

Programmeerimine 101

See ressurss hõlmab FIRST® robotikavõistluse programmeerimise põhitõdesid. See hõlmab C++, Java/Kotlini ja LabVIEW-d.

Esimene tase: roboti tööle panemine

1. Programmeerimiskeele valimine: Java, C++ või LabVIEW

- **Java** on tekstikeel, mida tavaliselt õpetatakse keskkoolides ja mida kasutatakse AP CS eksamil. See on "turvaline" keel, mis töötab oma virtuaalses keskkonnas. Kuigi see ei mõjuta üldiselt FIRST Robotics Competitioni, tähendab see virtuaalne keskkond, tuntud ka kui JVM, seda, et Java-programmid on arvutusmahukate ülesannete täitmisel märgatavalt aeglasemad kui kompileeritud keeled. Java valitakse sageli selle kasutusmugavuse ja ristühilduvuse tõttu. Java-d kasutavad meeskonnad on [254,125,503,4911](#) [ja 1241](#).
- **C++** on kiire tekstiline programmeerimiskeel. Seda kasutatakse tööstuses reaajas süsteemide jaoks selle võimsuse ja tõhususe tõttu, kuid õppimiskõver on palju järsem kui Java. C++ arenes välja C-programmeerimiskeelest ning ajalooliste ja tänapäevaste funktsioonide segu põhjustab mõnikord segadust tekitavat süntaksit ja/või ootamatut käitumist. FIRST Robotics Competitioni meeskonnad kasutavad seda peamiselt selle kiiruse, paindlikkuse ja ulatuslike matemaatiliste raamatukogude tõttu. C++ kasutavate meeskondade hulka kuuluvad [971](#) ja [1678](#).
- **LabVIEW** on graafilise andmevoo programmeerimiskeel, mille on välja töötanud National Instruments. Inseneridele ja tehnikutele kasutatavad instrumendid (NI). LEGO WeDo ja FIRST LEGO League Jr ja FIRST LEGO League jaoks kasutatavad Mindstorms keeled on LabVIEW tuletised; Nii et nendest programmidest tulevad õpilased võivad selle tuttavaks pidada. LabVIEW diagrammil on väga lihtne kasutada täiustatud arvutusfunktsioone, näiteks koodilõikude paralleelset käivitamist. Kuigi sellised funktsioonid on võimsad, toovad need sageli kaasa uusi probleeme, millega tuleb tegeleda. NI pakub siiski ulatuslikke silumistööriistu. LabVIEW keskkonnal ja keelel on oma õppimiskõver ja ainulaadsed väljakutsed. FIRST Robotics

Competitioni meeskonnad kasutavad seda peamiselt tänu lihtsustatud graafilisele süntaksile ja ulatuslikele inseneriteekidele. LabVIEW-d kasutavad meeskonnad on [33](#), [359](#), [624](#), [1986](#) ja [2468](#).

- **Keele valik sõltub alati sellest, mis on teie meeskonna jaoks kõige lihtsam.** Näiteks võib sageli olla mõttekas keelt valida, sest programmeerimismentor on selle keele ekspert. Teisest küljest võib olla mõttekas valida antud keele õppimise lihtsuse põhjal. Olenemata sellest, milline on teie otsus, pidage meeles, et programmeerimiskeele valik on teie meeskonna töökeskkonna ja inimeste jaoks spetsiifiline. Kõik keeled on võimekad, hästi toetatud ja piisavalt võimsad FIRST Robotics Competitioni kasutamiseks.

2. Programmeerimiskeele õpetamine

- Programmeerimise õpetamisel ESIMENE Robotikavõistlusel on kaks erinevat ainet, mida tuleb õpetada. Esimene on programmeerimiskeele enda semantika ja süntaks ning teine on liidestamine ESIMENE Robotika võistluse komponendid. C++ keele õppimise juhend on leitav [Siin](#) Java [Tema](#)___e, ja LabVIEW [Siin](#).

3. Stardikoodi valimine

- Java ja C++ jaoks on neli erinevat "klassi", mida saab robotiga liidestamisel kasutada. Nende klasside võrdluse ja nende tähenduse leiate [siit](#).
- LabVIEW puhul sisaldavad stardimallid ajastatud, iteratiivsete ja kudemis-/katkestamismehhanismide segu. Malle kirjeldatakse [siin](#).

4. Kui teie meeskond on programmeerimiskeele valinud ja kodeerimist alustanud, on FIRST Robotics Competition Docs sait hea ressurss arenduskeskkonna seadistamiseks ja koodi robotisse viimiseks. Need juhendid on hindamatud FIRST Robotics Competitioni programmeerimisel.

- [Alustamine](#)
- [C++\Java](#)
- [LabVIEW](#)

5. Koodi saamine oma robotile!

- Java ja C++
- Kui kasutate gradleRIO-t, tippige VS-koodi lihtsalt ".\gradlew deploy" Konsool ja teie kood on robotil.

- Aga oota! Teie kood ei tee veel midagi. Mõned lihtsad näited draivikoodi kohta leiate [siit](#). Katkendites olev kood kuulub kas Robot.java või Robot.cpp, mis tuleks automaatselt luua koos gradle/Eclipse'i projektiga.
- LabVIEW
- Arendamiseks peaksite käivitama allikast, nagu siin [näidatud](#).
- Kui olete lõpetanud, juurutate ehitatud käivitatava faili, nagu siin [näidatud](#).

6. Mehhanismide kood

- Java ja C++ jaoks saab siit kopeerida ja kleepida lihtsat draivikoodi.
- LabVIEW puhul on lihtne draivikood TeleOp VI mallis.
- Enamikul FIRST Robotics Competitioni robotitel on peale jõuülekande ka muud mehhanismid. See võib olla kõike alates pöörlevast hoorattast kuni pneumaatilise katapuladini. Kõik need mehhanismid peaksid olema juhitud autonoomselt või teleopis. PWM-i kaudu kiiruseregulaatoreid kasutavate mehhanismide juhtimiseks on C++ ja Java juhend [siin](#) ning LabVIEW jaoks [siin](#).
- Kui kasutate kiiruseregulaatoreid CAN-i asemel, peate kas järgima siinset [juhendit](#), et käsitleda neid PWM-kiiruse regulaatoritena, või kasutama Phoenixi API-d, mille dokumentatsioon on lingitud [siin](#).

7. Autonoomne

- Juhendi autonoomsete toimingute tegemiseks Java ja C++ programmeerimisel leiate [siit](#).
- Meeskond 1619 on koostanud ka lihtsa koodi Java automaatse joone ületamiseks, mille leiate [siit](#).
- LabVIEW mallid sisaldavad autonoomset koodi roboti paigal raputamiseks. Paljude ülesannete täitmiseks saate muuta mootori võimsuse väärtusi ja ajastust. [Siin](#) on näide 2468 autonoomsest koodist aastast 2018.

Teine tase: kohandatud arhitektuur ja suletud ahelaga mootori juhtimine

1. Kohandatud arhitektuuri kasutamine

- Sageli ei piisa saadaolevatest robotiklassidest. Näiteks võite soovida käivitada teleopi perioodiliselt ja autonoomselt järjest. Kui see nii on, on tõenäoliselt aeg minna üle kohandatud arhitektuurile.
- Kohandatud arhitektuur struktureerib sisuliselt kogu koodi kohandatud viisil.

- Mõned näited kohandatud arhitektuuridest hõlmavad 1678 koodi, mis on [siin](#).
1678 kood põhineb 971 koodil, mis on [siin](#).
- 254-l on ka kohandatud arhitektuur. Nende 2019. aasta kood on leitav [Siin](#). -
[33](#), [624](#), ja [1986](#) ja 2468

2. PID juhtimine

- PID-juhtimine võimaldab teil juhtida mehhanismi asukoha, mitte pinge alusel. PID-i abil saate käskida käel pöörata 30 kraadi, selle asemel, et käskida tal otse pinget väljastada. See on eriti kasulik autonoomsetes. Võimalus käskida robotil sõita 0,5 sekundi täisvõimsuse asemel 5 meetrit, võimaldab paremat korratavust.
- Mõned kasulikud dokumendid PID-i kohta on järgmised:
- [Wesley ajaveeb](#)
- [CSIM-i PID mannekeenidele](#)

3. Liikumismaagia (ainult CAN)

- Kui kasutate TalonSRX kiiruseregulaatorit, on soovitatav kasutada MotionMagicut mehhanismide, eriti käe või lifti juhtimiseks. MotionMagic on sisuliselt 1 kHz PID-silmus, mis järgib automaatselt genereeritud trapetsikujulisi liikumisprofiile. Kui neil sõnadel pole mõtet, ärge muretsege! Vaadake ülaltoodust teavet PID ja [siin](#) [s](#) liikumisprofiile selgitav dokument. - Motion Magicu dokumentatsioon asub [Tema](#)e.

Kolmas tase: täiustatud sõiduteed, MP-juhtimine ja üksuste testimine

1. Sõiduteed ja nende järgimine

- Mõnikord ei piisa toorest PID-st jõuülekande autonoomseks juhtimiseks. Näiteks võite soovida, et robot läheks ümber lüliti ja võtaks kuubiku tagant üles. Puhas viis selleks oleks sõidutee loomine. Sõidutee on sisuliselt punktide kogum, mida jõuülekande PID-silmus järgib, ja punktid viivad lõpliku eesmärgini. PathWeaver on graafiline tööriist, mis kasutab teeki nimega PathFinder, mis genereerib sellised teed ja salvestab need sõelutavasse faili. Üksikasjalikud juhised PathWeaveri kasutamise kohta leiate [Tema](#) e.
- Kui punktid on genereeritud, on nende jälgimiseks mitmeid viise. Need ulatuvad PID-i kasutamisest punktide otseseks jälgimiseks kuni teejälgimise algoritmi lisamiseni punktide töötlemiseks enne nende PID-silmusesse andmist. Sellise tee jälgimise algoritmi näite võib leida [Tema](#) [e](#) (ekvivalent 5.12). Muud

populaarsed lähenemisviisid teekonna järgimiseks on järgmised [adaptiivne puhas jälitamine contro](#)l. 254-I on mugav adaptiivse puhta jälitamise rakendus, mille võib leida [Tema](#) e.

2. Mudelipõhine juhtimine

- Mudelipõhine juhtimine on samm kaugemale PID-st. See võimaldab hoida koodis süsteemi matemaatilist mudelit ja värskendada mudelit anduri andmetega. Sellise mudeli abil saab palju täpsemalt juhtida mehhanismi asukohta, kiirust, kiirendust jne. Mõned meeskonnad, kes kasutavad mudelipõhist juhtimist, on 1678 ja 971.
- Kasulikud ressursid mudelipõhise juhtimise õppimiseks on järgmised:
- [Wesley ajaveeb](#)
- [See MIT-i jaotusmaterjal](#)

3. Ühiktestimine

- Sageli soovite oma koodi enne robotile juurutamist testida. See võib katastroofi ära hoida. Ühiktestimine on termin koodi osade testimiseks eraldiseisvate programmidenä. Näiteks võite testida koodi seda osa, mis juhib lifti, kuid mitte seda osa, mis paneb mõned tuled vilkuma. Katsemehhanismid *ESIMENE* Robootika võistlust täiustatakse oluliselt mudelipõhine juhtimine, kuna mudelit saab kasutada mehhanismi simulatsioonina, mis tähendab, et kogu mehhanismi saab testida uskumatu vastupidavusega. Mõned kasulikud ühiktestimise teegid on järgmised:
- [Google'i test](#)



Kompassiliidu kohta

Compass Alliance'i asutasid 10 meeskonda kogu maailmast missiooniga aidata FIRST Robotics Competitioni meeskondadel püsida ja kasvada. Kasvav ressursihoidla ja 24/7 kõnekeskus annavad igaühele mis tahes oskustasemega inimestele tööriistad, et õppida midagi uut või õppida rohkem kõikjalt maailmast. Kaugmeeskonnad, kellel puuduvad mentorid, saavad registreeruda Tag Teami, kes on nende kaugteejuht kogu hooaja vältel ja abikeskused määravad täpselt, kust pääseda ligi kohalikele teenustele, mida teised FIRSTi meeskonnad pakuvad. Hear For You pakub ressursse ja tööriistu, mis aitavad meeskondadel ja vabatahtlikel arendada vaimset heaolu oma meeskondades ja üritustel. Lisateavet Compass Alliance'i kohta, kvaliteetset abi leiате ja osalete aadressil www.thecompassalliance.org

Teave selle ressursi kohta

Selle ressursi koostas The Compass Alliance FIRSTi toel ja ülevaatel . Kui teil on selle ressursi kohta küsimusi, võtke ühendust thecompassalliance@gmail.com või firstroboticscompetition@firstinspires.org.

Versioonide ajalugu

Läbivaatamine #	Läbivaatamise kuupäev	Revisjoni märkused
1.0	Detsember 2018	Esialgne väljalase