Programming a Robot Using C++

Patrick Fairbank

January 8, 2011

Patrick Fairbank

- 10 years of *FIRST* experience
- Mentor for Team 296, 2004 – 2006
  - 2006 World Champions
- Mentor for Team 1503, 2007 – Present
  - 2 regional finalists
  - 2010 Woodie Flowers Finalist Award at the Waterloo Regional
- University of Waterloo undergrad student
  - Mechatronics Engineering, Class of 2011
  - Co-ops: General Motors, Sony, Google, PCAS

FIRST

Overview

- Introduction to robot programming
- Why choose C++?
- Programming basics
- Coding demonstration

FIRST

Programming in FRC

- The robot has mechanical systems and electrical hardware, but needs a program to tell it what to do
- The program collects inputs from the drivers and sensors, and uses them to decide what motor output should be
- Different programming “languages”:
  - LabVIEW
  - C++
  - Java

FIRST

Programming Languages

- LabVIEW
  - Developed by National Instruments
  - Used in laboratory instrumentation, industrial control applications
  - Graphical: programs written by placing blocks and dragging wires between them

FIRST

Programming Languages

- C++
  - Developed in the 1980's as an extension to C
  - Object-oriented programming language
  - Used everywhere – from operating systems to applications to web services
- Java
  - Another object-oriented programming language, used as widely as C++
  - Often used as a teaching language

FIRST

Why C++/Java over LabVIEW?

- LabVIEW isn't any simpler to learn
- C++ or Java help is easy to find
- It takes a lot longer to build and deploy LabVIEW programs vs. C++ or Java
- C++/Java skills are more useful for students to learn
- IDEs for C++/Java are lighter-weight

FIRST

Why C++ over Java?

- My team's reasons:
  - Currently, better support for C++
  - More teams use C++ than Java
- However, you may have other reasons to want to use Java
- If so, most of the concepts in this presentation will still be relevant

FIRST

C++ Basics

FIRST

Variables

- Used to store information (data)
- Different types: e.g. int (integer), bool (true or false) float (decimal number)
- Can also create custom types (e.g. classes – discussed later)

```
int myVar;
myVar = 5;

float myOtherVar = 15.03;
```

FIRST

Comments

- Sections of text ignored by the robot
- Used to illustrate and explain things in plain English to people looking at the code

```
int sensors; // Number of sensors.

/* This next block of code gets the
   value of the joystick Y-axis */
Joystick* stick = new Joystick(1);
float value = stick.GetY();
```

FIRST

Conditionals

- Used to make decisions in programs
- Comparisons using variables and numbers are made

```
if (myVar > 5) {
// do something
}
else if (myVar < 2) {
// do something else
}
else {
// do another thing
}
```

FIRST

Classes

- Representation of physical “things” in a program
- Used like custom variable types
- Examples: Joystick, Victor, Gyro, Relay

```
Victor* theMotor;
Joystick* stick;
Gyro* gyro;
```

FIRST

Functions

- Represent individual tasks
- Used to do things or get information

```
int Add(int a, int b) {
  return a + b;
}
int sum = Add(723, 780);

void StartMotor() {
  motor.Set(0.5);
}
```

FIRST

Files

- C++ has two different types of files
- Header (.h) files summarize the structure of classes
- Code (.cpp) files contain actual code
- By convention, each class has a .h file and a .cpp file
- Example: class Robot has Robot.h and Robot.cpp

FIRST

WPILib

- Already-written code provided by FIRST to make robot programming easier
- Consists of classes that represent all common robot hardware
- Example: Compressor, DigitalInput, DriverStation, Solenoid, Accelerometer

FIRST

Wind River Workbench

- The Windows program used to write robot programs and download them to the robot

FIRST