编程 101

该资源将涵盖 FIRST® 机器人竞赛中的编程基础知识。它涵盖了 C++、Java/Kotlin 和 LabVIEW。

第一关：让你的机器人启动并运行

1. 选择编程语言：***Java***、***C++*** 或 ***LabVIEW***

Java 是一种文本语言，在高中被广泛教授，并在 AP CS 考试中使用。它是一种“安全”语言，在自己的虚拟环境中运行。虽然它不会影响FIRST机器人竞赛的整体效果，但这种虚拟环境（也称为JVM）意味着在用于计算密集型任务时，Java程序明显比编译语言慢。Java 通常因其易用性和交叉兼容性而被选中。使用 Java 的团队包括 254、125、503、4911 和 1241。

C++ 是一种快速的文本编程语言。由于其功能和效率，它在工业上被用于实时系统，但学习曲线比 Java 陡峭得多。C++ 是从 C 编程语言演变而来的，历史和现代特性的混合有时会导致令人困惑的语法和/或意外行为。FIRST机器人竞赛团队使用这种语言主要是因为它的速度、灵活性和广泛的数学库。使用 C++ 的团队包括 971 和 1678 。

LabVIEW是 National开发的一种图形数据流编程语言。供工程师和技术人员使用的仪器（NI）。用于 LEGO WeDo 和 FIRST LEGO League Jr 和 FIRST LEGO League 的 Mindstorms 语言是 LabVIEW 的衍生产品;因此，这些程序的学生可能会发现LabVIEW很熟悉。在LabVIEW图表中，利用高级计算功能非常容易，例如并行运行代码片段。虽然功能强大，但这些功能往往会带来新的问题。但是，NI提供了广泛的调试工具。LabVIEW环境和语言有其自身的学习曲线和独特的挑战。FIRST机器人竞赛团队主要使用这种语言，因为它具有简化的图形语法和广泛的工程库。使用LabVIEW的团队包括 33、359、624、1986 和 2468。

你选择哪种语言总是取决于对你的团队来说最简单的语言。例如，选择一种语言通常是有意义的，因为编程顾问是该语言的专家。另一方面，根据学习特定语言的难易程度进行选择也可能是有意义的。无论您做出什么决定，请记住，编程语言的选择是特定于工作环境和团队中的人员的。所有的语言都具有功能，得到良好的支持和强大的功能，足以在FIRST机器人竞赛中使用。

2. 教授编程语言

- 在教授FIRST机器人竞赛的编程时，需要教授两个不同的主题。第一个是编程语言自身的语义和语法，第二个是与FIRST Robotics Competition的组件的接口。学习 C++ 语言的指南 因此，转到 Java 因此 和 LabVIEW 因此 达到。

3. 选择您的起始代码

- 对于 Java 和 C++，在与 Bot 交互时可以使用四个不同的“类”。可以在此处找到这些类的比较以及它们的含义 。
- 对于LabVIEW，入门模板包括计划、迭代和生成/取消机制的混合。此处介绍了这些模板。

4. 一旦您的团队选择了一种编程语言并开始编码，***FIRST Robotics Competition Docs*** 网站就是一个很好的资源，教你如何设置你的开发环境并将代码上传到你的机器人。这些指南对于***FIRST***机器人竞赛的编程非常宝贵。

- 开始
- C++\Java
- LabVIEW的

5. 将代码上传到您的机器人！

- Java 和 C++
  如果您使用的是 -gradleRIO，只需在 VS Code 中键入“./gradlew deploy”。控制台和您的代码将位于机器人上。

  - 但是等等！你的代码还没有做任何事情。可以在此处找到一些驱动程序代码的简单示例。snippets 中的代码属于 Robot.java 或 Robot.cpp，必须使用 gradle/Eclipse 项目自动生成。

- LabVIEW的
  - 对于开发， 您必须从源代码运行它 ，如此处所示。
  - 完成后， 您将部署一个已创建的可执行文件，如此处所示 。

6. 机构代码

- 对于 Java 和 C++， 可以在此处复制和粘贴简单的驱动程序代码。
- 对于LabVIEW，TeleOp VI的模板中提供了简单的驱动程序代码。
- 大多数FIRST Robotics Competition机器人的机构都在传动系统外部驱动。这可以是从旋转飞轮到气动弹射器的任何东西。所有这些机制都必须能够自主或远程控制。要通过PWM使用速度控制器控制机制， 这里有一个针对C++和Java的指南，这里是LabVIEW的指南。
- 如果您通过CAN使用速度控制器，则应按照此处的指南进行操作，或使用Phoenix API（其文档 链接在此处）将它们视为PWM速度控制器。

7. 自主性

- 有关如何在 Java 和 C++ 编程中执行自主操作的指南 可以 在此处找到。
- Team 1619 还编译了一些用于在 Java 中交叉自动线的简单代码， 你可以在这里找到 。
- LabVIEW模板包含自主代码，用于将机器人摇晃到位。您可以更改发动机功率值和计时以执行许多任务。这是 2018 年自主代码 2468 的示例。

第二层：定制架构和闭环电机控制

1. 使用自定义架构

- 通常，现有的机器人类别是不够的。例如，您可能希望定期进行远程操作并按顺序自主运行。如果是这样的话，可能是时候转向自定义架构了。
- 自定义架构本质上是以自定义方式配置所有代码。
- 特定架构的一些示例包括 1678 的代码，此处包含该代码。1678 的代码基于此处的 971 代码。
- 254 还具有特殊的架构。他们的 2019 年代码 可以在这里找到 。
- 33、624、1986和2468

*2. PID*控制

- PID 控制允许您根据位置而不是电压来控制机构。使用 PID，您可以告诉手臂旋转 30 度，而不是直接告诉它输出电压。这在自治系统中特别有用。能够告诉机器人行驶 5 米而不是全功率 0.5 秒，可以提高可重复性。
- 以下是一些用于 PID 的有用文档：
    - 卫斯理的博客
    - CSIM的傻瓜PID

3. Motion Magic（仅限 *CAN*）

- 如果您使用的是 TalonSRX 速度控制器，建议您使用 MotionMagic，尤其是用于控制杠杆或升降机等机构。MotionMagic 本质上是一个 1KHz PID 环路，它遵循自动生成的梯形运动曲线。如果这些话没有意义，请不要担心！有关 PID 和 这里 有一个文档描述了事务配置文件。

- 可以在此处找到 Motion Magic 的文档 。

第三级：高级驾驶路径、**MP** 控制和单元测试

1. 驾驶道路并遵循它们

- 有时，原始 PID 不足以自主控制传动系统。例如，您可能希望机器人绕过开关并从后面拿起一个立方体。做到这一点的一个巧妙方法是创建一条驾驶路径。驱动路径本质上是传动系统 PID 回路将遵循的一系列点，这些点将通向最终目的地。PathWeaver 是一个图形工具，它使用一个名为 PathFinder 的库来创建此类路径并将它们保存在可解析文件中。有关如何使用 PathWeaver 的详细说明 ，请参阅此处。

- 创建积分后，有多种方法可以跟踪它们。这些范围从直接使用 PID 跟踪点到添加路径跟踪算法以在将点提供给 PID 循环之前处理点。这种路径跟踪算法的一个例子可以在这里找到（式5.12）。其他流行的道路跟踪方法包括 自适应纯跟踪控制。254 上提供了方便的自适应纯跟踪实现，可以在这里找到。

2. 基于模型的控制

- 基于模型的控制超越了 PID。它允许将系统的数学模型保留在代码中，并使用传感器数据更新模型。使用这样的模型，可以更精确地控制机构的位置、速度、加速度等。一些使用基于模型的控制的团队包括 1678 和 971。
- 用于学习基于模型的控制的有用资源包括：
  - 卫斯理的博客
  - 这本麻省理工学院的小册子

3. 单元测试

- 大多数情况下，您希望在将代码部署到机器人之前对其进行测试。这可以防止一场灾难。单元测试是一个术语，用于将代码部分作为独立程序进行测试。例如，您可能想要测试使电梯正常工作的代码部分，而不是使几盏灯闪烁的部分。基于模型的控制大大增强了FIRST机器人大赛的机构测试，因为模型可以用作机构的模拟，这意味着可以以令人难以置信的鲁棒性测试整个机构。一些有用的单元测试库包括：
- Google测试

关于指南针联盟

Compass Alliance 由来自世界各地的 10 支团队创立，旨在帮助 FIRST 机器人大赛团队生存和发展。不断增长的资源库和 24/7 呼叫中心为任何技能水平的任何人提供了从世界任何地方学习新事物或了解更多信息的工具。没有导师的远程团队可以注册一个 Tag 团队，该团队将在整个赛季中进行远程指导，帮助中心决定在哪里访问其他 FIRST 团队提供的本地服务。Hear For You 提供资源和工具，帮助团队和志愿者在团队和活动中改善他们的心理健康。您可以了解有关 The Compass Alliance 的更多信息，找到高质量的帮助，并 参与 www.thecompassalliance.org。

关于此资源

该资源由 Compass Alliance 在 FIRST 的支持和概述下制作。如果您对此资源有疑问，请联系 thecompassalliance@gmail.com 或 firstroboticscompetition@firstinspires.org。

修订记录

校订#	修订日期	修订说明
1.0	破。2018	初始版本