

Intro to Programming for FTC

Kanthan Rajendran
Infinity Tech Mentor



FTC Canton Kickoff Event
September 7, 2024



Agenda

- FTC Control System
- Op Modes



INFINITY TECH

FTC TEAM 13684

FTC Control System



- Driver Hub or Driver Station (Android Phone)
- Control Hub or Robot Controller (Android Phone)
 - Expansion Hub



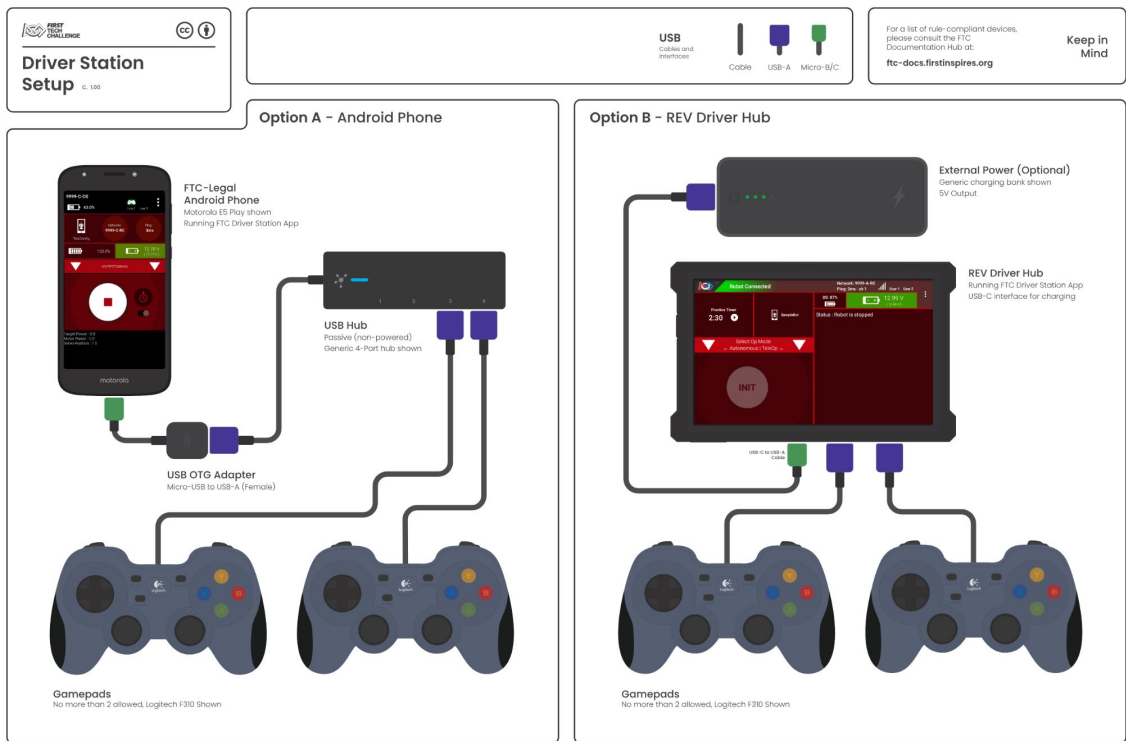
INFINITY TECH
FTC TEAM 13684

On the Robot

- 1 Control Hub
- 1 Expansion Hub
- 12 V Battery
- Up to 8 motors
- Up to 12 servos
- Touch, Distance, Color Sensors
- Webcams



Driver Station



With the Driver Team

- Driver Hub
- Game Pads

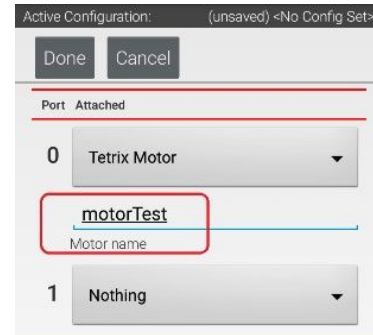
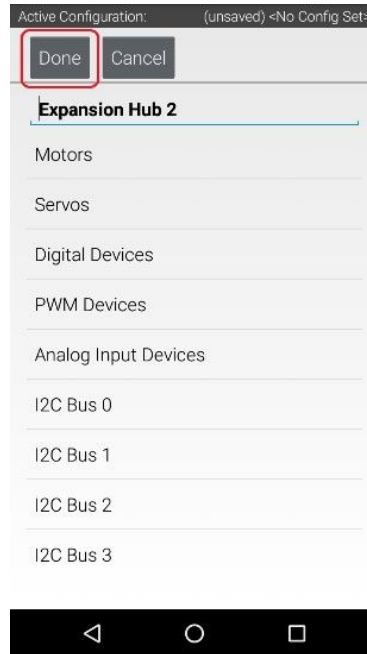
OpModes are Launched from the Driver Hub



INFINITY TECH
FTC TEAM 13684

Configuration File

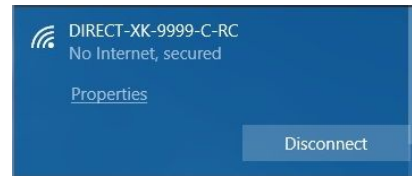
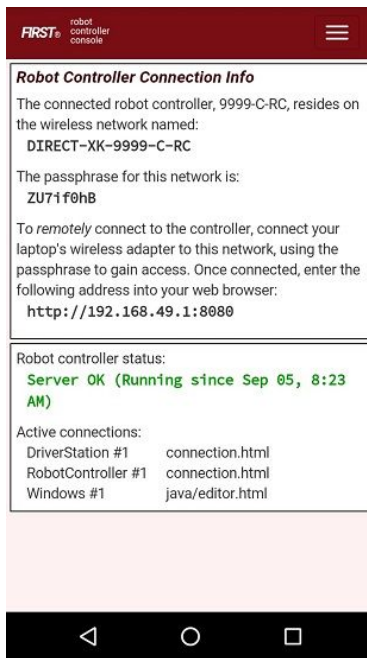
- Tells **Control Hub / Robot Controller** which port **sensors, motors, servos**, and any other connected devices can be found.



INFINITY TECH
FTC TEAM 13684

Connecting to Laptop

- Connect a laptop to the Control Hub / Robot Controller
- Navigate to the URL shown on a Java enabled browser (like Google Chrome)



INFINITY TECH
FTC TEAM 13684

Op Mode

- Teams write programs called ***op modes*** (which stands for “operational modes”) to specify the behavior for their robot.
- These op modes run on the **Robot Controller** after being selected on the **Driver Station**.
- Programming Options
 - Blocks
 - OnBotJava
 - Android Studio



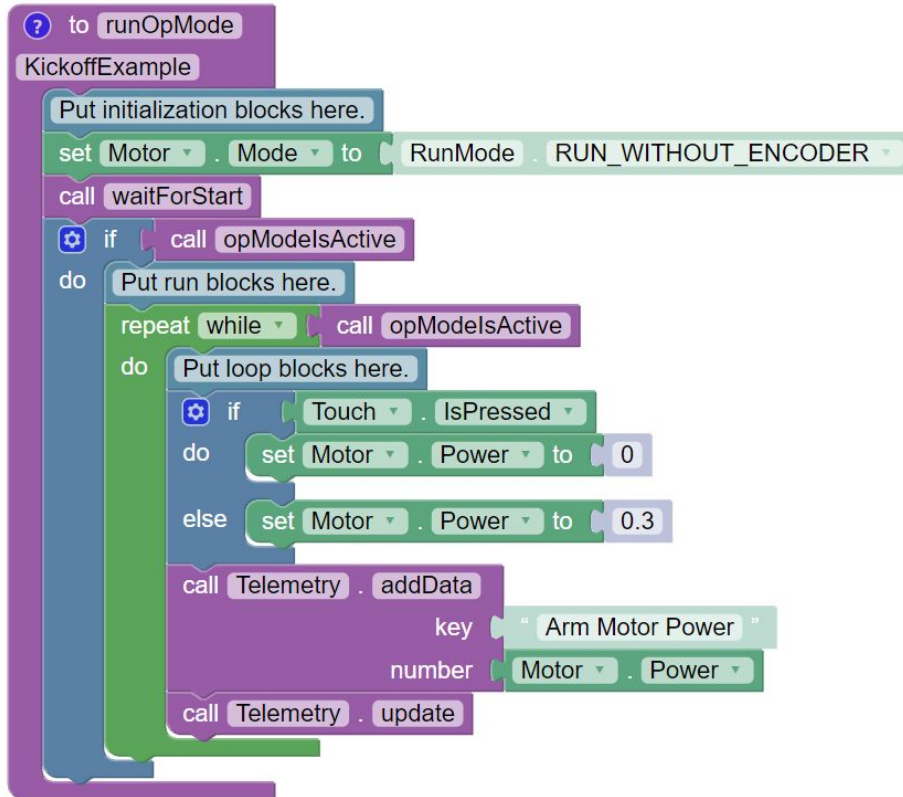
Increasing complexity & functionality



INFINITY TECH

FTC TEAM 13684

Blocks - Motor with Touch Sensor



INFINITY TECH
FTC TEAM 13684

OnBot Java - Motor with Touch Sensor

```
package org.firstinspires.ftc.teamcode;

import com.qualcomm.robotcore.eventloop.opmode.TeleOp;
import com.qualcomm.robotcore.hardware.DcMotor;
import com.qualcomm.robotcore.hardware.TouchSensor;
import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;

@TeleOp

public class KickoffProgram extends LinearOpMode{
    //Defines variables for the touch sensor and motor
    TouchSensor touch;
    DcMotor motor;

    @Override
    public void runOpMode(){
        //Get the touch sensor and motor from the hardware map
        touch = hardwareMap.get(TouchSensor.class, "Touch");
        motor = hardwareMap.get(DcMotor.class, "Motor");
    }
}
```

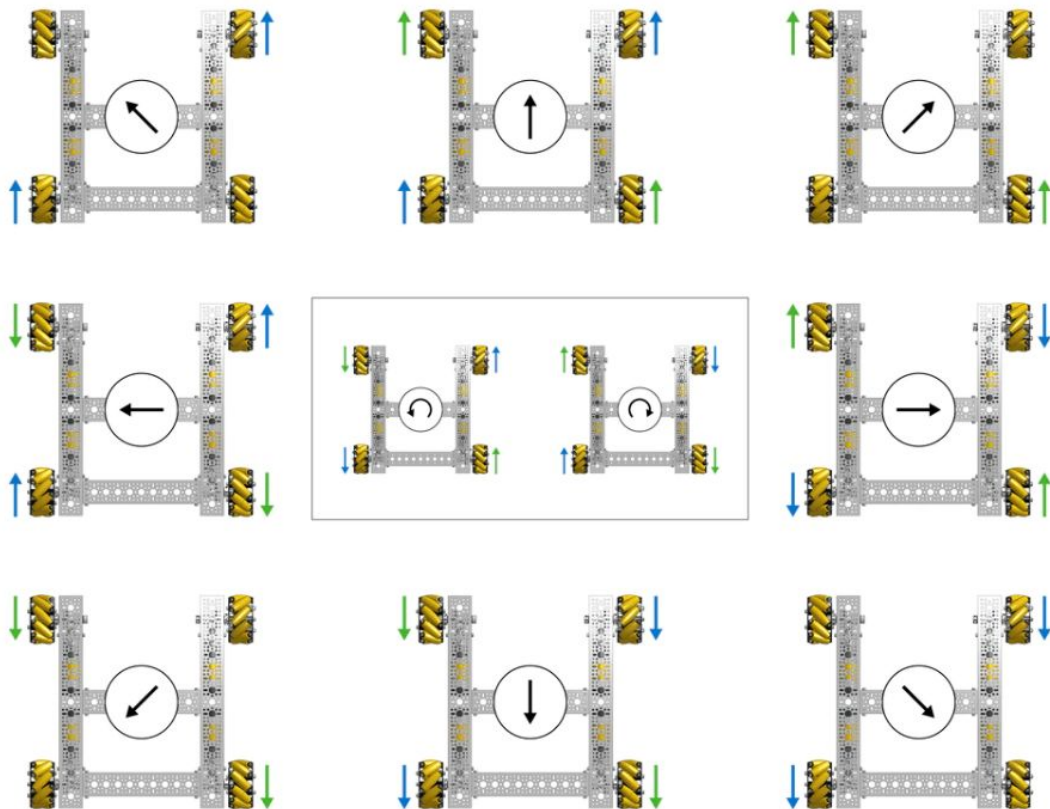
```
waitForStart();
while(opModeIsActive()){
    //If the touch sensor is pressed, stop the motor
    if (touch.isPressed()){
        motor.setPower(0);
    }
    //Otherwise run the motor
    else{
        motor.setPower(0.3);
    }
}
```



INFINITY TECH

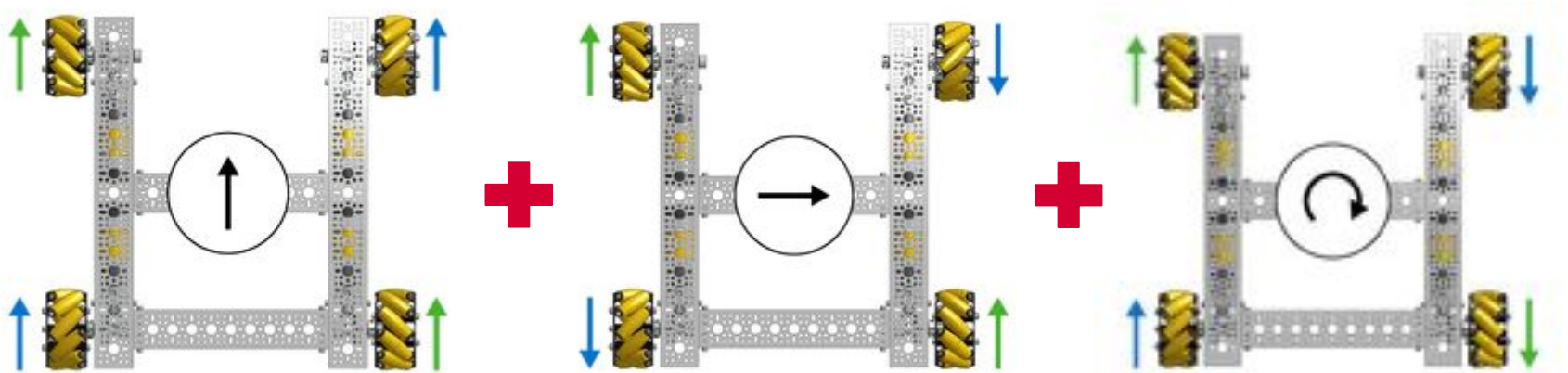
FTC TEAM 13684

Mecanum Drive



INFINITY TECH
FTC TEAM 13684

Mecanum Drive



$$\text{frontLeftMotor} = y + x + rx$$

$$\text{backLeftMotor} = y - x + rx$$

$$\text{frontRightMotor} = y - x - rx$$

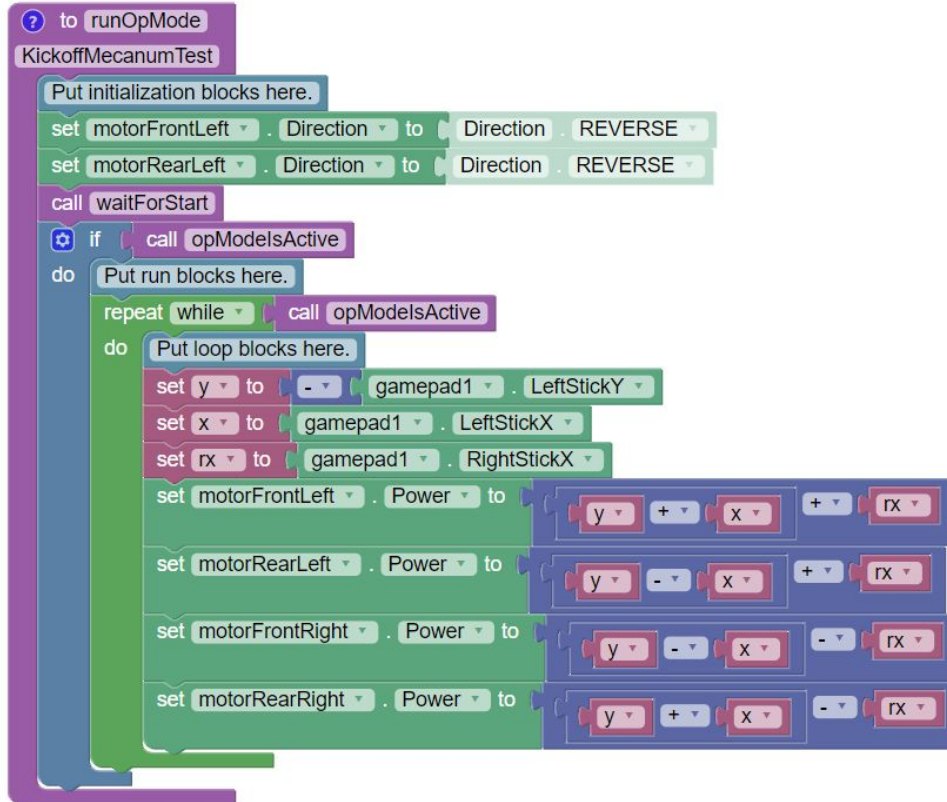
$$\text{backRightMotor} = y + x - rx$$



INFINITY TECH

FTC TEAM 13684

Blocks - Mecanum Drive



INFINITY TECH
FTC TEAM 13684

OnBot Java - Mecanum Drive

```
package org.firstinspires.ftc.teamcode;

import com.qualcomm.robotcore.eventloop.opmode.TeleOp;
import com.qualcomm.robotcore.hardware.DcMotorSimple;
import com.qualcomm.robotcore.hardware.DcMotor;
import com.qualcomm.robotcore.eventloop.opmode.OpMode;

@TeleOp
public class KickoffMecanumTest extends OpMode {

    //Declares the variables for each of the motors
    private DcMotor frontLeftMotor;
    private DcMotor frontRightMotor;
    private DcMotor rearLeftMotor;
    private DcMotor rearRightMotor;

    @Override
    public void init() {
        // Gets the motors from the hardware map, aka the programming board
        frontLeftMotor = hardwareMap.get(DcMotor.class, "motorFrontLeft");
        frontRightMotor = hardwareMap.get(DcMotor.class, "motorFrontRight");
        rearLeftMotor = hardwareMap.get(DcMotor.class, "motorRearLeft");
        rearRightMotor = hardwareMap.get(DcMotor.class, "motorRearRight");
    }
}
```



INFINITY TECH

FTC TEAM 13684

OnBot Java - Mecanum Drive

```
// These lines reverse the direction of the motors
frontLeftMotor.setDirection(DcMotorSimple.Direction.REVERSE);
rearLeftMotor.setDirection(DcMotorSimple.Direction.REVERSE);

public void loop() {
    double pf = 0.5;

    // Assigns gamepad inputs
    double x = gamepad1.left_stick_x;
    double y = gamepad1.left_stick_y;
    double rx = gamepad1.right_stick_x;

    // Sets power to the motors
    frontLeftMotor.setPower((y + x + rx) * pf);
    frontRightMotor.setPower((y - x - rx) * pf);
    rearLeftMotor.setPower((y - x + rx) * pf);
    rearRightMotor.setPower((y + x - rx) * pf);
}
```



INFINITY TECH

FTC TEAM 13684

Resources

- FIRST Tech Challenge Programming Resources
 - https://ftc-docs.firstinspires.org/en/latest/programming_resources/index.html
- Game Manual 0
 - gm0.org
 - Great resource for FTC teams
- Learn Java for FTC
 - Free programming guide written by FTC mentor
 - <https://github.com/alan412/LearnJavaForFTC>



INFINITY TECH

FTC TEAM 13684