RFID RC522, RDM6300 VE EM4095'in ARDUINO SHIELD TASARIMI VE YAZILIM TESTLERİ

ARDUINO SHIELD DESIGN AND SOFTWARE TESTS OF RFID RC522, RDM6300 AND EM4095

MUHAMMED ARİF KESKİN YUSUF ARLI

201813152043 201813152053

ABSTRACT

In this our study, Arduino Shield of RFID RC522, RDM6300 AND EM4095 was designed by usign Altium Designer. Provided by the Australian software company Altium Limited, Altium Designer is an open circuit board, FPGA, and design embedded software for Microsoft Windows. Radio-frequency identification (RFID) technology is used to transfer information wirelessly through radio waves for the purpose of data identification and tracking.We designed an arduino shield to connect RFID modules and integrated connections more easily and simply. Finally, We carried out the software tests of the module and the integrated.
Keywords: Arduino Shield, Altium Designer, Software, RFID technology.

ÖZET

Yaptığımız bu çalışmada, Okulumuzun laboratuvarında kullanılan RFID RC522, RDM6300 ve EM4095'in kablo kullanmadan daha kolay şekilde bağlantılarının gerçekleşmesi için Arduino Shield tasarımını gerçekleştirdik. Bu tasarımı yaparken PCB devre tasarımı imkanı sunan Altium Designer programını kullandık. Sonrasında ise Arduino ile bu RFID modüllerin yazılım testlerini yaptık ve çalışır durumda olup olmadığını gözlemledik.
Anahtar Kelimeler: Arduino Shield, Altium Designer, Yazılım Testleri, RFID Teknolojisi

İÇİNDEKİLER

1.GİRİŞ

RFID teknolojisinde temel olarak RFID etiketi ve RFID okuyucusu en kritik bileşenlerdir. Bunlara ayrıca RFID yazıcısı, RFID anteni, sistemin kullanacağı yazılımı ekleyebiliriz. Bir RFID etiketi çip, güç kaynağı ve antenden oluşmaktadır. Etiketi, bu sayede RFID okuyucularıyla iletişim kurabilir ve veri aktarıp alabilir. RFID, üzerinde mikroişlemci ile donanmış etiket taşıyan bir nesnenin, bu etikette taşıdığı kimlik yapısı ile hareketlerinin izlenebilmesine imkan veren radyo frekansları ile çalışan teknolojiye verilen addır. RFID kelimesinin açılımı İngilizce olarak Radio Frequency Identification'dır. Oldukça eski bir teknoloji olan RFID'nin kullanımı, İkinci Dünya Savaşı yıllarına kadar uzanmaktadır. Ancak etiketlerin maliyetlerinin yüksekliği ve kullanım zorluğu, RFID teknolojisinin uzunca bir süre tarafından kullanılmamasına neden olmuştur. RFID etiketlerinin ayırt edici bir diğer özelliği de çalıştıkları frekans aralıklarıdır. Günümüzde yaygın olarak kullanılan etiketlerin frekansları daha çok HF (High Frequency) aralığındadır. Bir etiketin maliyeti sadece aktif ya da pasif olmasına göre değil, çalıştığı frekansa göre de değişmektedir. Endüstride 13.56 MHz'lik etiket kullanılan projeler, şirketlere maliyetleri açısından da oldukça çekici gelmektedir. Ancak bazı uygulamalarda UHG frekans aralığı (800-900 MHz'lik) kullanılması performans adına daha iyi sonuçlar vermektedir. Herhangi bir projede hangi frekans aralığı kullanan etiketlere karar verilmesi, o projenin düşünülen ortamda gerçekleştirilip gerçekleştirilmeyeceği kararı gibi kritik kararlar tamamen tecrübe ve bilgi birikimi ile donatılmış uzman ekipler sayesinde olmalıdır. Aksi halde RFID projeleri, maliyetleri ve sonunda yaşanan koca bir başarısızlık ile birçok şirket kuruma faydadan çok zarar getirmiştir.

Yaptığımız projede bize birçok imkan tanıyan, sektörde kullanımı oldukça yaygın ve PCB devre tasarım programı olan Altium Desginer'ı kullandık. Altium Desginer hızla artan teknolojik gelişmeler ve aynı hızda artan karmaşık ürün yapılarına cevap verebilmek için, zamana karşı yarışan mühendislerin hızını arttıran birçok özelliğe sahip tasarım programıdır. Altium Designer, bu bilinçle hazırlanmış bir sistem olup, çok karmaşık ve gelişmiş PCB tasarımlarının en kısa zamanda tamamlanabilmesi için mühendislerin hızını maksimize eden bir kullanım ve bütünleşik (unfied) bir ürün geliştirme sistemi mimarisine sahip. Bu özelliği sayesinde, Altium Designer, dünyada ve Türkiye'de lider elektronik firmalarının tercih ettiği, konsept tasarımından imalata kadar gerekli bütün araçları bünyesinde barındıran, komple ve profesyonel PCB tasarım yazılımıdır.

Bu yazımızda, projede kullanılan program ve malzemelerin genel bir açıklaması, Altium Designer programının avantajları, RFID teknolojisi hakkında genel bilgilerden bahsedilmiştir. İkinci bölümde tasarımda kullandığımız RC522, RDM6300 ve EM4095'in özellikleri anlatılmaktadır. Üçüncü bölümde ise Altium tasarımında yaptığımız Şematik çizimleri ve PCB tasarımına yer verilmektedir. Son bölümde ise Arduino'da kullanılan malzemelerin yazılım testleri hakkında bilgilere yer verilmiştir.

2.MALZEMELERİN ÖZELLİKLERİ

2.1 RFID RC522

RC522 RFID kartı, NFC frekansı olan 13,56 MHz frekansında çalışan tagler üzerinde okuma ve yazma işlemini yapabilen, düşük güç tüketimli, ufak boyutlu bir karttır. Arduino başta olmak üzere birçok mikrodenetleyici platformu ile beraber rahatlıkla kullanılabilir. Enerjisiz çalışmasına en olanak tanıyan özelliği bilgiyi depolayan ve işleyen bir mikroçip ve bir sinyal almak ve iletmek için bir anten içermesidir. Okuyucu maksimum 10Mbps veri hızına sahip 4 pinli Seri Çevresel Arabirim (SPI) üzerinden mikrodenetleyici ile iletişim kurabilir. Ayrıca I2C ve UART protokolleri üzerinden de iletişimi destekler. Modülün çalışma voltajı 2.5 ile 3.3V arasındadır.

Özellikleri:

- Çalışma Frekansı: 13,56 MHz
- Çalışma Akımı: 13-26mA
- Haberleşme Protokolü: SPI
- Kart Boyutları: 40x60mm
- Desteklenen Kartlar: mifare1 S50, mifare S70 mifare ultralight, Mifare pro ve mifare desfire
- Uyku Akımı: 80uA
- Okuma Aralığı 5cm

❖ Datasheet

Şekil 1. RC522 Modülü

2.2 RDM6300

RDM6300 125 kHz kart okuyucu, mini-modül, okuma/yazma kartlarından ve 125 kHz'le uyumlu sadece okuma etiketli kartlardan okuma işlemi yapması için tasarlanmıştır. Bu ürün ofis / ev güvenliği, kişisel kimlik, erişim kontrolü, anti-sahtecilik, interaktif oyuncak ve üretim kontrol sistemleri gibi birçok alanda rahatlıkla uygulanabilir. Ayrıca bu ürünü Arduino ve Raspberry gibi geliştirme kartlarıyla da uyumludur.

Özellikleri:

- Çalışma Gerilimi: +5V
- 125 kHz okuma frekansı
- Maksimum etkili mesafesi 50mm
- Harici anten destekler
- Kod çözme aralığı 100ms den az
- UART arayüzü sayesinde rahat kullanım
- Sadece okuma ve okuma/yazma etiketleriyle uyumlu EM4100'ı destekler
- Ağırlık: 15gram
- Ebatlar: 3.8x1.8x1.2 cm

❖ Datasheet

Şekil 2. RDM6300

2.3 EM4095

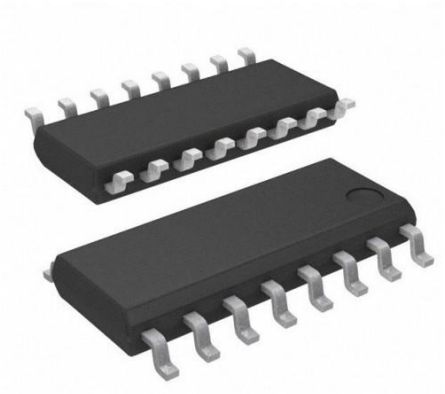
EM4095 (Önceki Adı P4095) çipi, RFID baz istasyonunda kullanılması amaçlanan bir CMOS tümleşik alıcı-verici devresidir. RFID Sistem teknolojisine hızlı bir adım için geliştirme kiti olan EM4095 tasarlanmıştır. EM okuyucu çipini içeren Em4095 RS232 arayüzü ile programlanabilir. Tamamen bir okuyucu modül olarak kullanılabilir ya da mikrodenetleyici ile programlanabilir. Aşağıda geliştirme kiti olan EM4095 ile kullanılabilen çip türleri:

- EM4100 / EM4102
- EM4005 / EM4105
- EM4150
- EM4170
- EM4025
- EM4056

Özellikleri:

- 100-150 kHZ taşıyıcı frekans aralığı
- SO16 paket kılıfa sahip
- Uyku modu tüketimi 1uA
- -40 ila +85 ˚C sıcaklık aralığı
- USB uyumlu güç kaynağı aralığı
- Harici kuvars gerekmez
- Çoklu aktarıcı protokol uyumluluğu (Örn: EM4102, EM4200, EM4450 ve EM4205/EM4305)

❖ Datasheet



Şekil 3. EM4095

3.ALTIUM İLE PCB TASARIMI

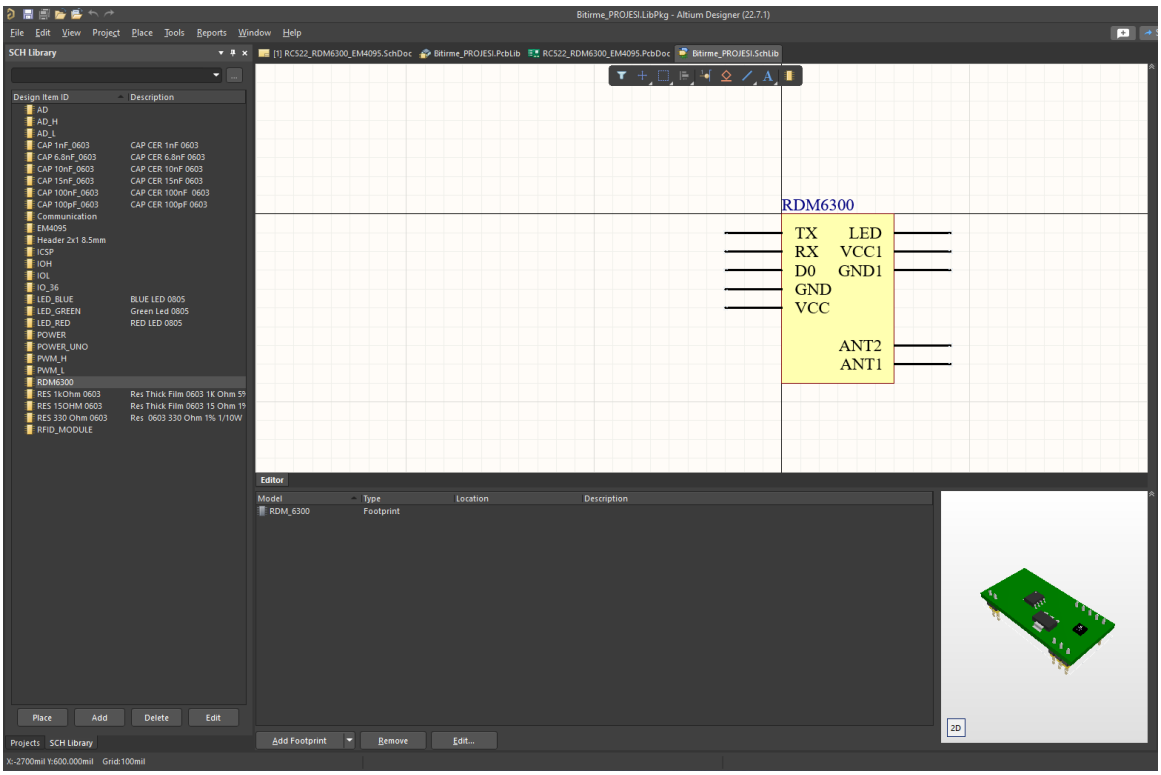
Altium Designer aslında elektronik ürün geliştirmek için eksiksiz bir ortam oluşturmak için gerekli olan tüm araçları bir araya getiren bir yazılım entegrasyon platformudur. Altium Designer, şematik ve HDL tasarım yakalama, devre simülasyonu, sinyal bütünlüğü analizi PCB tasarımı ve FPGA tabanlı gömülü sistem tasarımı ve geliştirmeden tüm tasarım görevleri için araçları içerir. Buna ek olarak Altium Designer ortamı çok çeşitli kullanıcı gereksinimleri karşılamak üzere özelleştirilebilir.

3.1 Kütüphane Oluşturma

3.1.1 Şematik Kütüphanesi

Şematik kütüphane Düzenleyicisi, şematik komponentleri oluşturmak, değiştirmek ve bileşen kitaplıklarını yönetmek için kullanılır. Bileşenler, Şematik Kitaplığı Düzenleyicisindeki tasarım nesneleri ile oluşturulur. Bileşenler bir devre şeması kitaplığından diğerine veya devre şeması düzenleyicisinden devre şeması kitaplığı düzenleyicisine kopyalanabilir ve yapıştırılabilir.

İlk olarak yapmamız gereken şey eğer hali hazırda elimizde bir kütüphane dosyası varsa onu açmak olacaktır. Eğer yoksa buradan bir Schematic Lib dosyası oluşturabilirsiniz. Hemen ardından Sol tarafta karşımıza çıkan alanda **Add** diye bir buton karşımıza çıkmaktadır. Oradan kütüphanenize istediğiniz kadar komponent ekleyebilirsiniz. Bundan sonra karşımıza boş bir alan çıkmaktadır. İşte orası bizim komponentlerimizin şematik gösterimleri yer alacaktır. Gösterimlerini, pin dizilimlerini, designator ayarlarını, komponentin açıklamalarını vs, her türlü şeyi burada belirtip, üzerinde istediğimiz gibi oynayabiliriz.



Şekil 4. Malzeme Kütüphanesi

3.1.2 Footprint Kütüphanesi

Footprint oluşturmada malzemenin datasheet'i bizim için çok önemlidir. Çünkü tasarım sırasında belirleyeceğimiz her türlü pad dizilimi, uzunluğu vb. şeyler, tamamen datasheet'e bağlı bir şekilde ilerleyecektir. Bu incelemeleri yaptıktan sonra çizimimizi 4 adımda tamamlayacağız.

- Padleri oluşturun
- Komponentin uzunluklarını ve alanını tanımlayın
- STEP bilgilerini ekleyin
- Footprint'i Kaydedin

Gerekli mesafe birimlerini belirledikten sonra tasarıma başlayabiliriz. İlk olarak PCB lib dosyamızda **Add** kısmına tıklayarak boş bir komponent footprint alanı oluşturuyoruz. Ardından **P** tuşuna basarak hızlıca Pad yazan kısma tıklıyoruz. Artık padleri yerleştirmeye başlayabiliriz. İlk pad yerleştikten sonra grid'in birimine dikkat ederek diğer padler yerleştirilmektedir. Sonrasında layer (katman) düzenlemesine de dikkat edilmelidir. Diğer katmanlarda işlemler yapıldıktan sonra 3D model için Step dosyasını yükleriz.

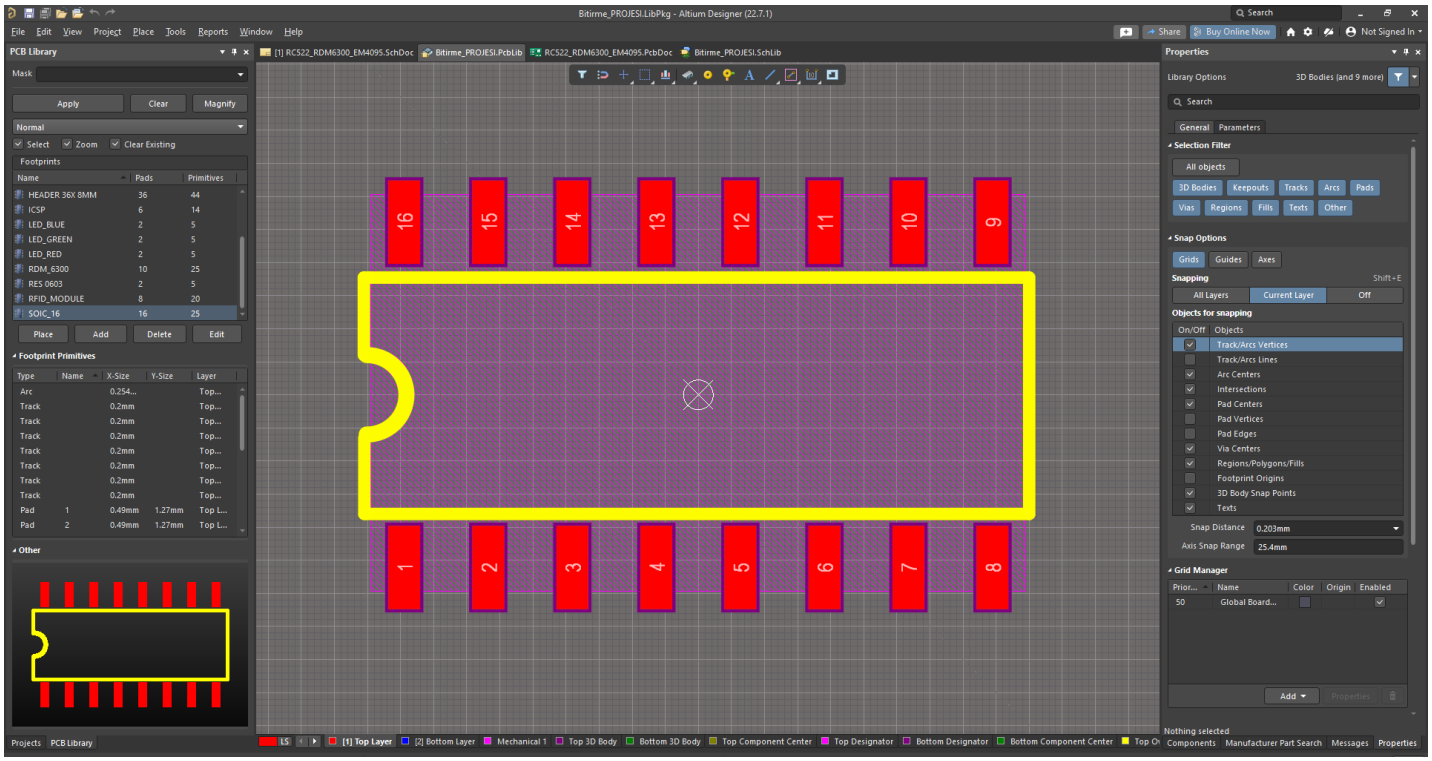
Step dosyası, footprintini çizdiğimiz komponentin 3 boyutlu görüntüsü için gerekli olan dosyadır. Altium için (*. step) uzantılı dosyalar indirilmelidir. Step dosyası, footprint, şematik vb. için gerekli olan dosyaları indirmek için kullanabileceğimiz internet siteleri mevcuttur. Bunlar:

- SnapEDA
- 3D Content Central
- GRABCAD COMMUNITY

Ayrıca şematik çiziminde kullandığınız komponentler hakkında bilgi edinmek, komponentin stok bilgisi, fiyatı, üretici firmanın bize sağladığı datasheet ve malzeme özelliklerini de bu sitelerden öğrenebiliriz. Bunlar:

- Digi-Key
- Farnell
- Mouser Electronics

Footprintleri tamamladıktan sonra ilk aşamada yaptığımız şematik kütüphanesindeki komponentlere tek tek footprintleri ekleriz. Bu işlemden sonra kütüphane oluşturma kısmını bitirmiş olduk. Bir sonraki işlem oluşturmuş olduğumuz kütüphaneleri şematik kısmına aktaracağız.

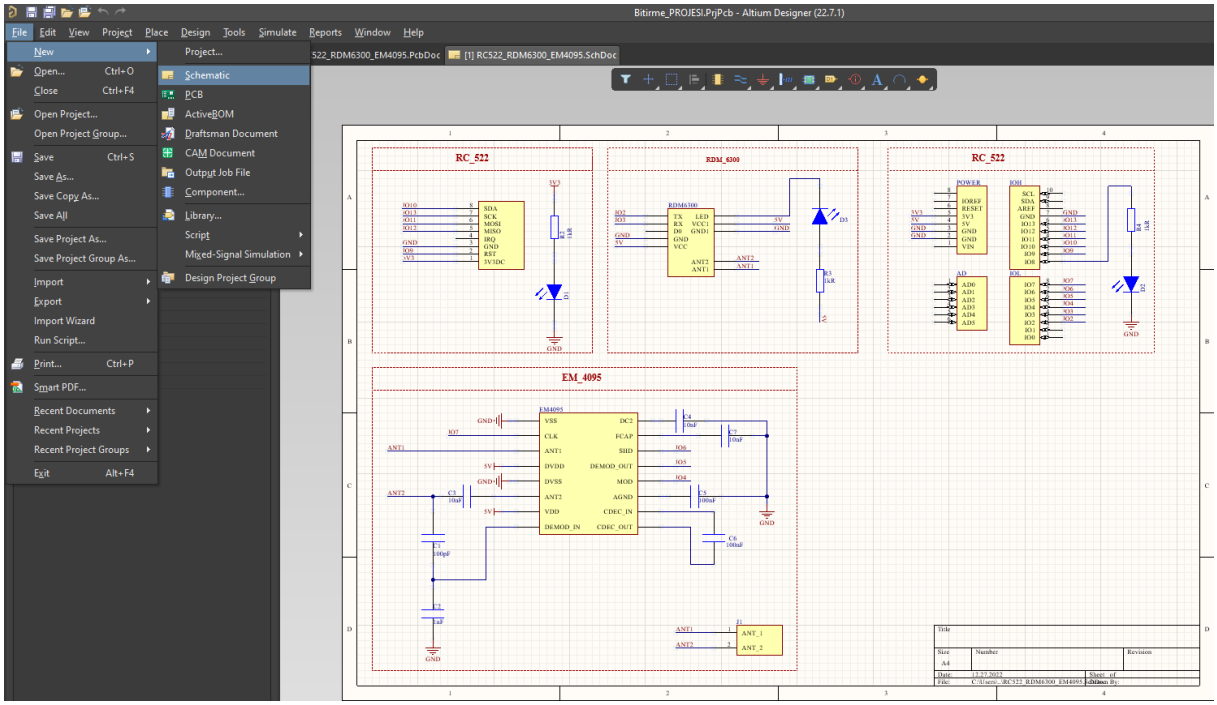


Şekil 5. Footprint Kütüphanesi

3.2 Şematik Tasarım

Şematik Tasarım; birçok elektronik sembolün (komponent) birbirleriyle özel bir şekilde ilişkilendirilmesiyle, ürüne ait, elektronik devrenin mantıksal temsili şeması ve dolayısıyla elektronik ürün tanımlanmış olur. Şematik devrenin tamamlanmasıyla, devre simülasyonu gibi kontroller yanı sıra, kullanıcı dostu arayüzü sayesinde tamamlanan elektronik devre, tasarımdan PCB araçlarına kolaylıkla aktarılır.

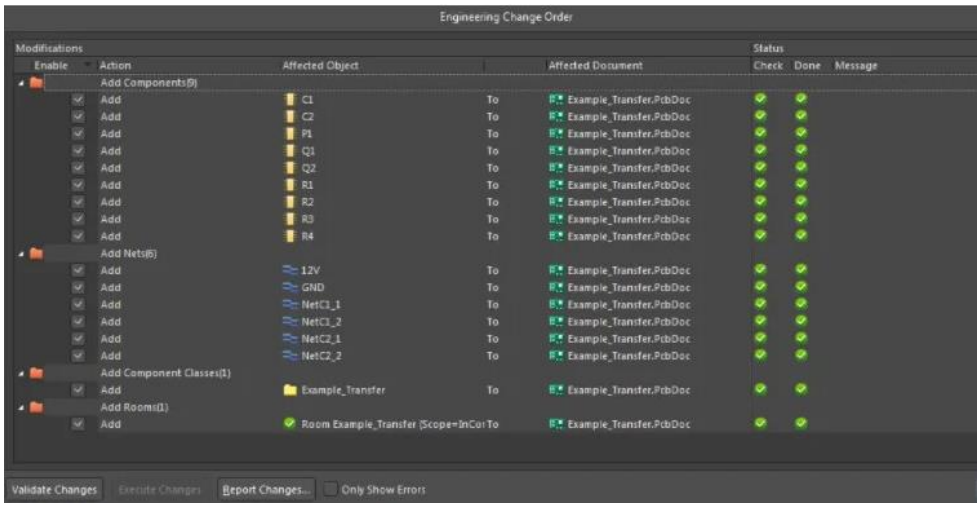
İlk olarak şematik eklemek için ekranın solunda görünen Project isimli klasöre sol tıklanır Add New to Project seçeneğinden Schematic Library seçilir. Bu aşamadan sonra açtığımız sayfaya önceden hazırladığımız kütüphaneleri eklememiz gerekir. Bunun iççinde yine sol üst ekranda Project seçeneğinden Compile Integrated Library seçilir ve önceden hazırladığımız komponentler şematik kısmına aktarılmış olur. Son olarak bu malzemeleri ekranın sağ tarafında bulunan Components sekmesinden çalışma alanımıza sürükleyerek şematiğimizi oluştururuz.



Şekil 6. Şematik Arayüzü

3.3 PCB Tasarımı

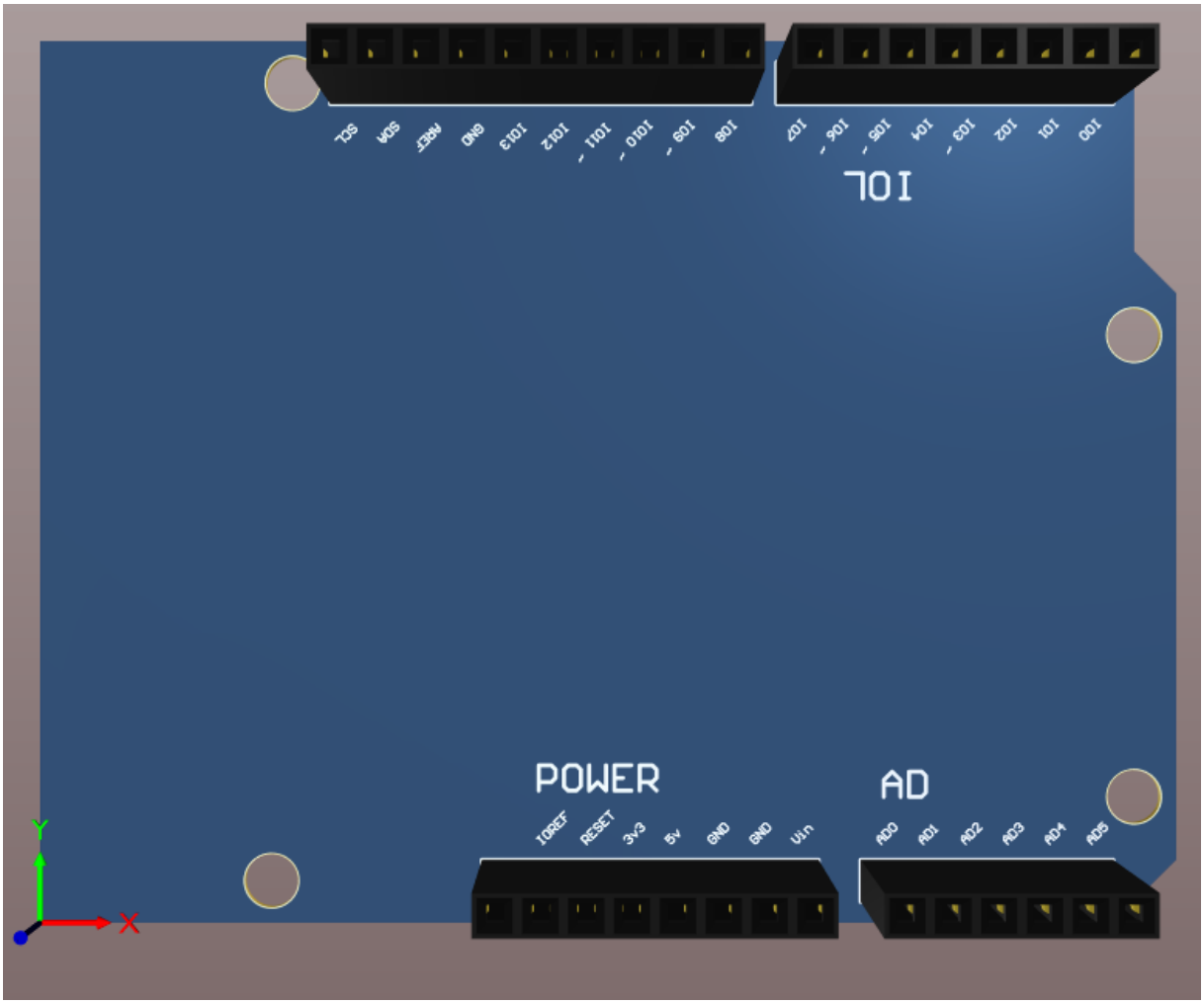
Tasarıma başlamadan önce ilk yapmamız gereken şey her zaman olduğu gibi **Menu» Create Project** kısmından yeni bir proje oluşturmak veya hali hazırda olan projemize **Projects** geçiş yapmak olacaktır. Bu noktada aslında yapmamız gereken ilk şey, tasarlamış olduğumuz şematiği, PCB sayfasına import etmek olacaktır. Bunun için Altium Designer bize çok büyük kolaylık sağlayacaktır. Sadece küçük bir ara yüzle hem import işlemini gerçekleştireceğiz hem de hatalı, eksik bağlantı varsa bize onları önceden belirtecektir.



Şekil 7.Hata Ekranı

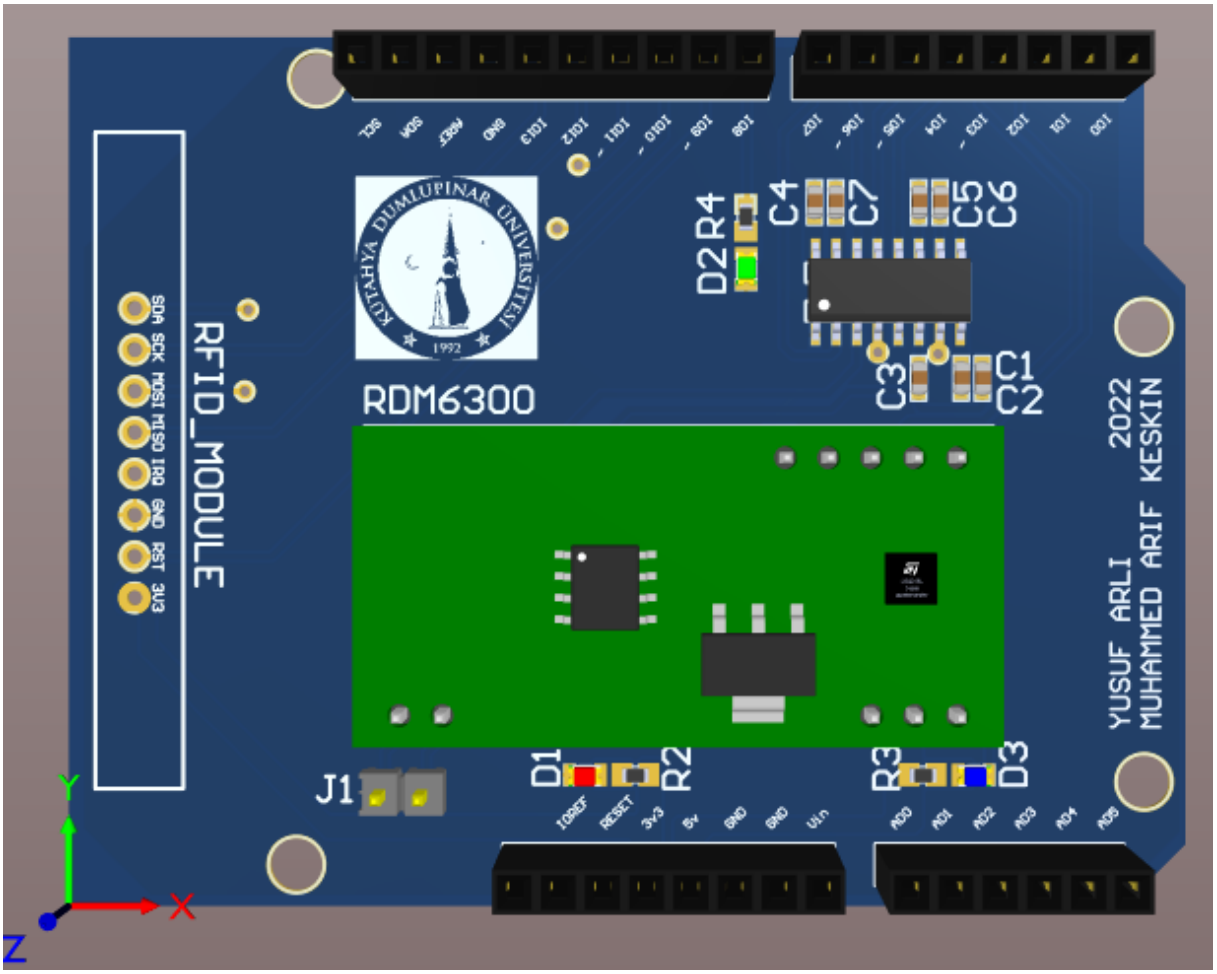
Burada yapacağımız tek şey **Execute Changes** butonuna basmak olacaktır. Sonrasında gerekli olan bütün işlemleri, Altium kendi başına bizim için halledecektir. Bu işlem tamamlandıktan sonra otomatik olarak PCB Ekranı açılacaktır. İşte orada artık yavaştan PCB Tasarımına başlayabiliriz. Bu noktada karşımıza yerleştirdiğimiz komponentlerin footprintleri ve PCB Kartımızın default hali çıkacaktır.

Arduino Shield tasarımı yaptığımız için kartımızın boyutları da Arduino kartının ölçülerinde olması gerekiyor. Bunun için kartımızın boyutunu bu ölçülere göre tasarladık. Bu aşamadan sonra hazırlamış olduğumuz komponentleri kartımıza yerleştirmek ve yolları çizmek kalıyor.



Şekil 8. Arduino Kartın Çizimi

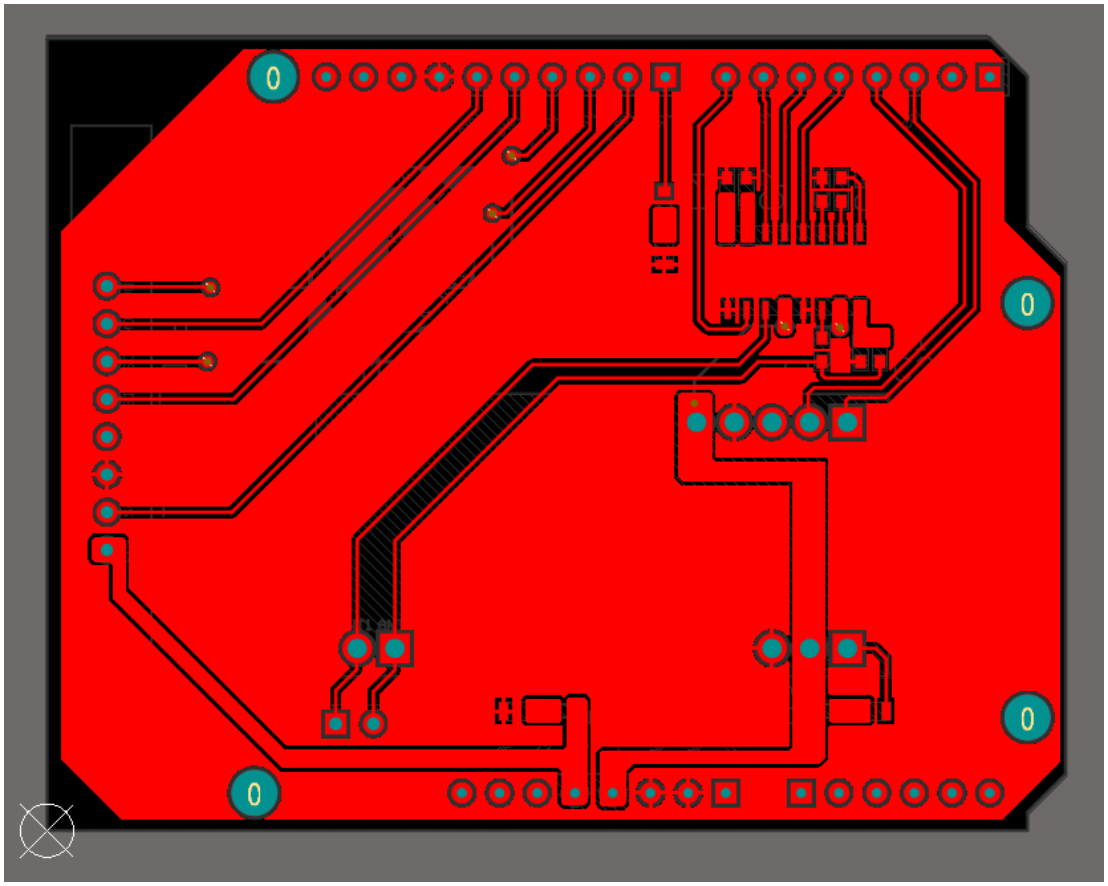
Hazırlamış olduğumuz shield tasarımında kullanacağımız komponentlerin daha işlevsel kullanımı ve bağlantılarının daha kolay yapılması için komponentler buna göre yerleştirilmiştir. Kartın üzerinde 2 modül ve 1 entegre bulunmasından dolayı her biri için kendi boyutlarına ve yanlarında kullandığımız malzemelere göre bir yerleşim düzeni oluşturulmuştur. RC522 boyutlarından dolayı kartın kenarında, RDM6300 ise genişliğinden dolayı kartın ortasına yerleştirdik. Pin bağlantılarından ve yanında kullanmak zorunda olduğumuz komponentlerinden dolayı da EM4095'i sağ üst köşeye yerleştirdik. EM4095 entegresi ve RDM6300 modülünde kullanılan antenleri kart üzerinde ikisinin de kullanabileceği ortak bir pinde birleştirdik. Bununla birlikte kart üzerindeki alandan tasarruf etmiş olduk. Kartın üzerinde görsellik açısından güzel gözükmesi için okulumuzun logosunu sol üstte bulunan Place sekmesinden Graphics seçeneğine tıklayarak resmi Top Overlay katmanına yerleştirdik. Son olarak Text ile tüm komponentlerin isimlerini, bağlantı yapılacak kısımları ve isimlerimizi belirttik.



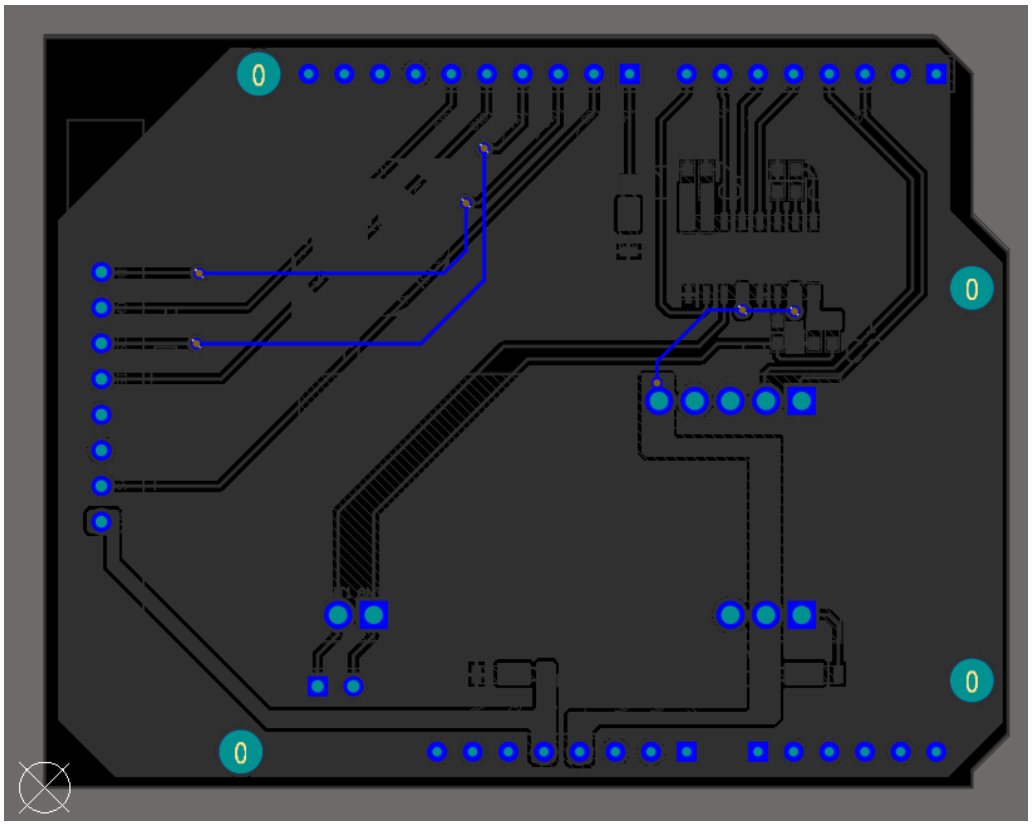
Şekil 9. Komponentlerin Yerleşimleri

3.4 Bağlantı Yollarının Çizilmesi

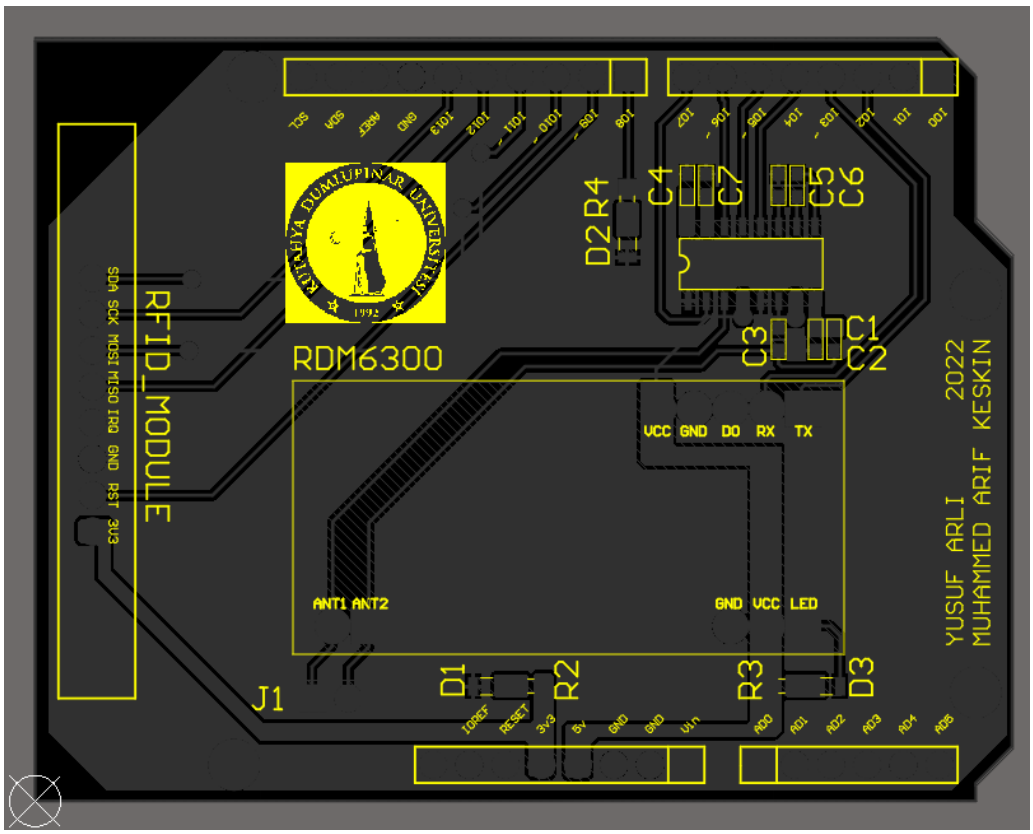
Elemanlar uygun konumlara getirildikten sonra aralarında gerekli yollar çizilir. Yol bir pad‟den diğerine çizilir. Yol çizerken, yolun görünen yüzde mi, arka yüzde mi çizileceğine karar verilmelidir. “Top Layer” PCB‟nin görünen yüzünde yol çekmek için tercih edilir. Gördüğümüz yüzün arka kısmında yol çizilmek istenir ise de “Bottom Layer” kullanılır. Yol çizilir iken kalınlığı önemlidir, çizime başlamadan önce yolların alabileceği en büyük ve en küçük genişlik tanımlanmalıdır. Bu tanımlamalar “Design-Rules”tan yapılabilir. Yol çizilirken Tab tuşuna basılır ise, yol genişliği; via eklenirken de Tab‟a tıklanması durumunda ise via özellikleri değiştirilebilir. Devre elemanını üzerine tıkladıktan sonra basılı tutup klavyede “Space” tuşuna basılır ise eleman 90 derece döner. Yol veya eleman yerleştirir iken, elemanların ve yolların birbirine göre konumlarını doğru belirlemek adına programın cetvelini kullanabiliriz. Cetvel “ctrl+M” kısa yolu ile çağrılabilir. Ölçüm almak istediğimiz noktaların önce birinin üzerine tıklar sonra diğer noktanın üzerine tıklarız, bu anda ekranda iki nokta arası mesafe görülebilir. Çizilen yollar 13 Mil kalınlığında tercih edilmiştir. Bağlantıların daha rahat yapılabilmesi için bazı yolların Bottom Layer’dan geçirmeye karar verdik. Bunun içinde katmanlar arasında geçiş yapmak için Through Via kullandık. Via Diameter 0.9mm, Hole Size ise 0.6mm tercih edilmiştir. Tüm elemanların bağlantılarını yaptıktan sonra GND pinlerinin hepsini birleştirmek için Place Polygon Plane seçeneğinden Polygon Pour seçilerek Polygon çizimi yapılır.



Şekil 10. Top Layer



Şekil 11. Bottom Layer



Şekil 12. Top Overlay

3.5 Kural Tanımlamaları

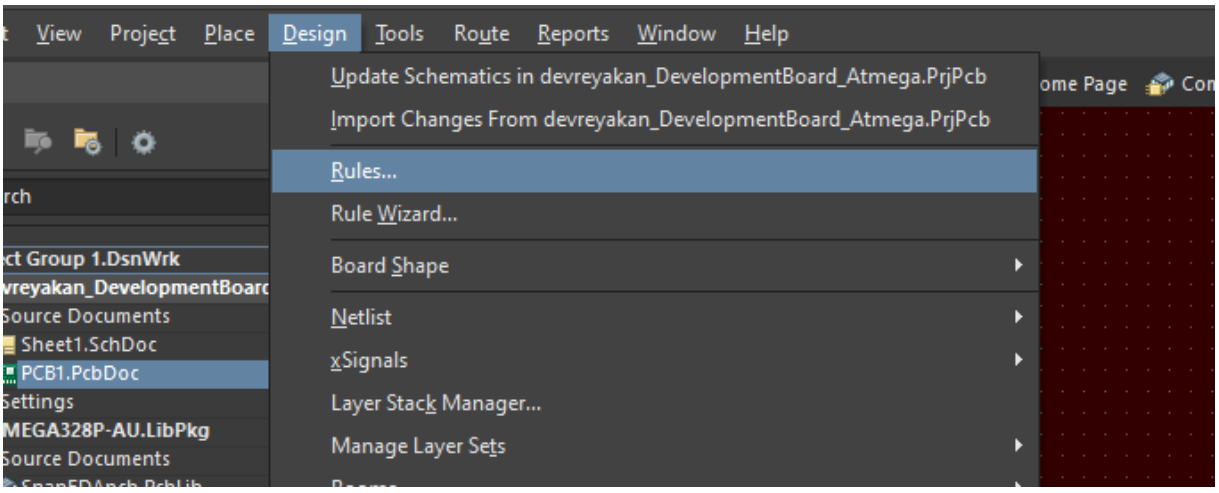
Altium Designer Programının PCB Düzenleyicisi, bir tasarımın gereksinimlerini tanımlamak için Tasarım Kuralları (Rules) kavramını kullanır. Bu kurallar toplu olarak PCB Düzenleyicisinin izlemesi için bir 'komut seti' oluşturur. Yol çizerken kalınlıklar, açıklıklar, düzlem bağlantı stilleri, stiller üzerinden yönlendirme vb. Gibi tasarımın her yönünü kapsar ve kuralların çoğu çevrimiçi Tasarım Kuralı Denetleyicisi (DRC) tarafından gerçek zamanlı olarak izlenebilmektedir. Tasarım kuralları, belirli nesneleri hedefler ve hiyerarşik bir şekilde uygulanır. Yani kuralların işleyiş biçimi bile belli bir öncelik sırasına göre değişmektedir. PCB Editörü kurallara dayalı olduğundan dolayı, tasarımın başlangıcında kuralları ayarlamak için zaman ayırmak, güvenli ve etkili bir şekilde tasarlama işine devam etmenizi sağlayacaktır.

Herhangi bir kural, bir nesnenin özniteliği olarak eklenmez. Bunun yerine genel kural kümesine eklenir ve ardından o nesneye uygulanacak şekilde kapsamı belirlenir. Bu durum, kuralların birden çok nesneye uygulanmasına, değiştirilmesine veya farklı nesnelere uygulanmasına izin verir; aksi takdirde, kural niteliklerini tek tek nesne düzeyinde değiştirmek zorunda kalırsanız, bunu yapması daha zahmetli olacaktır.

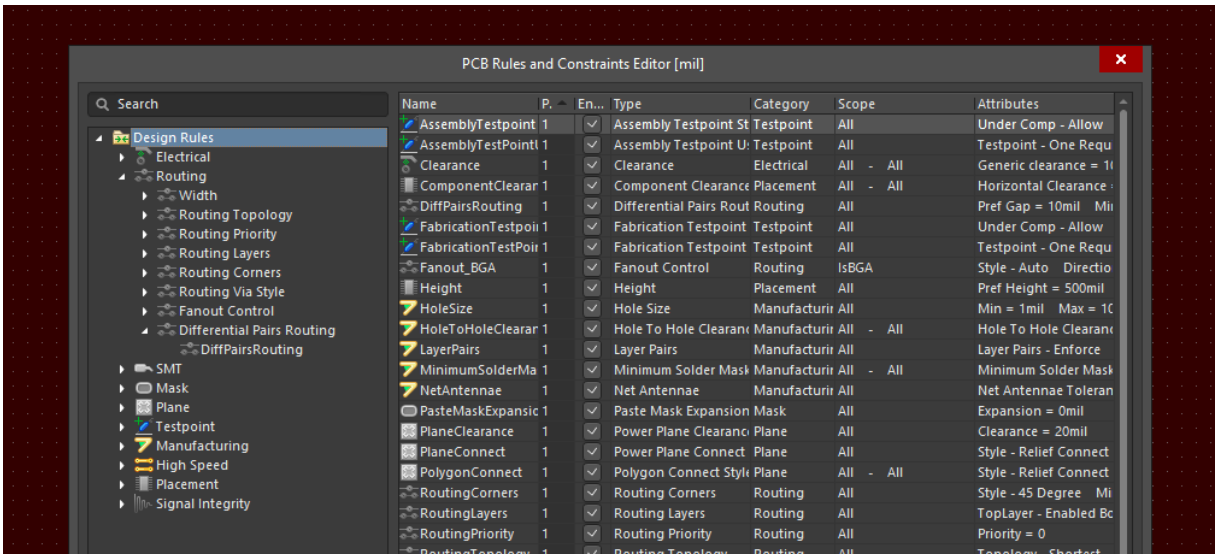
Aynı türde, birden fazla kural tanımlanabilir ve farklı nesne kümelerine hedeflenebilir. Böylece board kısıtlamalarının tanımı üzerinde tam kontrol sağlanır. Örneğin yolları farklı genişliklerde, farklı katmanlarda yönlendirmek için farklı genişlik kuralları tanımlanabilir.

Herhangi bir tasarım nesnesi, genel ve daha özel durumlara hitap eden aynı türden birden çok kural tarafından hedeflenebilir. Herhangi bir kural çekişmesini çözmek için kural önceliği kullanılır. Sistem basitçe kuralları en yüksekten en düşüğe doğru gözden geçirir ve kapsam ifadeleri kontrol edilen nesneler ile eşleşen ilkini seçer.

İlk olarak ara yüzümüzün Design kısmından Rules sekmesine basıyoruz. Sonrasında zaten karşımıza ilk ana ara yüzümüz çıkmaktadır.



Şekil 13. Rules Sekmesi

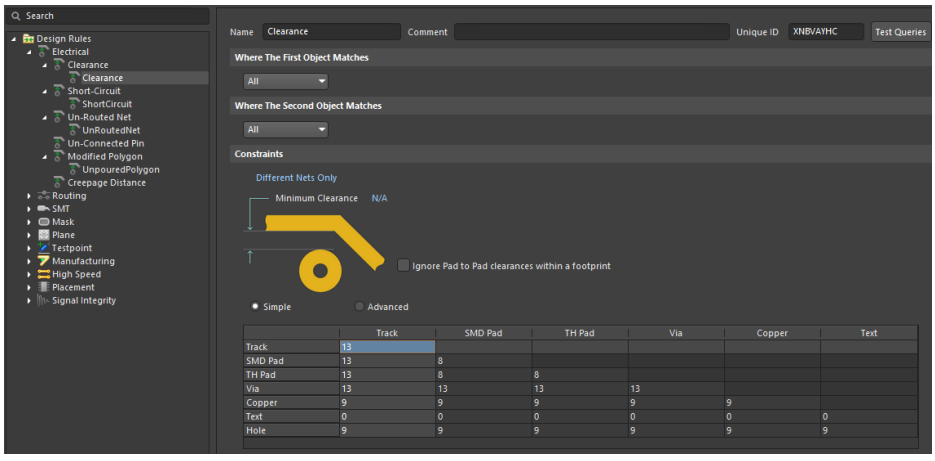


Şekil 14. Rules Ekranı

Soldaki dallanarak detaylandırma sistemi, farklı kural kategorilerini listeler. Mevcut bireysel kural türlerini görüntülemek için istediğiniz bir kategoriyi genişletin. Siz genişlettikçe halihazırda belirlenmiş olan değerleri göreceksiniz. Sağ tarafta ise, sol tarafta seçmiş olduğumuz ana başlığın kuralları ile ilgili bütün önemli detaylar sunulmaktadır.

Yeni kural oluşturmak istediğiniz zaman yapmanız gereken tek şey New Rule sekmesine basmak olacaktır. Bu sayede istediğiniz başlığın altına kural oluşturabilirsiniz.

Yeni kural oluşturduktan sonra üzerine tıklayın istediğiniz gibi düzenleyebilirsiniz. Örnek olarak eğer Routing 'den Width kısmını seçerseniz. Ardından top layer kısmında max width'i 1mm olarak belirlerseniz, PCB tasarımına geçtiğinizde çizdiğiniz hiçbir yol kalınlığı 1mm'yi geçmeyecektir. Veya min. width'i 0.250mm derseniz, aynı şekilde kalınlık asla 0.250mm'den aşağı olmayacaktır. Bunun aslında bir sürü sebebi olabilir ama bunlardan en çok karşımıza çıkanlarından biri üreticimizin belirlemiş olduğu sınırlardır. Örnek veriyorum: X bir PCB üretim firmasında çizilebilecek min. yol kalınlığı 0.15mm ise, orada 0.14mm kalınlığında bir PCB üretimi yapmanız mümkün değildir. İşte bu durumlarda o sıkıntılara düşmemeniz için bu Rule kısmı size yardımcı olmaktadır.



Şekil 15. Rules

4. ARDUINO İLE YAZILIM TESTLERİ

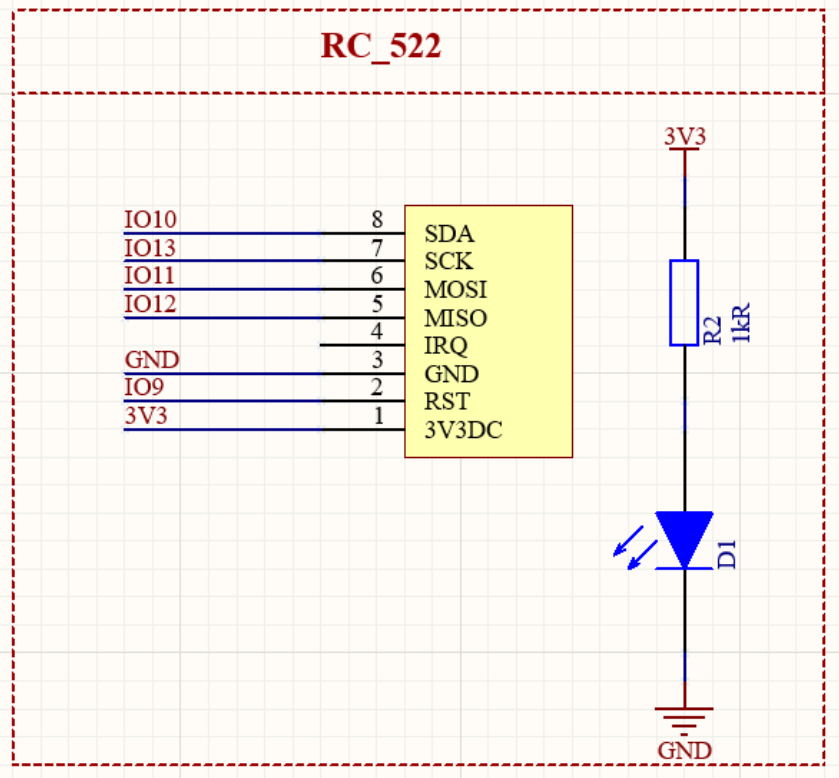
4.1 RC522

RFID RC522 modülü ile günlük hayat kullandığımız birçok ürünün kontrolünü sadece bir 13,56 MHz NFC Kart veya 13,56 MHz NFC anahtarlık ile kontrol edebilirsiniz. 13,56 MHz HF (yüksek frekans), yakın algılama olanağı sağlar. RFID modülleri genel olarak kapı kilitlerinde kullanılsa da kişinin hayal gücüne bağlı olarak farklı uygulamalar ve projeler geliştirilebilir.

4.1.2 RC522 Bağlantıları

RC522 bağlantılarını aşağıdaki şematikte ayrıntılı şekilde görebilirsiniz. Ayrıca yaptığımız şematikte 3.3V pinine karta güç geldiğini anlamamız için bir led bağlantısı gerçekleştirdik. VCC modüle güç sağlar. Bu 2.5 ila 3.3 volt arasında olabilir. RST Sıfırlama ve kapatma için bir giriştir. RST, Arduino'daki herhangi bir dijital pime bağlanabilir. Bu pin enerjisi azaldığında, sistem kapanma etkinleştirilir. Bu, osilatör dahil tüm dahili akımları ve pinleri kapatır ve modül sıfırlanır. GND toprak pinidir ve Arduino'daki GND pinine bağlanması gerekir. IRQ RFID etiketi modül menziline geldiğinde mikrodenetleyiciyi uyarabilen bir kesme pinidir ama bu kartta bu pini kullanmayı tercih etmedik.

NOT: Tasarladığımız kartta RST ve SDA pinlerini 9. ve 10. pinlere tanımladık. Bu yüzden kendi projenizde bu shield kartı kullandığınızda kod ekranında bu pinlere göre tanımlama yapmanız gerekmektedir.



Şekil 16. RC522 Bağlantıları

Şimdi SPI iletişimi için kullanılan pinlere bakıyoruz. RC522 modülü çok fazla veri aktarımı gerektirdiğinden, mikro denetleyici SPI pinlerine bağlandığında en iyi performansı verecektir. Her Arduino Kartında buna göre bağlanması gereken farklı SPI pinleri olduğunu unutmayın. UNO / Nano V3.0 gibi Arduino kartları için bu pinler dijital 13 numaralı pin SCK, 12 numaralı pin MISO, 11 numaralı pin MOSI ve 10 numaralı pin SS.

4.1.3 SPI Haberleşme

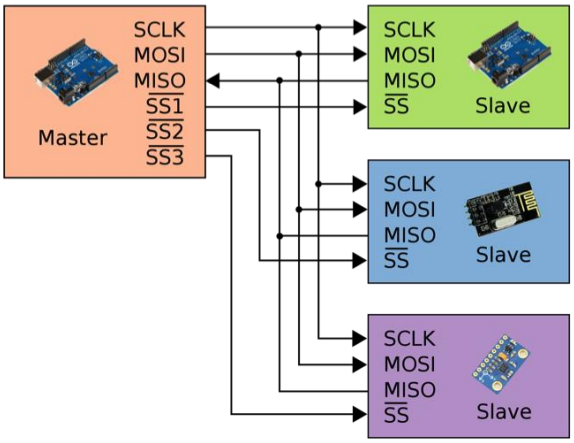
SPI (Serial Peripheral Interface), Arduino'nun desteklediği senkron seri haberleşme türlerinden biridir. Özellik ve kullanım olarak I2C'ye benzer. Bir Arduino'nun diğer Arduino veya sensörlerle kısa mesafede haberleşmesini sağlar. SPI protokolünde de I2C'de olduğu gibi bir adet Master cihaz bulunur. Bu cihaz hatta bağlı çevresel cihazları kontrol eder.

Master ve çevresel cihazlara bağlanan MISO (Master In Slave Out), MOSI (Master Out Slave In) ve SCK (Serial Clock) olmak üzere üç adet SPI hattı bulunur.

- **MISO**: Çevresel cihazlardan (slave) yollanan verilerin master cihaza aktarıldığı hattır.
- **MOSI**: Master cihazdan yollanan verilerin çevresel cihazlara aktarıldığı hattır.
- **SCK**: SPI haberleşmesinde senkronu sağlayan saat sinyalinin bulunduğu hattır. Saat sinyali master cihaz tarafından üretilir.

MISO ve MOSI hatlarından da anlaşıldığı gibi SPI protokolünde I2C'den farklı olarak veri hatları tek yönlüdür. Ayrıca çevresel cihazların (slave) adreslerinin olmasına gerek yoktur. Her çevresel cihazın seçim ayağı bulunur. Bu ayağa, SS (Slave Select) denir. Bu hattın sayısı kullanılan çevresel cihazların sayısı kadardır. Her cihaz için master cihazından ayrı SS hattı çıkar. SS hattı LOW (0 volt) düzeyinde olan çevresel cihaz, master cihaz ile iletişime başlar.

Örnek bir SPI haberleşme hattı aşağıdaki resimde gösterilmiştir. Resimde de görüleceği üzere, Master cihazdan çevresel cihaz sayısı kadar SS çıkışı bulunur. Master cihaz iletişime geçmek istediği çevresel cihazın SS pinini LOW (0 Volt) düzeyine çeker.



Şekil 17. SPI Haberleşme

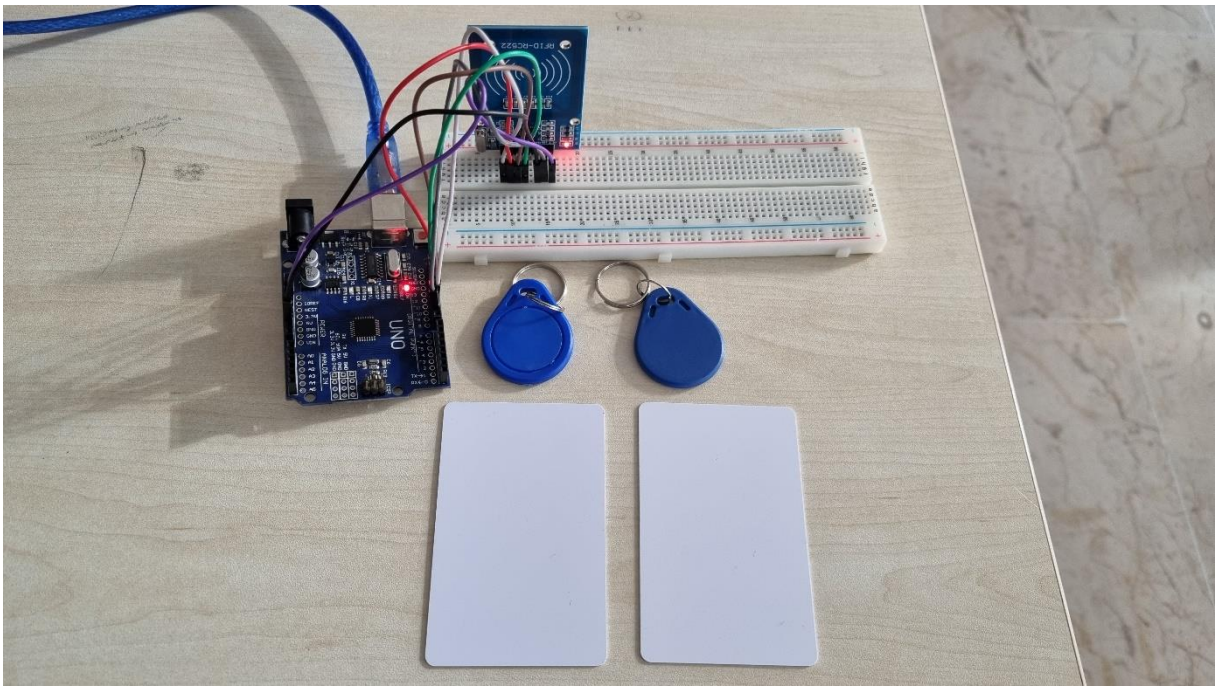
SPI Fonksiyonları

SPI protokolünü öğrendiğimize göre şimdi haberleşmenin Arduino kısmına bakalım. Arduino'da SPI fonksiyonlarını kullanabilmemiz için öncelikle "SPI.h" kütüphanesini projemize eklememiz gerekir. Kütüphane projeye eklendiğinde aşağıdaki fonksiyonlar kullanılabilir.

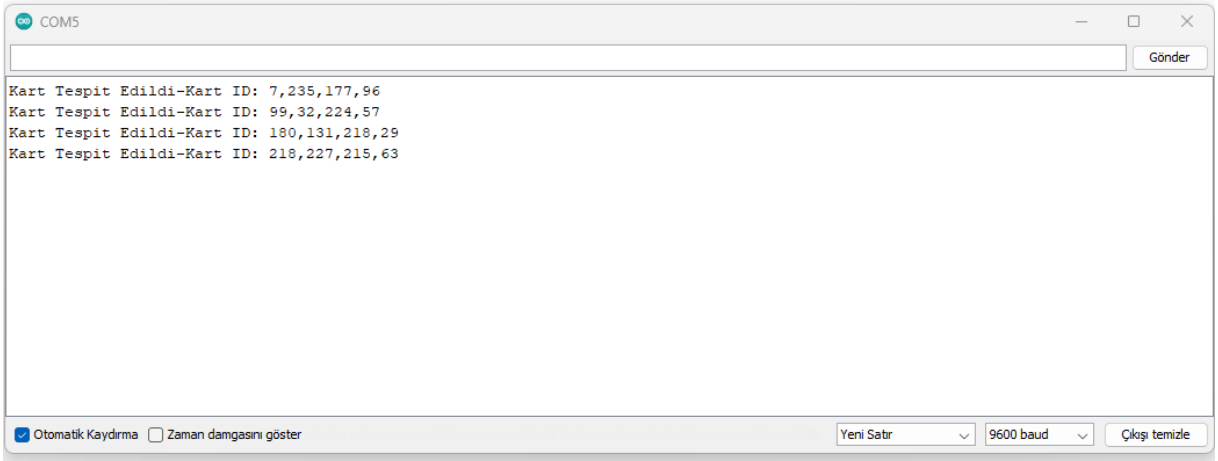
- **SPI.begin()**: SPI haberleşmesini başlatır ve SPI pinlerini başlangıç konumlarına alır.

4.1.4 RC522 Yazılım Testi

```
#include <SPI.h>
#include <MFRC522.h>
int SdaPin=10;
int RstPin=9;
MFRC522 rfid(SdaPin, RstPin); //RFID nesnesi oluşturup bağlantı pinlerini tanımladık.
void setup() {
Serial.begin(9600);
SPI.begin(); //SPI haberleşmesini başlattık.
rfid.PCD_Init(); //RFID Kart okuyucuyu başlattık.
}
void loop() {
if (rfid.PICC_IsNewCardPresent()) {
if (rfid.PICC_ReadCardSerial()) {
Serial.print("Kart Tespit Edildi-Kart ID: ");
Serial.print(rfid.uid.uidByte[0]);
Serial.print(",");
Serial.print(rfid.uid.uidByte[1]);
Serial.print(",");
Serial.print(rfid.uid.uidByte[2]);
Serial.print(",");
Serial.println(rfid.uid.uidByte[3]);
}
rfid.PICC_HaltA();
}
}
```



Şekil 18. Arduino Bağlantısı



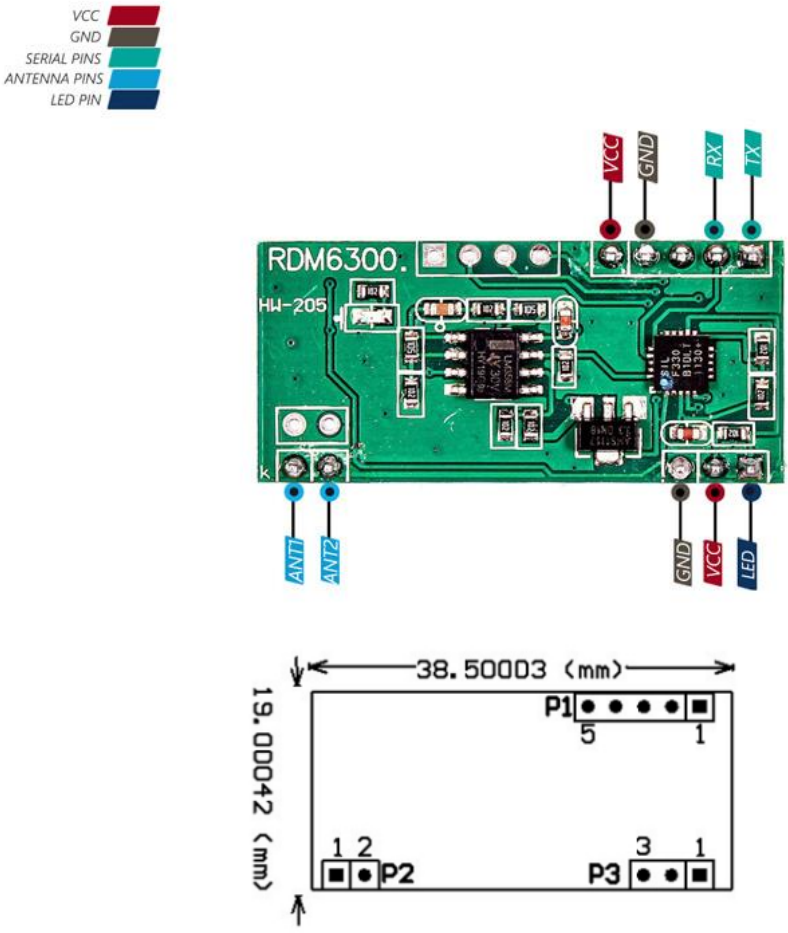
Şekil 19. Seri Port Ekranı

Yukarıdaki resimlerde görüldüğü üzere RC522 Shield kart tasarımında devre çalıştırılmadan önce Breadboard üzerinde denemeleri yapıldı. Bunun sonucunda Şekil19' da görüldüğü üzeri Seri Port ekranında okuttuğumuz kartların ID numaralarını görebildik.

4.2 RDM6300

4.2.1 RDM6300 Bağlantıları

Yapmış olduğumuz projede kullanılan RDM6300 modülü UART ile haberleştiğinden bağlantılarını da buna göre yapmamız gerekiyordu. Bu pinleri shield kartımızda bulunan herhangi 2 dijital pine bağladık. Bu pinler Arduino Shield üzerindeki IO2(TX) ve IO3(RX) pinleridir. VCC modüle güç sağladığı için 5V değerindeki pine bağlanmalıdır. GND toprak pinidir. Arduino üzerindeki GND ile bağlanmalıdır. NFC özelliği taşıyan kartımızı okuttuğumuzda bize geri bildirim vermesi için kart üzerinde bulunan LED pinine led bağlantısı gerçekleştirdik ve 5V ile bağladık. Son olarak shield üzerinde de yerleşimini yaptığımız 2 adet anten bağlantısı bulunmaktadır.



P1	P2	P3
1: TX	1: ANT1	1: LED
2: RX	2: ANT2	2: +5VDC
3: NC		3: GND
4: GND		
5: +5VDC		

Şekil 20.RDM6300 Pinout

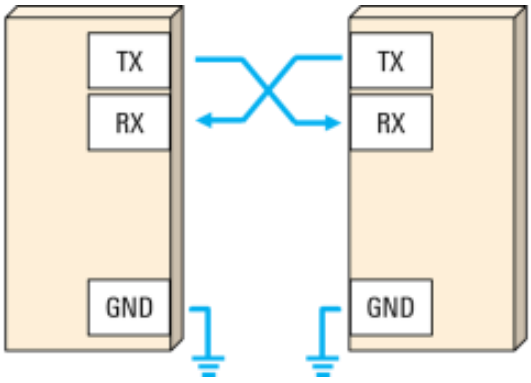
4.2.2 UART Haberleşme

UART, evrensel asenkron alıcı / verici anlamına gelir ve iki cihaz arasında seri veri alışverişi için bir protokol veya kurallar kümesi tanımlar. UART çok basittir ve her iki yönde de iletmek ve almak için verici ve alıcı arasında yalnızca iki kablo kullanır. Her iki uçta da toprak bağlantısı vardır. UART'ta iletişim tek yönlü (veriler yalnızca bir yönde gönderilir), yarı çift yönlü (her iki taraf konuşur ancak bir seferde yalnızca bir tane) veya tam çift yönlü (her iki taraf aynı anda iletebilir) olabilir. UART'taki veriler çerçeveler şeklinde iletilir. Bu çerçevelerin formatı ve içeriği kısaca açıklanmış ve açıklanmıştır.

UART'ın en büyük avantajlarından biri asenkron olmasıdır – verici ve alıcı ortak bir saat sinyalini paylaşmaz. Bu, protokolü büyük ölçüde basitleştirse de verici ve alıcıya belirli gereksinimler getirir. Bir saati paylaşmadıklarından, aynı bit zamanlamasına sahip olmak için her iki uç da aynı, önceden ayarlanmış hızda iletmelidir.

UART, en eski seri protokollerden biriydi. Bir zamanlar her yerde bulunan seri portlar neredeyse her zaman UART tabanlıdır ve RS-232 ara yüzleri, harici modemler vb. kullanan cihazlar, UART'ın kullanıldığı yerlerin yaygın örnekleridir.

Son yıllarda, UART'ın popülaritesi azaldı: SPI ve I2C gibi protokoller, çipler ve bileşenler arasında UART'ın yerini alıyor. Bir seri bağlantı noktası üzerinden iletişim kurmak yerine, çoğu modern bilgisayar ve çevre birimi artık Ethernet ve USB gibi teknolojileri kullanıyor. Bununla birlikte, UART, çok basit, düşük maliyetli ve uygulanması kolay olduğu için hala daha düşük hızlı ve daha düşük verimli uygulamalar için kullanılmaktadır.



Şekil 21.Uart Haberleşme

4.2.3 RDM6300 Yazılım Testi

Aşağı kullanılabilecek 2 tane Arduino kodu paylaşılmıştır. İlk kodun çalışması için kartınızın ID numarasını bilmeniz gerekiyor.

```
#include <SoftwareSerial.h>

SoftwareSerial rfid(3,2); // RX, TX

void setup(){

  Serial.begin(9600);

  rfid.begin(9600);

}

void loop()

{

  if (rfid.available()){

    if(rfid.find("18013121829")){

      Serial.println("OK");

    } } }
```

Github Nedir?

GitHub; kalabalık bir ekibin sürüm kontrol sisteminde bir yazılım geliştirirken kullanabilecekleri **internet tabanlı bir depolama servisidir.** GitHub, Git yönetim sistemi ile geliştirilen yazılımlarda kullanılır. Açık kaynak kodlu bir servis olması nedeniyle pek çok yazılım ekibi tarafından kullanılmakta ve geliştirilmektedir.

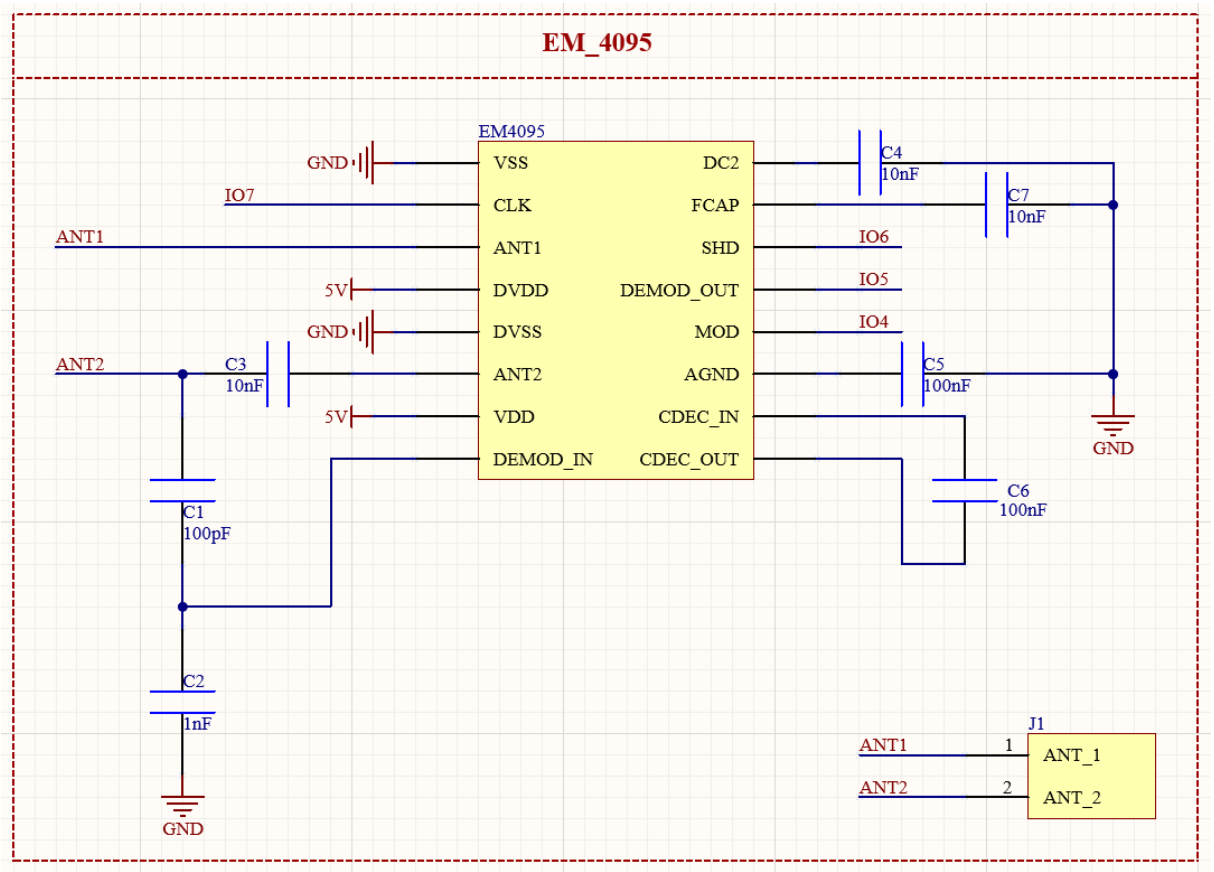
Bizde bu yüzden çok uzun kod satırlarımızı bu araştırmamızın içine koymak yerine Github linki bırakmayı tercih ettik. RDM6300 ve EM4095'i Github internet adresinde bulabilirsiniz. (Yapmanız gereken GitHub simgesine tıklamak)

- GitHub

4.3 EM4095

4.3.1 EM4095 Bağlantıları

EM4095 entegresinin bağlantılarını yaparken datasheet de bize tavsiye edildiği gibi kapasitör eklenmesi gereken pinlere kapasitör bağlantılarını yaptık. Bunları yaparken bize hangi pine kaç farad kapasitör bağlamamızı datasheet kısmında denklemler ile vermiş. Bu denklemlerden yararlanarak ilk önce kapasitör değerlerini bulduk sonra da gerekli olan yerlere hesaplamamıza göre bağlantılarını gerçekleştirdik. EM4095 entegresinin SHD pinini Arduino Shield IO6 pinine, DEMOD_OUT pinini IO5 pinine, MOD pinini IO4 pinine ve CLK pinini IO7 pinine bağlantısını gerçekleştirdim. Ayrıntılı olarak aşağıda **Şekil22** de verilmiştir.

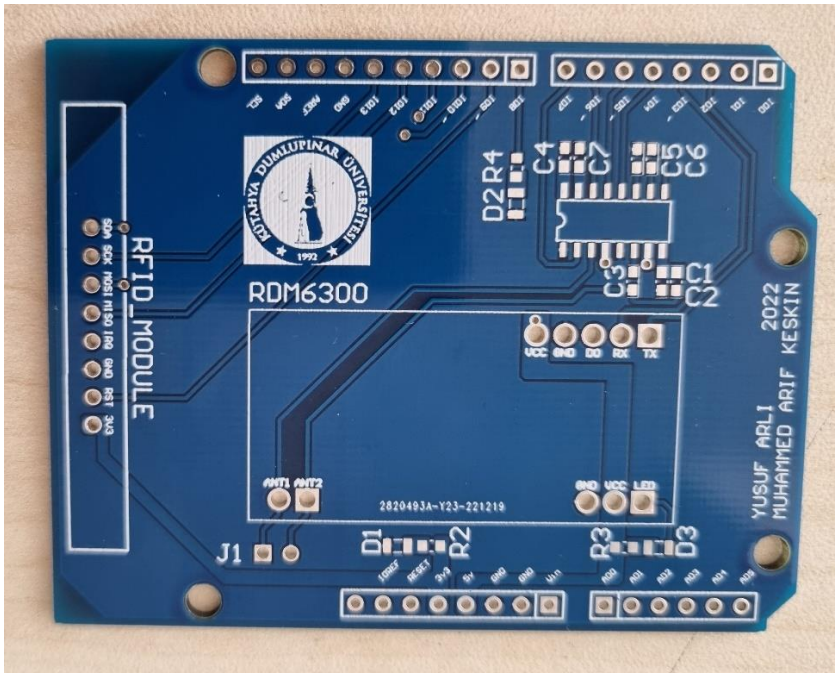


Şekil 22. EM4095 Bağlantıları

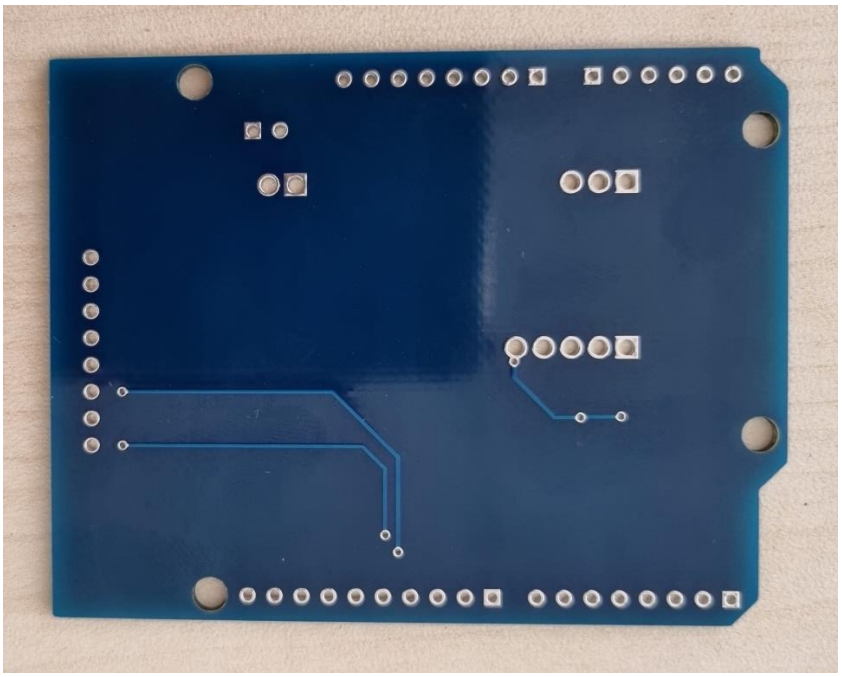
4.3.2 EM4095 Yazılım Testi

➢ GitHub

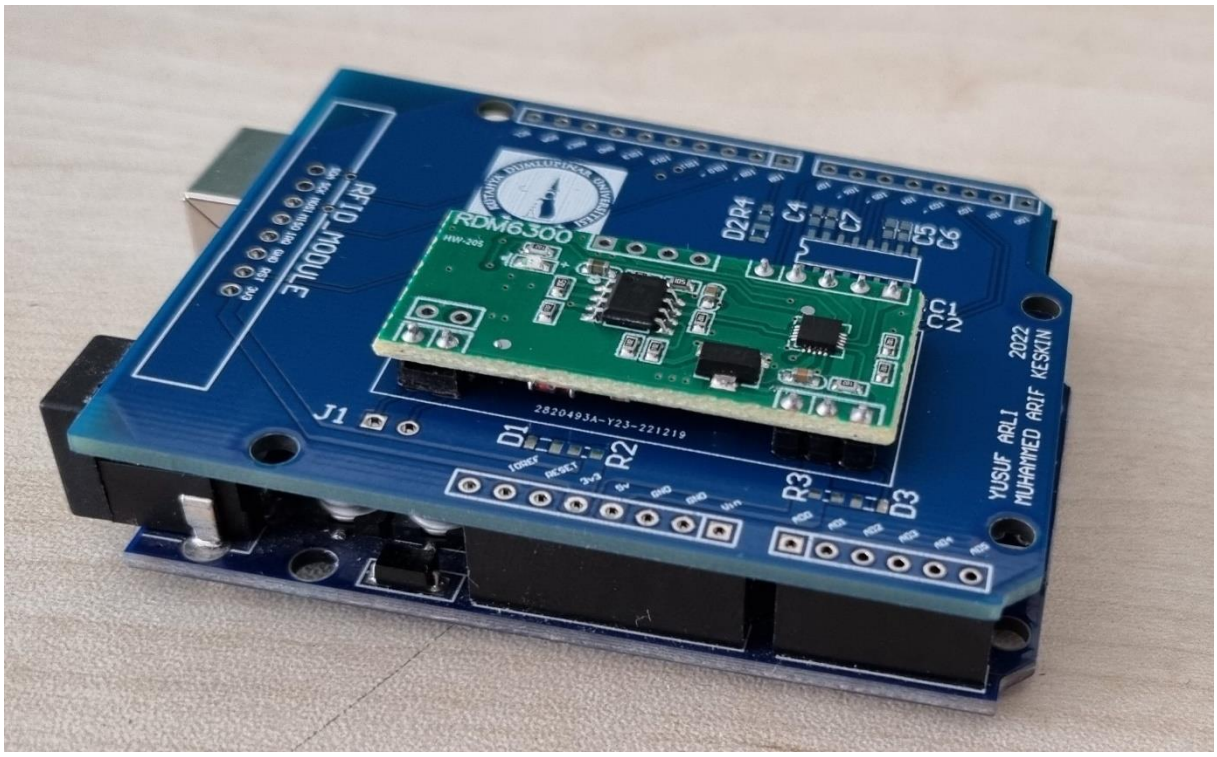
SONUÇLAR



Şekil 23



Şekil 24



Şekil 25

Şekil 26

KAYNAKLAR

- Rfid-turkiye
- infoma.biz
- Robotistan
- Emmicroelectronic
- AppNote 405 datasheet
- EM4095 datasheet
- Devreyakan
- gelecegiyazanlar.turkcell