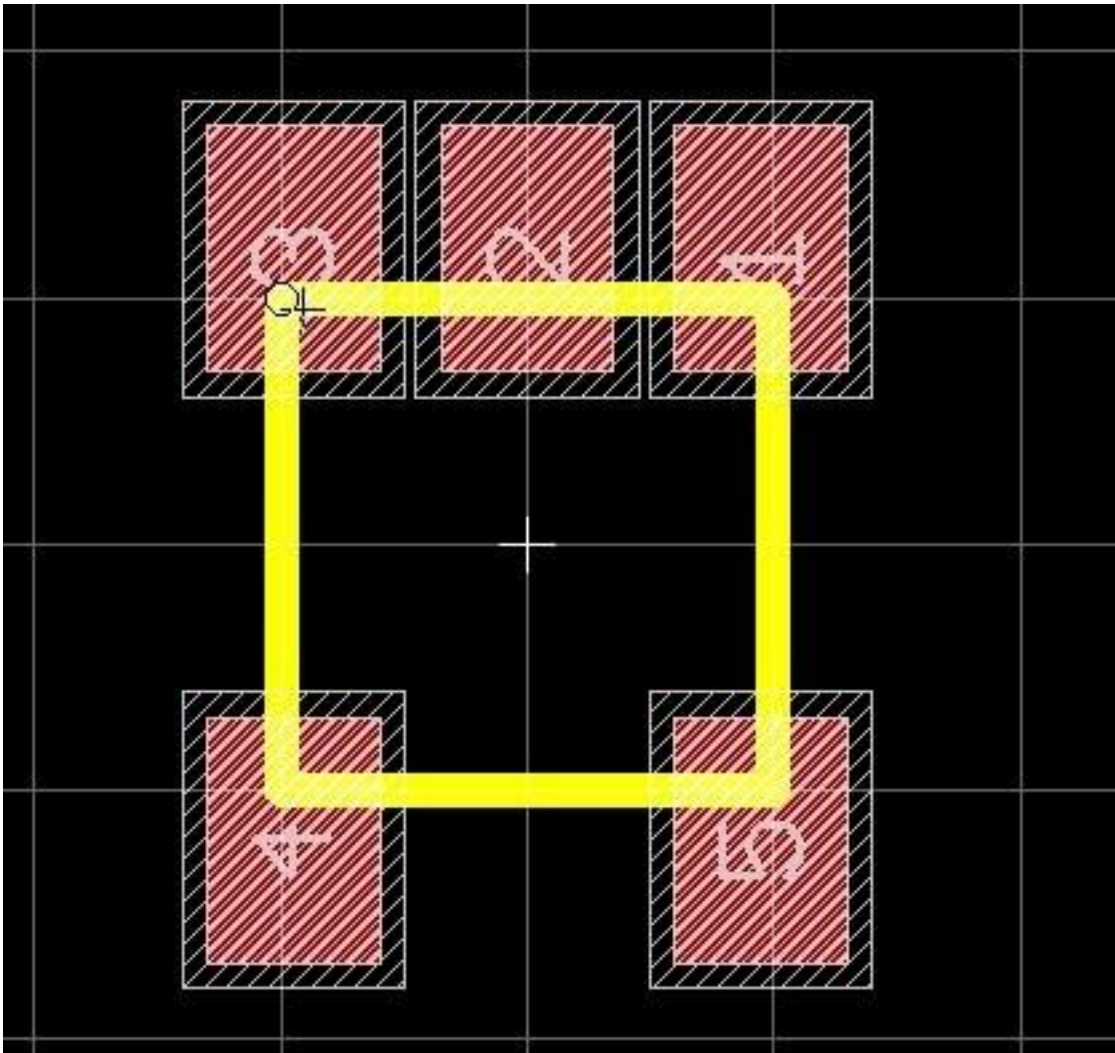


Sonra "tDocu" katmanını daha belirgin olması için sarı renk seçiyoruz.

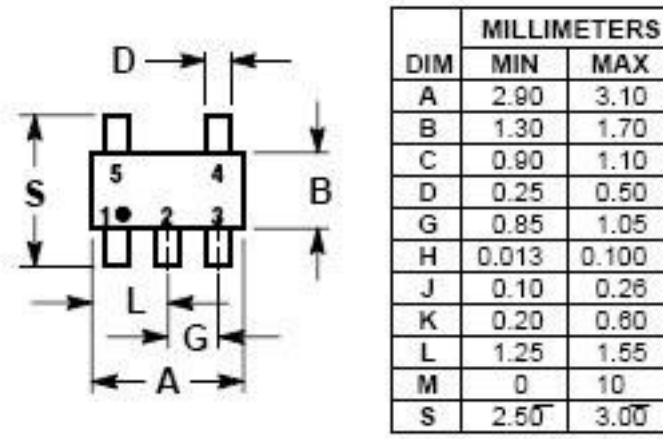




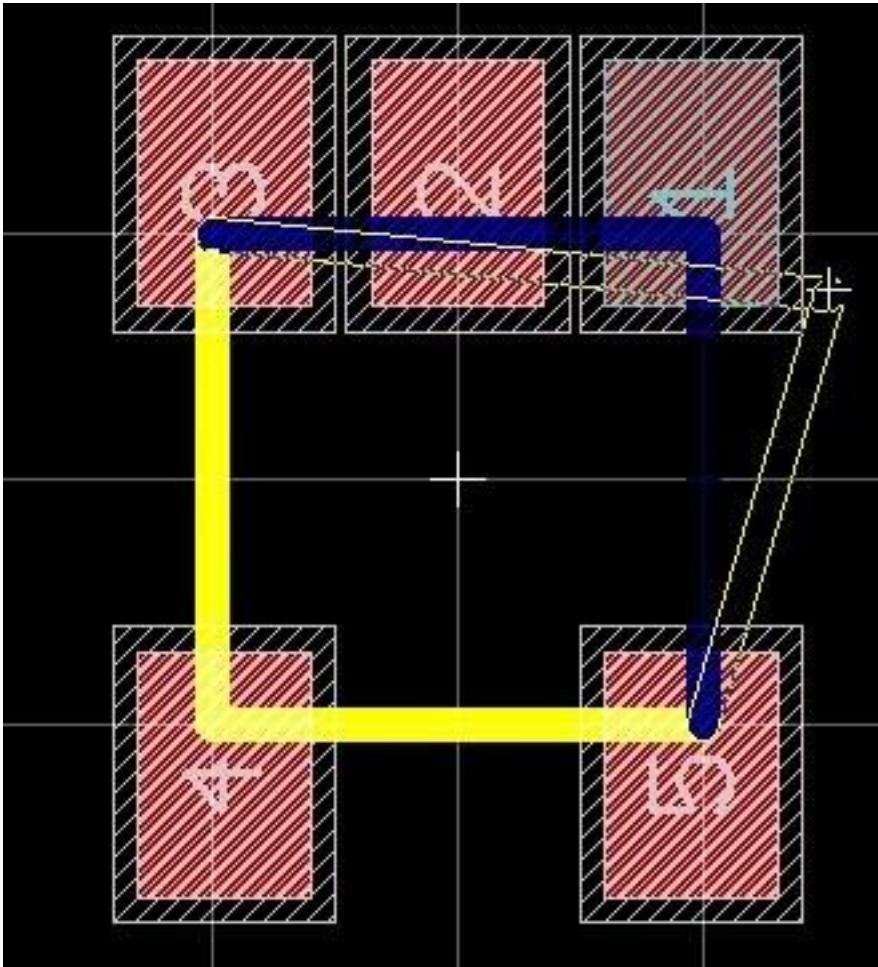
Ve oluşturduğumuz malzemenin boyutlarında "Wire" butonu ile bir kutu çiziyoruz.

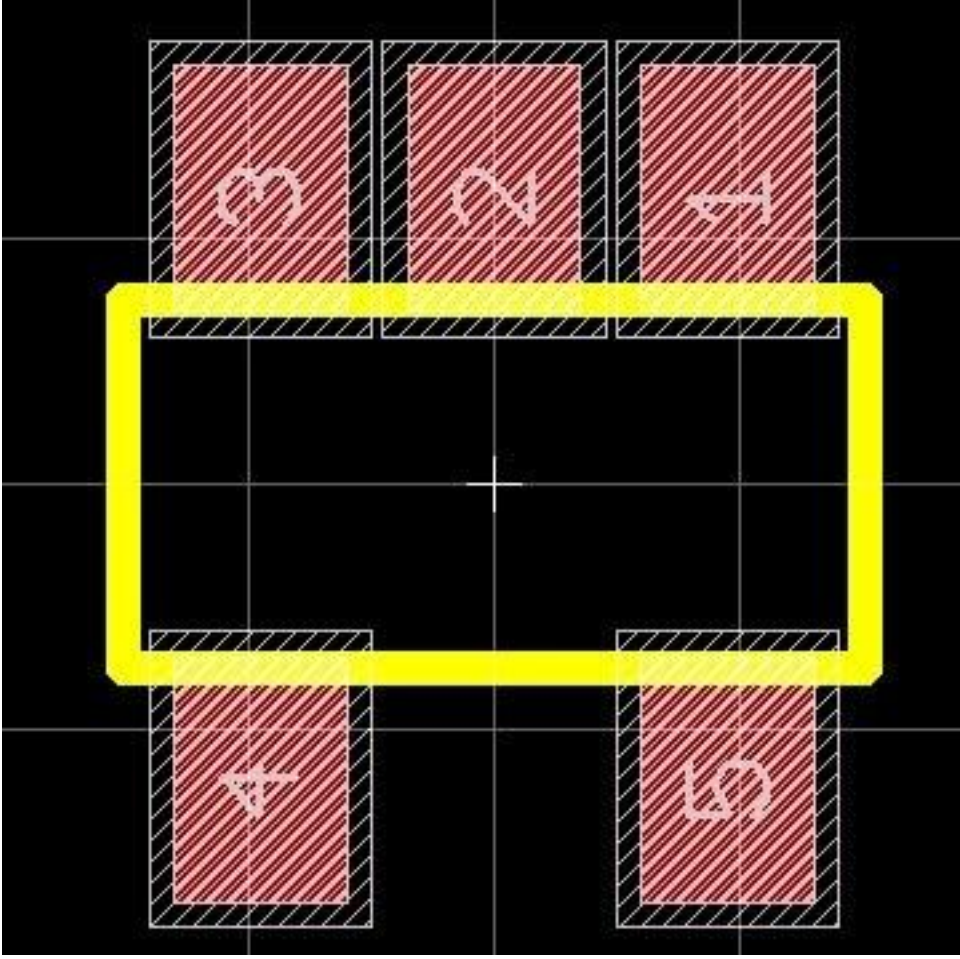


Metric mod'da doğru boyutlar aşağıdaki gibi olmalıdır.

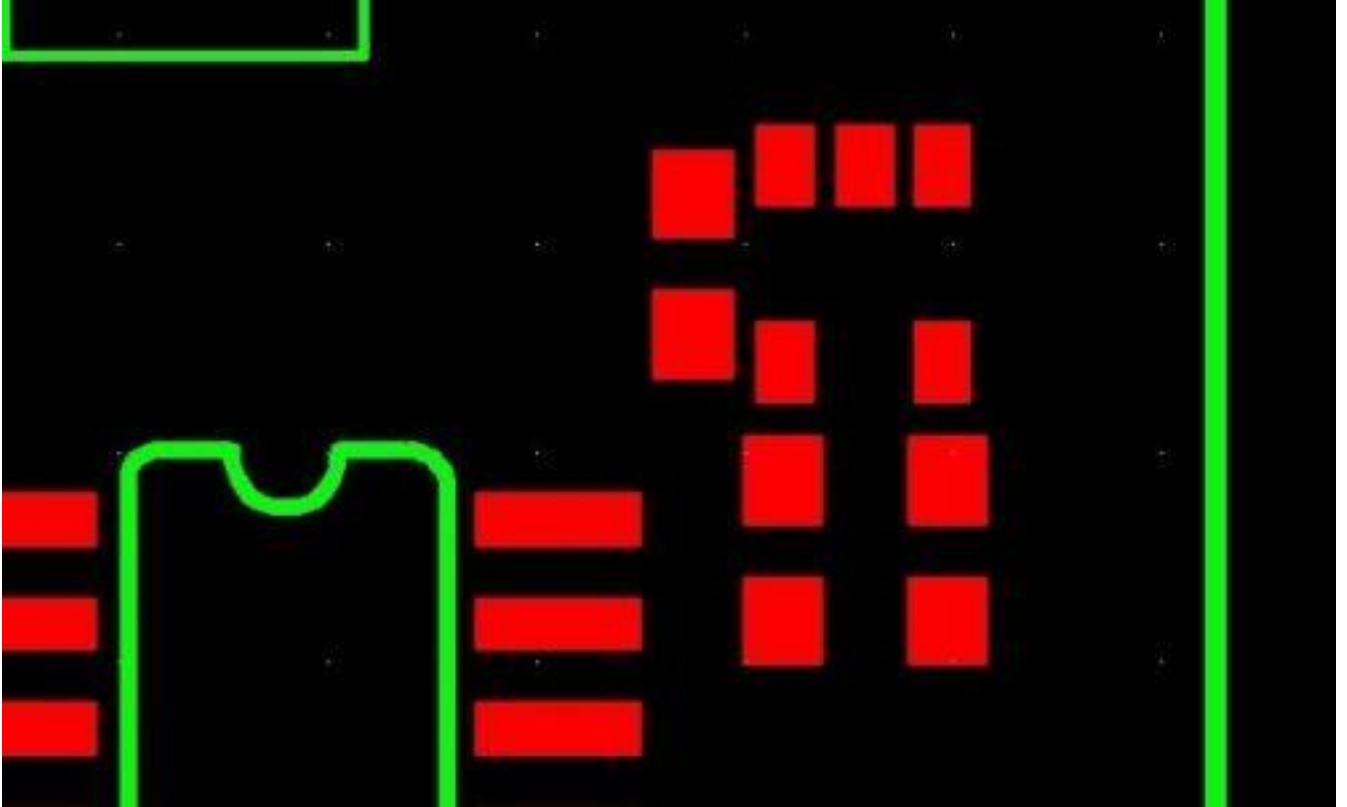


Mouse ile bu kutuyu büyüterek uygun şekle getirebilirsiniz.

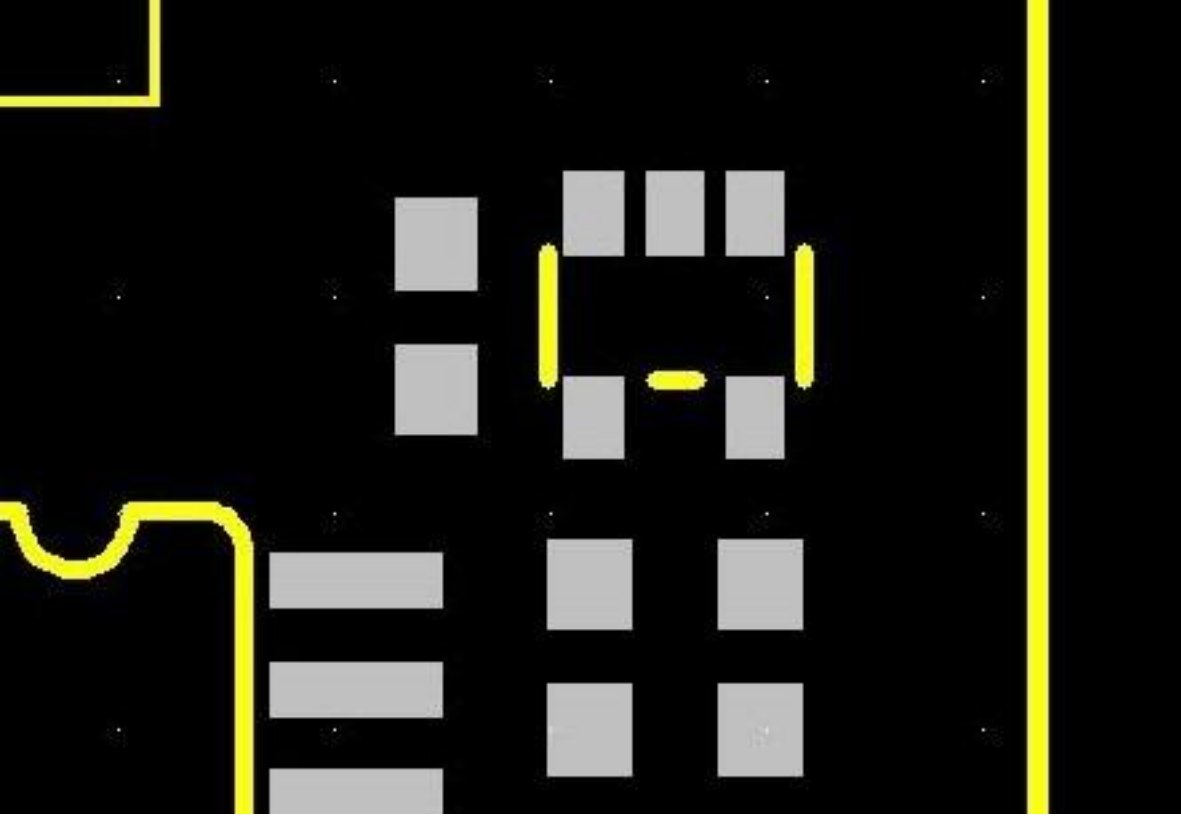




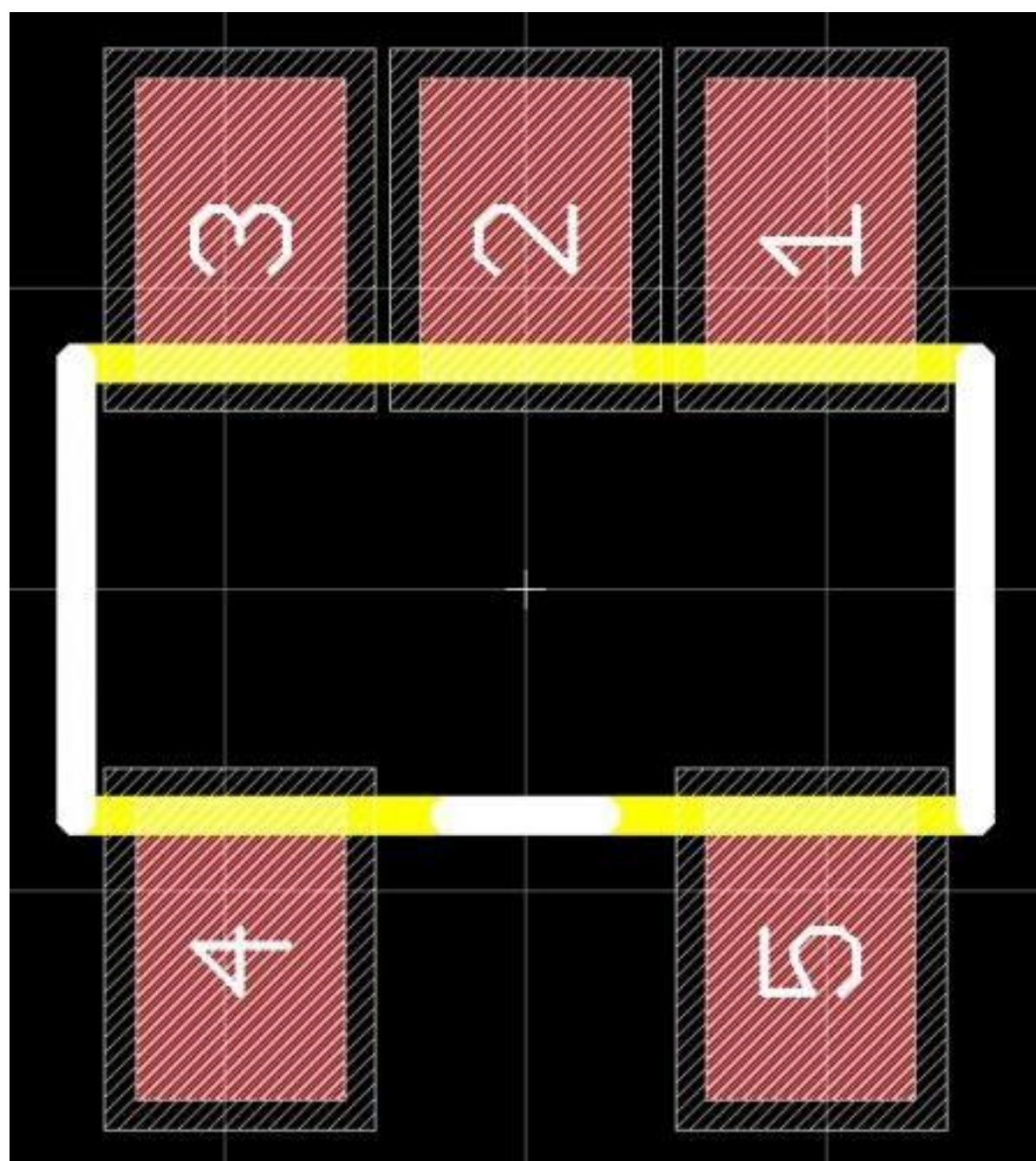
Oluşturduğumuz malzeme baskı devre çıktısında aşağıdaki gibi görünecektir.

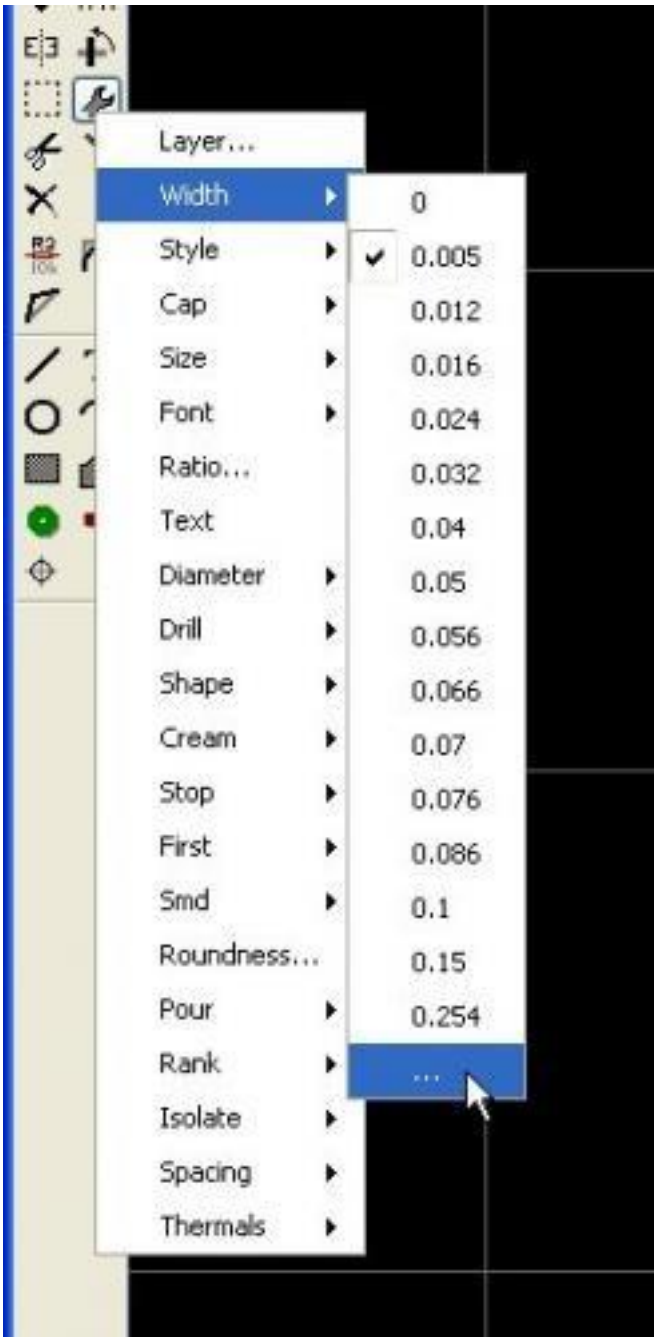


Fakat malzeme çerçevesi ekranda görünmemektedir. Normalde bu görünüm aşağıdaki gibi olmalıdır.

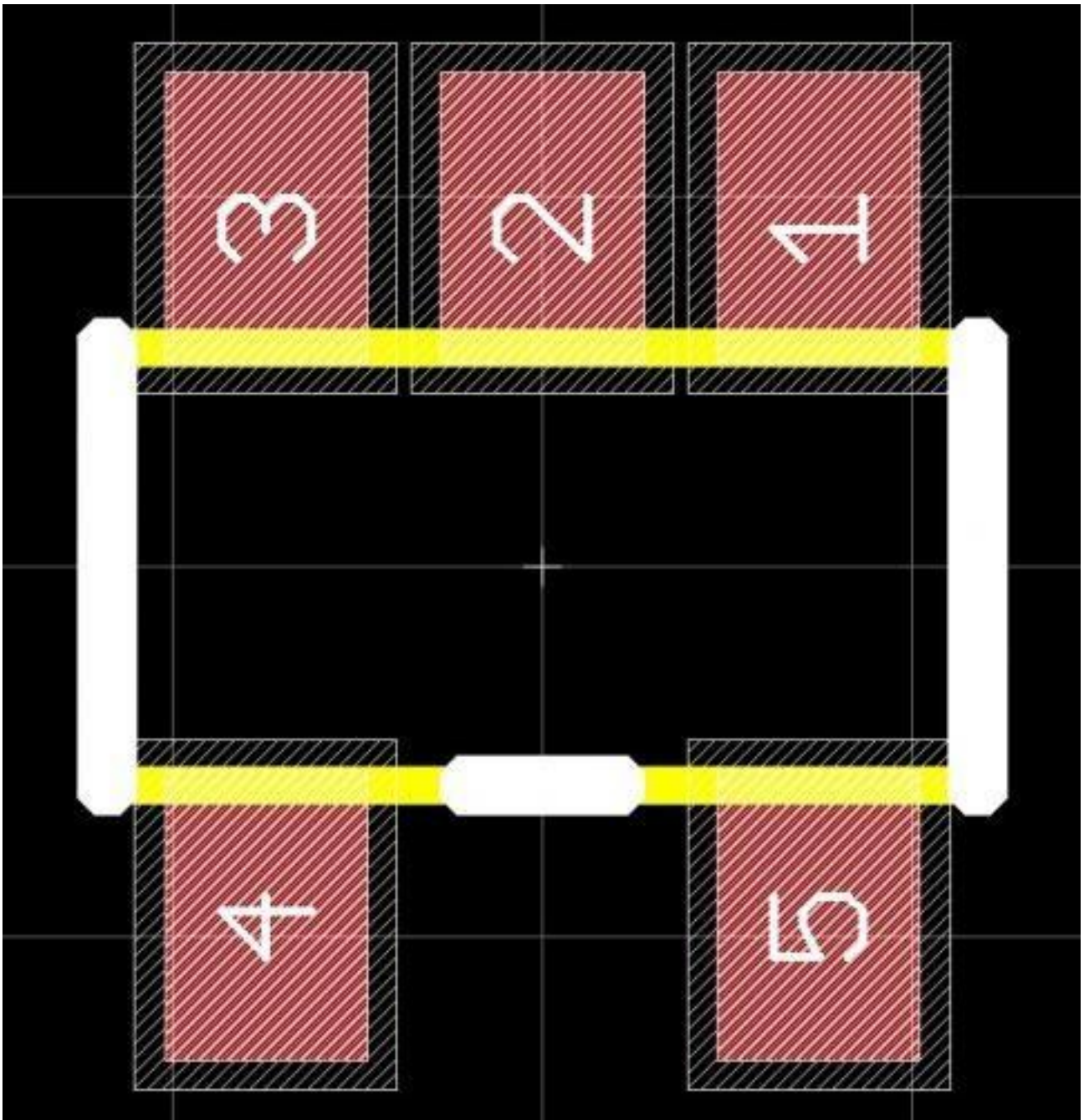


"tDocu" Katmanına tıkayıp Width sekmesine girersek bu değerin 0 olduğunu görürüz. Bunu 0.005 olarak değiştirelim.



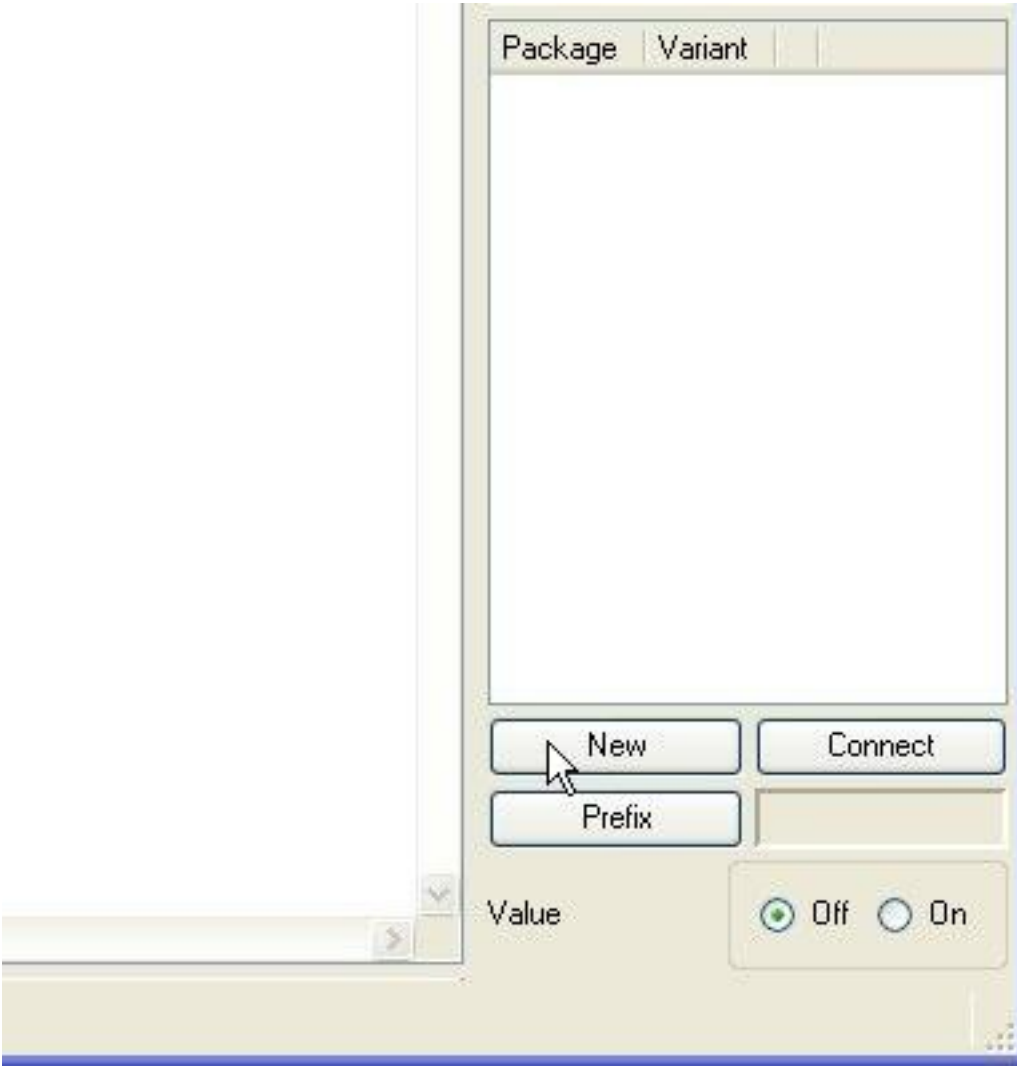
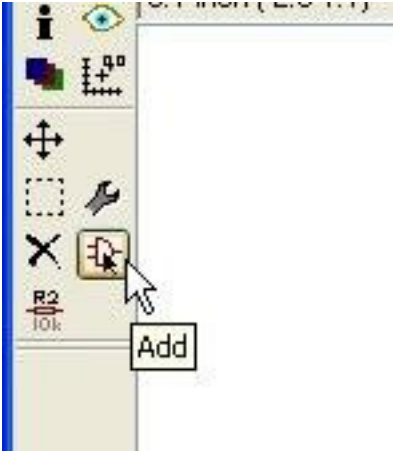


Yeni değeri girdikten sonra malzeme çerçevesi aşağıdaki gibi olacaktır.

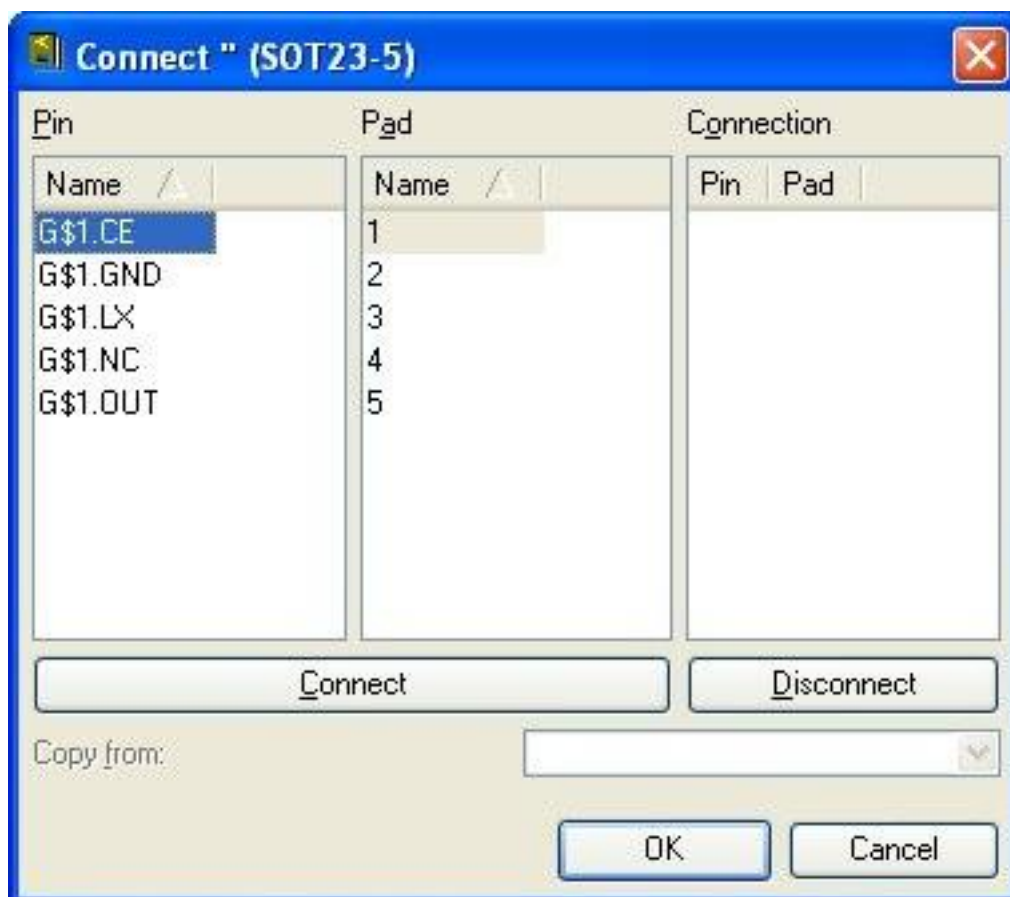
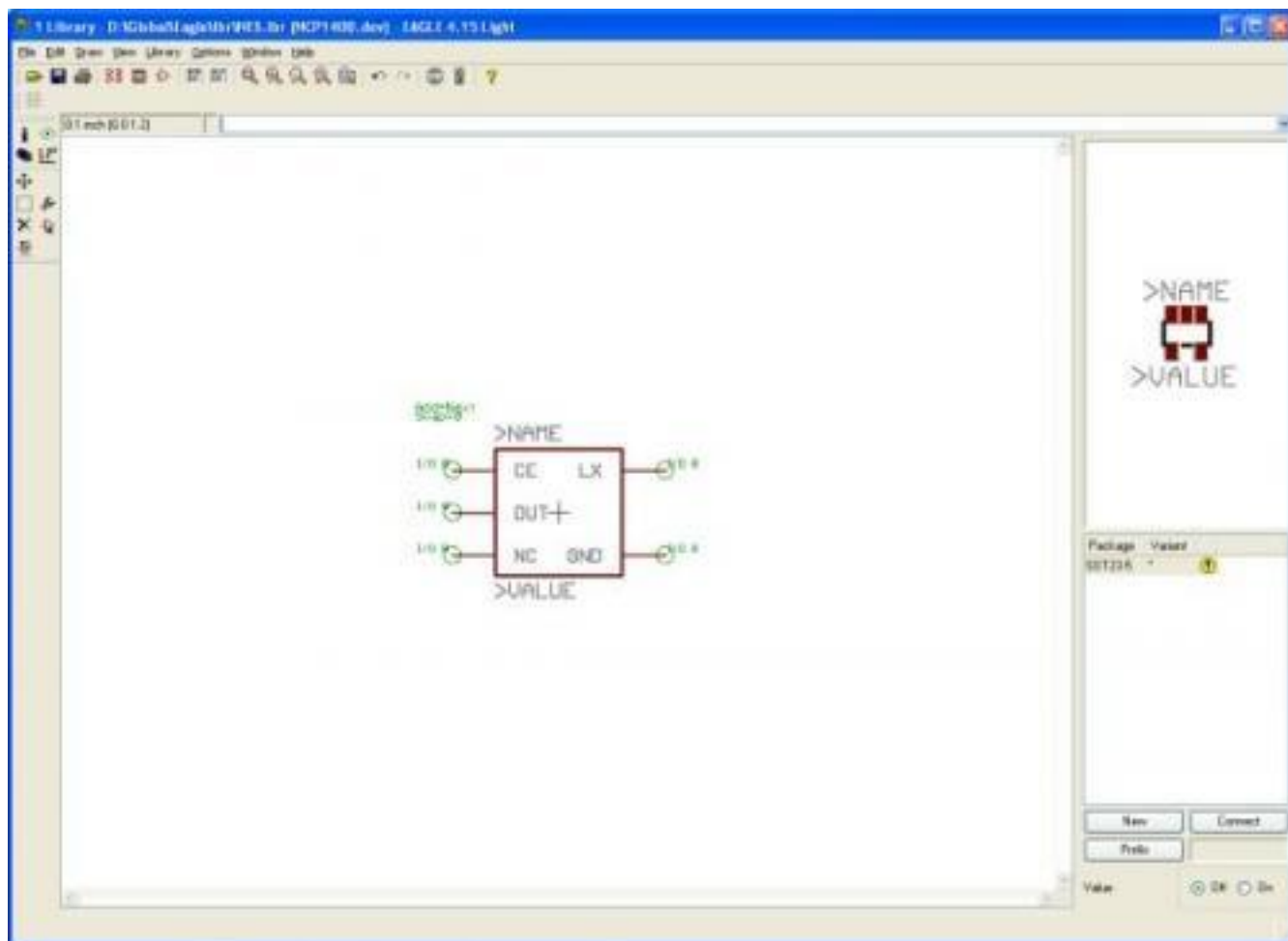


Son aşama olarak da >NAME ve >VALUE bilgilerini girip malzeme oluşturma işlemini tamamlıyoruz. Şimdi oluşturduğumuz bu malzemeyi kütüphanede birleştirmeliyiz.

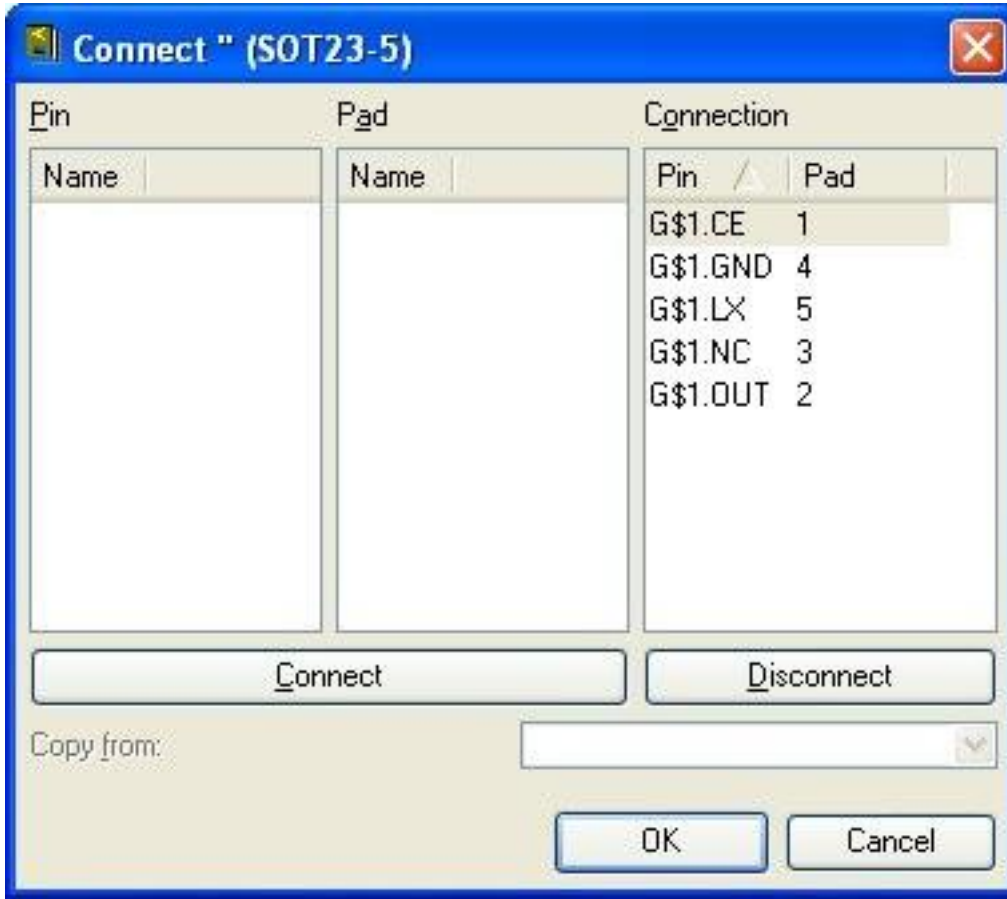




"New" butonuna tıkladıktan sonra karşımıza aşağıdaki ekran gelecektir. Burada sağ altta bir uyarı işareti çıkacaktır. Bu uyarı işareti şematik çizim ile kılıf tasarımı arasında pin bağlantılarının yapılmadığını belirtmektedir. "Connect" butonuna tıklayarak bu bağlantıları yapabilirsiniz.



Datasheete göre doru bilgiler girilerek ařağıdaki gibi bir bağılantı oluřurmalısınız.



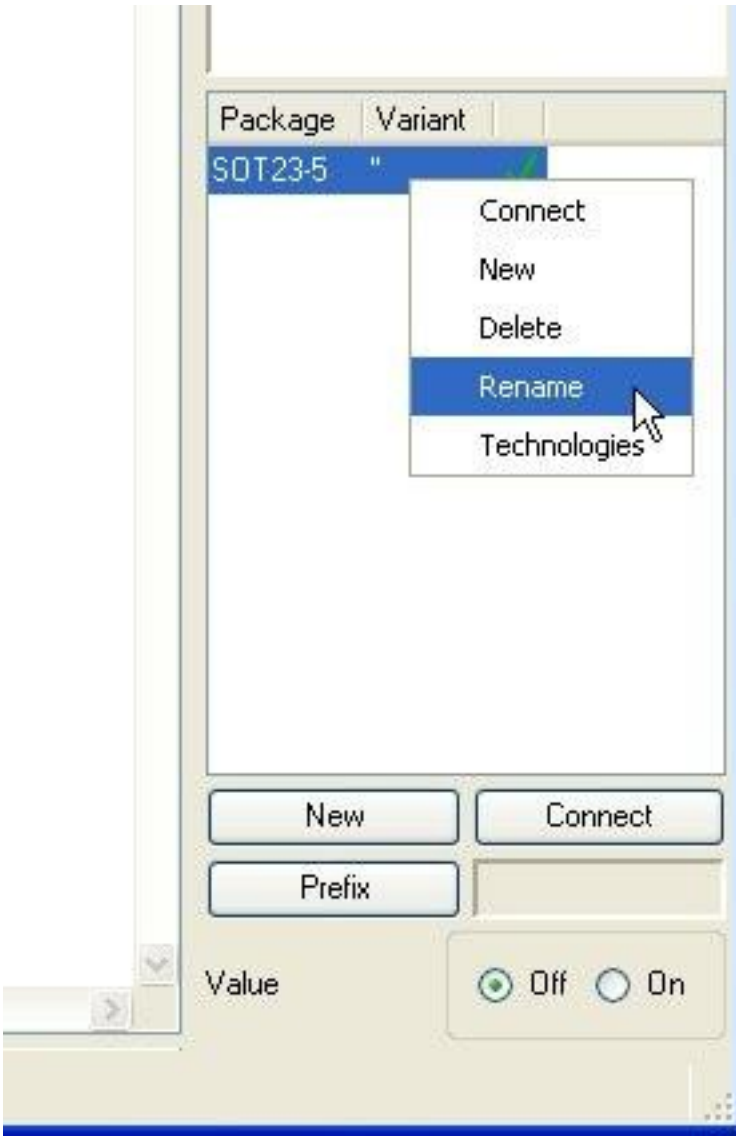
Pin		Pad		Connection	
Name		Name		Pin /	Pad
				G\$1.CE	1
				G\$1.GND	4
				G\$1.LX	5
				G\$1.NC	3
				G\$1.OUT	2

Connect Disconnect

Copy from:

OK Cancel

"Rename" sekmesinden isim girmelisiniz(" işaretinden farklı olmak zorundandır).



Bu aşamadan sonra yeni bir malzeme oluşturma işlemini tamamlamış oluyoruz. Artık bu malzemeyi kütüphaneden istediğiniz projeye ekleyebilirsiniz.

Control Panel - EAGLE 4.15 Light

File View Options Window Help

Name /	Description
--------	-------------

- + Libraries
- + Design Rules
- + User Language Programs User Language Programs
- + Scripts
- + CAM Jobs
- Projects
 - +  Simon
 - +  Empty Project

Close Project

New

Rename

Delete

Edit Description

Use all

Use none

Schematic

Board

Library

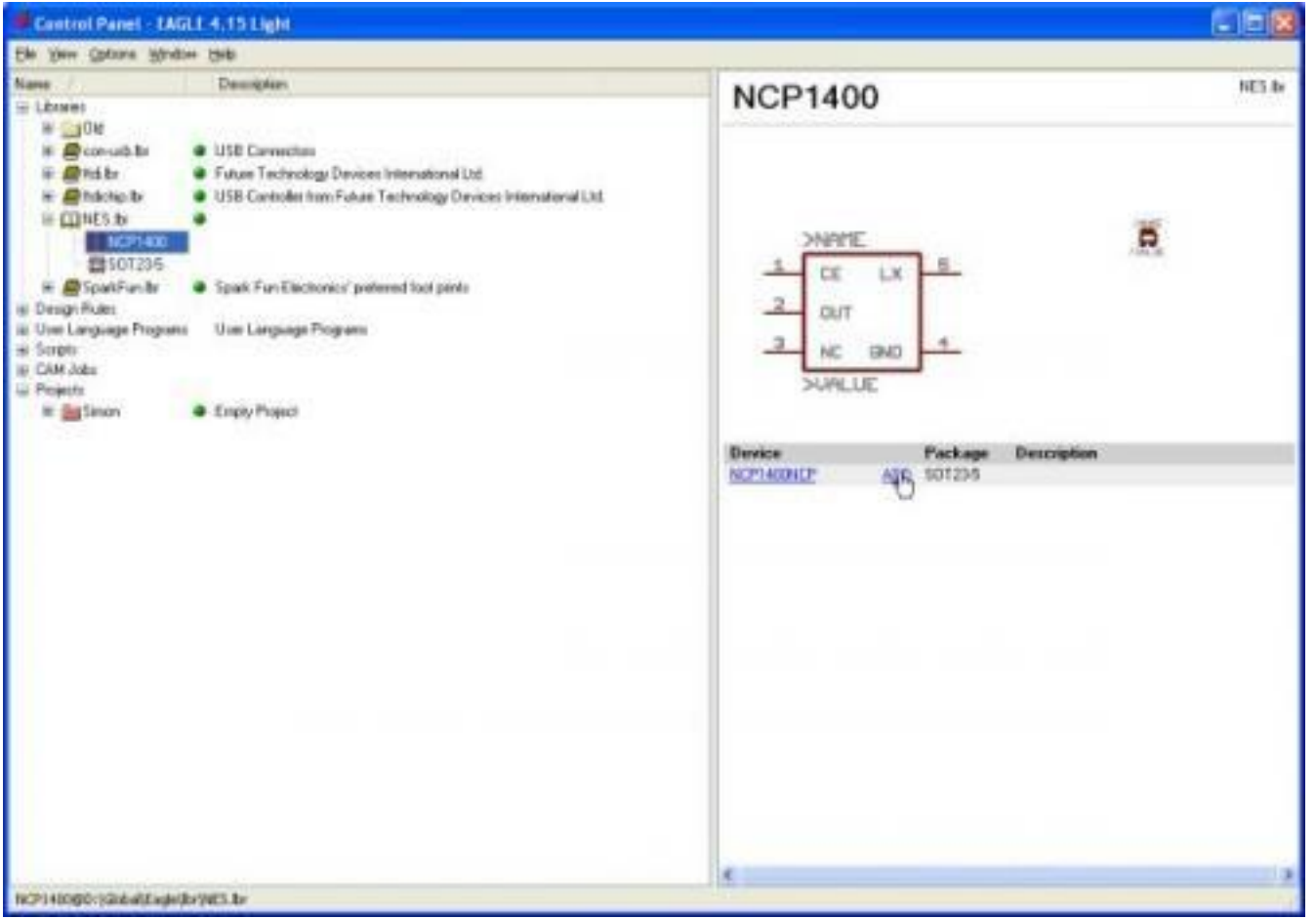
CAM Job

ULP

Script

Text

Folder



Eagle Eğitimi için gerekli dosyalara buradan ulaşabilirsiniz.

Püf Noktaları, Tecrübeler ve Dikkat Edilmesi Gerekenler

1. Devredeki tüm GND ler ortak olmalıdır.
2. Seri iletişimin çalışıp çalışmadığını kontrol etmek için hızlıca TX ve RX uçlarını birbirine bağlayabilirsiniz.
3. 9" kablo yerine standart olan 2" kabloları kullanabilirsiniz.
4. Breadboard üzerindeki kısa devre olasılıklarını en aza indirgeyin.
5. Yeni bir şey öğrenirken hep basit projeler ile işe başlayın.
6. Araştırmalarınızda Google en iyi arkadaşınızdır.
7. for(x=0;x
8. Lehimleme işlemi sırasında süngerinizi her zaman ıslak tutun.
9. Lehimleyeceğiniz yüzeyi önceden havya ile biraz ısıtın.
10. Malzemelerin baskı devre çıktılarında ölçüleri 1:1 olup olmadığını muhakkak kontrol edin.
11. TX ve RX bağlantılarının doğru olup olmadığını kontrol edin.

12. Yapacağınız boradlarda devre üzerine programlama çıkışı(ISP) koymayı ihmal etmeyin.
13. Malzeme yerleşim planını çıktı almayı unutmayın.
14. Seçtiğiniz konektörün pin diziliminin sizdeki konektör ile aynı olup olmadığını kontrol edin.
15. Eagle'da "Always Vector" seçeneğini her zaman aktif edin.
16. PCB yi basmadan önce gerber dosyalarını muhakkak kontrol edin.