FIRST Tech (FTC) Robotics: New Programming Platform Workshop

FTC Team 9901
Techie Titans

Aug 14, 2016
We are an FTC team with 7 team members, from grade 7 to 11 – from 6 different schools

- John Champe High school (Chantilly)
- Thomas Jefferson High School
- Lunsford Middle school (Chantilly)
- Mercer Middle school (Chantilly)
- Eagle Ridge Middle school (Ashburn)
- Frost Middle School (Fairfax)

We are part of Nova-Labs Robotics, a non-profit and well-known Maker-Space in this area
Achievements last season:

➔ Qualified to East Super-Regional Championship
➔ Qualified to VA and MD State Tournament
➔ Awards -
  ◆ Inspire Award 2nd
  ◆ Winning Alliance
  ◆ Control Award
  ◆ Think Award
  ◆ Innovate Award 2nd
  ◆ Connect Award 3rd
FTC Team 9901

About us - team members

Jasmine, 11th
Sonika, 9th
Vinay, 8th
Ridha, 8th
Faraaz, 8th
Prashanth, 7th
Arsh, 7th
FTC Team 9901

About us - Coaches and Mentors

Coaches

Hossain (Tauhid) Rahman
https://www.linkedin.com/in/tauhid-hossain-rahman-pmp-pmi-acp-csm/

Farzana Afrin
https://www.linkedin.com/in/farzana-afrin

M. Ruhul Chowdhury
https://www.linkedin.com/in/ruhul-chowdhury

Jagan Manickam
https://www.linkedin.com/in/jagan-manickam

Mentors
Agenda

Part 1:
➔ Brief overview of New platform
➔ Software development Environment
  ◆ ZTE Speed Phone
  ◆ FTC Apps (Driver Station and Robot Controller)
  ◆ Android Studio
  ◆ FTC SDK
  ◆ Event driven and linear programming model
➔ Example Op Modes
➔ Build, Deployment and Drive!!
➔ Test and Debug
➔ Tele Op
Part 2:

- Resources and helpful links
  - Starting a new team
  - How to Get Organized
- Award categories
- Engineering Notebook requirements
- Lesson learned
- Q&A
Non Goals

➔ App Inventor
➔ Compatibility/reusability with legacy HW
➔ Advanced Topics (If time permits)
  ◆ Parallel threads
  ◆ Motor Calibration, Stalling
  ◆ Autonomous techniques (i.e. Line tracker, IR Beacon follower)
Attendee Poll

- Experience: Rookie Team? 1-2 years? More than 2 year?
- Received the Kit? Tetrax? Matrix?
- Installed Android Studio? Built an app?
- Installed the FTC App?
- Ran a OpMode?
- Wrote and tested an OpMode?
Part 1

FTC New Programming Platform
New Platform

Overview

➔ Based on the Android OS and Java
➔ Uses two phones/tablets: Robot Controller and Driver’s Station
➔ Devices are connected over USB
FTC Hardware

First learning setup -
Our last season’s robot run:

https://www.youtube.com/watch?v=PBJYaj82Op0

https://www.youtube.com/watch?v=WML5Yn8Rv80
There are two (3) allowed Android devices that Teams will use to control their Robot:

- ZTE Speed.
- Motorola Moto G (3rd Generation)
- Google Nexus 5
Download Driver and controller App from Google app store
FTC Apps

Driver Station

- Closed Source
- No code is deployed
- Front end app:
  - Program(OpMode) selection, Start and Stop
  - Gamepad is connected via micro USB
  - Telemetry (message from robot) is displayed
- Setup for Wifi communication to Robot Controller
FTC Apps

Robot Controller

➔ Partial Open Source
➔ Part of Robot Assembly:
  ◆ Connects to Power module/USB Hub
  ◆ Integrates and executes programs (OpModes)
  ◆ Broadcasts Telemetry messages to Driver station
➔ Setup and configuration of HW (i.e. Motors, sensors).
Early summer we spent some time learning Java Basics and Android Studio development environment.

Our programming coach developed a lesson plan for the team members to follow.

Leveraged online resources. We liked the following:

- [http://stackoverflow.com/](http://stackoverflow.com/)

2 projects we did:

- Simple Calculator
- Tic Tac Toe
Android and Java

Android Studio

➔ Team members installed Java and Android Studio using installation instructions in FTC Manual (Page 14 - 21)

➔ We deployed and tested Apps in:

◆ Built in Emulator/Geny Motion
◆ ZTE phone
Android and Java

Our learning: Tic Tac Toe

➔ 3 Team members and Coach came up with 4 different and working solutions!

➔ We learned:
  ◆ Class - its structure and pieces
  ◆ Member Variables. Explore different types of Variables and where they are used.
  ◆ Methods - Its structure and pieces
  ◆ Java Data Types
  ◆ Controls (if, for, while, switch/case etc.)
FTC Software

Basics

➔ Developed by Qualcomm for the FTC program
➔ Published in GitHub: https://github.com/ftctechnh/ftc_app
➔ Includes:
  ◆ Robot Controller Source Code in an Android Studio Project that teams will use to create their own programs (Op Modes)
  ◆ Sample programs (Op Modes)
  ◆ Documentation
FTC Software

Download

Can be downloaded as:
- .zip
- Forked (to your github) and cloned to desktop
FTC Software

GIT Management Model

➔ Set a GROUP in github to manage and share code changes

➔ Kids picked it up quickly!!
FTC Software

Git Management Model

➔ All git(and github) features are in Android Studio
➔ Google did a great job!!
An Op Mode is what teams use in order to create custom behavior for their robot. It is a Java class.

Op Modes are similar to the tele-op and autonomous programs that teams wrote for their LEGO NXT controllers.

During a match, Op Modes are executed on the robot controller, but are selected by the team from the driver station.

Two Types: Event based and Linear
Your code (Op Modes) are integrated in same project with FTC code

Only registered Op Modes will be available to Driver station
FTC Software

Code Separation

➔ Our own namespace/package: `com.techietitans.opmodes`

➔ Easier to manage future updates to FTC software
FTC Software
Event based Op Mode

➔ Inherited from OpModes base class

➔ Loop() method is continuously executed until the program terminates

➔ HW communication (Sensor reads, Motor control) is done at the end of each loop() execution

➔ Useful for program with distinct states and state transition
FTC Software

Event based Op Mode life cycle

Op Mode selected

Start Button click

Stop Button click
FTC Software

Linear Op Mode

➔ Commands are executed sequentially one after the other
➔ Similar to the model used to program a LEGO NXT with a tool like RobotC
➔ Inherited from LinearOpModes base class
➔ Can use blocking statements like `Thread.sleep()`
➔ HW communication (Sensor reads, Motor control) is done on demand, as needed
➔ Useful for Autonomous
```java
public class tt_L_A_DemoBotSensors extends LinearOpMode {

    @Override
    public void runOpMode() throws InterruptedException {
        // Wait for the start button to be pressed
        waitForStart();

        while (opModeIsActive()) {
            // Code goes here
        }
    }

    // Code goes here
}
```
[Code DEMO - Vinay]
HW configuration

Robot Controller App

- **Scan** to auto discover connected controllers
- Select a controller
- Enter name for devices (motor, sensor)
HW configuration

Hardware map in the Op Mode

```java
public class PushBotDriveTouch extends LinearOpMode {
    DcMotor leftMotor;
    DcMotor rightMotor;
    TouchSensor touchSensor;

    @Override
    public void runOpMode() throws InterruptedException {
        // Get references to the motors from the hardware map
        leftMotor = hardwareMap.dcMotor.get("left_drive");
        rightMotor = hardwareMap.dcMotor.get("right_drive");

        // Reverse the right motor
```
Execution Preparation

Robot Controller - Ready

- With our desired configuration file active - we are ready to start

- We can have multiple configuration files, and swap between them as needed

- An error here indicated that there is a mismatch between the file and attached HW
Execution Preparation

**Driver Station - Pair Wifi**

- Initiate Pairing from FTC Driver Station app
- Accept on Robot Controller
Execution Preparation

**Driver Station - Joystick**

- Connect Gamepad/Joysticks to driver station ZTE phone using OTG USB Hub
- Enable Driver 1 with START+A, Driver 2 with START+B
This is the Modern Robotics Core Device Discovery. It is used to test the electronics.
We have located a motor controller on the discovery tool and when you click on it this is what you will see.
We have found the core servo controller so now we will click on it and the screen on the right will show up.
Test and Debug

Calibration

➔ Wheel Rotation
➔ Color Sensor
[DEMO -- Faraaz]
Tele Op

Focus areas

- Navigation - moving around in the field
- Attachment Controls
- Gamepad Assignments
- Execution Strategy
**Tele Op**

**Navigation**

---

### Tank Drive
- Point turning
- Swing turning
- Precise controls on each side of the drive train

### Race car drive
- Less room to mess up going forwards or backwards
- Easy to learn

---

**Left Side of drive train**
- Forwards
- Backwards

**Right side drive train**
- Turning
● We needed the controls to be convenient
● We needed the gamepads to have equal controls.
● But we didn't want synchronization to be required between drivers
● Wanted to avoid toggle controls
Tele Op
Gamepad Assignments

Gamepad1
- Left Tank drive
- Right Tank drive
- Slow rolling wheels
- Zipline

Gamepad2
- Right hooks
- Left hooks
- Bar hook open
- Bar hook closed
- Tape measure
- Tape aim up
- Tape aim down
- Right zipline close
- Right zipline open
- Left zipline close
- Left zipline open
Drive

[ Drive DEMO]
Programming Q&A
Part 2

- Resources and helpful links
  - Starting a new team
  - How to Get Organized
- Engineering Notebook requirements
- Award Categories
- Lesson Learned
* Intellitek training: [http://ftc.edu.intelitek.com/](http://ftc.edu.intelitek.com/) (select course, login as guest)
  - Module 1: HW and Control Systems
  - Module 3: Java Programming with Android Studio

* Game Manual
  - [FTC Game manual part 1 2016 - 2017](http://ftcforum.usfirst.org/forum.php) (released already)
  - FTC Game manual part 2 2016 - 2017 (Coming 9/10/2016)

* Unofficial forums (unmoderated):
  - [https://www.chiefdelphi.com/forums/](https://www.chiefdelphi.com/forums/) [FIRST Tech Challenge]
  - [https://www.reddit.com/r/FTC/](https://www.reddit.com/r/FTC/)


* PushBot build guide:
* Starting a new team

http://www.firstinspires.org/ftc-start-a-team
- Registration
- Mentor Manual
- Team Budget
- Fundraising
- Robot Building Resource (http://www.firstinspires.org/node/5181)
FTC Team 9901

How to Get Organized

http://techietitans-ftc.com (our Website)

https://www.facebook.com/TechieTitans9901 (our Facebook page)

https://www.youtube.com/ (our YouTube channel)

https://github.com/ (code repository - revision control and code management)

https://drive.google.com/drive (document storage in Cloud, for sharing)

https://slack.com (messaging app, for team collaboration)
A complete documentation of the Team’s journey throughout the season.

- Sketches
- Discussions
- Design evolution
- Obstacles
- Team member’s thoughts throughout the journey for the entire season.

- A new notebook should be created for each new season.
- Through explanations and thought processes

All Awards, other than robot game competition, are given based on Engineering Notebook content (and the team presentation/interview).

- Can be either HANDWRITTEN or ELECTRONIC.

- Detailed rules, examples, and templates for documents are in “Engineering Notebook Guidelines”
FTC Team 9901
Award Categories

- Robot Game Award
  Winning Alliance
  Finalist Alliance

- Individual Awards
  Inspire Award
  Think Award
  Connect Award
  Rockwell Collins Innovate Award
  PTC Design Award
  Motivate Award
  Control Award

- Advancement criteria is based on award ranking (see “Game Manual Part 1”)
# FTC Team 9901

## Awards and the Engineering Notebook

<table>
<thead>
<tr>
<th>Engineering Notebook Requirements by Award</th>
<th>Team Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inspire Award</strong></td>
<td>- Business Section</td>
</tr>
<tr>
<td>• Engineering Notebook must be submitted, and must include an Engineering Section, a Team Section and a Business or Strategic Plan. The entire Engineering Notebook must be high quality, thoughtful, thorough, detailed and well organized.</td>
<td>- Engineering Sect</td>
</tr>
<tr>
<td><strong>Think Award</strong></td>
<td>- Eng. Section</td>
</tr>
<tr>
<td>• Engineering Notebook must have an Engineering Section that includes entries describing underlying science, mathematics, and game strategies.</td>
<td>- Design Process</td>
</tr>
<tr>
<td>• Engineering Notebook must demonstrate that the Team has a clear understanding of the engineering design process, with pictures or drawings and details documenting all stages of Robot design.</td>
<td>- Team’s Journey</td>
</tr>
<tr>
<td>• Notebook must recount the Team’s journey, experience and lessons learned throughout the season.</td>
<td></td>
</tr>
</tbody>
</table>
# FTC Team 9901

**Awards and the Engineering Notebook**

<table>
<thead>
<tr>
<th>Award</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect Award</td>
<td>An Engineering Notebook must be submitted and must include a Business or Strategic plan that identifies their future goals and the steps they will take to reach those goals. The plan could include fundraising goals, sustainability goals, timelines, outreach, and community service goals.</td>
</tr>
<tr>
<td>Rockwell Collins Innovate Award</td>
<td><em>Team</em> must submit an Engineering Notebook with an Engineering Section that documents the design process and how the <em>Team</em> arrived at their design solution.</td>
</tr>
<tr>
<td>PTC Design Award</td>
<td><em>Team</em> must submit an Engineering Notebook with an Engineering Section that includes detailed <em>Robot</em> design drawings.</td>
</tr>
<tr>
<td>Control Award</td>
<td>The Engineering Notebook must include an Engineering Section that documents the control components.</td>
</tr>
</tbody>
</table>
FTC Team 9901

Lesson Learned

* Understand the game requirements
* Use a test / prototype environment or board
  - to work on programming in parallel
* Build prototypes with cardboard, plastic pieces (avoid tetrix parts)
* Test incrementally
* Think about maintenance of your design
  - complex vs. simple (reliable, easy to maintain)
  - need to sustain multiple tournaments
* Tournament dates and location (only 1 in NoVA region)
* Tournament readiness (taking everything needed)
  - inventory, checklist, team roster, inspection checklist
* Final robot for competition
  - inspection checklist, team # display (only one robot)
Q & A