The Patronum Bots
FTC Team #5972
Setups for Multiple Color Sensors

If your team wants to use more than one Modern Robotics (MR) color sensor on your robot, there are several steps that you will need to follow. A special thanks to Colton Mehlhoff from Modern Robotics for helping us understand how to make all of this work. Here is a quick outline of the steps we followed to make it work for us:

- Download Modern Robotics' **Core Device Discovery** software
- Use the MR software to change the I2C address(es) for one (or more) of the MR color sensors
- Create a program (we use Android Studio) that "recognizes" each MR color sensor based on their unique I2C addresses

Now we are going to walk through all the steps in detail.

1) Download Modern Robotic's Core Device Discovery software

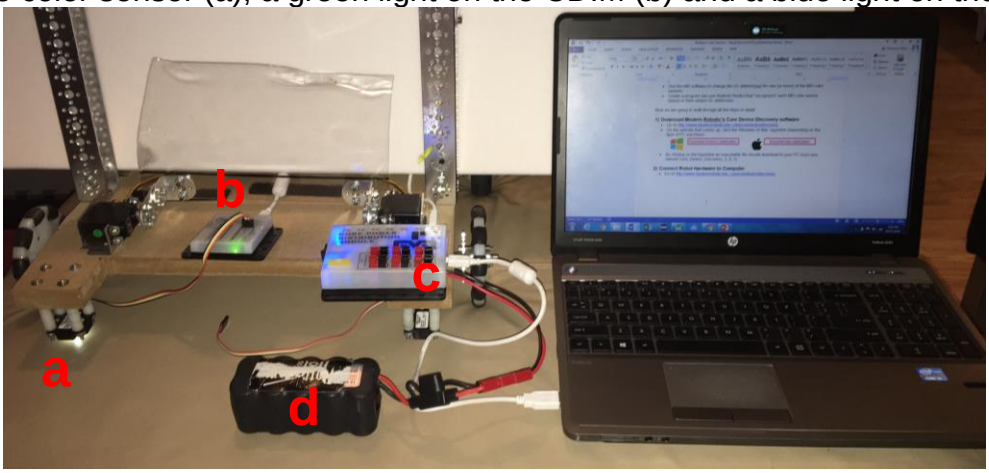
- Go to http://www.modernroboticsinc.com/coredevicediscovery
- On the website that comes up, click the Windows or Mac hyperlink (depending on the type of PC you have)

- By clicking on the hyperlink an executable file should download to your PC (ours was named Core_Device_Discovery_2_0_1)

2) Connect Robot Hardware to Computer

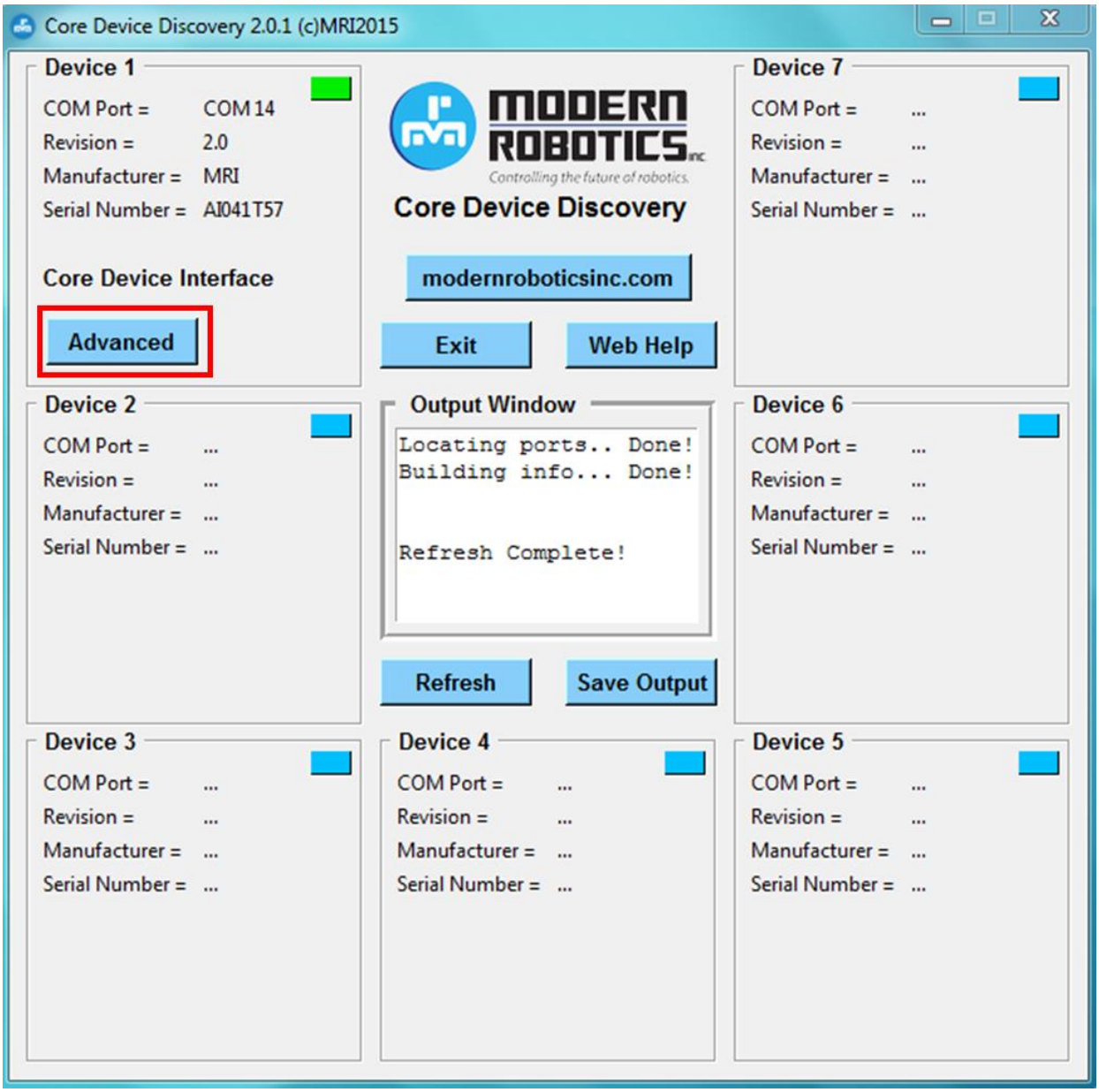
Connect the following hardware to your programming computer:

a. ONE color sensor plugged into Core Device Interface Module (CDIM). Do NOT plug in more than one sensor to start. We plugged our first sensor into CDIM's I2C Port 1.
b. Connect CDIM to the Core Power Distribution Module (PDM). We plugged the CDIM into Port P0 of the PDM.
c. Connect the PDM to the computer using the same kind of cable you use to connect any of the modules to the PDM.
d. Connect a 12v battery to the PDM and turn on the PDM. You should see the LED on the color sensor (a), a green light on the CDIM (b) and a blue light on the PDM (c).



3) Open the Core Device Discovery Software

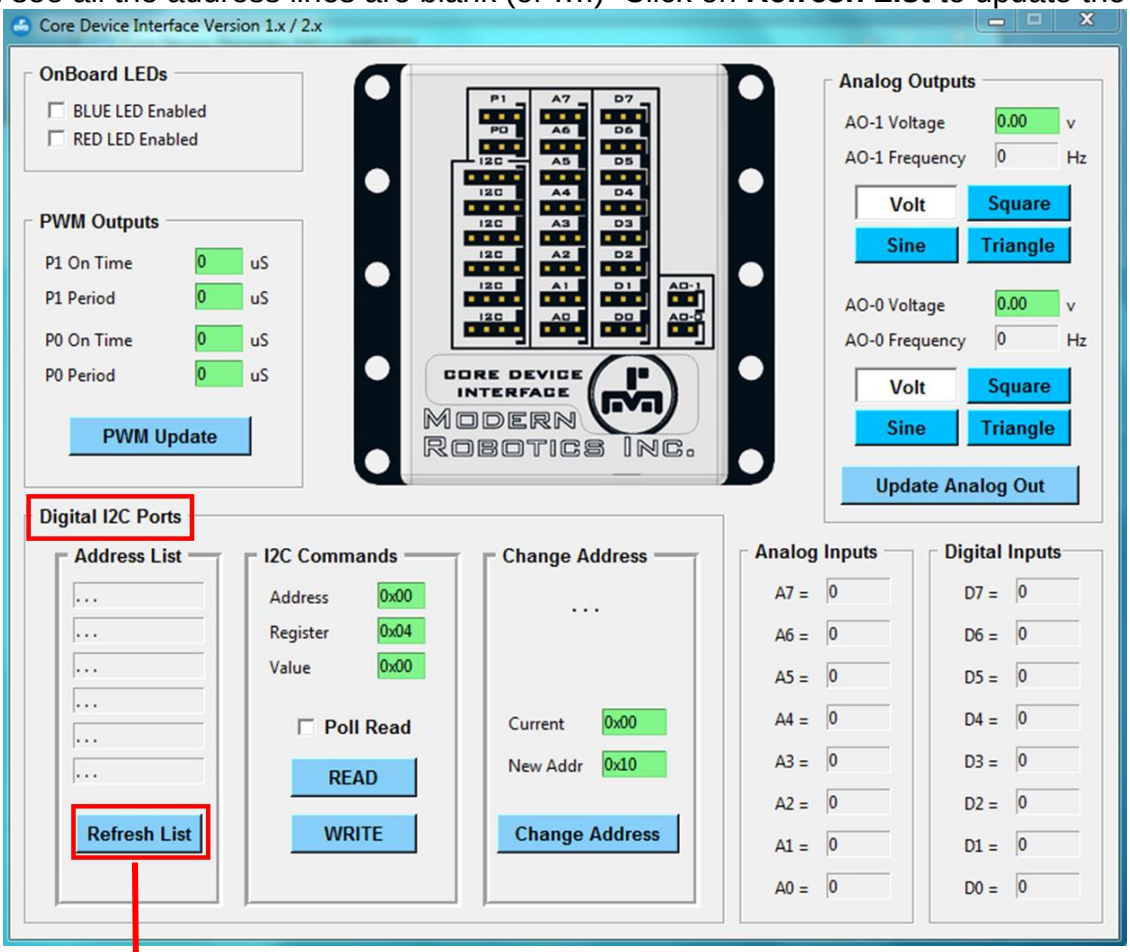
Open the executable file that was downloaded to your computer during Step 1. You may get a couple pop-up messages asking if it is OK to run the program. After saying Yes/OK, you should then see a screen that looks like this:

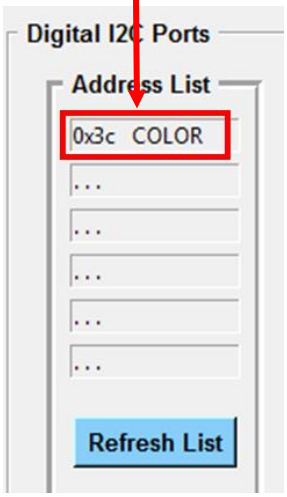


The software is designed to show up to 7 devices that are connected to the Power Distribution Module. In our scenario, we only have the CDIM connected. Click on the **Advanced** button to see more details about the Core Device Interface Module.

4) Viewing I2C Address of Color Sensor

Below is the CDIM details screen that is then displayed. For our testing, we are only concerned with the information in the bottom left of the screen in the **Digital I2C Ports** section. You will see all the address lines are blank (or ….) Click on **Refresh List** to update the list.



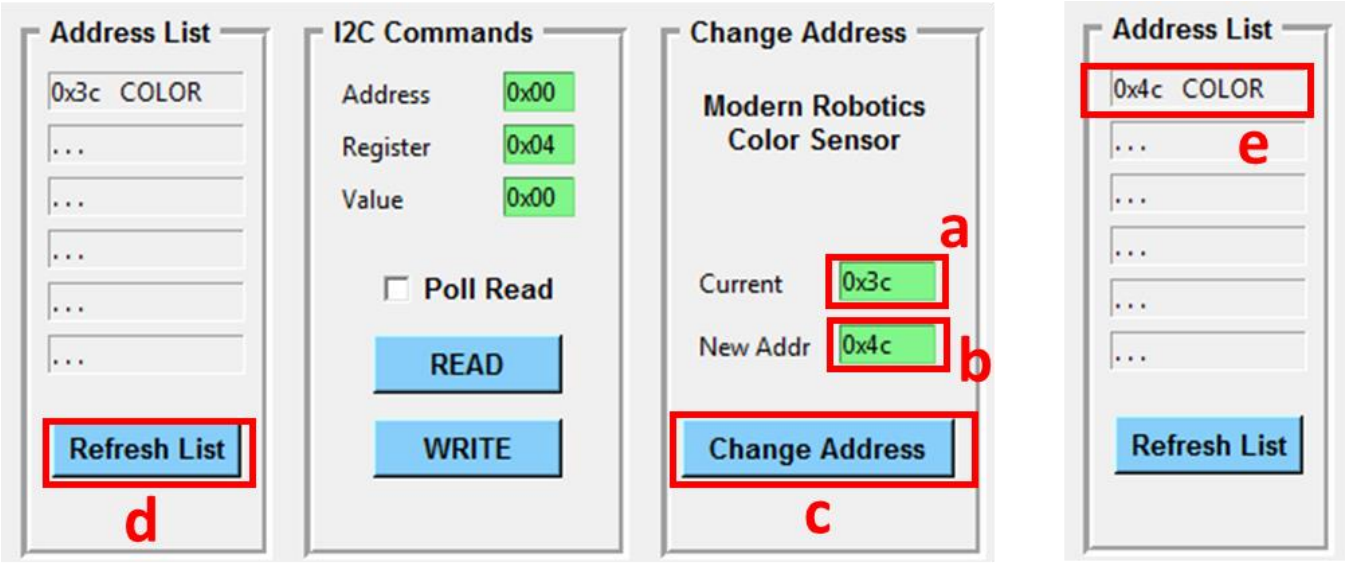


You should then see that the software recognizes a color sensor with an address of **0x3c**. It is important to know that this is the default, 8-bit hexadecimal address assigned to the MR color sensors. If you had more than one color sensor plugged in right now, the software would still only recognize ONE color sensor because they all start out with the same 0x3c I2C address. This is what we need to modify to use more than one color sensor.

5) Changing I2C Address of Color Sensor

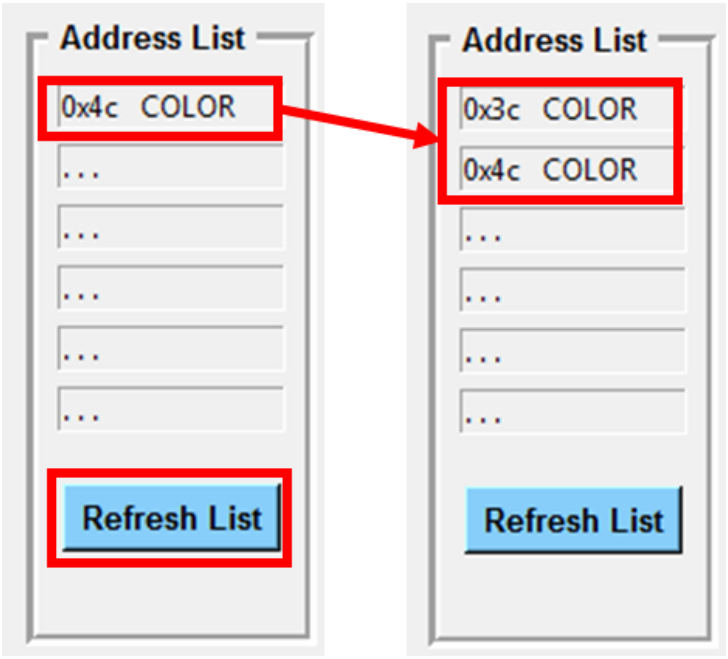
To change the address of your color sensor, follow these steps:

a. Type in current I2C address of color sensor in space shown below (in this case: 0x3c)
b. Type in new I2C address of color sensor in space shown below (we used **0x4c**)
c. Click on Change Address
d. Click on Refresh List
e. The address list will then show the color sensor with the new I2C address



6) Connecting a Second (or More) Color Sensor

- Plug a second color sensor into the CDIM (we plugged our second color sensor into Port 0). You should see the LED light up on the second color sensor when it is connected.
- Go back to the software program and click **Refresh List** again and you should then see two color sensors recognized with two different I2C addresses.
- If you are going to have more than two color sensors on your robot, repeat the steps to change the I2C address on each sensor until you have added them all and they appear on the Address List with unique addresses



7) Creating a Program to Test Two Color Sensors

Now that the color sensors have unique I2C addresses, we can write a simple program to display some unique values from each sensor on our drive station phone. Our team uses Android Studio, so that is what we will be showing here. Below is the program that we wrote to test and confirm that our two color sensors would work independently.

```
package org.firstinspires.ftc.robotcontroller.external.samples;

import com.qualcomm.robotcore.eventloop.opmode.OpMode;
import com.qualcomm.robotcore.hardware.ColorSensor;
import com.qualcomm.robotcore.hardware.I2cAddr;

public class TwoColorSensors extends OpMode {

    ColorSensor Color0;
    ColorSensor Color1;

    public void init() {
        Color0 = hardwareMap.colorSensor.get("LeftColor");
        Color0.setI2cAddress(I2cAddr.create7bit(0x1e));  //7-bit address for 0x3c
        Color1 = hardwareMap.colorSensor.get("RightColor");
        Color1.setI2cAddress(I2cAddr.create7bit(0x26));  //7-bit address for 0x4c
    }

    public void loop() {
        telemetry.addData("Left Side Reads: ", Color0.alpha());
        telemetry.addData("Right Side Reads: ", Color1.alpha());
    }
}
```

I've highlighted three lines in this program that were new to us and are required for the I2C address changes to work in Android Studio. The most important thing to point out is that the I2C addresses declared in the program are **different** than what you assigned to the color sensors in the MR Core Device Discovery software. The reason for the difference is that the Modern Robotics software assigned an 8-bit hexadecimal address, but the Android Studio requires the address to be declared as a 7-bit hexadecimal address.

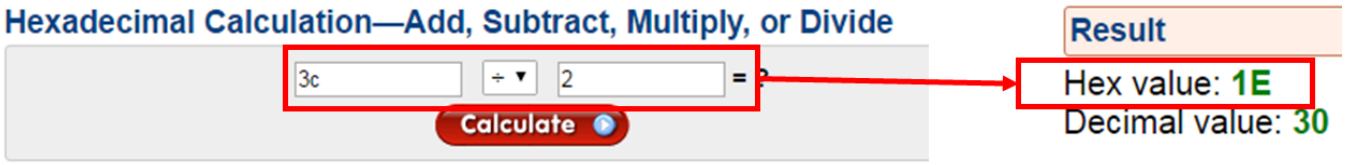
It really wasn't clear to us why this is the case, but in the next step we will show you how to convert from 8-bit hexadecimal to 7-bit hexadecimal addresses.

REMINDER: Also remember to add your new program to the registerMyOpModes file. We named our program **Test 2 Sensors** and then downloaded the program onto the robot controller phone.

8) Converting from 8-bit to 7-bit Hexadecimal Address

To convert from 8-bit to 7-bit hexadecimal, you need to divide the 8-bit address by two. While that may sound simple, it was not very clear/obvious to us how to do this. We found a website that helped us do the conversion at: http://www.calculator.net/hex-calculator.html

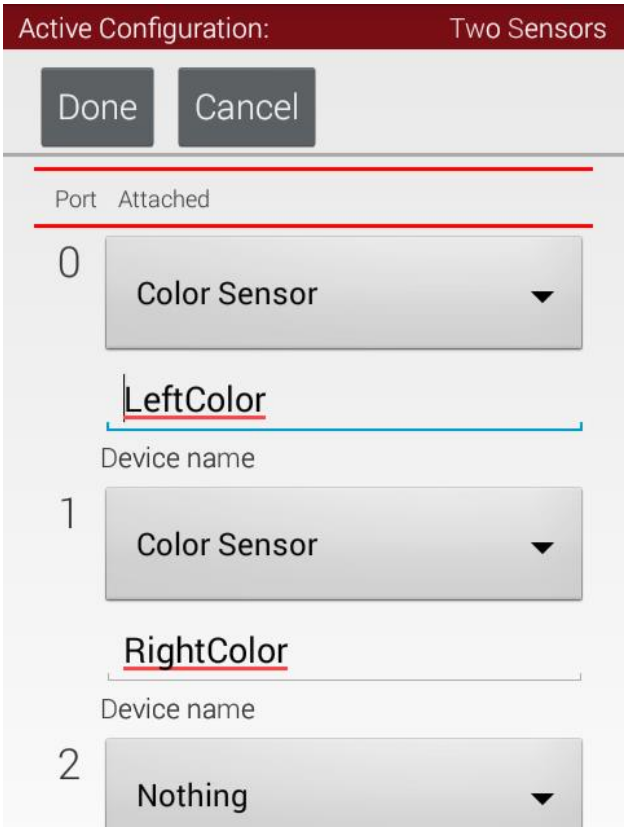
We needed to convert the following two 8-bit addresses: **0x3c** and **0x4c** Below we show how we used the website to calculate the 7-bit address of 0x3c. In the calculator, you just enter the last 2 alphanumeric characters of the 8-bit address and click **Calculate**. The result of 1E means that 0x3c (8-bit) = 0x1e (7-bit).



We followed the same steps for the second address of 0x4c and determined that the equivalent 7-bit address was 0x26. So that is why you see 0x1e and 0x26 declared as the I2C addresses in our program on the previous page.

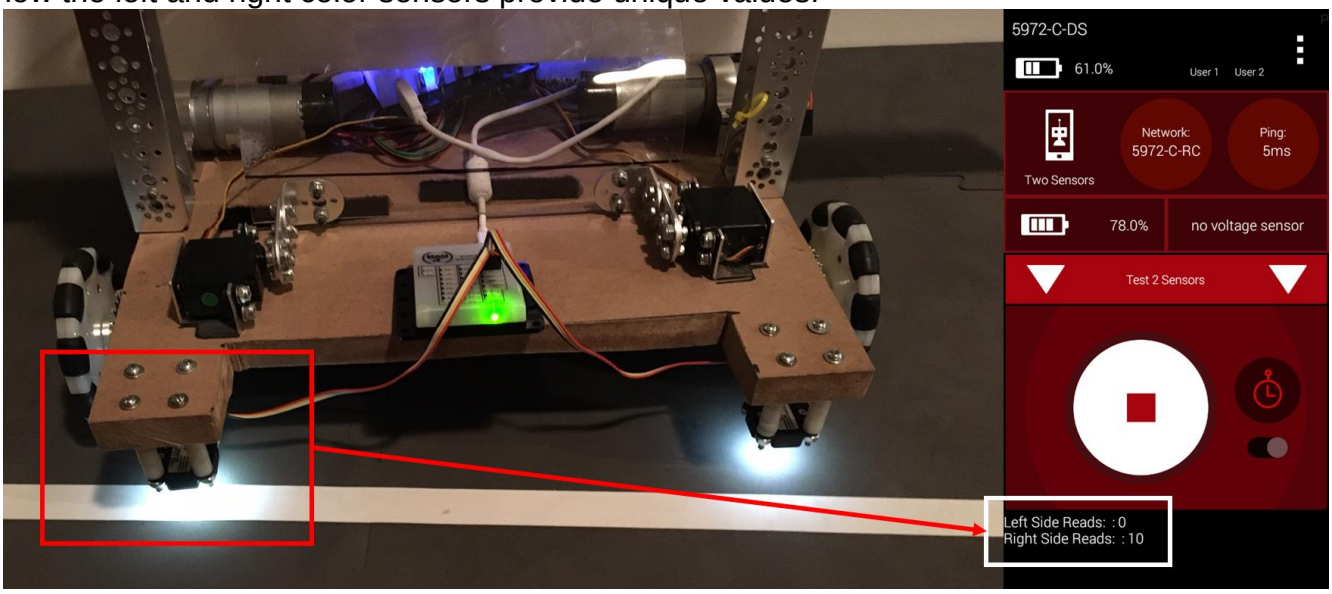
9) Create Configuration on Robot Controller Phone

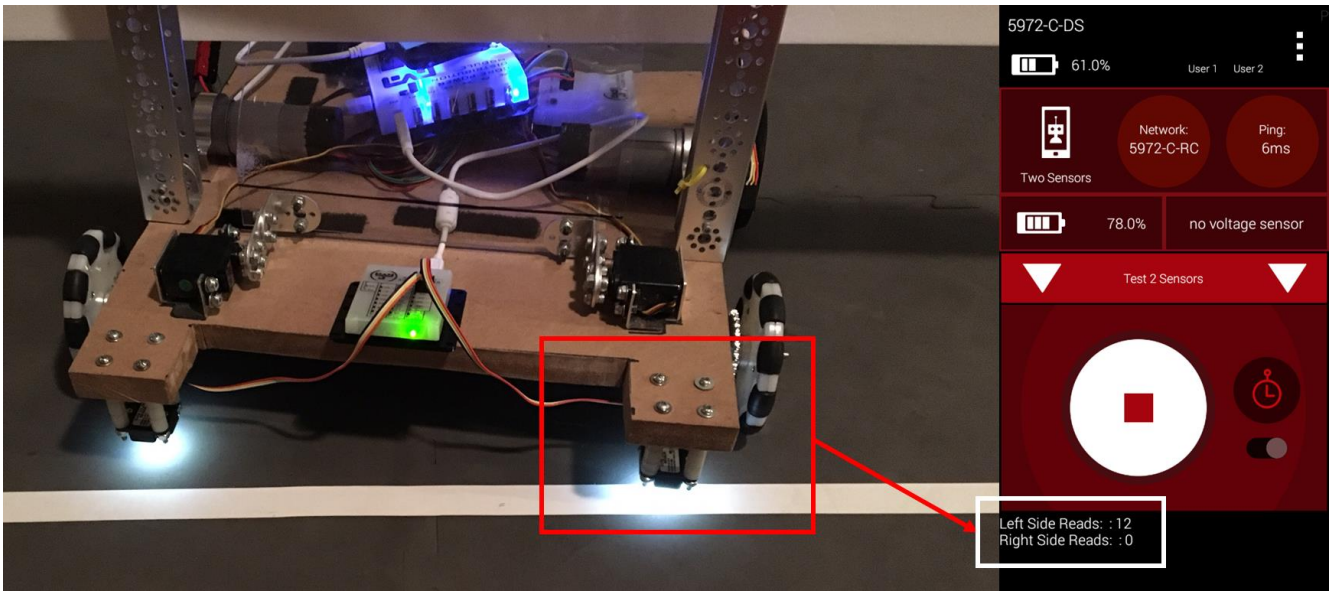
- Connect the robot controller phone to the PDM and make sure the power is turned on the PDM.
- To run our test program, we created the following configuration on our robot controller phone and named the configuration **Two Sensors**



10) Running Program and Seeing Unique Values for Color Sensors

After all of this preparation, we are now ready to see how the color sensors work independently of each other. When we run the **Test 2 Sensors** program, you can see below how the left and right color sensors provide unique values.





Hopefully this information has been helpful and will allow your team to configure multiple sensors to use on your robot.

Good luck to everyone this season!