Programlama 101

Bu kaynak FIRST® Robotics Competition'da programlamanın temellerini kapsayacaktır. C++, Java/Kotlin ve LabVIEW konularını kapsamaktadır.

Birinci Seviye: Robotunuzu Çalıştırın

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

Java liselerde yaygın olarak öğretilen ve AP CS sınavlarında kullanılan metinsel bir dildir. Kendi sanal ortamında çalışan "güvenli" bir dildir. Genel olarak 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 seçilmektedir. Java kullanan takımlar arasında 254, 125, 503, 4911ve 1241 bulunmaktadır.

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 sözdizimine ve/veya beklenmedik davranışlara yol açar. FIRST Robotics Competition ekipleri öncelikle hızı, esnekliği ve kapsamlı matematiksel kütüphaneleri nedeniyle bu dili kullanmaktadır. C++ kullanan takımlar arasında 971 ve 1678 bulunmaktadır.

LabVIEW, National tarafından geliştirilen grafiksel bir veri akışı programlama dilidir. Mühendisler ve teknisyenler tarafından kullanılmak üzere Instruments (NI). LEGO WeDo ve FIRST LEGO League Jr ve FIRST LEGO League için kullanılan Mindstorms dilleri LabVIEW'in türevleridir; dolayısıyla bu programlardan gelen öğrenciler LabVIEW'i tanıdık bulabilir. Bir LabVIEW diyagramında, kod parçalarını paralel olarak çalıştırmak gibi gelişmiş bilgi işlem özelliklerinden yararlanmak çok kolaydır. Güçlü olsa da, bu tür özellikler genellikle başa çıkılması gereken yeni sorunlar ortaya çıkarır. Ancak NI, kapsamlı hata ayıklama araçları sağlar. LabVIEW ortamı ve dili, kendi öğrenme eğrisi ve benzersiz zorluklarıyla birlikte gelir. FIRST Robotics Competition ekipleri, basitleştirilmiş grafik söz dizimi ve

kapsamlı mühendislik kütüphaneleri nedeniyle öncelikle bu dili kullanmaktadır. LabVIEW kullanan takımlar arasında 33, 359, 624, 1986 ve 2468 bulunmaktadır.

Hangi dili seçeceğiniz her zaman ekibiniz için neyin en kolay olduğuna bağlıdır. Örneğin, bir programlama danışmanı o dilde uzman olduğu için bir dil seçmek genellikle mantıklı olabilir. Öte yandan, belirli bir dili öğrenmenin kolaylığına göre seçim yapmak da mantıklı olabilir. Kararınız ne olursa olsun, programlama dili seçiminin çalışma ortamına ve ekibinizdeki kişilere özgü olduğunu unutmayın. Bütün dilleri FIRST Robotics Competition kullanımı için yetenekli, iyi desteklenmiş ve yeterince güçlüdür.

2. Programlama Dilinin Öğretilmesi

- FIRST Robotics Competition için programlama öğretirken, öğretilmesi gereken iki farklı konu vardır. Birincisi programlama dilinin kendi semantiği ve sözdizimi, ikincisi ise FIRST Robotics Competition bileşenleri ile arayüz oluşturmaktır. C++ dilinin öğrenilmesine ilişkin bir kılavuza buradan, Java'ya buradan ve LabVIEW'e buradan ulaşabilirsiniz.

3. Başlangıç kodunuzu seçme

- Java ve C++ için, Robot ile arayüz oluştururken kullanılabilecek dört farklı “sınıf” vardır. Bu sınıfların bir karşılaştırması ve ne anlama geldikleri burada bulunabilir.
- LabVIEW için, başlangıç şablonları zamanlanmış, yinelemeli ve doğma/iptal 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 nasıl kod yükleyeceğiniz konusunda iyi bir kaynaktır. Bu kılavuzlar FIRST Robotics Competition programlaması için çok değerlidir.

- Başlarken
- C++\Java
- LabVIEW

5. *Robotunuza kod yükleyin!*

- Java ve C++

  -gradleRIO kullanıyorsanız, VS Koduna "./gradlew deploy" yazmanız yeterlidir. Konsol ve kodunuz robotta olacak.

  - Ama durun! Kodunuz henüz hiçbir şey yapmıyor. Sürücü kodu için bazı basit örnekler burada bulunabilir. Parçacıklardaki kod, gradle/Eclipse projesi ile otomatik olarak oluşturulması gereken Robot.java veya Robot.cpp'ye aittir.

- LabVIEW

  - Geliştirme için, burada gösterildiği gibi kaynaktan çalıştırmalısınız.

  - Tamamlandığında, burada gösterildiği gibi oluşturulmuş bir yürütülebilir dosyayı dağıtacaksınız.

6. *Mekanizmalar için kodlar*

- Java ve C++ için, basit sürücü kodu buradan kopyalanıp yapıştırılabilir.
- LabVIEW için, basit sürücü kodu TeleOp VI'daki şablonda bulunmaktadır.
- FIRST Robotics Competition robotlarının çoğunda aktarma organları dışında harekete geçirilen mekanizmalar vardır. Bu, dönen bir volandan pnömatik bir mancınığa kadar her şey olabilir. Tüm bu mekanizmalar otonom veya teleop olarak kontrol edilebilmelidir. PWM üzerinden hız kontrolörleri kullanarak mekanizmaları kontrol etmek için burada C++ ve Java için ve burada LabVIEW için bir kılavuz bulunmaktadır.
- CAN üzerinden hız kontrolörleri kullanıyorsanız, bunları PWM hız kontrolörleri olarak ele almak için ya buradaki kılavuzu izlemeli ya da belgeleri burada bağlantılı olan Phoenix API'yi kullanmalısınız.

7. *Otonom*

- Java ve C++ programlamada otonom eylemlerin nasıl yapılacağına dair bir rehbere buradan ulaşabilirsiniz.
- Team 1619 ayrıca Java'da otomatik çizgiyi geçmek için bazı basit kodlar derlemiştir, bunlara buradan ulaşabilirsiniz.
- LabVIEW şablonları, robotu yerinde sallamak için otonom kod içerir. Birçok görevi yerine getirmek için motor gücü değerlerini ve zamanlamayı değiştirebilirsiniz. İşte 2468'in 2018'den otonom kodunun bir örneği.

İkinci Seviye: Özel Mimari ve Kapalı Döngü Motor Kontrolü

1. Özel bir mimari kullanma

- Çoğu zaman, mevcut robot sınıfları yeterli değildir. Örneğin, periyodik olarak teleop ve sıralı olarak otonom çalıştırmak isteyebilirsiniz. Eğer durum buysa, muhtemelen özel bir mimariye geçme zamanı gelmiştir.
- Özel mimari esasen tüm kodun özelleştirilmiş bir şekilde yapılandırılmasıdır.
- Bazı özel mimari örnekleri arasında 1678'in burada yer alan kodu bulunmaktadır. 1678'in kodu 971'in buradaki kodunu temel alır.
- 254'ün de özel bir mimarisi vardır. Onların 2019 kodu 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 bir voltaj çıkışı yapmasını söylemek yerine 30 derece dönmesini söyleyebilirsiniz. Bu özellikle otonom sistemlerde kullanışlıdır. Bir robota 0,5 saniye boyunca tam güç yerine 5 metre sürmesini söyleyebilmek, gelişmiş tekrarlanabilirlik sağlar.
- PID için bazı yararlı belgeler şunlardır:
    - Wesley'in Blogu
    - CSIM'in Aptallar için PID'si

3. Motion Magic (Yalnızca CAN)

- TalonSRX hız kontrol cihazı kullanıyorsanız, özellikle kol veya asansör gibi mekanizmaları kontrol etmek için MotionMagic kullanmanız önerilir. MotionMagic esasen otomatik olarak oluşturulan trapezoidal hareket profillerini takip eden 1KHz PID döngüsüdür. Bu kelimeler bir anlam ifade etmiyorsa endişelenmeyin! PID hakkında bilgi için yukarıya bakın ve burada hareket profillerini açıklayan bir belge var.
- Motion Magic için belgeler burada yer almaktadır.

Üçüncü Seviye: Gelişmiş Sürüş Yolları, MP Kontrolü ve Birim testi

1. Sürücü yolları ve onları takip etme

- 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. Tahrik yolu, esasen aktarma organı PID döngüsünün takip edeceği bir dizi noktadır ve noktalar nihai hedefe götürecektir. PathWeaver, bu tür yolları oluşturan ve bunları ayrıştırılabilir bir dosyaya kaydeden PathFinder adlı bir kütüphane kullanan grafiksel bir araçtır. PathWeaver kullanımı ile ilgili ayrıntılı talimatları burada bulabilirsiniz.
- Noktalar oluşturulduktan sonra, bunları 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 değişir. Böyle bir yol takip algoritmasının bir örneği burada bulunabilir (eşitlik 5.12). Yol takibine yönelik diğer popüler yaklaşımlar arasında uyarlanabilir saf takip kontrolü yer almaktadır. 254'te uyarlanabilir saf takibin kullanışlı bir uygulaması bulunmaktadır ve burada bulunabilir.

2. Model bazlı kontrol

- Model tabanlı kontrol, PID'nin bir adım ötesidir. 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 bulunmaktadır.
- Model tabanlı kontrolü öğrenmek için faydalı kaynaklar şunlardır:
  - Wesley'in Blogu
  - Bu MIT broşürü

3. Birim testleri

- Çoğu zaman, kodunuzu robota yerleştirmeden önce test etmek istersiniz. Bu bir 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 sönmesini sağlayan kısmını test etmek istemeyebilirsiniz. FIRST Robotics Competition için mekanizmaların test edilmesi, model tabanlı kontrol ile büyük ölçüde geliştirilmiştir, çünkü model mekanizmanın bir simülasyonu olarak kullanılabilir, bu da tüm mekanizmanın inanılmaz bir sağlamlıkla test edilebileceği anlamına gelir. Bazı faydalı birim test kütüphaneleri şunlardır:
- GoogleTest

Compass Alliance Hakkında

Compass Alliance, FIRST Robotics Competition takımlarının varlıklarını sürdürmelerine ve büyümelerine yardımcı olmak amacıyla dünyanın dört bir yanından 10 takım tarafından kurulmuştur. Büyüyen bir Kaynak Deposu ve 7/24 Çağrı Merkezi, herhangi bir beceri seviyesindeki herkese dünyanın herhangi bir yerinden yeni bir şeyler öğrenmek veya daha fazla bilgi edinmek için araçlar sağlar. Mentoru olmayan uzak ekipler, sezon boyunca uzaktan rehberlik edecek bir Etiket Ekibine kaydolabilir ve Yardım Merkezleri, diğer FIRST ekiplerinin sunduğu yerel hizmetlere nereden erişileceğini belirler. Hear For You, ekiplerin ve gönüllülerin ekiplerinde ve etkinliklerde zihinsel sağlıklarını geliştirmelerine yardımcı olacak kaynaklar ve araçlar sağlar. The Compass Alliance hakkında daha fazla bilgi edinebilir, kaliteli yardım bulabilir ve www.thecompassalliance.org adresinden katılım sağlayabilirsiniz.

Bu Kaynak Hakkında

Bu kaynak, FIRST'ün desteği ve genel bakışıyla Compass Alliance tarafından hazırlanmıştır. Bu kaynak hakkında sorularınız varsa, lütfen thecompassalliance@gmail.com veya firstroboticscompetition@firstinspires.org ile iletişime geçin.

Revizyon Geçmişi

Revizyon #	Revizyon Tarihi	Revizyon Notları
1.0	Ara. 2018	İlk Sürüm