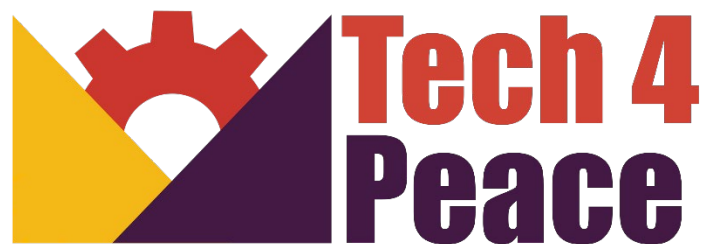


# T4P – Rookie

**Tasarım 101**

**Prototipleme 101**

**Kodlama 101**



# Tasarım 101

Tasarım her robotun başlangıç noktasıdır. CAD programı, beyaz tahta ya da bir mendilin arkasına yapılsa bile süreç aynı hep olacaktır. Mekanik tasarım çok büyük bir konudur ve her yanını tek bir yerde ele almak imkansızdır. Onun yerine, bu kaynağın amacı size bir FIRST Robotik Yarışması durumunda tasarım öğrenmek için bir başlangıç yeri vermektir.

## Giriş

Tasarım öğrenmeye başlamadan önce doğru yerde olduğunuzdan emin olun: Bu kaynak spesifik olarak FIRST Robotik Yarışması tasarımı için, yazılım üzerinden CAD için kaynaklar sağlayan farklı bir kaynak bulunmaktadır. Başarılı bir robot için stratejiler öğrenmeyi umut ediyorsanız, doğru yerdesiniz. Tasarım yaparken size yardım edecek yazılım öğrenmek istiyorsanız, CAD kaynaklarına bakın.

Robot tasarımı, öğretmesi zor olan bir konudur. Kusursuz bir mekanizma bulmanın bir bilimi yoktur - bolca deneyim, iyi bir tahmin ve deneme yanılma ister. Bu el kitabı size o süreçte yardımcı olabilecek kaynaklar sağlamak için tasarlandı. Ama size tam olarak nasıl kazanan bir robot yapmayı öğretemez. O, sizin ve takımınızın yaratıcılığına kalmış.

FIRST Robotik Yarışmasında tasarıma diğer iyi genel bakışlar için Team 1114'ün sayfasına bakın: Takım 1114 - Robot Tasarımı ve Takım 3874'ün ayrıntılı sunumu: Takım 3874 - Tasarım Sunumu Seviye 0: Birinci Plan

## Seviye 0: İlk Plan

FIRST Robotik Yarışması'ndaki nerdeyse her durum gibi, tasarımda en zorlayan faktör zamandır. Bu nedenle, robot tasarımının ilk adımları çoğu zaman en önemliler olmaktadır. Sonsuz vaktimiz olsa, olabilecek her robotun listesini yapıp, hepsini inşa edip, hangisinin en iyi olduğunu test edebiliriz. Bunun yerine, bir tanesini (biraz yinelemeyle) seçmemiz gerekiyor. Takımınızın becerilerinin dürüst bir değerlendirmesini yapmanız gerekecektir. Sahada çok fazla şeye meydan okumaya çalışırsanız, robotunuzun hepsinde düşük performans göstermesi kaçınılmazdır. Çok az şey yapmaya kalkarsanız, robotunuz maksimum potansiyeline ulaşmamış olur. Neyin en etkili olacağına karar vermek için bu değerlendirmeyi yapmak kolay bir şey değildir. Aşağıdaki kaynaklar ilk adımlarınızda size yardımcı olacaktır:

### Oyun Stratejisi

Bir tasarım projesi hakkında bilmeniz gereken ilk ve en önemli şey hedeflerdir. Robotunuz ne yapmasını istiyorsunuz. Stratejik tasarım takımın bu soruyu cevaplamasına yardımcı olacak bir süreçtir.

Takım 1114 oyun analizi hakkında bu sunumu veriyor: Takım 1114 - Efektif FIRST Stratejileri

## Stratejik Tasarım

Robotunuzun ne yapacağı kararına varılınca, bir sonraki soru bunu nasıl yapacağı olur. Bazı durumlarda cevap bellidir ve çok az/hiç tasarım tartışmasını ya da prototiplemeye gerek kalmaz. Diğer durumlarda olabilecek birden fazla çözüm vardır ve hangisinin çalışma ihtimalinin en fazla olduğuna karar vermeniz gerekir.

Mantıklı ilk adım, yapılması gereken şeyleri mekanizmalara gruplandırmaktır. Bazen tek bir mekanizma içeri alma ve atış gibi bir sürü görevi üstlenebilir. Ama çoğu durumda mekanizmaları birleştirmeye çalışmanın eklediği zorluğa değmeyebilir.

Ardından, mekanizmalar için beyin fırtınası fikirleri gelir. Çoğu zaman, en iyi başlangıç noktası, sizden önce benzer bir sorunu çözen önceki kişilerden ilham almaktır:

- 3 Günde Robot (RI3D), aynı oyunu oynadıkları için yıla özel mekanizmalar için harika bir kaynaktır:  
Team RI3D - 2018 Robot Gösterimi Videosu

- Ortak mekanizmalar söz konusu olduğunda - aktarma organları, asansörler, top giriş/atıcılar,vb-önceki FRC takımları en zor sorunların çoğunu çözmüştür:

- Önceki yıllardan tanınmış FIRST Robotics Competition robot modellerinden oluşan bir koleksiyon, çevrimiçi bir CAD platformuna aktarılmıştır. Erişmek için, bu bağlantıdan bir hesap oluşturduktan sonra genel arama çubuğuna 'TCA FIRST Robotics Competition' ifadesini arayın. İşte 1678'in 2018 Robotuna bir bağlantı örneği

- 3847, robot mekanizmalarının iyi organize edilmiş bir fotoğraf galerisini tutar:  
o Team 3847 - Robot Mekanizmaları

- Team 254'ün teknik dosyaları harika bilgiler içeriyor: o Team 254 - Robotlar

- Team 971'in robot tasarımıyla ilgili sayfasının yanı sıra: o Team 971 - Robot Design

- Bazen, en iyi ilham ticari bir üründen gelir. İlham uçaklardan, otomobillerden, binalardan veya endüstriyel robotlardan gelebilir.

Örn: Team 1678'in 2018 arkadaş kaldırma sistemi, inanılmaz derecede hafif kalırken güç ve dayanıklılık için bir uçak kanadından sonra modellenmiştir.

- En etkili tasarımlardan bazıları basitlik, tekrarlanabilirlik ve doğruluğu birleştirir. 1678 stratejik tasarım sunumu, ekiplerin bu tasarımlara yaklaşmasına yardımcı olacak yönergeler sağlar:

- Team 1678 - Stratejik Tasarım Sunumu

- Team 1678 - Stratejik Tasarım Videosu

- Tasarım, birbiriyle çelişen fikirlere sahip büyük bir grup olarak tartışılırken, genellikle tek bir çözüme ulaşmak zordur. Bazen, herkesin kabul etmesi için yeterli zaman ve mantıklı argümanlar gerekir. Önerilen tüm tasarım fikirlerinin bir takas matrisi oluşturmak, her seçeneğin artılarını/eksilerini tanımlamaya yardımcı olabilir ve ideal bir çözümü aydınlatır. Grup hala anlaşamıyorsa, ekibinizin iki seçeneği olacaktır: ya kararı baş tasarımcıya bırakın ya da popüler bir oylama yapın.

- Genel Robot Düzeni

- İlk, kavramsal tasarımda gerçeklikten kopmak çok kolaydır. Her robotun boyut kısıtlamaları vardır ve bunların tüm tasarım süreci boyunca akılda tutulması gerekir. Robotu önden, yandan ve üstten ölçeklendirecek şekilde çizmek, fikirlerinizin gerçek dünyada uygulanabilir olduğundan emin olmanın etkili bir yoludur.

- Ek olarak, bu ilk tasarım aşamasında elektronik, pil, tampon vb. gibi temel robot bileşenlerini göz ardı etmek kolaydır. Her bileşenin/mekanizmanın yalnızca alana sahip olduğundan değil, aynı zamanda gerektiğinde onarım için erişilebilir olduğundan emin olmak önemlidir. Bu, önemli bileşenlerin konumlarına karar vermek ve hiçbir şeyin müdahale etmeyeceğinden emin olmak için ideal zamandır.

- Doğru bir ölçek tutmak çok kolay olduğundan, genellikle 3 boyutlu tasarım yazılımı kullanmak bunu yapmanın en etkili yoludur. 973 bu konuyla ilgili bir video sağlar:

- Team 973 - Eski RAMP Tasarlayan Robotlar Seviye 1: Mekanizma Tasarımı

## Seviye 1: Mekanizma Tasarımı

Tasarım sürecinin bu noktasında, robotun ne yapması gerektiğine dair somut bir listeye, robotun neye benzeceğine dair genel bir fikre ve çalışacağına inandığınız mekanizmalar için bir dizi konseptte sahip olmalısınız.

Her mekanizma farklı olsa da, aşağıda bir mekanizma tasarlama sürecini kolaylaştırmaya yardımcı olacak bir tasarım iş akışı verilmiştir. Adımların katı olması amaçlanmamıştır; bazı mekanizmalar prototipleme

gerektirmez, bazıları ise prototiplemeyi asla durdurmaz. Bununla birlikte, genel akış hemen hemen tüm durumlar için geçerlidir.

### **Prototipleme**

- Prototipleme, oluşturduğunuz mekanizmalar için kavramları test etmek için ilk fırsattır. Bir prototipin başarısız olmasını ne kadar çok sağlarsanız, o kadar fazla bilgi toplayabilir ve daha sonra daha az başarısızlık yaşarsınız.

- Prototip oluşturmanın anahtarı her zaman hızlı yinleme yeteneğidir. Modeli tam olarak geliştirmek için zaman harcamak genellikle buna değmez. Bunun yerine, ahşap ve vida gibi ucuz inşaat malzemeleri kullanarak inşa edin ve tekerlekli atıcılara el matkaplarıyla güç vermeyi düşünün. Robot hareketini test etmek için prototipi önceki bir robota veya güç verilmeyen bir döner şaseye monte edin. Yinlemeyi aklınızda bulundurun: Prototipin ayarlanmasını kolaylaştırmak için az miktarda karmaşıklık eklemek çoğu zaman buna değer.

Prototiplemeyle ilgili Ek Kaynaklar:

- Çizgilerin Arkasında S02E03 - Etkili Prototipleme
- Bağlantı TBD - FIRST/TCA Prototyping 101 Kaynak

### **Mekanizma Detayları**

- İşlevsel bir prototip oluşturulduktan sonra bile, prototipten öğrenilen dersler kullanarak onu sağlam, üretilebilir bir mekanizmaya dönüştürmek zorlayıcı olabilir. Her mekanizma farklıdır, bu nedenle ne yazık ki bunu yapmanın her zaman işe yarayacak "doğru" bir yolu yoktur.

- FIRST Robotics Competition' daki yaygın yapım yöntemlerini anlamak, bu adım için harika bir başlangıç noktasıdır. Her zaman geçerli olmasa da, genellikle doğru yönü gösterirler:

■ Çoğu mekanizmada çerçeve, basit alüminyum boru, köşebentler ve altıgen millerden yapılabilir. Alüminyum boru, boru boyunca her 0,5 inçte bir delikler yerleştirmek gibi standart bir delik deseni ile birlikte hafif, güçlü ve kolayca genişletilebilir.

■ Bazı durumlarda, kırılması yerine bir mekanizmanın bükülmesi tercih edilir. En yaygın durum, robotun çok dışına uzanan emme mekanizmasıdır. Ortalama maç boyunca kötüye kullanılması muhtemeldir ve esneklik, sert bir vuruştan sonra işlevsel kalmasını sağlayacaktır. Bu uygulamalar için genellikle cevap polikarbonattır.

■ Son olarak, en basit yapı malzemesini asla gözden kaçırmayın: kontrplak. Çalışması kolaydır, hafiftir ve genellikle yeterince güçlüdür. Bazı uygulamalarda, kibrit delinmiş kontrplak, genellikle alüminyumla ilgili olan tüm modelleme, cep açma ve işleme sürecinin yerini alabilir.

■ Daha fazla yaygın yapı örneği için Spectrum robot kitaplığına bakın: Team 3847 - Robot Mechanisms

■ FIRST Robotik Yarışması COTS parçaları

COTS parçaları veya ticari kullanıma hazır parçalar terimi, FIRST Robotik Yarışması'nda robotta satın alınan herhangi bir bileşeni tanımlamak için kullanılıyor. En yaygın bağlantı elemanları, tekerlekler, güç aktarım parçaları ve diğer yapı malzemeleri hakkında bilgi sahibi olmak, bir mekanizmanın nasıl oluşturulacağına karar verirken daha fazla seçenek sağlayacaktır.

● Birçok tedarikçi, ihtiyaç duyacağınız parçaların çoğu için 3D modellere sahiptir. Mevcut bileşenlerine aşina olmak, bir sorun için en iyi çözüme hızla ulaşmanıza yardımcı olabilir. 3D modeller, tedarikçi web sitelerinde, genellikle teknik belgelerde veya ilgilendiğiniz parça veya montaj için indirilebilir içerikte bulunabilir.

## Güç

Hemen hemen her mekanizmanın hareket eden bir parçası olacaktır. Bu hareket dört güç kaynağından birinden gelmelidir: pil, basınçlı hava, yay veya yerçekimi. Depolanan enerjiyi istediğiniz harekete dönüştürmek mekanizma tasarımının en zorlu yönlerinden biri olabilir.

Bir bütün olarak güç aktarımına genel bir bakış için 1678'in sunumuna bakın:

● Team 1678 - Modifiye Güç Aktarımı

2 farklı pozisyona sahip kısa, lineer hareketler için pnömatik çoğu zaman ideal çözüm. 1114'ün sunumu harika bir rehber sunuyor:

● Takım 1114 - Pnömatik

Döner hareket söz konusu olduğunda, zorluk her zaman doğru hızda döndürmedir. Bunun gerçekleşmesi için genellikle önemli azaltmalar gerekir. Bazı takımlar bu hedefe ulaşmak için kendi şanzıman kutularını tasarlamayı tercih ediyor. Özelleştirilmiş şanzıman kutularının avantajları olsa da, birçok ekip bunlara çok fazla zaman harcar, bu zaman diğer, daha önemli şeylere harcanabilir. COTS şanzıman kutuları harika bir alternatif sunar.

Özel şanzıman kutuları hakkında daha fazla bilgi için bu kaynağın 2. Seviyesine bakın.

Son olarak, yay gücü söz konusu olduğunda bir çok seçenek mevcuttur. Standart sarmal yaylar, sıkıştırma veya gerilimde kuvvet sağlayabilir. Tek yönde ekstra kuvvet sağlamak için şaftın etrafına monte edildiklerinde pnömatik ile birlikte de kullanılabilirler. Cerrahi boru sadece gerilimde kuvvet sağlar, ancak beraber çalışması kolaydır ve bir yay kuvvetini "yeniden yönlendirmek" için kablolar ve makaralarla birleştirilebilir. Bu, büyük bir kuvvete ihtiyacınız varsa ancak alanla kısıtlıysanız yardımcı olabilir. Son olarak, inanılmaz yüksek kuvvetlerde gaz şokları mevcuttur. Kapalı olduklarından, 60 psi çalışma basıncı kuralı ile sınırlı değildirler, bu nedenle nispeten küçük bir yer kaplarken yüzlerce pound kuvvet aralığına kolayca ulaşabilirler.

## Sensörler

Sensörlerin ne olduğunu, nasıl çalıştıklarını veya bunların nasıl kullanılacağını anlamak biraz bu kaynağın kapsamı dışındadır. Bununla birlikte, sensörler söz konusu olduğunda bahsetmeye değer birkaç önemli tasarım notu vardır.

En yaygın sensör türü, açısal konumdaki değişiklikleri ölçmek için enkoderdir. Mümkün olduğu her yerde, enkoderleri olabildiğince hızlı dönen şaftlara, diğer bir deyişle motorda önemli bir azalma olmayan şaftlara yerleştirmek en iyisidir. Bu şekilde, daha yüksek bir çözünürlük yaşayacaklar.

Enkoderlerin montajı söz konusu olduğunda, ideal yöntem kodlayıcı tipine bağlıdır. Optik kodlayıcılar için kodlayıcı, şaftla fiziksel olarak ara yüz oluşturacak ve bu nedenle kendisini eş merkezli olmaya zorlayacaktır. Montajın tek görevi dönüşü engellemektir. Bu nedenle, montajınız esnek olmalı ve onu aşırı zorlamamalı ve milin kırılma olasılığını ortaya çıkarmamalıdır. İyi bir örnek, Team 971 tarafından sıklıkla kullanılan polikarb z-bükümleridir, CAD Depolarında bazı örnekleri görebilirsiniz.

Manyetik enkoderler için, enkoder mil ile arayüz oluşturmaz ve bu nedenle mıknatıstan sabit bir mesafe sağlamak için başka bir yerde tamamen sınırlandırılması gerekir. Enkoderde esneme olmadığından emin olmak için montaj için en az iki temas noktası olmalıdır.

Her tür sensörde, bir şeyler ters gittiğinde veya bir sensörü değiştirmeniz gerektiğinde sensörün kolayca erişilebilir olduğundan emin olun.

## Yineleme

Mekanizma tasarımındaki üçüncü ve en önemli adım yinelemedir. Bir mekanizmanın yapımı asla bitmez. Çalıştıktan sonra bile, üzerinde geliştirilebilecek şeyler her zaman vardır. Tasarım süreci sürekli kendini tekrar eder. Mekanizmayı test ettikten sonra, iyileştirme alanlarını belirleyin, zorlukların nasıl çözüleceğine dair daha fazla kavram geliştirin ve süreci baştan başlatın.

## Seviye 2: Ek Tasarım Araçları

- Tasarım hesaplamaları / özel dişli kutuları
  - Özel dişli kutuları tasarlama yeteneği, bir ekip içindeki mekanizmalar için büyük fırsatlar yaratır. Belirli oranları seçebilir veya gücünü, ağırlığını, boyutunu ve verimliliğini optimize edebilirsiniz. Bununla birlikte, bunlar hiçbir şekilde gerekli değildir: COTS dişli kutuları genellikle yeterlidir ve özel bir dişli kutusu geliştirmek için gereken kaynakları harcamak bazen diğer hedeflerden zaman alabilir.

○ İlk ve en önemli kaynak her zaman JVN'nin hesap makinesi olacaktır, burada bulunur:

- Chief Delphi - JVN Hesaplayıcı

o FIRST Robotics Competition tarafından onaylanan birçok motor için ve çoğu şanzıman kutusu tasarımı için geçerli olan motor bilgisi şu adreste bulunabilir:

■ [motors.vex.com](http://motors.vex.com)

o 973'ün google elektronik tabloları, motor eğrileri ve motor seçimi hakkındaki videoları önceki bağlantıyı anlamınıza ve uygulamanıza yardımcı olur. Ek olarak, West Coast Drive videolarında kendini çizen şanzımana değiniyor.

■ Team 973 - RAMP Videoları

### **3D Baskı**

o FIRST Robotik Yarışması kapsamında 3D baskı, çoğu uygulama için ancak son zamanlarda uygulanabilir hale geldi. Her yazıcı farklıdır ve her yazıcının sınırlarını ve neler başarabileceğini bilmek çok önemlidir. Doğru yapıldığında, 3D yazdırılmış bir parça zorlu bir mekanik soruna hızlı, hafif ve etkili bir çözüm olabilir.

o Markforged Blog'u, hangi yazıcıya sahip olduğunuzdan bağımsız olarak birçok harika baskı bilgisi sağlar. Bloglarındaki bazı önemli bağlantılar şunları içerir:

- 3D Basılı Birim Testleri ve Toleranslar
- 3D Basılı Parçalar Neden Bükülür ve Nasıl Durdurulur?
- Baskı Süresini Azaltma
- Gizli Bağlantı Elemanı Mukavemeti için Somunları 3D Basılı Parçalara Gömme

o 3D Hub'lar, yazdırma bilgisi için başka bir harika kaynaktır. Özellikle 6. Adım basılı parçalar ve yaygın basılı tasarım özellikleriyle ilgili bazı önemli püf noktalarına genel bir bakış sağlar:

■ 3D Hub'lar - 3D Baskı için Muhafaza Tasarımı • Kontrol edilebilirlik için tasarım

o Ekibiniz daha karmaşık yazılım sorunlarını çözmeye doğru ilerledikçe, daha iyi kontrol sağlamak için mekanizmalarınızda daha hassas olmanız gerekir. Travis Schuh, kontrol edilebilirlik için tasarım üzerine mükemmel bir sunum sağlar, burada bulunur:

■ Team 971 - Atölye Videoları



# Prototipleme 101

Eksiksiz bir ürün oluşturmada önce mekanizmalar, tasarımlar, hatta oynanış bile denenmeli ve test edilmelidir. Prototipleme, performansı doğrulamak ve fikirlerin uygulanabilirliğini test etmek için ön maketler oluşturma eylemidir.

## Seviye 0: Prototip Kavramları

Robotlar, oyunun nasıl oynanacağına dair bir strateji ile başlar. Bir sonraki adım, 'nasıl' sorusuna cevap vermektir. Prototipleme, ekibinizin fikirlerinizi test etmesine ve bunları tahmin edilen stratejinizle kıyaslamasına olanak tanır.

### 1. Prototipin Amacı

a. Her prototip, genel robot stratejinizden türetilen belirli hedeflere sahip olmalıdır. Prototip için test senaryoları, bu prototipin hedefe nasıl ulaştığını - bu mekanizmanın veya oyun tarzının strateji oturumunuz sırasında ekip tarafından oluşturulan gereksinimleri ne kadar etkili bir şekilde karşıladığını belirlemeye yardımcı olmalıdır.

2015'ten bu uzun Behind the Lines videosuna göz atın:

Behind the Lines S02E03 - Effective Prototyping

### 2. Neyin Prototipini Yapmalıyız?

a. Prototip için en önemli öğeler, oyun nesnelere veya alan öğeleriyle doğrudan etkileşime giren öğelerdir. Bunlar her yıl değişir ve başarılı bir sezon geçirmek için etkileşimde bulunmak çok önemlidir.

b. Şase veya dişli kutuları gibi bazı öğeler mümkün olduğunca geçmiş yılların deneyimlerine dayalı olarak tasarlanmalı ve umarım prototipleme gerektirmez. Bu, prototip oluşturma zamanınızı ve kaynaklarınızı en çok ihtiyaç duyulan yeni şeylere odaklamanıza olanak tanır.

### 3. Prototipin Test Edilmesi

a. Prototiplemenin ana amacı, test etmek, sonuçları kaydetmek ve konsepti geliştirmektir. Prototiplemenin etkin kullanımı, hem konsept üzerinde tekrarlamayı hem de uygun testlerin çalıştırılıp kaydedilmesini gerektirir.

b. Prototipinize karşı yürüttüğünüz tüm testler için anlamlı, tercihen sayısal veriler kaydedebildiğinizden emin olun. Bu veriler, oyun stratejiniz tarafından tanımlanan mekanizmanın gereksinimleriyle uyumlu olmalıdır.

c. Prototiplerinizi gerçek bir eşleşmeymiş gibi test edin. Sürücülerinize veya operatörlerinize, gerçek bir maçta olduğu gibi gerçekçi süreler, mesafeler ve görünürlük verildiğinden emin olun.

d. Tek bir değişiklik yaptığınızda veya ince ayar yaptığınızda, tüm anlamlı testleri yeniden çalıştırın. Değişikliğin bir miktar test tartışması oluşturması mümkündür. Bir öğeyi değiştirmenin başka bir öğede beklenmeyen bir davranış oluşturması da mümkündür. Her testi herhangi bir değişiklikten sonra çalıştırmak, gerilemelerin oluşmasını engeller.

e. Veri toplama ve veri analizi, ne istediğinizi size gösterecektir. Anlamlı sorular sormak ve anlamlı testler oluşturmak, prototipinizin etkin bir şekilde kullanılmasını ve ihtiyacınız olan verileri sağlamasını sağlayacaktır. Bir prototipin bir defada bir eylemi gerçekleştirmesi onu mükemmel yapmaz.

## Seviye 1: Prototipin İnşası

Bir robot gereksiminiz ve inşa etmek ve test etmek için genel bir konseptiniz olduğunda, eldivenleri takıp inşa etme zamanı! Odak, bir şeyleri hızlı bir şekilde oluşturmaya ve değiştirmeye ve değişiklikleri kaydetmeye olmalıdır. Konsept bu süreç boyunca gelişecektir.

### 1. Basit Bir Sürüm Yapın

a. Kafanızda oluşan fikirlerin başkalarına iletilmesi gerekir. Fikrinizin özünü oluşturmak için kâğıt, beyaz tahta veya CAD'deki basit şekiller üzerindeki eskizlerle başlayın.

b. Çiziminizi çoğaltmak için hızla oluşturulabilen ve parçalara ayrılabilen basit malzemeler kullanın.

c. Bu noktada güce veya aktarıma odaklanmayın, minimum uygulanabilir ürüne odaklanın - "çalışan" en basit ve en hızlı nesne.

ii. Takım 1678, prototiplemeyi mekanizmalarının ve prototiplerinin bir parçası olarak açıklıyor Sonbahar Atölyesi: Mekanik İmalat ve Prototipleme

## 2. Kullanışlı Malzemeler

a. Bina ve yapılar için karton, ahşap, PVC boru, koli bandı, yaylar, plastikler ve hatta eski prototipler ve önceki yıllardan "kırık" parçalar gibi malzemeleri kullanın.

b. Güç için milleri manuel olarak döndürebilir veya kaydırıcıları itebilir/çekebilirsiniz. Elektrikli matkapların dönen parçalar için millere takılması kolaydır. Daha yüksek doğruluklu prototipler için, bir pil paketiyle (ve bir açma-kapama düğmesi veya özel potansiyometre!)

c. Çoğu zaman, bu prototipler küçük, dayanıksız veya tutması zordur. Prototipi sert bir yüzeye veya eski bir tekerlekli şasiye tutturmak için kelepçeler, mingeneler veya ağırlıklar kullanın!

i. Ağırlıklar özellikle robot performansını belirlemek için kullanışlıdır. Takım 148, Robowranglers, her mekanizmayı oluşturmaya gerek kalmadan aktarma organlarının ve şasilerinin yük altında nasıl çalışacağını anlamak için laboratuvarlarında çıplak şasiye bantladıkları veya bağladıkları beton bir cüruf bloğuna sahipler.

ii. 2018'de birçok ekip, hareket halindeyken toplamada ne kadar etkili olacaklarını görmek için emme mekanizmalarını eski şasiye veya tekerlekli ahşap kalaslara bağladı.

iii. 3847 ekibinin 2018 blogunun bir kısmına buradan göz atın:

Team 3847 Blog - Ahşap Şasi

iv. Team 3847 ayrıca bileşenleri ölçekte hızlı bir şekilde oluşturmak ve değiştirmek için pvc ve 3D yazdırılabilir eklemleri kullanan proto-pipe adı verilen hızlı prototipleme konseptine sahiptir: GrabCAD - Spectrum Protopipe

### 3. Sıklıkla Yineleyin

a. Prototipiniz uzun süre aynı kalmamalıdır. Her test çalıştırmasında, kaydedilen her sonuçta, değiştirmeniz, ince ayar yapmanız ve yeniden test etmeniz gerekir.

b. Yazılı bir test planınız olsun ve her ince ayar için her testi yeniden çalıştırın. Değişikliği kaydedin, yeni testin sonuçlarını ve önceki permütasyonlarla karşılaştırın.

c. Her yineleme, tanımlanan ve denenmeye çalışılan bir "başarısızlıktan" oluşmalıdır. Her yineleme, aynı anda yalnızca bir öğeyi veya bileşeni değiştirmeyi içermelidir.

i. 2018 için, 4911 ekibi ve diğer birçok ekip, emme mekanizması prototipleri oluşturdu ve çekiş tekerlekleriyle bir dizi test yaptı, ardından sütunlu tekerleklerle değiştirildi, ardından yeşil tekerlekleri kullandı, ardından başka bir tekerlek denedi ve oyun öğelerini almanın verimliliğini kaydetti. Veriler, en iyi performansın, tutarlılığın ve uzun ömürlülüğün yeşil tekerleklerden geldiğini gösterdi.

## Seviye 2: Konsepti Geliştirmek

Temel kavramları ve prototipi oluşturduktan sonra, süreci iyileştirebilir ve prototipin aslına uygunluğunu artırabilirsiniz. Aslına daha yüksek uygunluklu prototipleri, sonunda pratik robotlarında veya rekabet robotlarında yollarını bulabilir ve temel kavramlar test edildikten sonra robot performansını iyileştirmeye devam etmek için kullanılır.

### 1. Faydalı Araçlar

a. Bazı ekipler, daha yüksek doğrulukta prototipler oluşturmak için karmaşık şekilleri hızla üretme amacıyla hassas makinelere dönebilir. Ahşap/plastik hala metalden daha ucuzdur ve çoğu durumda üretimi daha hızlıdır. CNC frezeleri ve yönlendiricileri kullanabilen ekipler, prototiplerinin hassasiyetini ve toleranslarını artırmak ve onları rekabet özelliklerine yaklaştırmak için bunu yaparlar.

b. Elinizde varsa eski robotları kullanın. Ekibinizin yaşına ve çalışma alanınızın depolama kapasitesine bağlı olarak, eski robotları etrafta ve çalışır durumda tutmak, prototip oluşturmaya ve test etmeye yardımcı olabilir. Zaten kodları ve güçleri var ve yeni prototip mekanizmanızı çalıştırmak için uyarlanabilirler.

c. Test sırasında, kamera tekrarları ve ağır çekim kameralar, prototipinizin oyun nesneleriyle nasıl etkileşime girdiği hakkında çok daha ince ayrıntıları ortaya çıkarabilir. Kamera incelemeleri, harika bir test kanıtı sağlar ve tekrarlanabilirlik için test yaparken nüansları gösterebilir.

i. 2017'de birçok takım, yakıt oyun parçalarının mekanizmalarından nasıl ve neden atıldığını görmek için ağır çekim kameralar veya cep telefonu videosundan ağır çekim çekim kullandı.

## 2. Daha Yüksek Sadakat Prototipi Oluşturma

a. Prototipiniz, bir kavram kanıtından, yüksek doğrulukta bir performans doğrulamasına doğru hareket etmelidir. İnce ayarlar ve değişiklikler, eşleşen üretim kalitesine odaklanmalıdır. İlk adımlar manuel kontrolleri kaldırmak, ardından FIRST Robotics Competition motorları, kontrolörleri ve kodu ile değiştirmek olmalıdır.

b. Kavram kanıtı tamamlandıktan sonra, bir sonraki hedef performansı artırmak olmalıdır. Prototipin daha hızlı çalışmasını sağlayın ve dayanıklılığı zorlamak için daha fazla döngü çalıştırın.

c. Malzemeleri yükseltmeye başlayın. Şimdi kayış, zincir, polikarbonat vb. kararların alınması gereken yer burasıdır. Ahşap yapılar daha hafif ve daha güçlü alüminyum ile değiştirilmelidir.

d. Prototipe karmaşıklık eklemeye başlayın. Hızlı pozisyon değişikliğine veya parça takasına izin veren bazı karmaşıklıklar belirlenmeli ve eklenmelidir - uzun vadede zaman kazandıracaktır.

i. Volan atış mekanizmalarına özel olarak, ekipler prototip oluştururken boşluğu/sıkıştırmayı ve çıkış açısını hızla ayarlamak için genellikle kızaklar ve kameralar kullanır.

ii. Farklı ekiplerin prototiplerini nasıl oluşturduğunu ve geliştirdiğini açıklayan bu slaytlara göz atın:  
Behind the Lines - Effective Prototyping Techniques

## Seviye 3: Ek Düşünceler

- Prototipler asla bitmez. Yarışma robotuna girdikten sonra hala ince ayar yapılabilir, değiştirilebilir ve test edilebilirler.



- Stratejik prototipleme diye bir şey var! Yeni oyunu “ oynamak ” için benzer oyun parçalarına sahip önceki robotları kullanın (ya da kullanmayın), farklı döngüleri deneyin, döngü sürelerini ve puanları karşılaştırın, oyun alanının belirli yollarının veya bölümlerinin ne kadar dar olduğunu test edin.
- Birçok ekibin prototipleri belgelenmiştir—sizin için pek çok işi bitirdiler! Asansörler, dört çubuklu bağlantılar, tırmanma/kaldırma mekanizmaları, volan atıcılar, yukarıdakilerin tümü için mevcut CAD vardır. Belirli öğeler için kavram kanıtını atlayın ve doğrudan daha yüksek kaliteli bir modele geçin.
- Prototipleri yarışmaya hazır bileşenlere dönüştürmek - AYNI OLMAK çok önemlidir. Boşluk, sürtünme, güç, boyutlar - bunlar, son versiyonunuzu üretirken aynı şekilde kopyalanmalıdır.

# Programlama 101

Bu kaynak, FIRST® Robotics Competition' daki programlamanın temellerini kapsayacaktır. C++, Java/Kotlin ve LabVIEW'i kapsar.

## Seviye 1: Robotunuzu Çalıştırmak

### 1. Bir Programlama Dili Seçme: Java, C++ veya LabVIEW

- **Java** liselerde yaygın olarak öğretilen ve AP Bilgisayar Bilimi sınavlarında kullanılan bir metin dilidir. Kendi sanal ortamında çalışan “güvenli” bir dildir. Genellikle FIRST Robotics Competition' ı etkilemese de, JVM olarak da bilinen bu sanal ortam, Java programlarının hesaplama açısından yoğun görevler için kullanıldığında derlenmiş dillerden belirgin şekilde daha yavaş olduğu anlamına gelir. Java genellikle kullanım kolaylığı ve çapraz uyumluluğu nedeniyle seçilir. Java kullanan ekipler arasında 254, 125, 503, 4911 ve 1241 bulunur.

- **C++** hızlı bir metinsel programlama dilidir. Gücü ve verimliliği nedeniyle endüstride gerçek zamanlı sistemler için kullanılır, ancak öğrenme eğrisi Java'dan çok daha diktir. C++, C programlama dilinden evrimleşmiştir ve tarihsel ve modern özelliklerin karışımı bazen kafa karıştırıcı sözdizimlerine ve beklenmedik davranışlara yol açar. FIRST Robotics Competition ekipleri öncelikle hızı, esnekliği ve kapsamlı matematik kütüphaneleri nedeniyle kullanıyor. C++ kullanan takımlar arasında 971 ve 1678 bulunur.

- **LabVIEW**, National Instruments (NI) tarafından mühendisler ve teknisyenler tarafından kullanılmak üzere geliştirilmiş bir grafik veri akışı programlama dilidir. FIRST LEGO League Jr ve FIRST LEGO League için kullanılan LEGO WeDo ve Mindstorms dilleri LabVIEW'in türevleridir; bu yüzden bu programlardan gelen öğrenciler onu tanıdık bulabilirler. Bir LabVIEW diyagramında, kod parçalarını paralel olarak çalıştırmak gibi gelişmiş bilgi işlem özelliklerinden yararlanmak çok kolaydır. Güçlü olmakla birlikte, bu tür özellikler genellikle üstesinden gelinmesi gereken yeni sorunlar ortaya çıkarır. Ancak NI, kapsamlı hata ayıklama araçları sağlar. LabVIEW ortamı ve dili, kendi öğrenme eğrisi ve benzersiz zorlukları ile birlikte gelir. FIRST Robotics Competition ekipleri, basitleştirilmiş grafik sözdizimi ve kapsamlı mühendislik kitaplıkları nedeniyle öncelikle bunu kullanır. LabVIEW kullanan takımlar arasında 33, 359, 624, 1986 ve 2468 bulunur.

- **Hangi dili seçmek her zaman ekibiniz için en kolay olana bağlıdır.** Örneğin, bir programlama danışmanı o dilde uzman olduğu için genellikle bir dil seçmek mantıklı olabilir. Öte yandan, belirli bir dili öğrenmenin kolaylığına göre seçim yapmak mantıklı olabilir. Kararınız ne olursa olsun, programlama dili seçiminin



çalışma ortamına ve ekibinizdeki kişilere özel olduğunu unutmayın. Tüm diller, FIRST Robotics Competition kullanımı için kabiliyetli, iyi desteklenmiş ve yeterince güçlüdür.

## 2. Programlama Dilinin Öğretimi

- FIRST Robotics Competition için programlama öğretirken, öğretilmesi gereken iki ayrı konu vardır. Birincisi, programlama dilinin kendisinin semantiği ve sözdizimidir. İkincisi, FIRST Robotics Competition bileşenleriyle arayüz oluşturmaktır. C++ dilini öğrenmek için bir kılavuz burada, Java burada ve LabVIEW burada bulunabilir.

## 3. Başlangıç kodunuzu seçme

- Java ve C++ için Robot ile arayüz oluştururken kullanılacak dört farklı "sınıf" vardır. Bu sınıfların bir karşılaştırması ve ne anlama geldikleri burada bulunabilir.

- LabVIEW için başlangıç şablonları, zamanlı, yinelemeli ve yeniden başlatma/iptal etme mekanizmalarının bir karışımını içerir. Şablonlar burada açıklanmıştır.

**4. Ekibiniz bir programlama dili seçtikten ve kodlamaya başladıktan sonra, FIRST Robotics Competition Docs sitesi, geliştirme ortamınızı nasıl kuracağınız ve robotunuza kodu nasıl alacağınız konusunda iyi bir kaynaktır. Bu kılavuzlar, FIRST Robotics Competition programlaması için çok değerlidir.**

- Başlarken
- C++\Java
- LabVIEW

## 5. Robotunuza kod yüklemek!

- Java ve C++
  - gradleRIO kullanıyorsanız, VS Koduna `./gradlew dağıtım` yazmanız yeterlidir. Konsol ve kodunuz robotta olacak.
  - Fakat bekle! Kodunuz henüz hiçbir şey yapmıyor. Sürücü kodu için bazı basit örnekler burada bulunabilir. Parçacıklardaki kod, gradle/Eclipse projesiyle otomatik olarak oluşturulması gereken Robot.java veya Robot.cpp'ye aittir.



- LabVIEW

- Geliştirme için burada gösterildiği gibi kaynaktan çalıştırmalısınız.
- Tamamlandığında, burada gösterildiği gibi yerleşik bir yürütülebilir dosyayı dağıtacaksınız.

## 6. Mekanizmalar için kod

- Java ve C++ için basit sürücü kodu buradan kopyalayıp yapıştırılabilir.

- LabVIEW için, basit sürücü kodu TeleOp VI'daki şablondadır.

- Çoğu FIRST Robotik Yarışması robotları, aktarma organları dışında harekete geçirilen mekanizmalara sahiptir. Bu, dönen bir volandan pnömomatik bir mançınığa kadar her şey olabilir. Tüm bu mekanizmalar otonom veya teleop olarak kontrol edilebilir olmalıdır. PWM üzerinden hız kontrolörleri kullanan mekanizmaları kontrol etmek için burada C++ ve Java için bir kılavuz ve LabVIEW için burada bir kılavuz bulunmaktadır.

- CAN üzerinden hız kontrolörleri kullanıyorsanız, bunları PWM hız kontrolörleri olarak ele almak için buradaki kılavuzu takip etmeli veya belgeleri burada bağlantılı olan Phoenix API'yi kullanmalısınız.

## 7. Otonom

- Java ve C++ programlamasında otonom eylemlerin nasıl yapılacağına ilişkin bir kılavuz burada bulunabilir.

- Team 1619, Java'da otomatik çizgiyi geçmek için burada bulunabilecek bazı basit kodlar da derledi.

- LabVIEW şablonları, robotu yerinde sallamak için otonom kod içerir. Birçok görevi gerçekleştirmek için motor gücü değerlerini ve zamanlamasını değiştirebilirsiniz. İşte 2018'den 2468'in otonom kodunun bir örneği.

## Seviye 2: Özel Mimari ve Kapalı Çevrim Motor Kontrolü

### 1. Özel bir mimari kullanma

- Çoğu zaman, mevcut robot sınıfları yeterli değildir. Örneğin, teleop'u periyodik olarak ve otonom olarak sırayla çalıştırmak isteyebilirsiniz. Bu durumda, özel bir mimariye geçmenin zamanı gelmiş olabilir.

- Özel bir mimari, esasen tüm kodu özelleştirilmiş bir şekilde yapılandırıyor.
- Bazı özel mimari örnekleri, burada bulunan 1678 kodunu içerir. 1678'in kodu, burada bulunan 971'in kodundan oluşuyor.
- 254 ayrıca özel bir mimariye sahiptir. 2019 kodları burada bulunabilir.
- 33, 624 ve 1986 ve 2468

## 2. PID Kontrolü

- PID Kontrolü, bir mekanizmayı voltaj yerine konuma göre kontrol etmenizi sağlar. PID kullanarak, bir kola doğrudan voltaj vermesini söylemek yerine 30 dereceye dönmesini söyleyebilirsiniz. Bu özellikle otonomda kullanışlıdır. Bir robota 0,5 saniye boyunca tam güç yerine 5 metre sürmesini söyleyebilmek, gelişmiş tekrarlanabilirlik sağlar.
- PID için bazı yararlı belgeler şunlardır:
  - Wesley'nin Blogu
  - Aptallar için CSIM'in PID'si

## 3. MotionMagic (Yalnızca CAN)

- TalonSRX hız kontrol cihazı kullanıyorsanız, özellikle kol veya asansör gibi mekanizmaları kontrol etmek için MotionMagic kullanılması önerilir. MotionMagic, otomatik olarak oluşturulmuş yamuk hareket profillerini izleyen esasen 1KHz PID döngüsüdür. Bu sözler bir anlam ifade etmiyorsa, endişelenme! PID hakkında bilgi için yukarıya bakın ve işte hareket profillerini açıklayan bir belge.
- Motion Magic için belgeler burada bulunur.

## Seviye 3: Gelişmiş Sürüş Yolları, MP Kontrolü ve Birim Testi

### 1. Şürüş yolları ve onları takip etmek

- Bazen ham PID, aktarma organlarını otonom olarak kontrol etmek için yeterli değildir. Örneğin, robotun anahtarın etrafından dolaşmasını ve arkadan bir küp almasını isteyebilirsiniz. Bunu yapmanın temiz bir yolu, bir sürüş yolu oluşturmaktır. Bir sürüş yolu, esas olarak, aktarma organları PID döngüsünün izleyeceği ve noktalar nihai hedefe götürecek bir dizi noktadır. PathWeaver, bu tür yolları oluşturan ve

bunları ayrıştırılabilir bir dosyaya kaydeden PathFinder adlı bir kitaplığı kullanan grafiksel bir araçtır. PathWeaver kullanımıyla ilgili ayrıntılı talimatlar burada bulunur.

- Noktalar oluşturulduktan sonra, onları takip etmenin çeşitli yolları vardır. Bunlar, noktaları doğrudan takip etmek için PID kullanmaktan, noktaları PID döngüsüne vermeden önce işlemek için bir yol takip algoritması eklemeye kadar uzanır. Böyle bir yol izleme algoritmasının bir örneği burada bulunabilir (eqn 5.12). Yol izlemeye yönelik diğer popüler yaklaşımlar, uyarlanabilir saf takip kontrolünü içerir. 254, burada bulunabilecek kullanışlı bir uyarlanabilir saf takip uygulamasına sahiptir.

## 2. Model tabanlı kontrol

- Model tabanlı kontrol, PID'nin ötesinde bir adımdır. Sistemin matematiksel bir modelinin kodda tutulmasına ve modelin sensör verileriyle güncellenmesine olanak tanır. Böyle bir model kullanılarak bir mekanizmanın konumu, hızı, ivmesi vb. çok daha hassas bir şekilde kontrol edilebilir. Model tabanlı kontrolü kullanan bazı ekipler arasında 1678 ve 971 bulunur.

- Model tabanlı kontrolü öğrenmek için faydalı kaynaklar şunlardır:
- Wesley'nin Blogu
- Bu MIT broşürü

## 3. Birim Testi

- Çoğu zaman, kodunuzu robota yerleştirmeden önce test etmek istersiniz. Bu felaketi önleyebilir. Birim testi, kodun bölümlerini bağımsız programlar olarak test etmek için kullanılan bir terimdir. Örneğin, kodun asansörü çalıştıran kısmını test etmek isteyebilirsiniz, ancak birkaç ışığın yanıp sönmelerini sağlayan kısmı değil. FIRST Robotics Competition için test mekanizmaları, modele dayalı kontrol ile büyük ölçüde geliştirilmiştir, çünkü model mekanizmanın simülasyonu olarak kullanılabilir, yani tüm mekanizma inanılmaz bir sağlamlıkla test edilebilir. Bazı yararlı birim test kitaplıkları şunları içerir:

- GoogleTest



### **Pusula İttifakı Hakkında**

"Compass Alliance", FIRST Robotics Competition ekiplerinin ayakta kalmasına ve büyümesine yardımcı olma misyonu ile dünyanın dört bir yanından 10 ekip tarafından kuruldu. Büyüyen bir Kaynak Deposu ve 7/24 Çağrı Merkezi, herhangi bir beceri seviyesindeki herkese dünyanın herhangi bir yerinden yeni bir şeyler öğrenmesi veya daha fazlasını öğrenmesi için araçlar sağlar. Mentorları olmayan uzak ekipler, sezon boyunca uzaktan rehberleri olması için bir Etiket Ekibine kaydolabilir ve Yardım Merkezleri, diğer FIRST ekiplerinin sunduğu yerel hizmetlere nereden erişebileceklerini saptayabilir. Hear For You, ekiplerin ve gönüllülerin ekiplerinde ve etkinliklerde zihinsel sağlık geliştirmelerine yardımcı olacak kaynakları ve araçları sağlar. The Compass Alliance hakkında daha fazla bilgi edinebilir ve kaliteli yardımı [www.thecompassalliance.org](http://www.thecompassalliance.org) adresinde bulabilirsiniz.

### **Bu Kaynak Hakkında**

Bu kaynak, FIRST'in desteği ve genel bakışıyla The Compass Alliance tarafından hazırlanmıştır. Bu kaynakla ilgili sorularınız varsa, lütfen [thecompassalliance@gmail.com](mailto:thecompassalliance@gmail.com) veya [firstroboticscompetition@firstinspires.org](mailto:firstroboticscompetition@firstinspires.org) ile iletişime geçin.

### **Revizyon Geçmişi**

Revizyon #	Revizyon Tarihi	Revizyon Notları
1.0	Aralık 2019	İlk yayın