FTC New Platform Workshop

presented

By

FTC TEAM #8565

New Platform Hardware

Evan Li / Patrick Lin / Abhishek Vijaykumar

New Platform Hardware (part I)

Evan Li

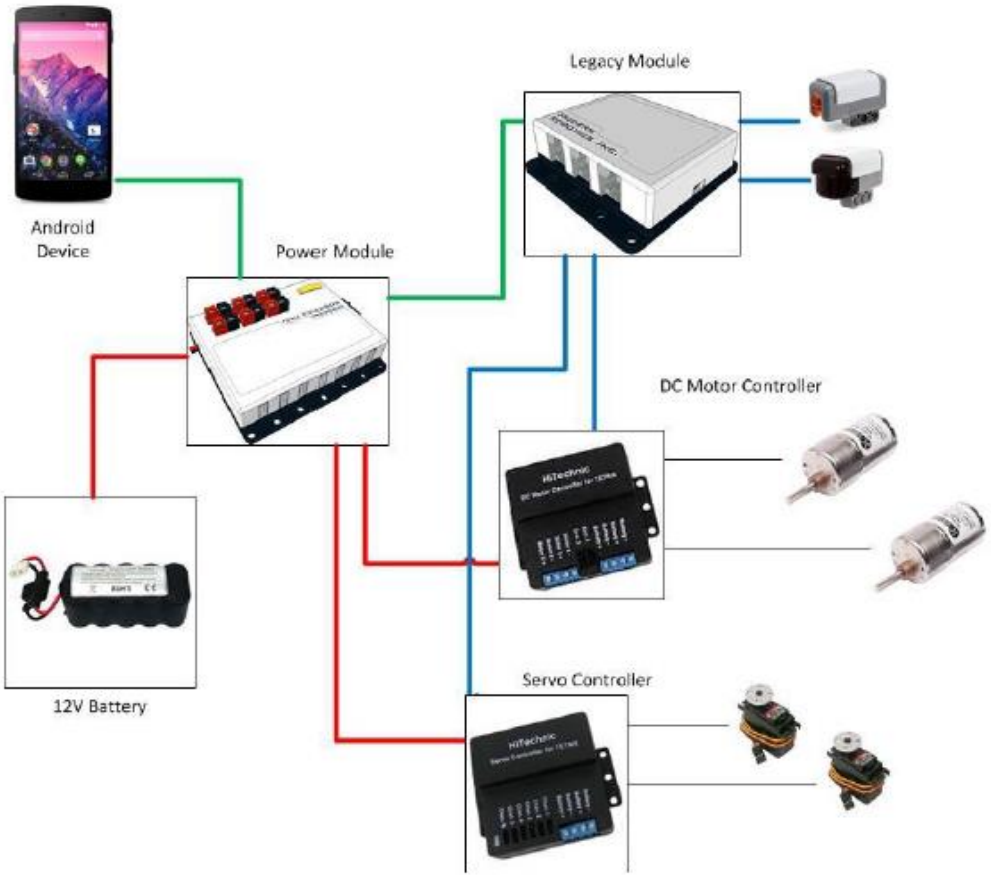
Overview

- Point to Point Wireless Connectivity
- New Hardware Modules
- Robot Controller and Driver Station
- USB Ports and Cables
- Sparring Robot Demo

Personal Connection

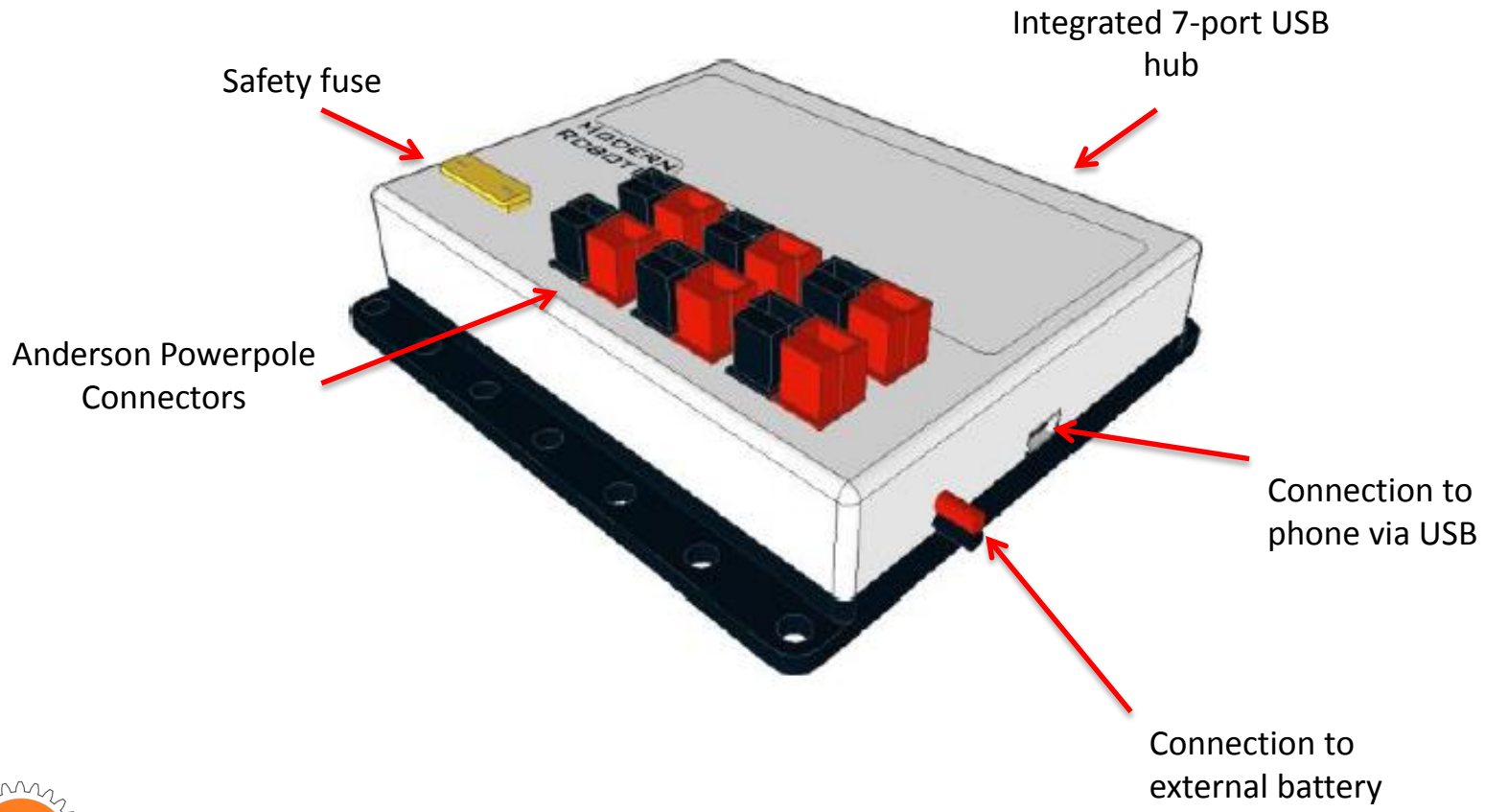
- The new FTC platform is a solution that is based on the Android operating system.

Robot Configuration with Legacy Controllers



New Hardware Models

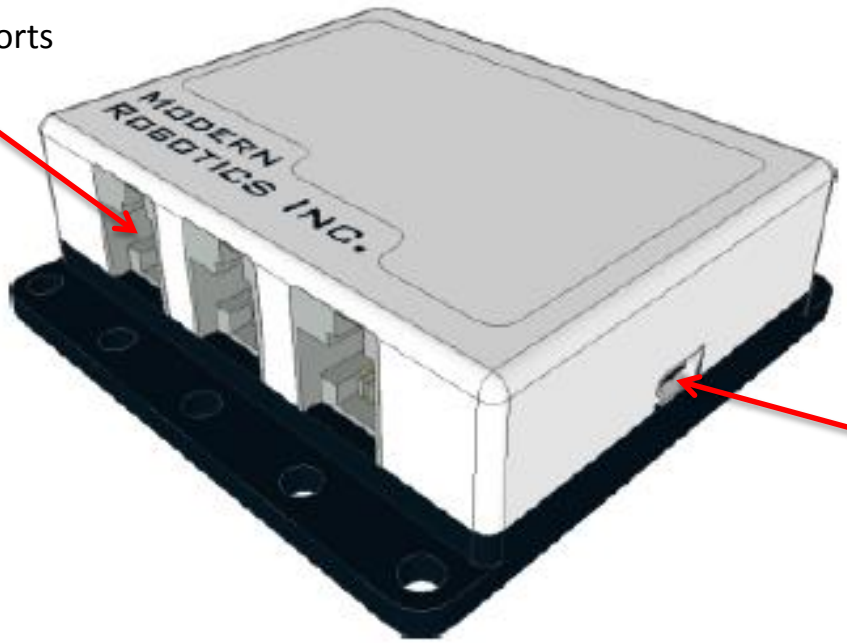
Power Distribution Module



New Hardware Models

Legacy Module

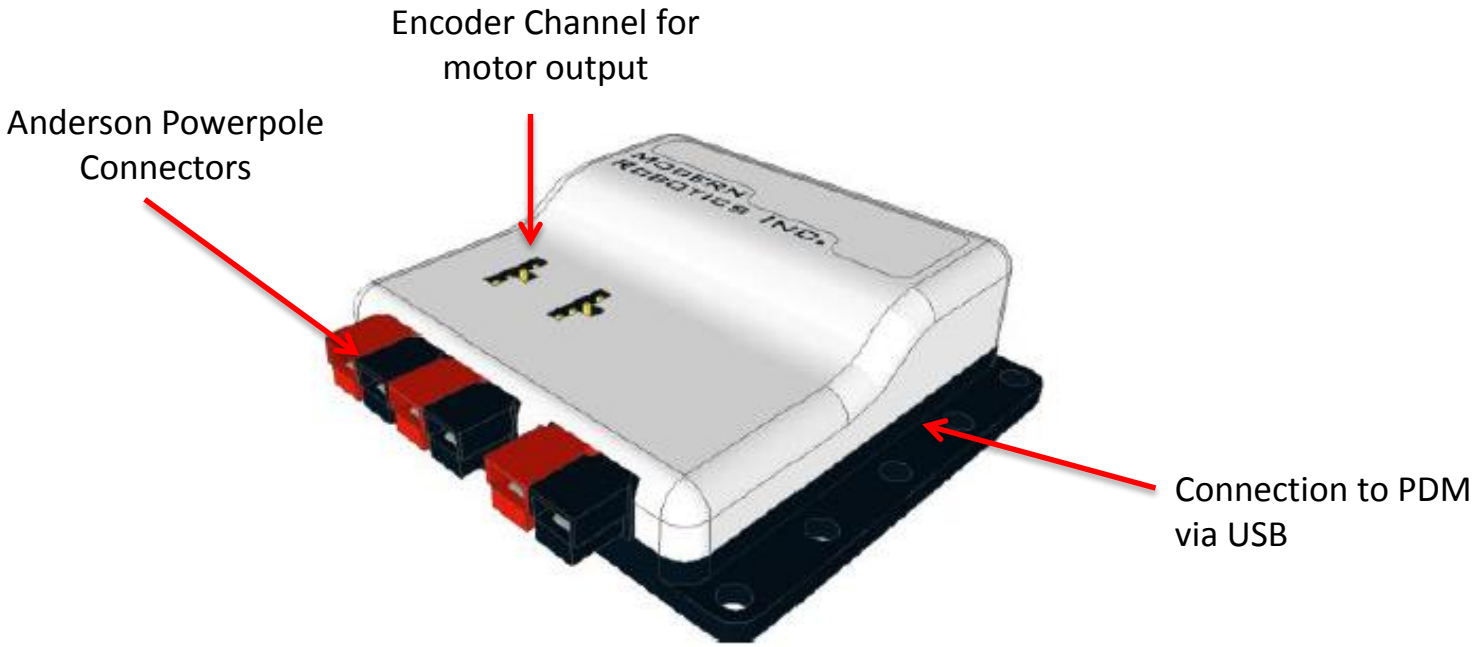
6 NXT-style
connector ports



Connection to PDM
via USB

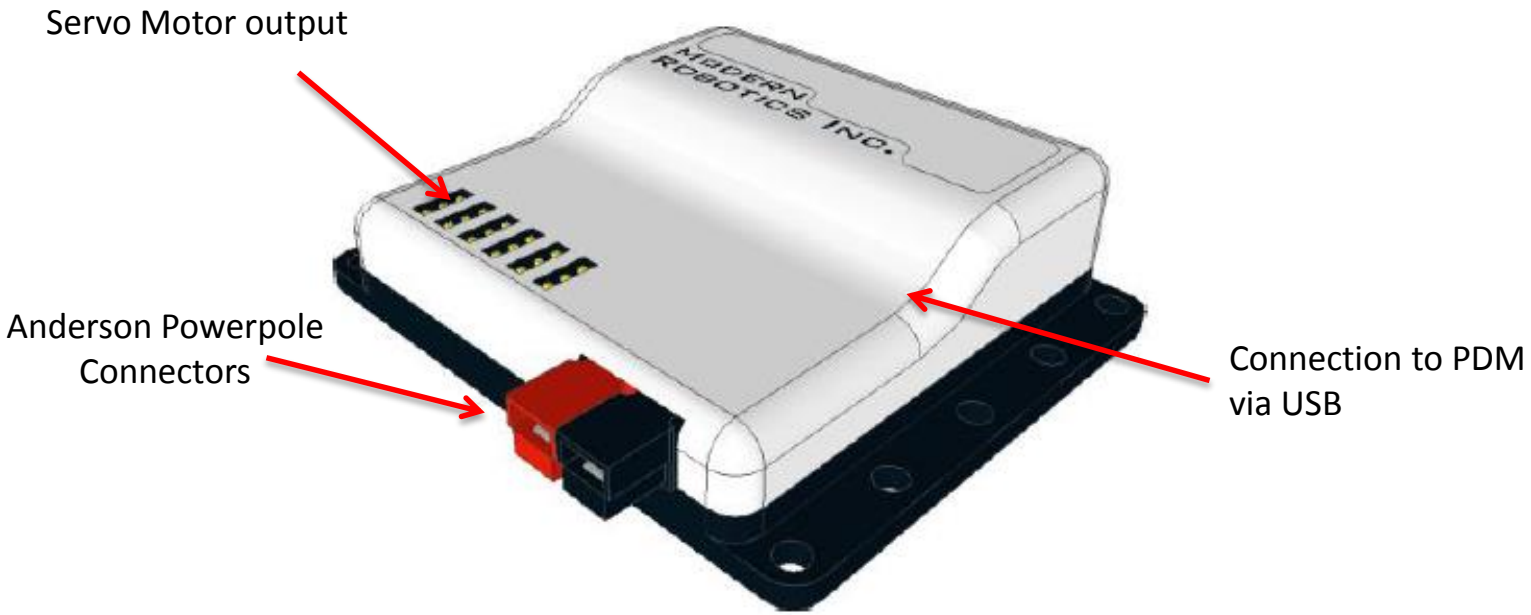
New Hardware Models

High Speed DC Motor Controller



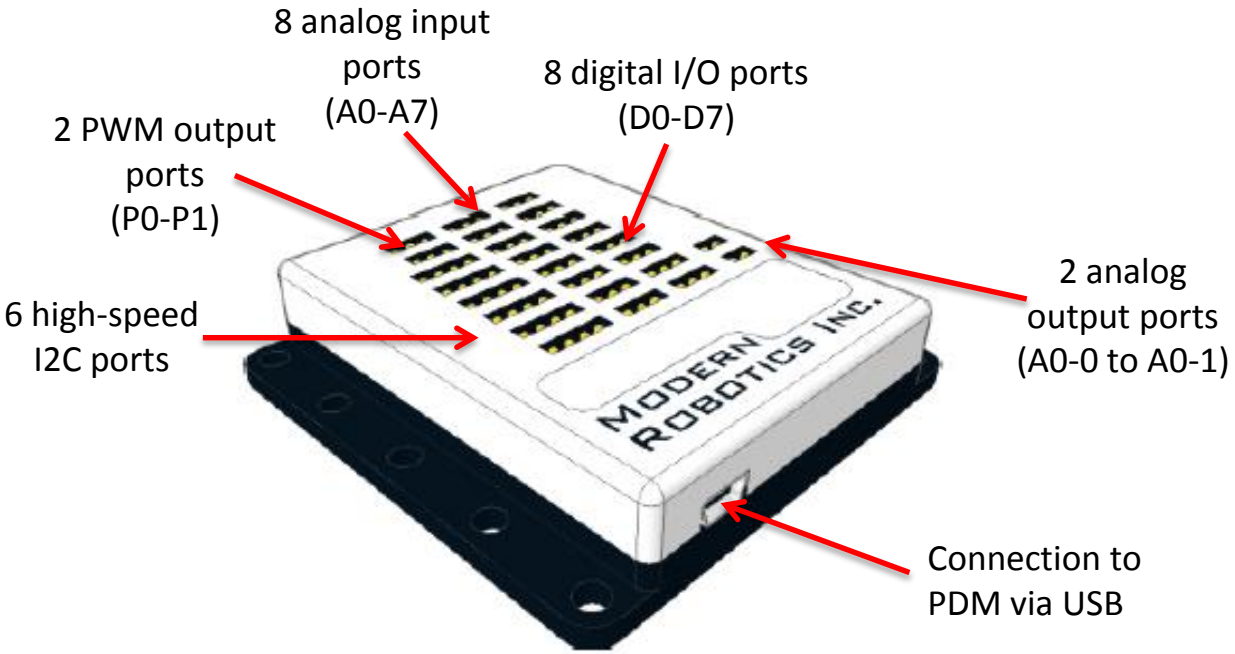
New Hardware Models

High Speed Servo Motor Controller



New Hardware Models

Advanced Sensor Module



Touch Sensor

Optical Distance Sensor

IR Seeker V3

Robot Controller (Phone)

- The robot controller acts as the “brains” of the robot. It handles communications with the driver station.

- It processes sensor data and user commands and sends instructions to the motor and servo controllers to make the robot move.

Robot

The robot **may** consist of the following components:

1. **Android Device**
2. **Robot Controller App**
3. **12V Rechargeable Battery**
4. **Power Distribution Module**
5. **Micro to Mini USB Adapter**
6. **USB Cables**
7. **Legacy Module**
8. **DC Motor Controller with Motors**
9. **Servo Controller with Servos**
10. **Hitechnic IR Seeker V2 Sensor**
11. **LEGO Light Sensor**
12. **NXT Cables**

Now we’re moving on to the Driver Station

Driver Station

- The driver station is the component that the teams use to provide input (from a pair of gamepads and/or a touch screen) to the Robot Controller.

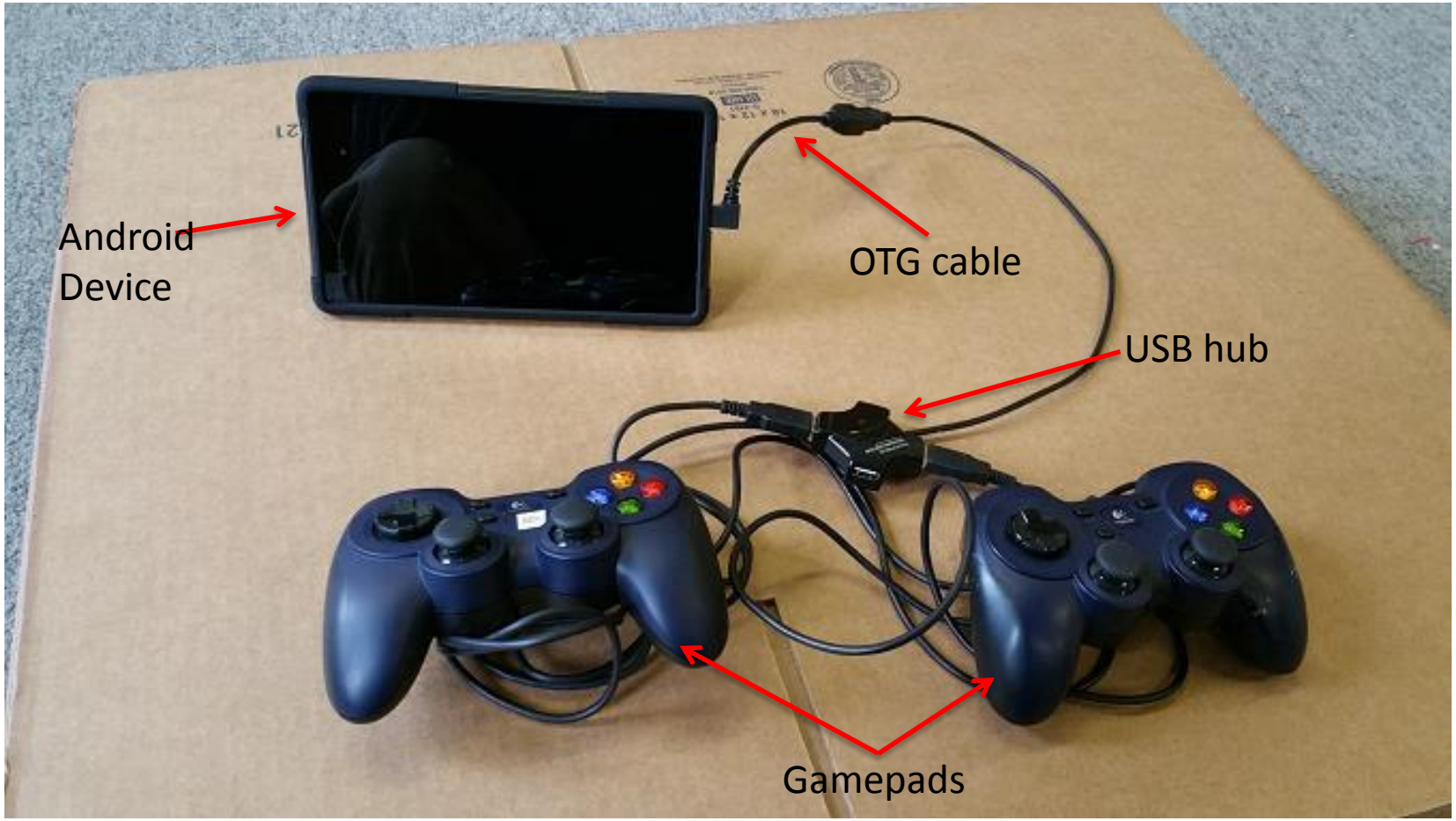
- The driver station also displays messages (status information, motor speed, sensor data, etc.) from the Robot Controller.

Driver Station

The driver station usually consists of the following components:

1. **Android Device**
2. **USB Gamepad Controllers**
3. **Non-Powered USB Hub**
4. **Micro USB Adapter (OTG cable)**
5. **Driver Station Android App**

Driver Station



The hardware of this driver station consists of an Android device, an OTG cable, a USB hub and two gamepads.

USB Ports and Hubs

- The next gen platform uses USB (which stands for Universal Serial Bus) to connect the Android devices to the robot and game controller hardware.
- The USB cables allow the devices to communicate at a very high speed.
- The USB cables also provide power to the devices that are “downstream” to the Power Module (flows all to one body).

USB Host and Client Mode

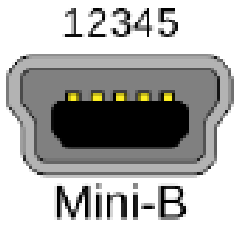
- A USB-enabled device such as a smartphone or tablet can act as a USB host or as a USB client device.
- A USB host is the device that initiates all communications on the bus. The USB client responds to communications from the host.
- The Android devices that are used for the next gen platform have the ability to act as either a USB host or a USB client.

So here are a few USB cables and their receptacles…

USB Micro Type B

requires…

USB Mini Type B



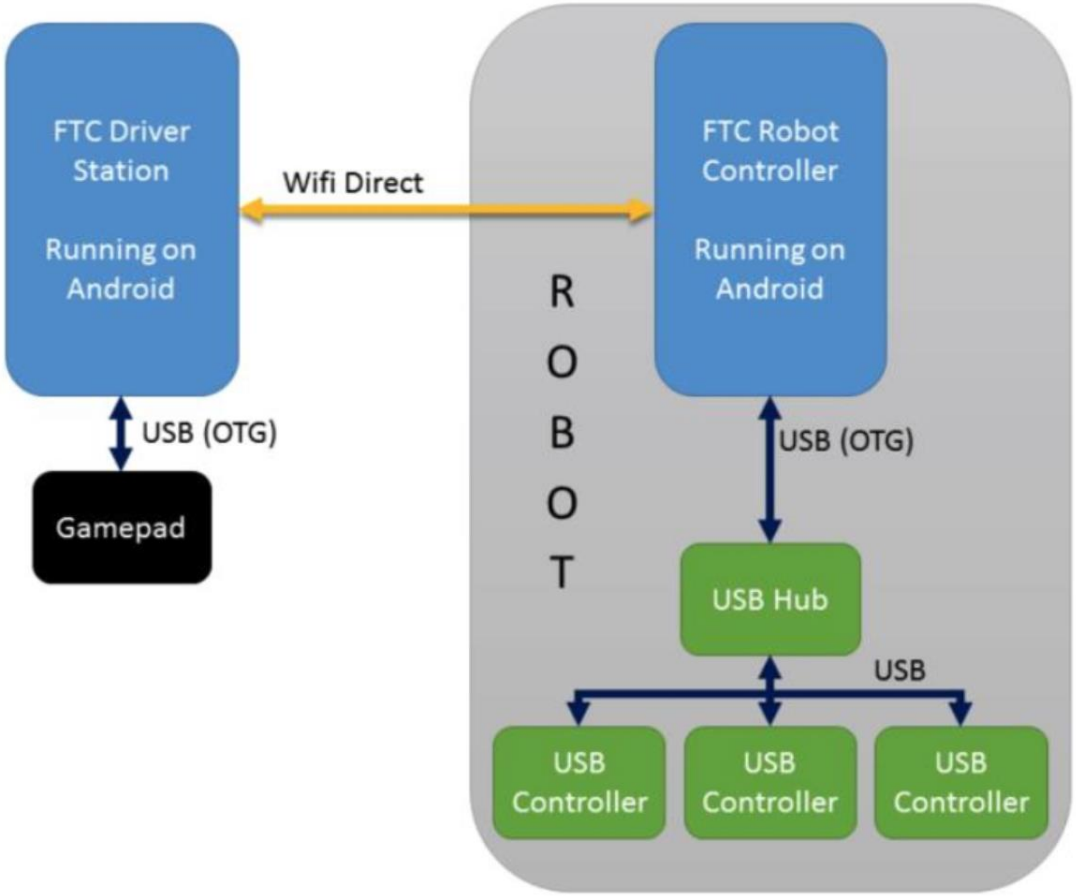
requires…

Micro USB OTG Adapter Cable

- A special adapter cable is used to connect the Android device to the USB hub for gamepads.

- OTG stands for On The Go

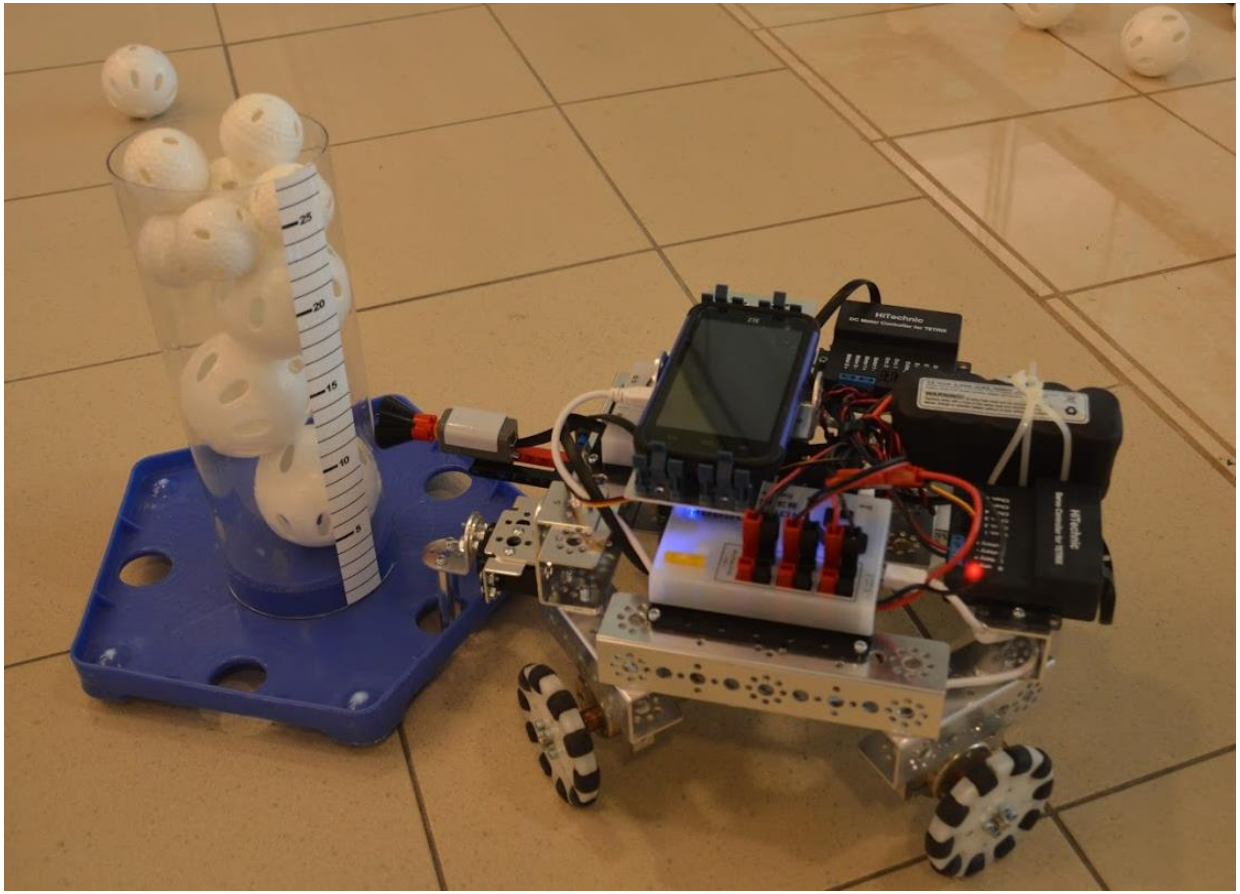
Robot Control Model



USB Ports and Hubs Reminder

- When the driver station is left connected to the robot controller while the robot is on and idle, then the robot controller and driver station drain lots of battery. After two hours of leaving it idle, the robot controller (phone) drained 35% battery and the driver station drained 25%. Please make sure that when you are not using the robot, unplug the phone, kill the app, and turn the robot off.

Sparring Robot Demo



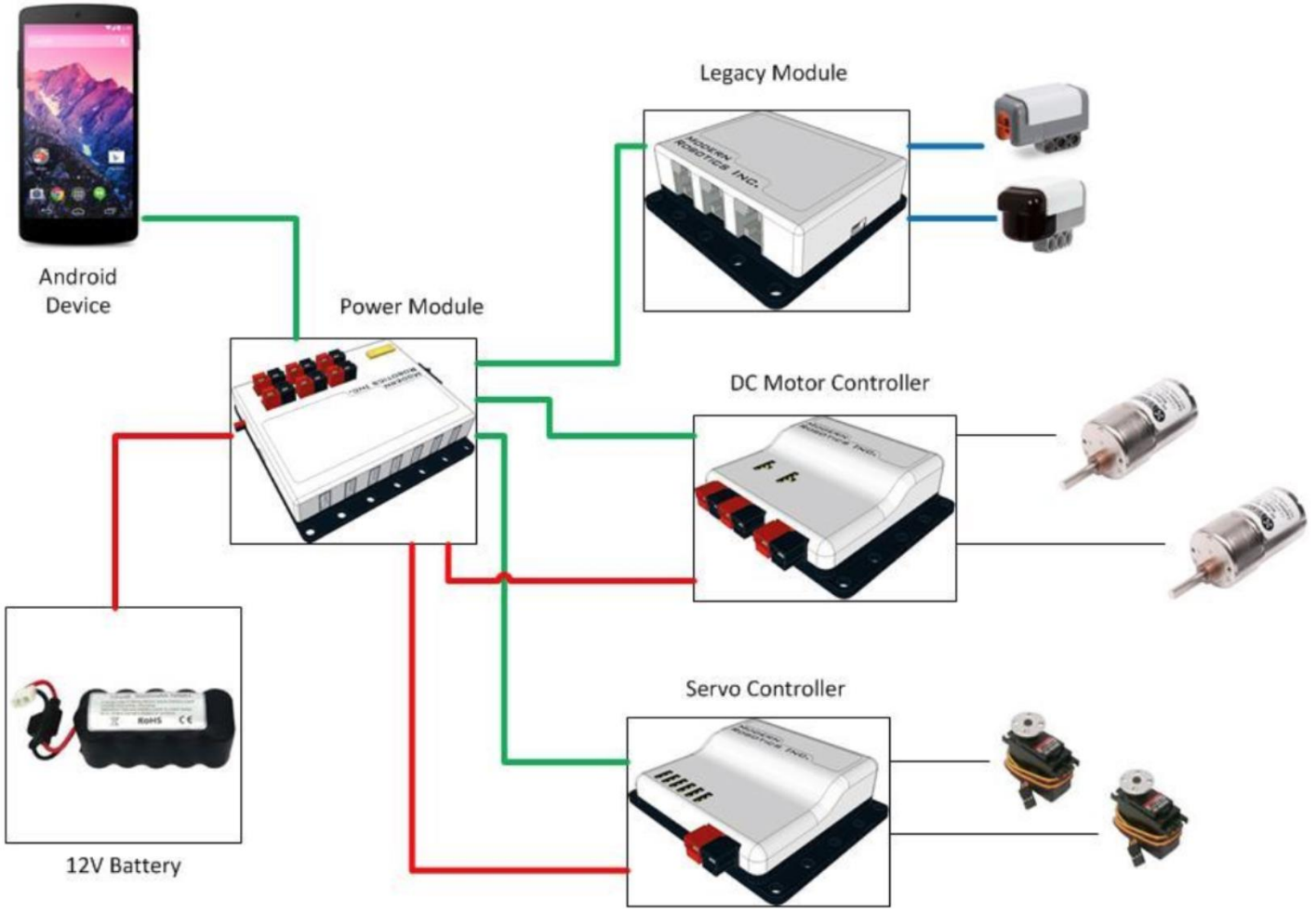
New Platform Hardware (part II)

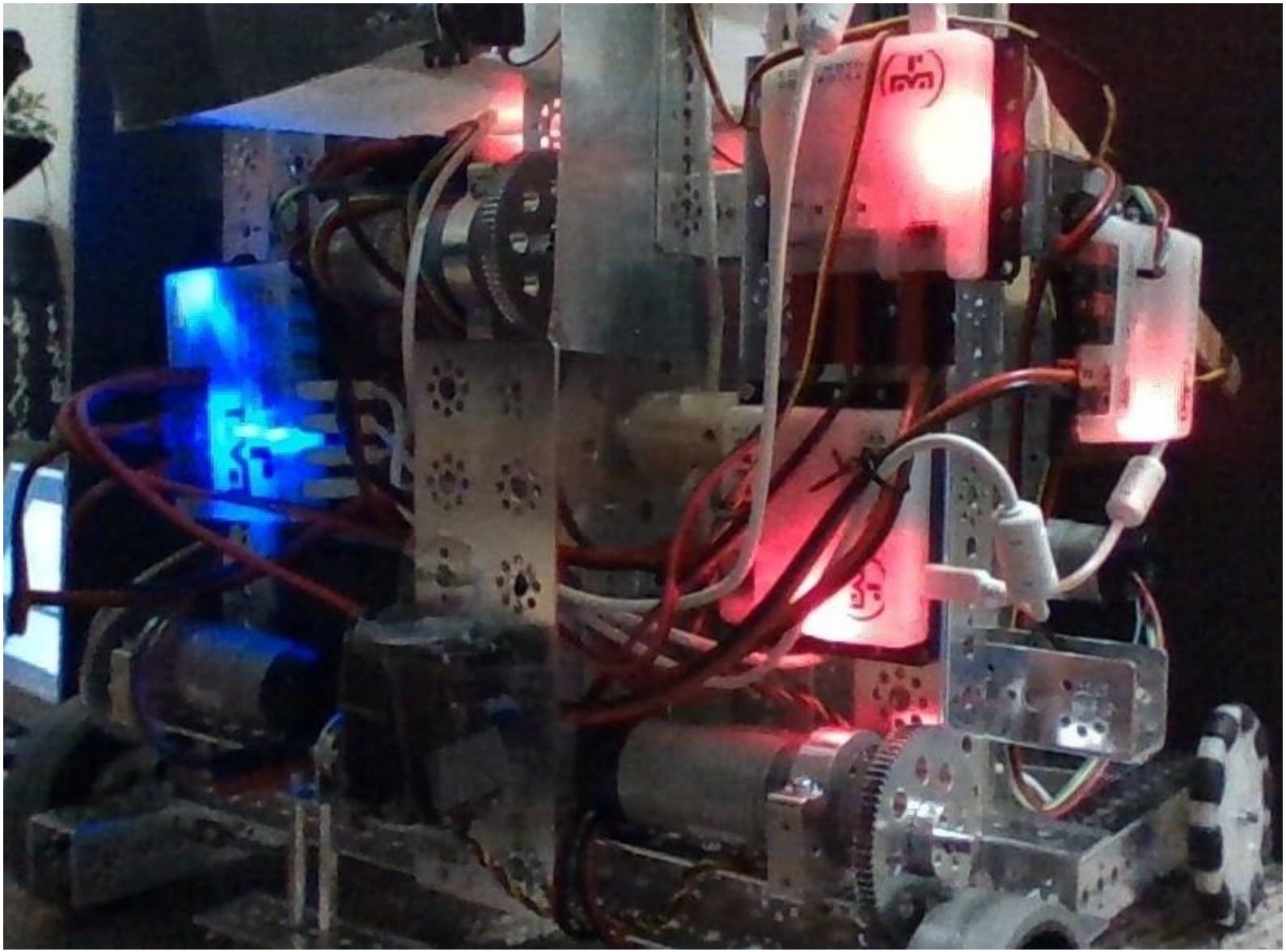
Patrick Lin

Overview

- Multiple DC Motor Controller Configuration
- Dynamic nature of new robot configuration
- Three Modes of DC Motor Controller
- Port Study
- Battery Study

Robot Configuration with New Controllers





Two step migration from old to new controllers

- Step 1: Add on Power Distribution Module(PDM) and Legacy Module(LM) with legacy controllers

- Step 2: Remove legacy controllers and LM, add on new controllers.

DC Motor Controller Naming Convention - Problem

When you attach multiple motor controllers to the PDM and scan, the Configuration **randomly** assigns names, “Motor Controller 1,2,3,4” to the controllers.

Because of this, you do not know which Controller on the robot is “Motor Controller 1”

DC Motor Controller Naming Convention – 3 Step Solution

- Step 1: Create a configuration file for each controller to find out the controller's serial number

- "Phone Name" > Phone > FIRST

Step 2

- Connect phone to PC as a media device (in the settings of the phone), look into the FIRST directory for the corresponding XML file, write down the serial number on the controller for easy reference.

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<Robot>
    <MotorController name="Motor Controller 2" serialNumber="AL00VDS8">
        <Motor name="sweeper" port="1" />
        <Motor name="NO DEVICE ATTACHED" port="2" />
    </MotorController>
```

Step 3

- Look at the official configuration file, and map out the serial number with the controller name, so you know what controller is “Motor Controller 1”.

Differences between new and old configuration

- Old: In Robot C it is a pragma statement, every time you change the configuration you have to change the code, pragma can not be shared among programs (unless you copy and paste).

Difference between new and old configuration

- New: In the new one, the configuration is in an XML file, an independent file, so you don't have to re-download the program if you change the configuration, as long as the hardware map/naming is the same.

- Multiple programs can use the same configuration file

Sample of Robot configuration xml file

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<Robot>
    <MotorController name="Motor Controller 2" serialNumber="AL00VDS8">
        <Motor name="sweeper" port="1" />
        <Motor name="NO DEVICE ATTACHED" port="2" />
    </MotorController>
    <MotorController name="Motor Controller 1" serialNumber="AL00YBZ2">
        <Motor name="linearmotor2" port="1" />
        <Motor name="rightwheel" port="2" />
    </MotorController>
    <ServoController name="Servo Controller 3" serialNumber="AL00VCT4">
        <Servo name="clamp" port="1" />
        <Servo name="boxservo" port="2" />
    </ServoController>
    <MotorController name="Motor Controller 4" serialNumber="AL00VGPB">
        <Motor name="leftwheel" port="1" />
        <Motor name="linearmotor" port="2" />
    </MotorController>
</Robot>
```

Three Modes of DC Motor Controllers

- Running Without Encoders
- Running With Encoders
- Run To Position

Running Without Encoders

This mode of a DC Motor Controller commands the DC motor to run a certain speed. Since there is no encoder, the robot doesn't know if motor is actually running at the specified speed.

- This mode is not accurate, especially when exposed to heavy loads or changing environments.

Running With Encoders

In this mode, you still tell it to move a certain speed. Encoder values are monitored by the controller.

- The controller will automatically adjust the output to maintain the desired speed.

Encoder Cable Reminder

- Make sure you put in the encoder cable the right way into the controller!

The **YELLOW** wire should be close to the side with the motor wires.

Run To Position

This mode of a DC Motor Controller is basically making the DC Motor a larger version of a servo motor. This needs an encoder cable.

- The DC Motor Controller moves to a certain position, and automatically adjusts to stay at the position

Ex:

```
Linear Slide2.setChannelMode(DcMotorController.RunMode.RUN_TO_POSITION);
```

Run To Position

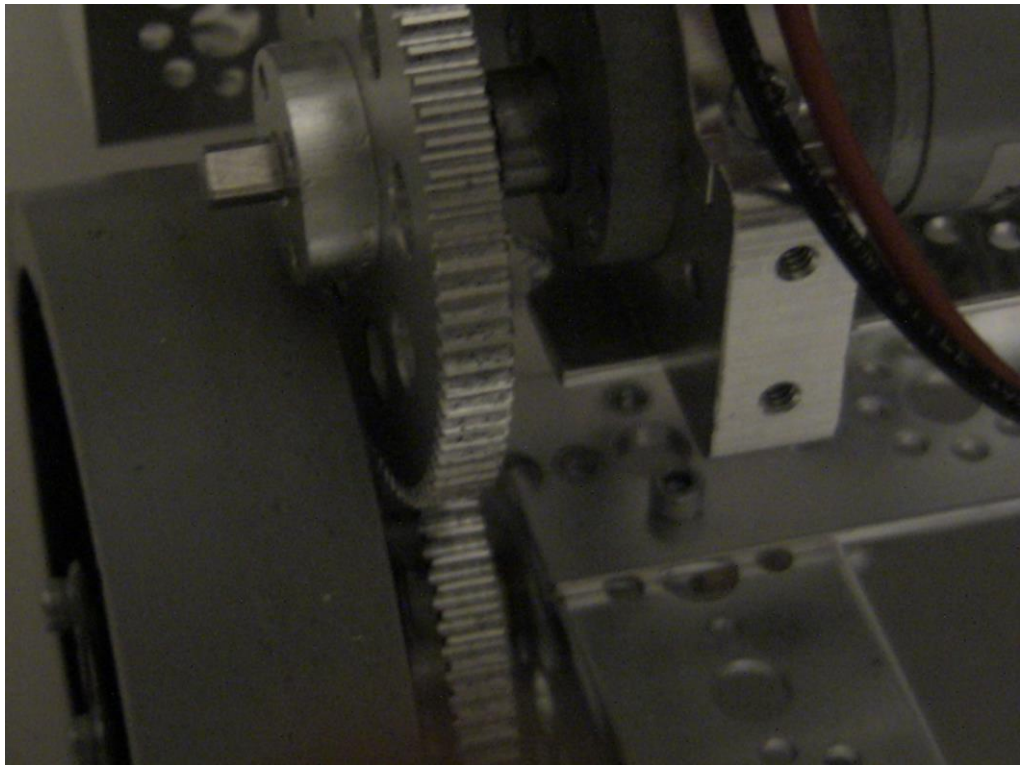
- If you set the DC Motor to run to a certain position, it will ignore the positive or negative sign of the power.

DC Controller Port Study

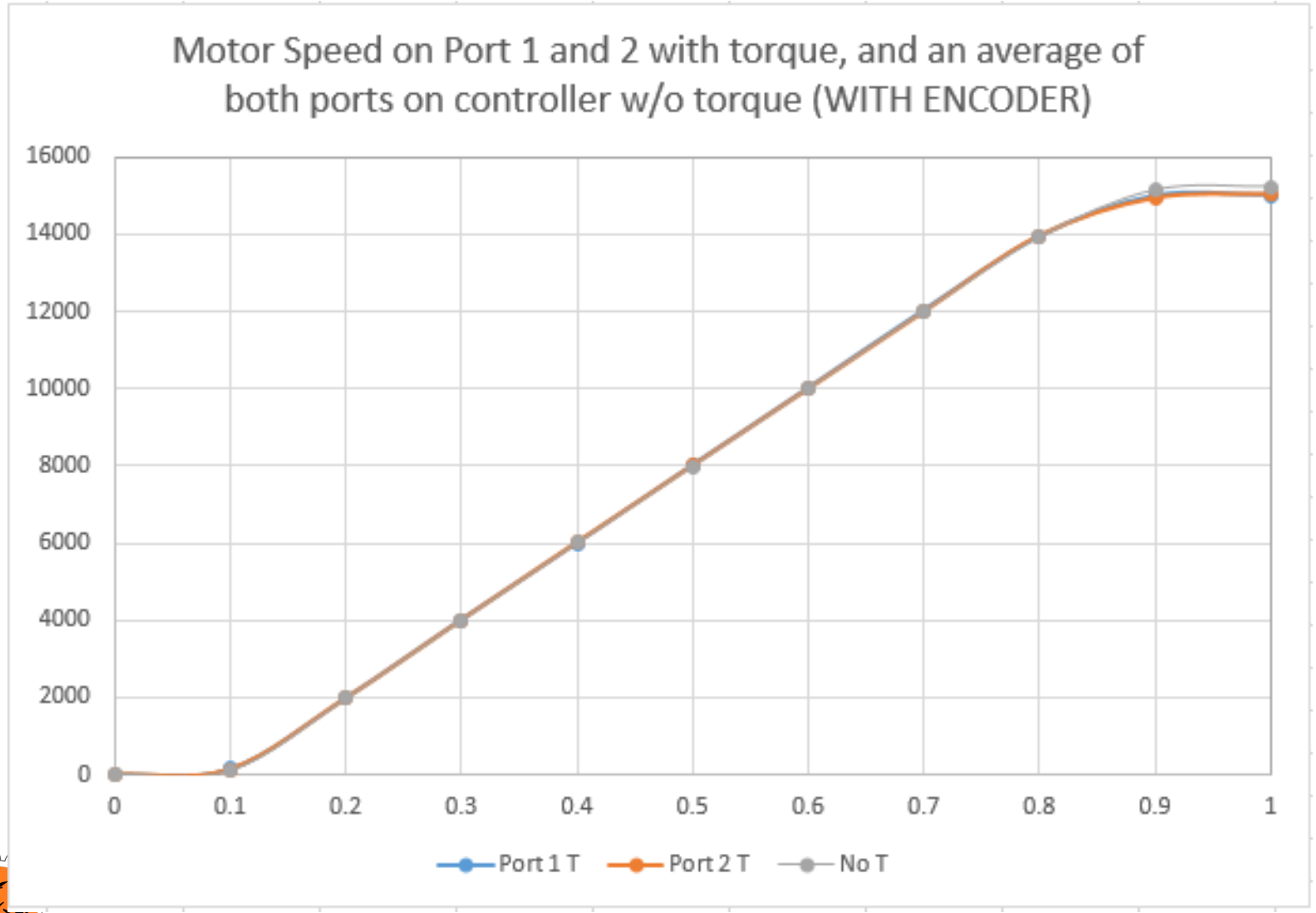
- Purpose: to see if the DC motor has any major jumps in output power at different specific speeds.
- Another reason is to make sure that both Port 1 and 2 are equal in power, because in the past our team has had a difference in power of the two ports on the Legacy Controllers.

Experiment Setup

Picture of slight load, (DC connected to top 80 tooth gear)



DC Motor Output Speed vs. Power

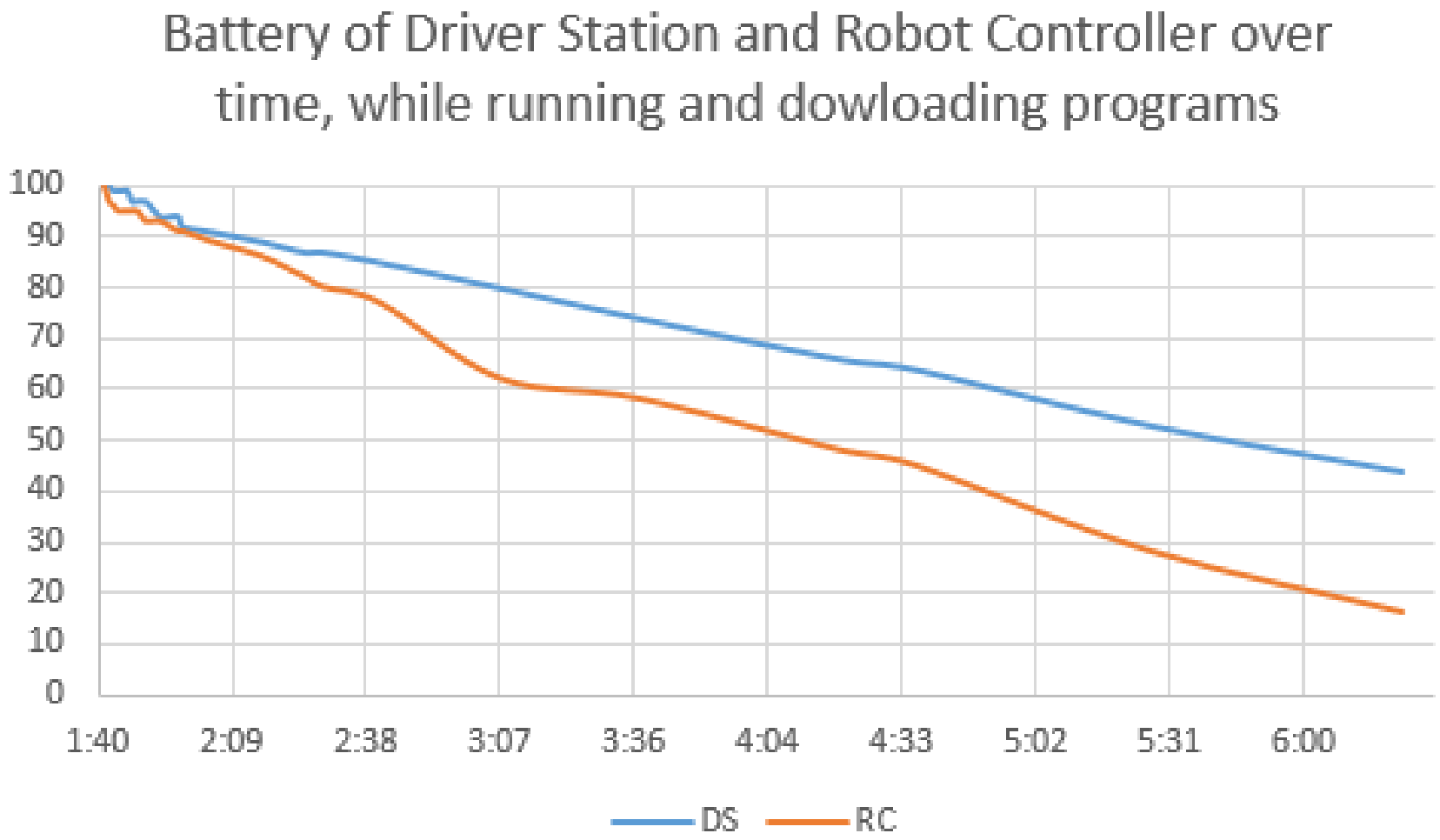


Battery Study

- Our team conducted a battery test just to see about how long it would take for the phones to run out of battery.

- Reason: During one of our demos, the both phones, especially the robot controller, ran out of battery completely in a short time so we wanted to know exactly the battery life.

Battery Study



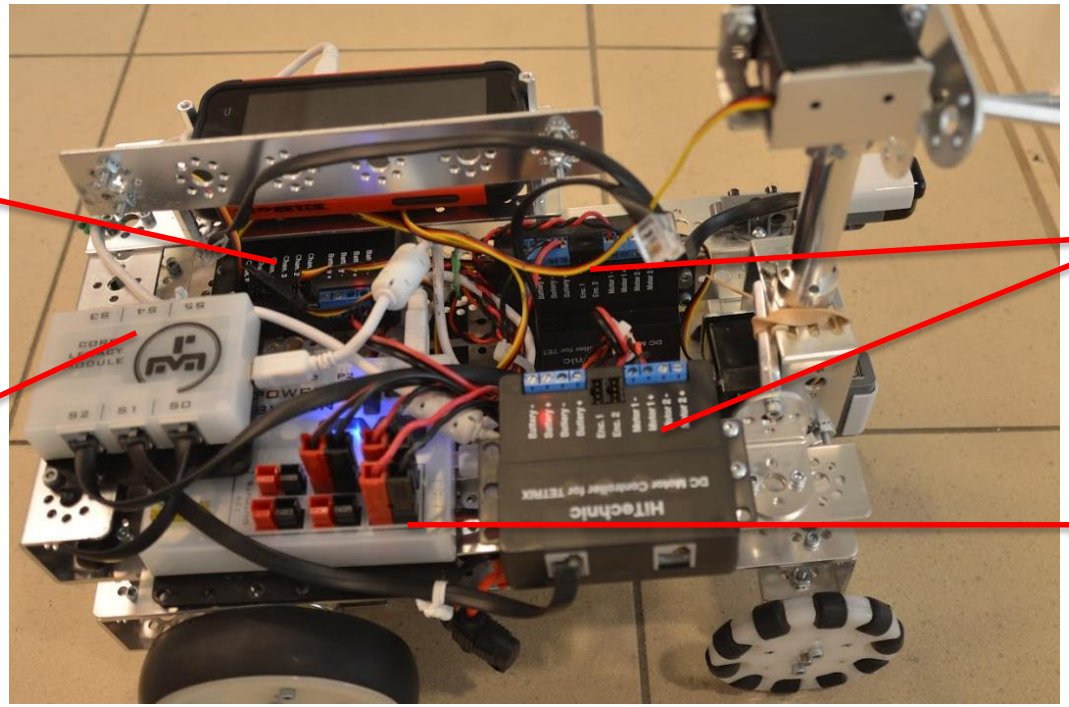
New Platform Hardware (part III)

Abhishek Vijaykumar

Overview

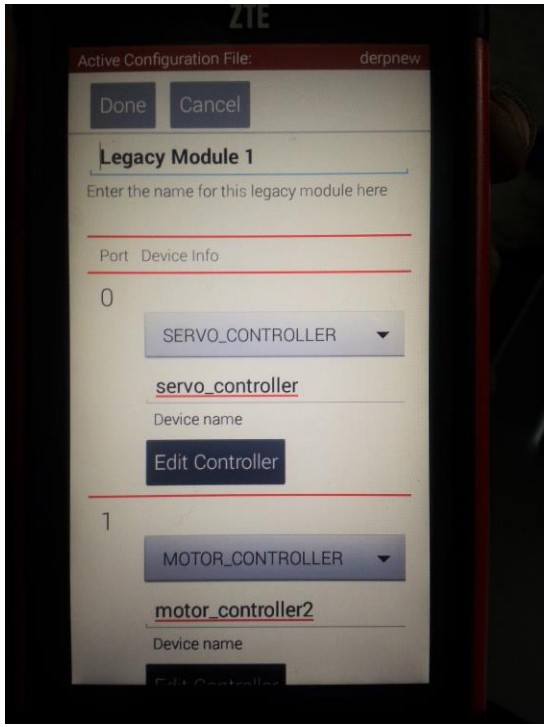
- K9 Special Build
- K9 Configuration
- K9 PTC Model
- K9 Demo
- Android Phone Mount
- Anderson Power Poles Installation

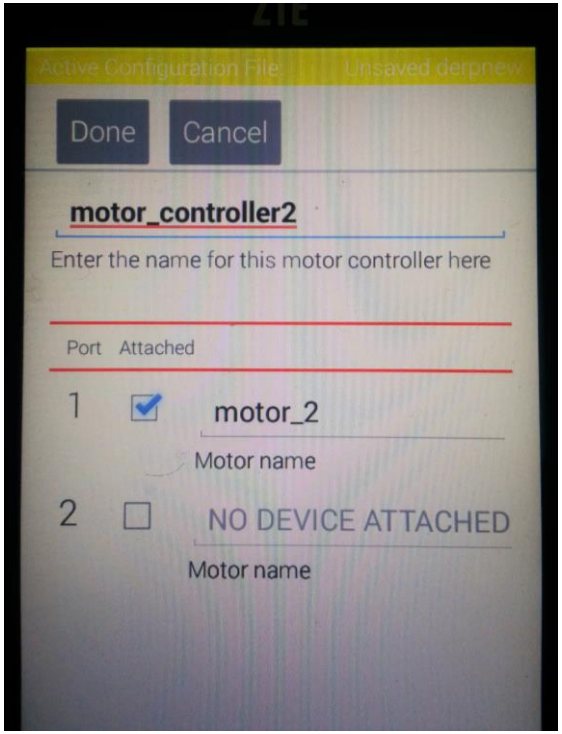
The Special K9 Build



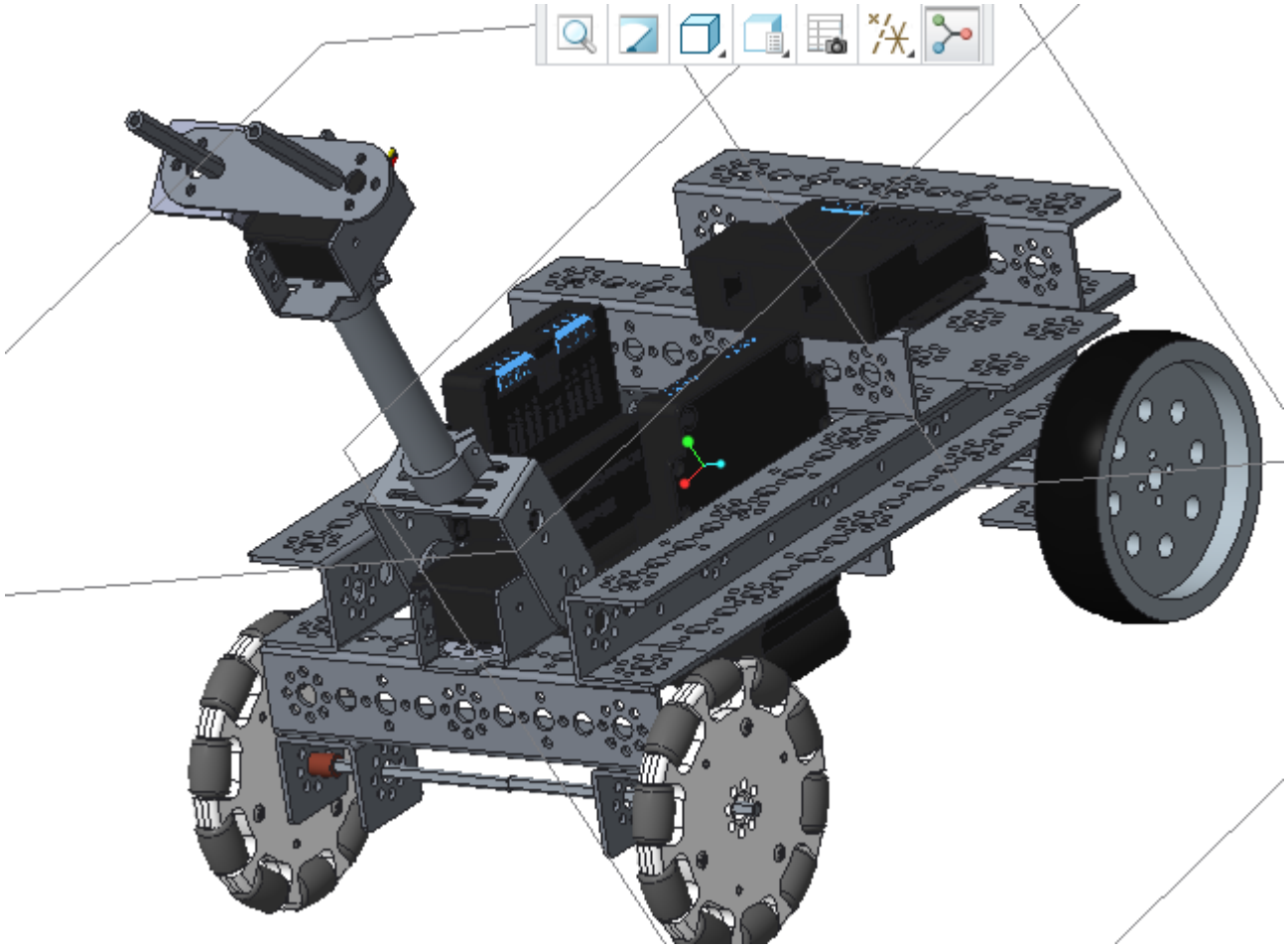
K9 Configuration

- The K9 Op Modes in the FTC robot controller app from google play works with this robot.





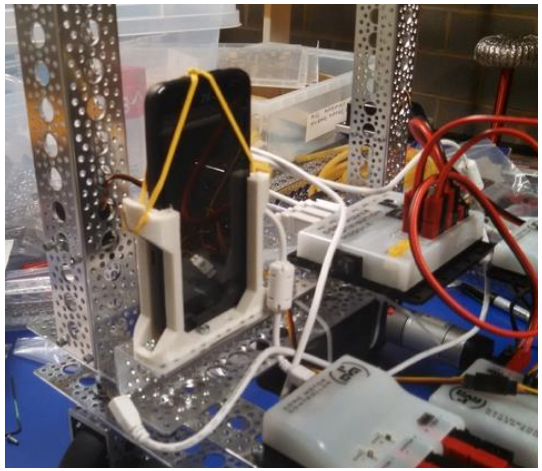
PTC Model of K9 Robot

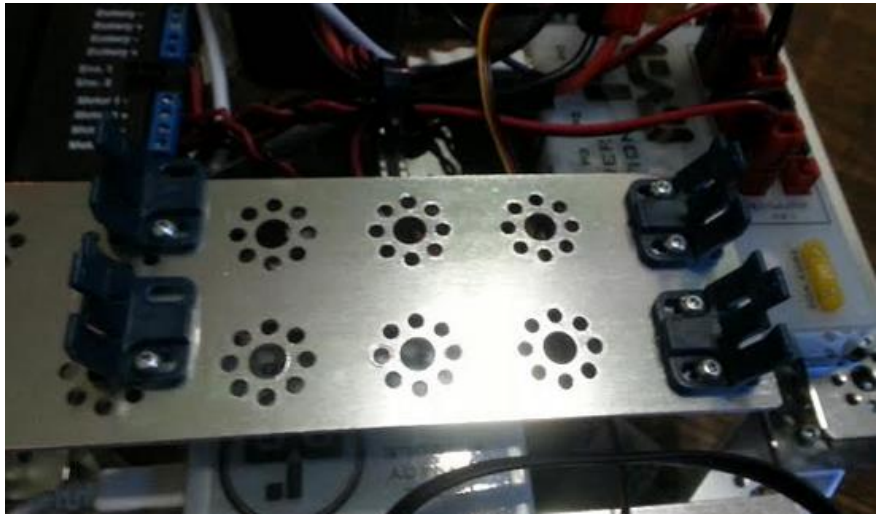


K9 Demo

- PTC Mechanism Demo
- Configure Hardware
- Physical Robot Demo

Different Android Phone Mounts





Our Own Phone Mount

Anderson Power Poles