

Lập trình 101

Tài nguyên này sẽ bao gồm những kiến thức cơ bản về lập trình trong Cuộc thi Robot FIRST®. Nó bao gồm C++, Java/Kotlin và LabVIEW.

Cấp độ một: Chạy Robot của bạn

1. Chọn ngôn ngữ lập trình: Java, C++ hoặc LabVIEW

- **Java** là một ngôn ngữ văn bản thường được dạy ở các trường trung học và được sử dụng cho các kỳ thi AP CS. Nó là một ngôn ngữ "an toàn" chạy trong môi trường ảo của riêng nó. Mặc dù nó thường không ảnh hưởng đến ĐẦU TIÊN Robotics Competition, môi trường ảo này, còn được gọi là JVM, có nghĩa là các chương trình Java chậm hơn đáng kể so với các ngôn ngữ biên dịch khi được sử dụng cho các tác vụ tính toán chuyên sâu. Java thường được chọn vì tính dễ sử dụng và khả năng tương thích chéo. Các nhóm sử dụng Java bao gồm [254](#), [125](#), [503](#), [4911](#), và [1241](#).
- **C++** là một ngôn ngữ lập trình văn bản nhanh. Nó được sử dụng trong công nghiệp cho các hệ thống thời gian thực vì sức mạnh và hiệu quả của nó, nhưng đường cong học tập dốc hơn nhiều so với Java. C++ phát triển từ ngôn ngữ lập trình C và sự pha trộn của các tính năng lịch sử và hiện đại đôi khi dẫn đến cú pháp khó hiểu và / hoặc các hành vi không mong muốn. Các đội Cuộc thi Robot FIRST chủ yếu sử dụng nó do tốc độ, tính linh hoạt và thư viện toán học phong phú. Các đội sử dụng C++ bao gồm [971](#) và [1678](#).
- **LabVIEW** là một ngôn ngữ lập trình luồng dữ liệu đồ họa được phát triển bởi National Instruments (NI) để sử dụng bởi các kỹ sư và kỹ thuật viên. LEGO WeDo và Ngôn ngữ Mindstorms được sử dụng cho FIRST LEGO League Jr và FIRST LEGO League là dẫn xuất của LabVIEW; vì vậy sinh viên đến từ các chương trình đó có thể thấy quen thuộc. Trong sơ đồ LabVIEW, rất dễ dàng để tận dụng các tính năng tính toán nâng cao, chẳng hạn như chạy song song các đoạn mã. Mặc dù mạnh mẽ, nhưng các tính năng như vậy thường đưa ra các vấn đề mới cần giải quyết. Tuy nhiên, NI cung cấp các công cụ gỡ lỗi mở rộng. Môi trường và ngôn ngữ LabVIEW đi kèm với đường cong học tập riêng và những thách thức riêng. Các đội Cuộc thi Robot FIRST chủ yếu sử dụng nó do cú pháp đồ họa đơn giản hóa và



thư viện kỹ thuật phong phú. Các nhóm sử dụng LabVIEW bao gồm [33](#), [359](#), [624](#), [1986](#) và [2468](#).

- **Việc chọn ngôn ngữ luôn phụ thuộc vào ngôn ngữ dễ dàng nhất cho nhóm của bạn.** Ví dụ, thường có thể có ý nghĩa khi chọn một ngôn ngữ vì một cố vấn lập trình là một chuyên gia về ngôn ngữ đó. Mặt khác, có thể có ý nghĩa khi lựa chọn dựa trên sự dễ dàng học một ngôn ngữ nhất định. Bất kể quyết định của bạn là gì, hãy nhớ rằng việc lựa chọn ngôn ngữ lập trình là cụ thể cho môi trường làm việc và mọi người trong nhóm của bạn. Tất cả các ngôn ngữ có khả năng, được hỗ trợ tốt và đủ mạnh để sử dụng Cuộc thi Robot FIRST.

2. Dạy ngôn ngữ lập trình

- Khi dạy lập trình cho ĐẦU TIÊN Cuộc thi Robot, có hai môn học riêng biệt cần được dạy. Đầu tiên là ngữ nghĩa và cú pháp của chính ngôn ngữ lập trình, và thứ hai là giao tiếp với ĐẦU TIÊN Các thành phần Cuộc thi Robot. Bạn có thể tìm thấy hướng dẫn học ngôn ngữ C++ [Ở đây](#), Java [bà ấy](#), và LabVIEW [Ở đây](#).

3. Chọn mã bắt đầu của bạn

- Đối với Java và C++, có bốn "lớp" khác nhau có thể được sử dụng khi giao tiếp với Robot. So sánh các lớp này và ý nghĩa của chúng có thể được tìm thấy [ở đây](#).
- Đối với LabVIEW, các mẫu bắt đầu bao gồm hỗn hợp các cơ chế hẹn giờ, lập đi lập lại và sinh ra/hủy bỏ. Các bản mẫu được mô tả ở [đây](#).

4. Khi nhóm của bạn đã chọn một ngôn ngữ lập trình và bắt đầu viết mã, trang web Tài liệu Cuộc thi Robot FIRST là một nguồn tài nguyên tốt về cách thiết lập môi trường phát triển và lấy mã vào rô-bốt của bạn. Những hướng dẫn này là vô giá đối với chương trình Cuộc thi Robot FIRST.

- [Bắt đầu](#)
- [C++\Java](#)
- [LabVIEW](#)

5. Nhận mã vào rô bốt của bạn!

- Java và C++
- Nếu sử dụng gradleRIO, chỉ cần gõ `./gradlew deploy` vào VS Code Bảng điều khiển và mã của bạn sẽ có trên robot.



- Nhưng chờ đã! Mã của bạn chưa làm gì cả. Một số ví dụ đơn giản cho mã ổ đĩa có thể được tìm thấy [tại đây](#). Mã trong đoạn mã thuộc về Robot.java hoặc Robot.cpp, mã này sẽ được tạo tự động bằng dự án gradle/Eclipse.
- LabVIEW
- Để phát triển, bạn nên chạy từ nguồn như được hiển thị [ở đây](#).
- Sau khi hoàn tất, bạn sẽ triển khai một tệp thực thi đã xây dựng như được hiển thị ở [đây](#).

6. Quy tắc cơ chế

- Đối với Java và C++, mã ổ đĩa đơn giản có thể được sao chép-dán từ [đây](#).
- Đối với LabVIEW, mã ổ đĩa đơn giản nằm trong mẫu trong TeleOp VI.
- Hầu hết các robot FIRST Robotics Competition đều có các cơ chế hoạt động khác ngoài hệ thống truyền động. Đây có thể là bất cứ thứ gì từ bánh đà quay đến máy bắn đá khí nén. Tất cả các cơ chế này phải có thể điều khiển được trong chế độ tự động hoặc trong teleop. Để kiểm soát các cơ chế sử dụng bộ điều khiển tốc độ qua PWM, có hướng dẫn cho C++ và Java [tại đây](#) và cho LabVIEW, [tại đây](#).
- Nếu sử dụng bộ điều khiển tốc độ qua CAN, bạn phải làm theo hướng dẫn [tại đây](#) để coi chúng là bộ điều khiển tốc độ PWM hoặc sử dụng API Phoenix, có tài liệu được liên kết [tại đây](#).

7. Tự trị

- Hướng dẫn về cách thực hiện các hành động tự động trong lập trình Java và C++ có thể được tìm thấy [tại đây](#).
- Đội 1619 cũng đã biên soạn một số mã đơn giản để vượt qua dòng auto trong Java, có thể tìm thấy [ở đây](#).
- Các mẫu LabVIEW bao gồm mã tự động để lắc rô-bốt tại chỗ. Bạn có thể sửa đổi các giá trị công suất động cơ và thời gian để hoàn thành nhiều nhiệm vụ. [Dưới đây](#) là một ví dụ về mã tự trị của 2468 từ năm 2018.

Cấp độ hai: Kiến trúc tùy chỉnh và điều khiển động cơ vòng kín

1. Sử dụng kiến trúc tùy chỉnh

- Nhiều khi, các lớp robot có sẵn là không đủ. Ví dụ: bạn có thể muốn chạy teleop định kỳ và tự động tuần tự. Nếu đúng như vậy, có thể đã đến lúc chuyển sang kiến trúc tùy chỉnh.

- Một kiến trúc tùy chỉnh về cơ bản là cấu trúc tất cả các mã theo cách tùy chỉnh.
- Một số ví dụ về kiến trúc tùy chỉnh bao gồm mã của 1678 ở [đây](#).
Mã của 1678 được xây dựng dựa trên mã của 971 ở [đây](#).
- 254 cũng có một kiến trúc tùy chỉnh. Mã 2019 của họ có thể được tìm thấy ở [đây](#). - [33](#), [624](#), và [1986](#) và 2468

2. Điều khiển PID

- Điều khiển PID cho phép bạn điều khiển một cơ chế dựa trên vị trí, thay vì điện áp. Sử dụng PID, bạn có thể yêu cầu cánh tay quay sang 30 độ, thay vì yêu cầu nó xuất điện áp trực tiếp. Điều này đặc biệt hữu ích trong chế độ tự động. Có thể yêu cầu rô-bốt lái xe 5 mét thay vì hết công suất trong 0,5 giây cho phép tăng cường độ lặp lại.
- Một số tài liệu hữu ích cho PID là:
 - [Blog của Wesley](#)
 - [PID của CSIM dành cho hình nộm](#)

3. Ma thuật chuyển động (Chỉ có CAN)

- Nếu sử dụng bộ điều khiển tốc độ TalonSRX, bạn nên sử dụng MotionMagic để điều khiển các cơ chế, đặc biệt là một cái gì đó như cánh tay hoặc thang máy. MotionMagic về cơ bản là một vòng lặp PID 1KHz theo cấu hình chuyển động hình thang được tạo tự động. Nếu những lời đó không có ý nghĩa, đừng lo lắng! Xem ở trên để biết thông tin về PID và [ở đây](#) s một tài liệu giải thích hồ sơ chuyển động.
 - Tài liệu cho Motion Magic được đặt [bà ấy](#) e.

Cấp độ ba: Đường dẫn truyền động nâng cao, điều khiển MP và kiểm tra đơn vị

1. Đường lái xe và đi theo chúng

- Đôi khi PID thô không đủ để điều khiển hệ thống truyền động một cách tự động. Ví dụ: bạn có thể muốn robot đi vòng quanh công tắc và nhặt một khối lập phương từ phía sau. Một cách rõ ràng để làm điều này là tạo ra một đường lái xe. Đường dẫn về cơ bản là một tập hợp các điểm mà vòng lặp PID của hệ thống truyền động sẽ đi theo và các điểm sẽ dẫn đến mục tiêu cuối cùng. PathWeaver là một công cụ đồ họa sử dụng một thư viện có tên là PathFinder tạo ra các đường dẫn như vậy và lưu chúng vào một tệp có thể phân tích được. Hướng dẫn chi tiết về cách sử dụng PathWeaver được tìm thấy [bà ấy](#) e.
- Khi các điểm đã được tạo, có nhiều cách để theo dõi chúng. Chúng bao gồm từ việc sử dụng PID để theo dõi trực tiếp các điểm, đến thêm thuật toán theo đường dẫn để xử lý các điểm trước khi đưa chúng vào vòng lặp PID. Một ví dụ về thuật

toán theo đường dẫn như vậy có thể được tìm thấy [bà ấy_e](#) (Eqn 5.12). Các cách tiếp cận phổ biến khác để theo dõi đường dẫn bao gồm [Nhược điểm theo đuổi thuần khiết thích ứng](#) [l.254](#) có một triển khai hữu ích của việc theo đuổi thuần túy thích ứng có thể được tìm thấy [bà ấy_e](#).

2. Điều khiển dựa trên mô hình

- Điều khiển dựa trên mô hình là một bước vượt ra ngoài PID. Nó cho phép giữ một mô hình toán học của hệ thống trong mã và cập nhật mô hình với dữ liệu cảm biến. Sử dụng một mô hình như vậy, người ta có thể kiểm soát vị trí, vận tốc, gia tốc, v.v. của cơ chế chính xác hơn nhiều. Một số nhóm sử dụng kiểm soát dựa trên mô hình bao gồm 1678 và 971.
- Các tài nguyên hữu ích để học kiểm soát dựa trên mô hình là:
- [Blog của Wesley](#)
- [Tài liệu phát tay MIT này](#)

3. Kiểm tra đơn vị

- Thông thường, bạn muốn kiểm tra mã của mình trước khi triển khai nó trên robot. Điều này có thể ngăn chặn thảm họa. Unit-testing là một thuật ngữ để kiểm tra các phần của mã dưới dạng các chương trình độc lập. Ví dụ: bạn có thể muốn kiểm tra phần mã chạy thang máy, nhưng không phải phần làm cho một vài đèn nhấp nháy. Cơ chế thử nghiệm cho **ĐẦU TIÊN** Cuộc thi Robotics được nâng cao đáng kể với điều khiển dựa trên mô hình, vì mô hình có thể được sử dụng như một mô phỏng của cơ chế, có nghĩa là toàn bộ cơ chế có thể được kiểm tra với độ bền đáng kinh ngạc. Một số thư viện kiểm thử đơn vị hữu ích bao gồm:
- [Kiểm tra GoogleTest](#)



Giới thiệu về The Compass Alliance

Liên minh Compass được thành lập bởi 10 đội từ khắp nơi trên thế giới với sứ mệnh giúp các đội Cuộc thi Robot FIRST duy trì và phát triển. Kho lưu trữ tài nguyên đang phát triển và Trung tâm cuộc gọi 24/7 cung cấp cho bất kỳ ai ở bất kỳ trình độ kỹ năng nào các công cụ để học điều gì đó mới hoặc học hỏi nhiều hơn từ mọi nơi trên thế giới. Các đội từ xa thiếu cố vấn có thể đăng ký Tag Team để trở thành hướng dẫn viên từ xa của họ trong suốt mùa giải và Trung tâm trợ giúp xác định nơi có quyền truy cập vào các dịch vụ địa phương mà các nhóm FIRST khác cung cấp. Hear For You cung cấp các tài nguyên và công cụ để giúp các nhóm và tình nguyện viên phát triển sức khỏe tinh thần trong nhóm của họ và tại các sự kiện. Bạn có thể tìm hiểu thêm về The Compass Alliance, tìm hỗ trợ chất lượng và tham gia tại www.thecompassalliance.org

Giới thiệu về tài nguyên này

Tài nguyên này được chuẩn bị bởi The Compass Alliance, với sự hỗ trợ và tổng quan của FIRST. Nếu bạn có thắc mắc về tài nguyên này, vui lòng liên hệ với thecompassalliance@gmail.com hoặc firstroboticscompetition@firstinspires.org.

Lịch sử sửa đổi

| Phiên bản # | Ngày sửa đổi | Ghi chú sửa đổi |
|-------------|---------------------|-----------------------|
| 1.0 | Tháng Mười Hai 2018 | Bản phát hành ban đầu |
| | | |
| | | |
| | | |

