Programare 101

Această resursă va acoperi elementele de bază ale programării în competiția FIRST® Robotics. Acesta acoperă C ++, Java / Kotlin și LabVIEW.

Nivelul unu: Punerea robotului în funcțiune

1. Alegerea unui limbaj de programare: Java, C ++ sau LabVIEW

- **Java** este un limbaj textual care este predat în mod obișnuit la licee și utilizat pentru examenele AP CS. Este un limbaj "sigur" care rulează în propriul mediu virtual. Deși nu afectează în general PRIM Robotics Competition, acest mediu virtual, cunoscut și sub numele de JVM, înseamnă că programele Java sunt considerabil mai lente decât limbajele compilate atunci când sunt utilizate pentru sarcini intensive din punct de vedere computațional. Java este adesea selectat datorită ușurinței sale de utilizare și compatibilității încrucișate. Printre echipele care utilizează Java se numără: 254, 125, 503, 4911, și 1241.
- **C++** este un limbaj de programare textual rapid. Este folosit în industrie pentru sisteme în timp real datorită puterii și eficienței sale, dar curba de învățare este mult mai abruptă decât Java. C++ a evoluat din limbajul de programare C, iar amestecul de caracteristici istorice și moderne duce uneori la sintaxă confuză și / sau comportamente neașteptate. Echipele FIRST Robotics Competition îl folosesc în primul rând datorită vitezei, flexibilității și bibliotecilor matematice extinse. Echipele care folosesc C++ includ 971 și 1678.
- **LabVIEW** este un limbaj de programare grafică dezvoltat de National Instrumente (NI) pentru utilizare de către ingineri și tehnicieni. LEGO WeDo și Limbajele Mindstorms folosite pentru FIRST LEGO League Jr și FIRST LEGO League sunt derivate din LabVIEW; astfel încât studenții care provin din aceste programe îl pot găsi familiar. Într-o diagramă LabVIEW, este foarte ușor să profitați de caracteristicile avansate de calcul, cum ar fi rularea bucăților de cod în paralel. Deși puternice, astfel de caracteristici introduc adesea noi probleme de rezolvat. Cu toate acestea, NI oferă instrumente extinse de depanare. Mediul și limbajul LabVIEW vin cu propria curbă de învățare și provocări unice. Echipele FIRST Robotics Competition îl folosesc în primul rând datorită sintaxei grafice

simplificate și bibliotecilor extinse de inginerie. Printre echipele care utilizează LabVIEW se numără 33, 359, 624, 1986 și 2468.

- **Alegerea limbii depinde întotdeauna de ceea ce este mai ușor pentru** echipa ***dvs.*** De exemplu, poate avea adesea sens să alegeți un limbaj, deoarece un mentor de programare este un expert în acel limbaj. Pe de altă parte, ar putea avea sens să alegeți pe baza ușurinței de a învăța o anumită limbă. Indiferent de decizia ta, amintește-ți că alegerea limbajului de programare este specifică mediului de lucru și oamenilor din echipa ta. Toate limbile sunt capabile, bine susținute și suficient de puternice pentru utilizarea FIRST Robotics Competition.

2. Predarea limbajului de programare

- Când predați programare pentru PRIM Competiția de robotică, există două materii distincte care trebuie predate. Primul este semantica și sintaxa limbajului de programare în sine, iar al doilea este interfața cu PRIM Componentele concursului de robotică. Un ghid privind învățarea limbajului C ++ poate fi găsit aiciJava ei e, și LabVIEW aici.

3. Alegerea codului de pornire

- Pentru Java și C ++, există patru "clase" diferite care pot fi utilizate atunci când interfațați cu robotul. O comparație a acestor clase și a ceea ce înseamnă ele poate fi găsită aici.
- Pentru LabVIEW, șabloanele de pornire includ un amestec de mecanisme temporizate, iterative și de reproducere/avort. Șabloanele sunt descrise aici.

4. ***Odată ce echipa dvs. a ales un limbaj de programare și a început să codifice, site-ul FIRST Robotics Competition Docs este o resursă bună despre cum să vă configurați mediul de dezvoltare și să obțineți cod pe robotul dvs. Aceste ghiduri sunt de neprețuit pentru programarea FIRST Robotics Competition.***
   - Introducere
   - C++\Java
   - LabVIEW

5. ***Obținerea codului pe robotul dvs.!***
   - Java și C++
   - Dacă utilizați gradleRIO, tastați "./gradlew deploy" în codul VS Consola, iar codul tău va fi pe robot.

- Dar așteaptă! Codul tău nu face încă nimic. Câteva exemple simple pentru codul de unitate pot fi găsite aici. Codul din fragmente aparține fie Robot.java, fie Robot.cpp, care ar trebui să fie creat automat cu proiectul gradle/Eclipse.
- LabVIEW
- Pentru dezvoltare, ar trebui să fugiți de la sursă, așa cum se arată aici.
- După finalizare, veți implementa un executabil construit așa cum se arată aici.

6. Cod pentru mecanisme

- Pentru Java și C ++, codul simplu de unitate poate fi copiat-lipit de aici.
- Pentru LabVIEW, codul simplu al unității se află în șablonul din TeleOp VI.
- Majoritatea roboților FIRST Robotics Competition au mecanisme acționate, altele decât trenul de rulare. Aceasta ar putea fi orice, de la un volant care se învârte la o catapultă pneumatică. Toate aceste mecanisme ar trebui să fie controlabile în mod autonom sau în teleop. Pentru a controla mecanismele folosind controlere de viteză prin PWM, există un ghid pentru C ++ și Java aici, iar pentru LabVIEW, aici.
- Dacă utilizați controlere de viteză peste CAN, trebuie fie să urmați ghidul de aici pentru a le trata ca controlere de viteză PWM, fie să utilizați API-ul Phoenix, a cărui documentație este legată aici.

7. Autonom

- Un ghid pentru modul de a face acțiuni autonome în programarea Java și C ++ poate fi găsit aici.
- Echipa 1619 a compilat, de asemenea, un cod simplu pentru a traversa linia auto în Java, care poate fi găsit aici.
- Șabloanele LabVIEW includ cod autonom pentru jiggling robotul în poziție. Puteți modifica valorile puterii motorului și sincronizarea pentru a îndeplini multe sarcini. Iată un exemplu de cod autonom 2468 din 2018.

Nivelul doi: arhitectură personalizată și controlul motorului cu buclă închisă

1. Utilizarea unei arhitecturi personalizate

- De multe ori, clasele de roboți disponibile nu sunt suficiente. De exemplu, poate doriți să rulați teleop periodic și autonom secvențial. Dacă acesta este cazul, este probabil timpul să treceți la o arhitectură personalizată.
- O arhitectură personalizată este în esență structurarea întregului cod într-un mod personalizat.
- Câteva exemple de arhitecturi personalizate includ codul lui 1678, care este aici.

  Codul din 1678 se bazează pe codul lui 971 care este aici.

- 254 are, de asemenea, o arhitectură personalizată. Codul lor din 2019 poate fi găsit aici. - 33, 624, și 1986și 2468

2. Controlul PID

- PID Control vă permite să controlați un mecanism bazat pe poziție, mai degrabă decât pe tensiune. Folosind PID, puteți spune unui braț să se întoarcă la 30 de grade, în loc să-i spuneți să emită direct o tensiune. Acest lucru este util în special în autonom. Posibilitatea de a spune unui robot să conducă 5 metri în loc de putere maximă timp de 0,5 secunde permite o repetabilitate îmbunătățită.
- Unele documente utile pentru PID sunt:
- Blogul lui Wesley
- PID CSIM pentru manechine

3. Motion Magic (numai CAN)

- Dacă utilizați un controler de viteză TalonSRX, se recomandă utilizarea MotionMagic pentru controlul mecanismelor, în special ceva precum un braț sau un lift. MotionMagic este în esență o buclă PID de 1KHz care urmează profiluri de mișcare trapezoidale autogenerate. Dacă aceste cuvinte nu au sens, nu vă faceți griji! Consultați cele de mai sus pentru informații despre PIP și aici"___s un document care explică profilurile de mișcare. - Documentația pentru Motion Magic se află ei__e.

Nivelul trei: Căi de acționare avansate, control MP și testare unitară

1. Conduceți căile și urmați-le

- Uneori, PID brut nu este suficient pentru a controla trenul de rulare în mod autonom. De exemplu, ați putea dori ca robotul să ocolească comutatorul și să ridice un cub din spate. O modalitate curată de a face acest lucru ar fi crearea unei căi de acționare. O cale de rulare este, în esență, un set de puncte pe care bucla PID a trenului de rulare le va urma, iar punctele vor duce la obiectivul final.

PathWeaver este un instrument grafic care utilizează o bibliotecă numită PathFinder care generează astfel de căi și le salvează într-un fișier analizabil. Instrucțiuni detaliate despre utilizarea PathWeaver se găsesc ei e.

- Odată ce punctele au fost generate, există o varietate de moduri de a le urma. Acestea variază de la utilizarea PID pentru a urmări direct punctele, până la adăugarea unui algoritm de urmărire a căii pentru a procesa punctele înainte de a le da buclei PID. Un exemplu de astfel de algoritm de urmărire a căii poate fi găsit ei e (EQN 5.12). Alte abordări populare pentru urmărirea căii includ Contro Adaptive Pure Pursuit l. 254 are o implementare la îndemână a urmăririi pure adaptive, care poate fi găsită ei e.

2. Control bazat pe model

- Controlul bazat pe model este un pas dincolo de PID. Acesta permite păstrarea unui model matematic al sistemului în cod și actualizarea modelului cu datele senzorului. Folosind un astfel de model, se poate controla mult mai precis poziția, viteza, accelerația unui mecanism etc. Unele echipe care utilizează controlul bazat pe model includ 1678 și 971.
- Resursele utile pentru învățarea controlului bazat pe modele sunt:
- Blogul lui Wesley
- Această fișă MIT

3. Testarea unitară

- De multe ori, doriți să testați codul înainte de a-l implementa pe robot. Acest lucru poate preveni dezastrul. Testarea unitară este un termen pentru testarea porțiunilor codului ca programe independente. De exemplu, poate doriți să testați porțiunea de cod care rulează ascensorul, dar nu și partea care face ca câteva lumini să clipească. Mecanisme de testare pentru *PRIM*

  Competiția robotică este mult îmbunătățită cu controlul bazat pe model, deoarece modelul poate fi folosit ca o simulare a mecanismului, ceea ce înseamnă că întregul mecanism poate fi testat cu o robustețe incredibilă. Unele biblioteci utile de testare a unităților includ:
- GoogleTest

Despre The Compass Alliance

Compass Alliance a fost fondată de 10 echipe din întreaga lume, cu misiunea de a ajuta echipele FIRST Robotics Competition să susțină și să crească. Un depozit de resurse în creștere și un Call Center 24/7 oferă oricui, de orice nivel de calificare, instrumentele necesare pentru a învăța ceva nou sau pentru a învăța mai multe de oriunde din lume. Echipele la distanță care nu au mentori se pot înscrie pentru o echipă de etichete care să fie ghidul lor de la distanță pe tot parcursul sezonului, iar centrele de ajutor identifică unde să obțină acces la serviciile locale oferite de alte echipe FIRST. Hear For You oferă resursele și instrumentele necesare pentru a ajuta echipele și voluntarii să dezvolte bunăstarea mentală în echipele lor și la evenimente. Puteți afla mai multe despre The Compass Alliance, puteți găsi asistență de calitate și vă puteți implica la www.thecompassalliance.org

Despre această resursă

Această resursă a fost pregătită de The Compass Alliance, cu sprijinul și prezentarea generală a FIRST. Dacă aveți întrebări despre această resursă, vă rugăm să contactați thecompassalliance@gmail.com sau firstroboticscompetition@firstinspires.org.

Istoricul reviziilor

Revizie #	Data revizuirii	Note de revizuire
1.0	Decembrie 2018	Versiunea inițială