



Autonomous Robot Racer: Scratch Mode



by TCPL

In this project, you'll learn how to make an autonomous wall-detecting robot. This robot uses an Arduino Nano and Scratch to avoid walls and race around the track. The Tompkins County Public Library uses this robot to hold weekly robot racing events. This is one of several autonomous robot racers developed at TCPL. More robot Instructables coming soon!

Supplies:

Laser Cut Files:

- [Chassis](#)
- [Wheels](#)

3D File:

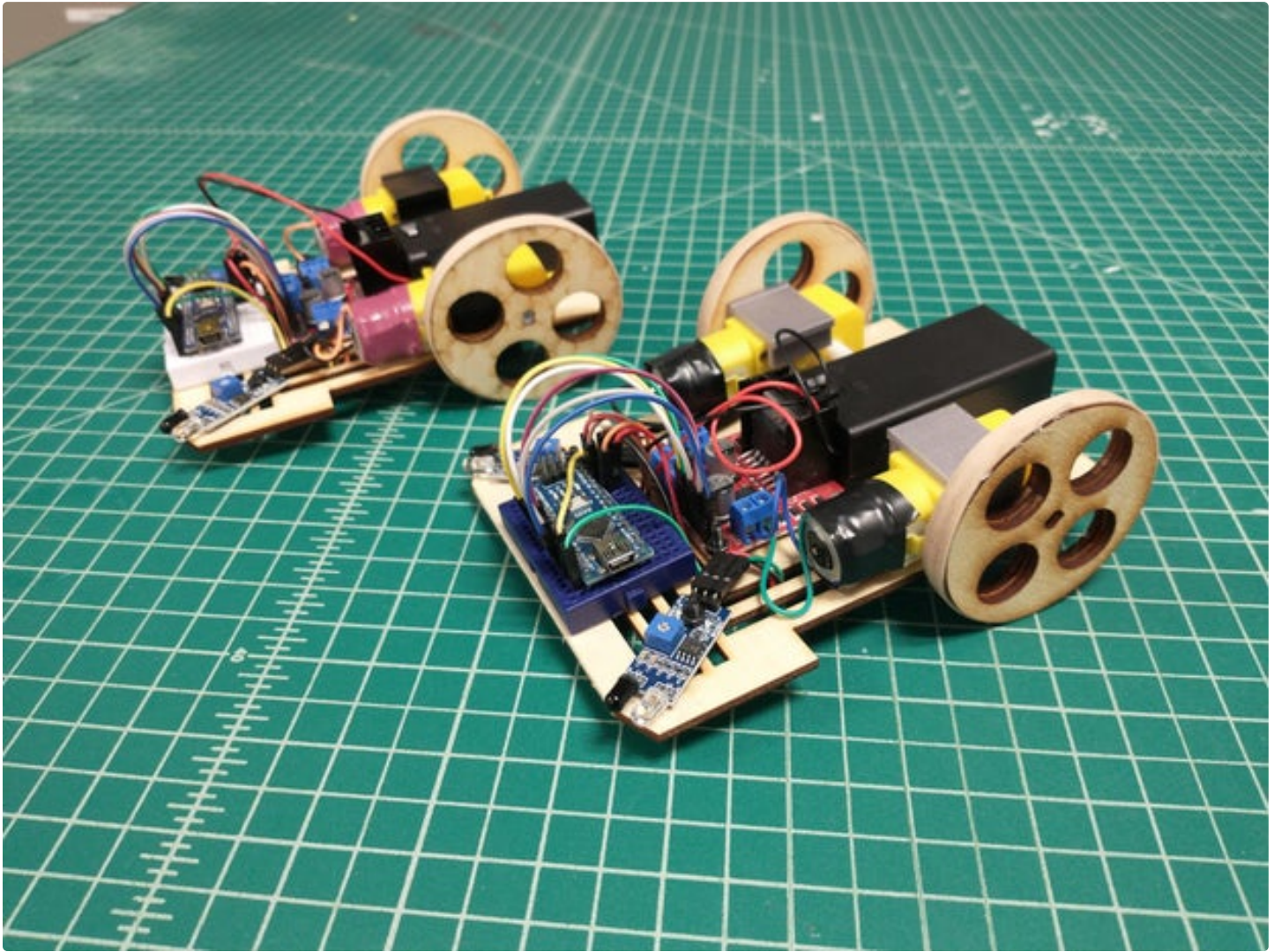
- [Breaks](#)
- [Motor Clips](#)

Electronics and Other Parts:

- [9V Battery Case with Switch](#)
- [L298N Motor Driver](#)
- [DC Motors with Gearbox](#)
- [Breadboard](#)
- [Breadboard wires](#)
- [Arduino Nano](#)
- [IR Obstacle Avoidance Sensors](#)
- [M3 nuts and bolts](#)
- [Size #64 rubberbands](#)
- [Rechargeable 9v battery](#)
- 12 x 12 plywood
- Electrical Tape

Tools:

- Wire stripper
- Plastic nippers
- Hot glue gun
- Phillips head screwdriver
- USB 2.0 cable

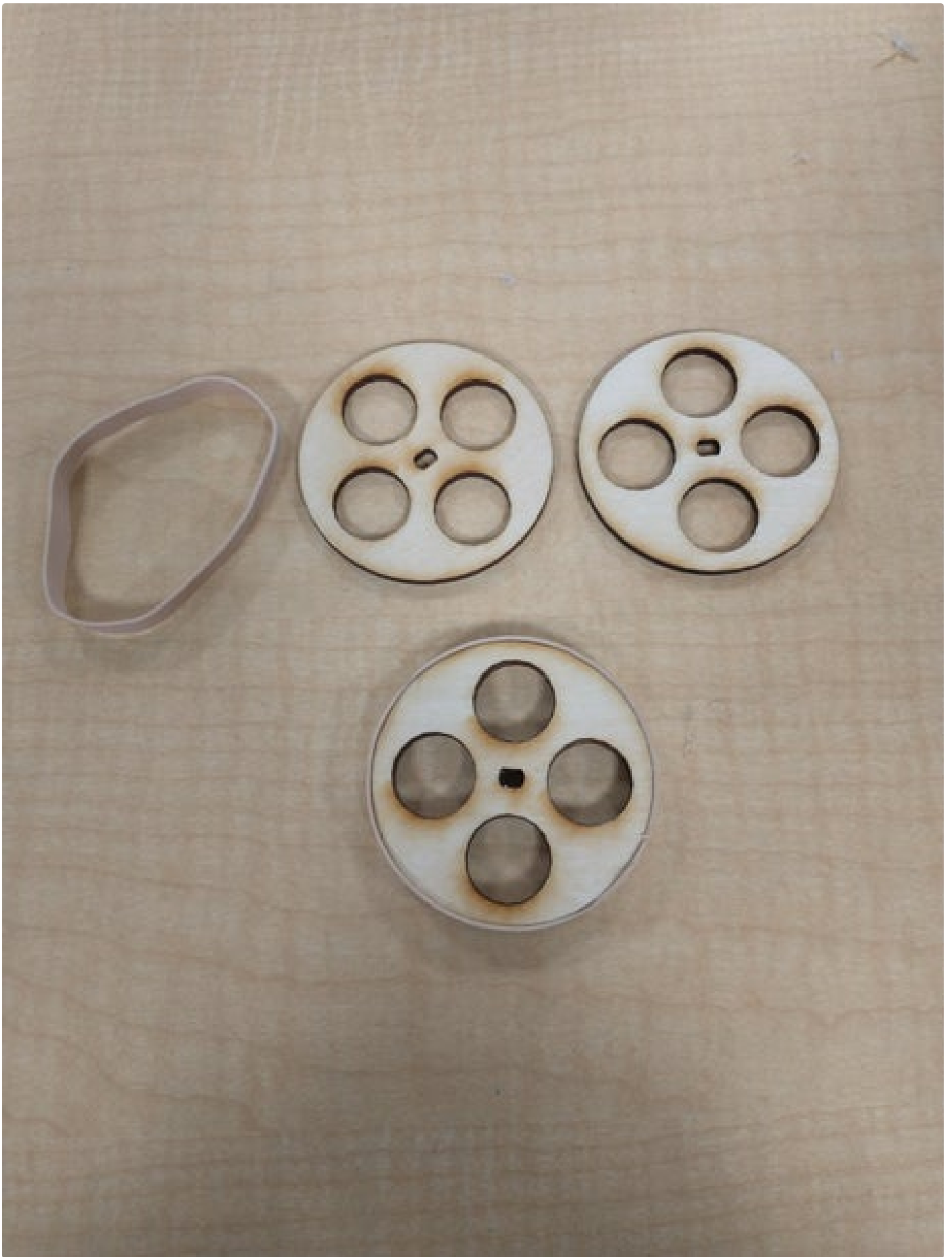


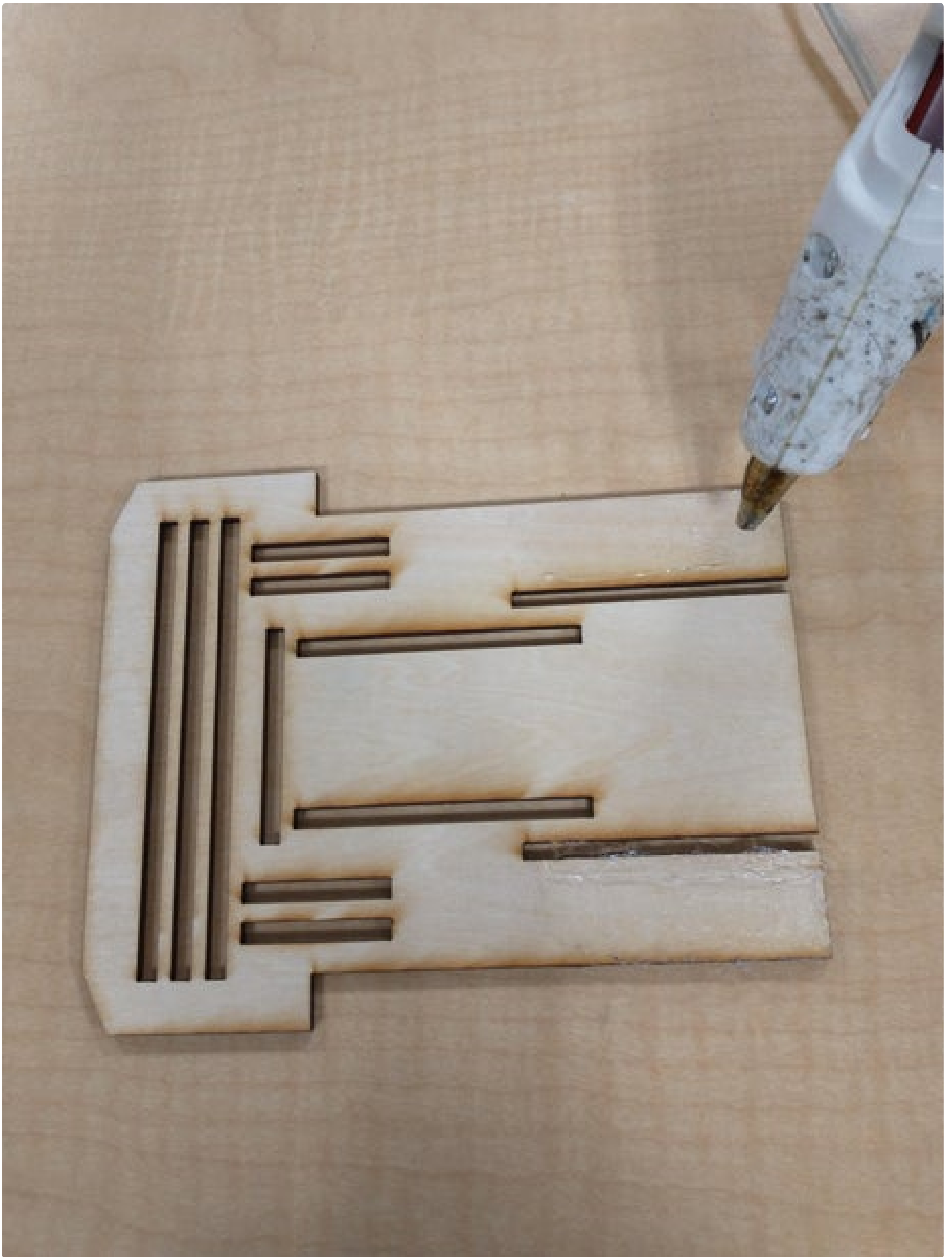
Step 1: Prepare Laser Cut and 3D Printed Parts

At TCPL we are privileged to have access to an Epilog Laser and Prusa 3D printers. Please check your local Makerspace or Library for public-access laser cutters and 3D printers.

1. Download the **Laser Cut Files** in the **Supplies** section from Tinkercad as an SVG file.
2. Upload files to your favorite vector software. TCPL uses [Inkscape](#). Prepare the files based on the requirements of the laser you are using.
3. Laser cut the parts on a 12x12 1/8 inch plywood board.
4. Once the parts are cut out hot-glue matching wheel pairs together.
5. Stretch a rubber band over each wheel.
6. Apply and spread hot glue along the rectangles at the back of the chassis. See the second photo for an example. This will keep the motors from sliding around and make them adjustable.
7. Download the **3D Print File** in the **Supplies** section from Tinkercad as an STL file.
8. Import the STL in your favorite slicer. Make sure to set the infill to 15% and supports on.
9. Once the print is complete clean off the supports.

Note: If you are unable to access a laser you can print the SVG files out and use a scroll or coping saw and drill to cut out the parts.



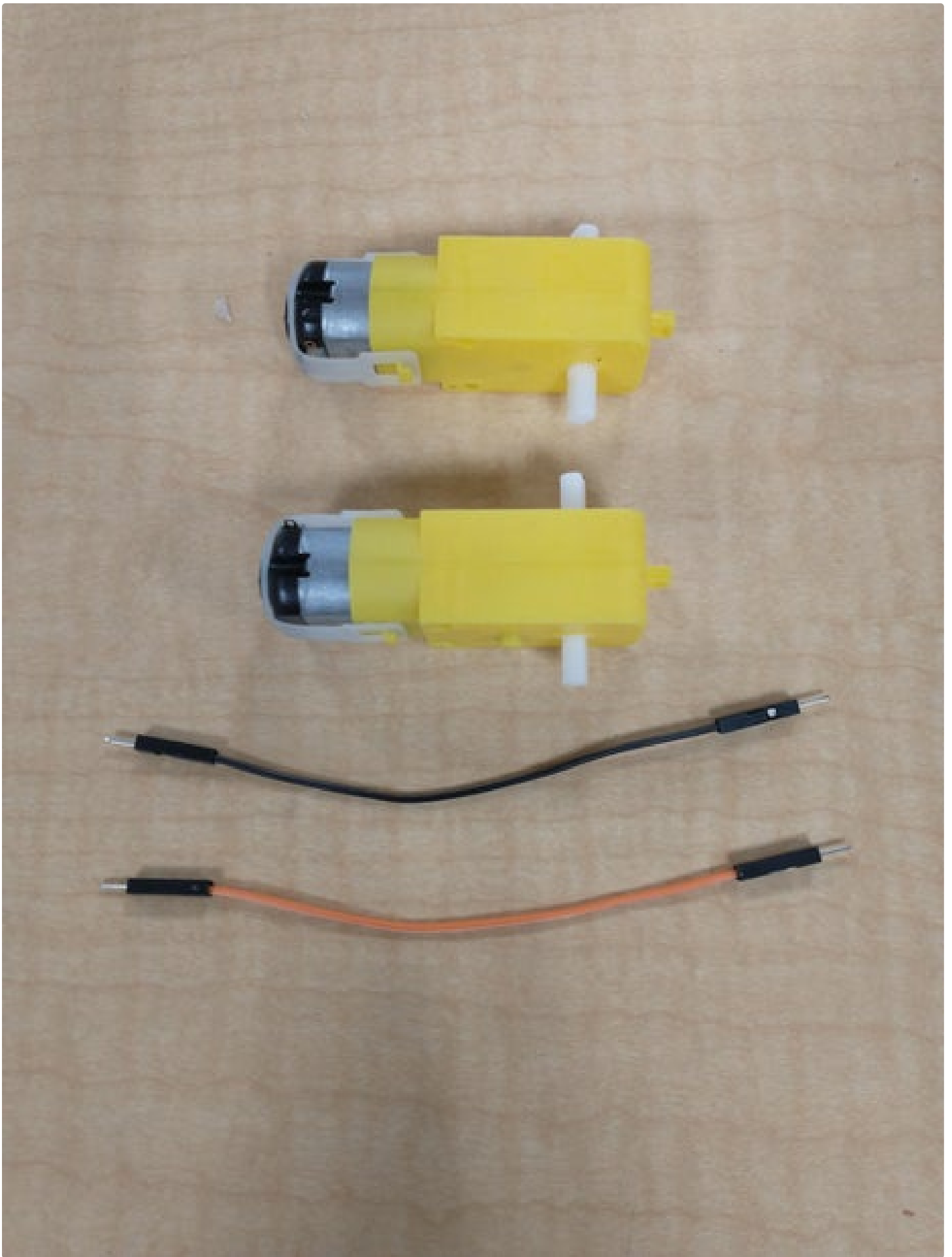


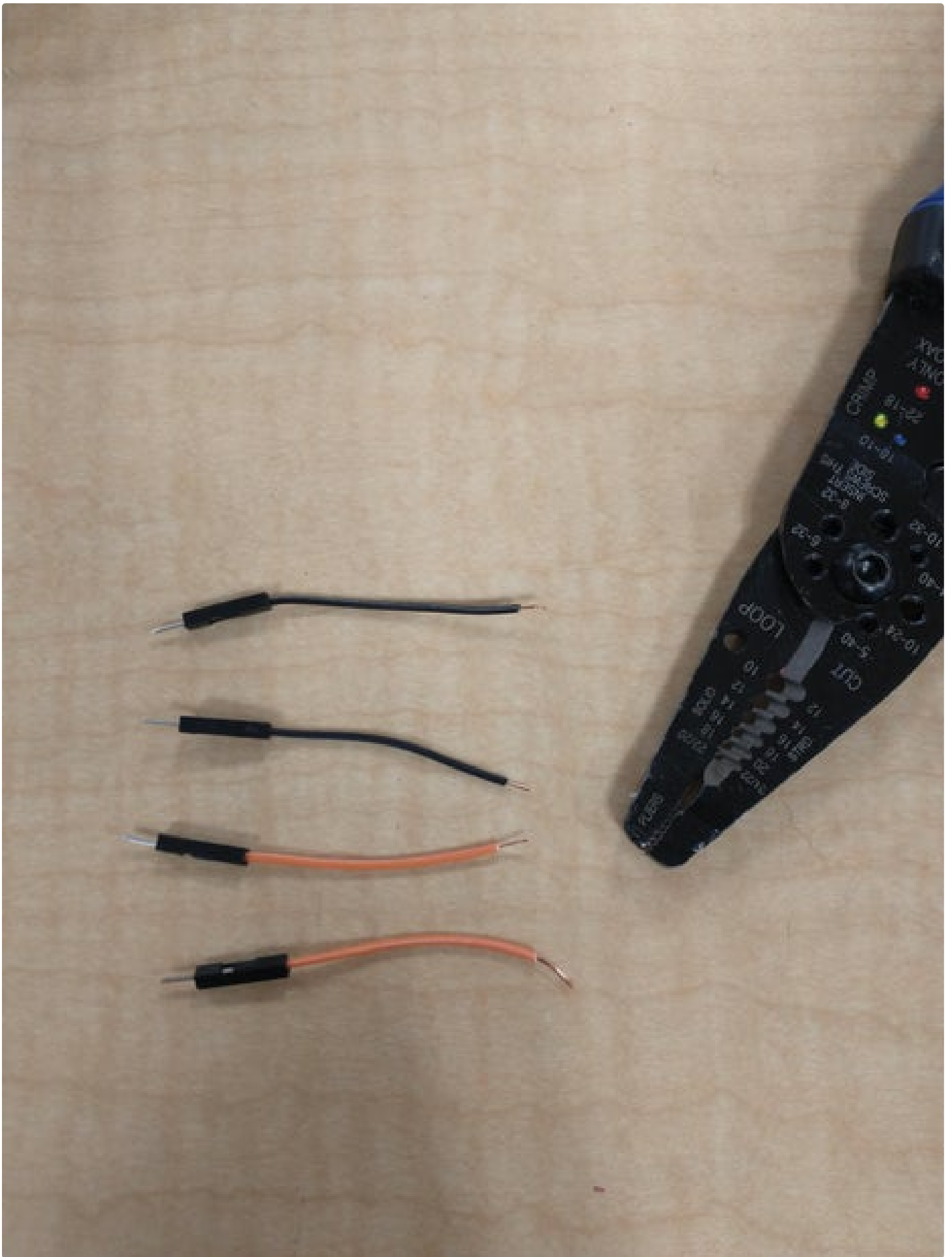


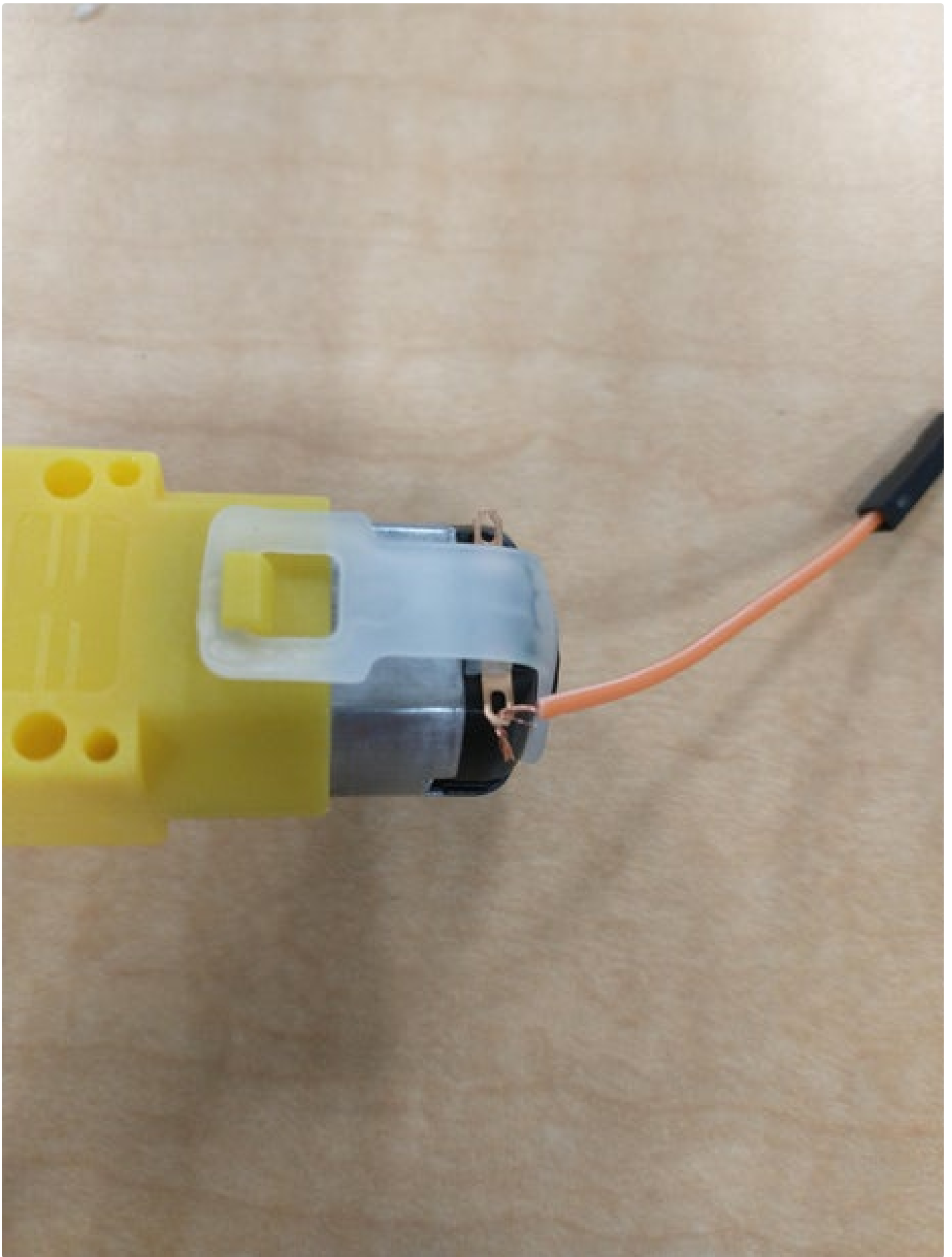
Step 2: Wire Motors

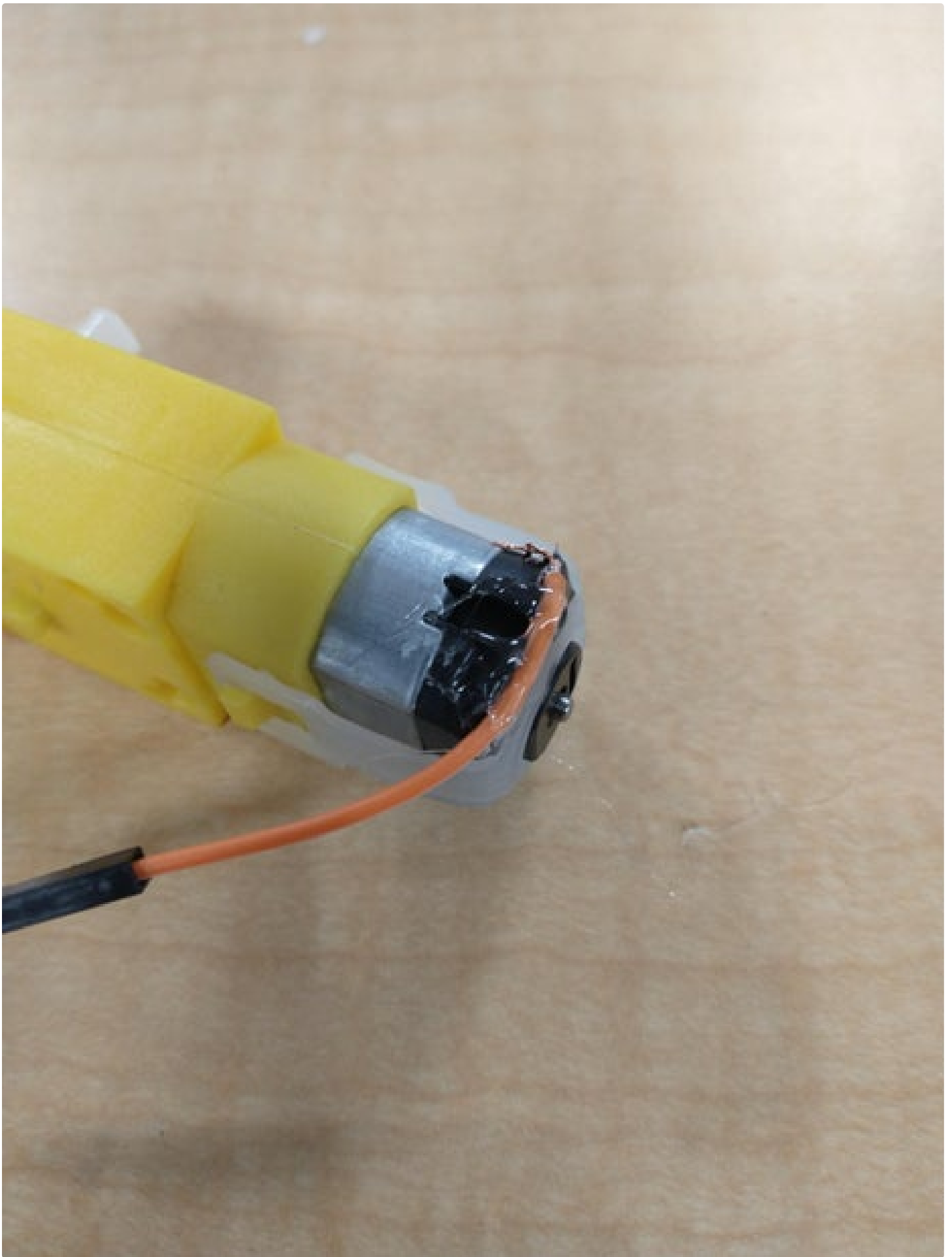
At TCPL we have developed a way to attach wires to the DC motors without the need for solder. This was done to streamline repairs during races. If you are comfortable using solder attach the wires that way.

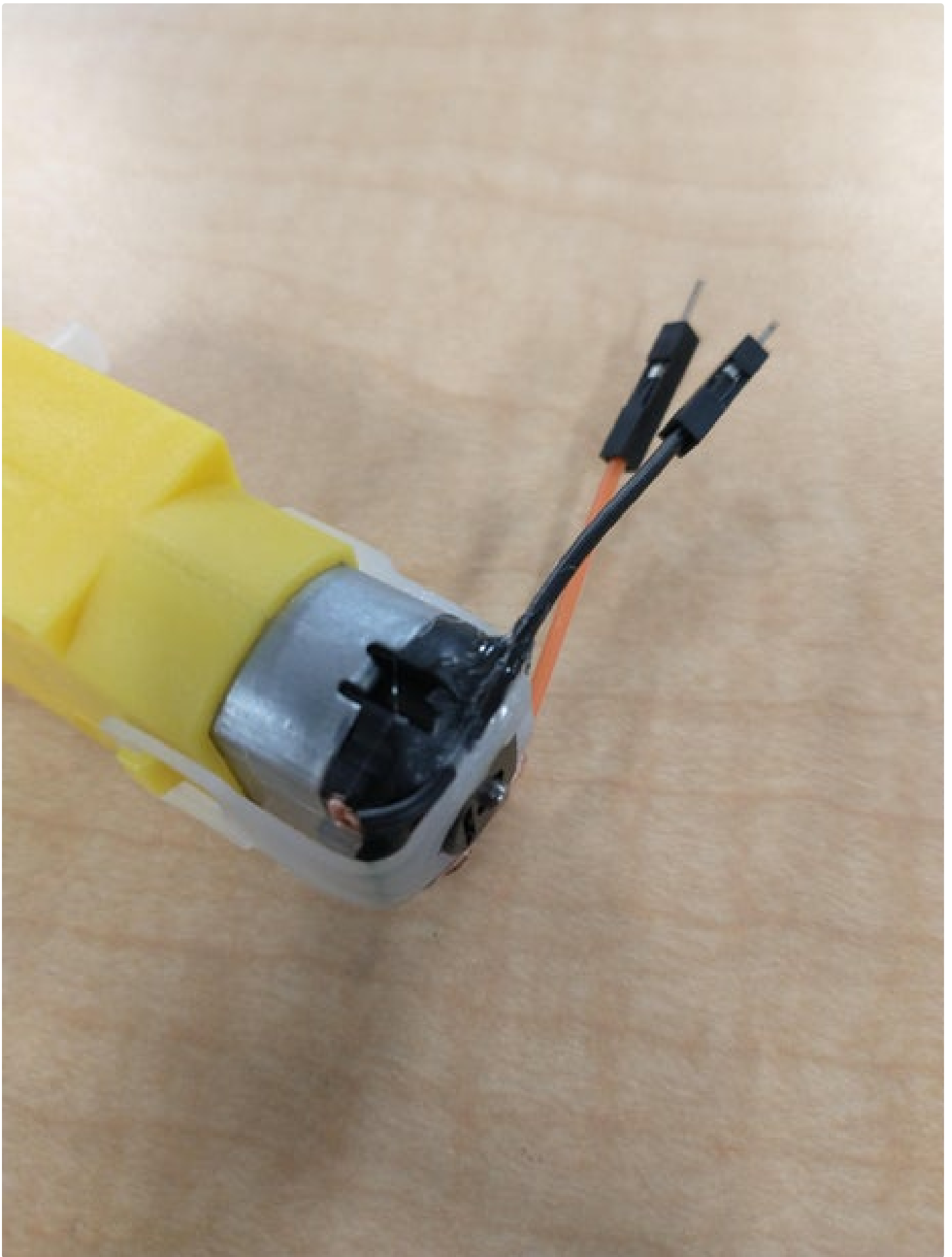
1. Take two M-to-M wires and cut them in half and strip each end without a pin.
2. Twist the wire brush into a tight wrap and slide it into the DC motor input. See the second picture for an example.
3. Once the brush is secure in the DC motor input lightly pull it down the side of the motor bell and glue it in place. Be careful not to get glue on the motor input of the shaft. See the third picture for an example.
4. Repeat the process for the other DC motor input so there are two wires attached to it.
5. Tightly wrap the electrical tape around the bell of the motor to cover the exposed wire brushes.
6. Repeat this process for the second motor.
7. Once both motors are ready slide a motor clip on each. You will see a round circle protruding from one side of the motor, the open side of the clip goes around this. See the 9th picture for an example.
8. Slide a clip onto each motor.

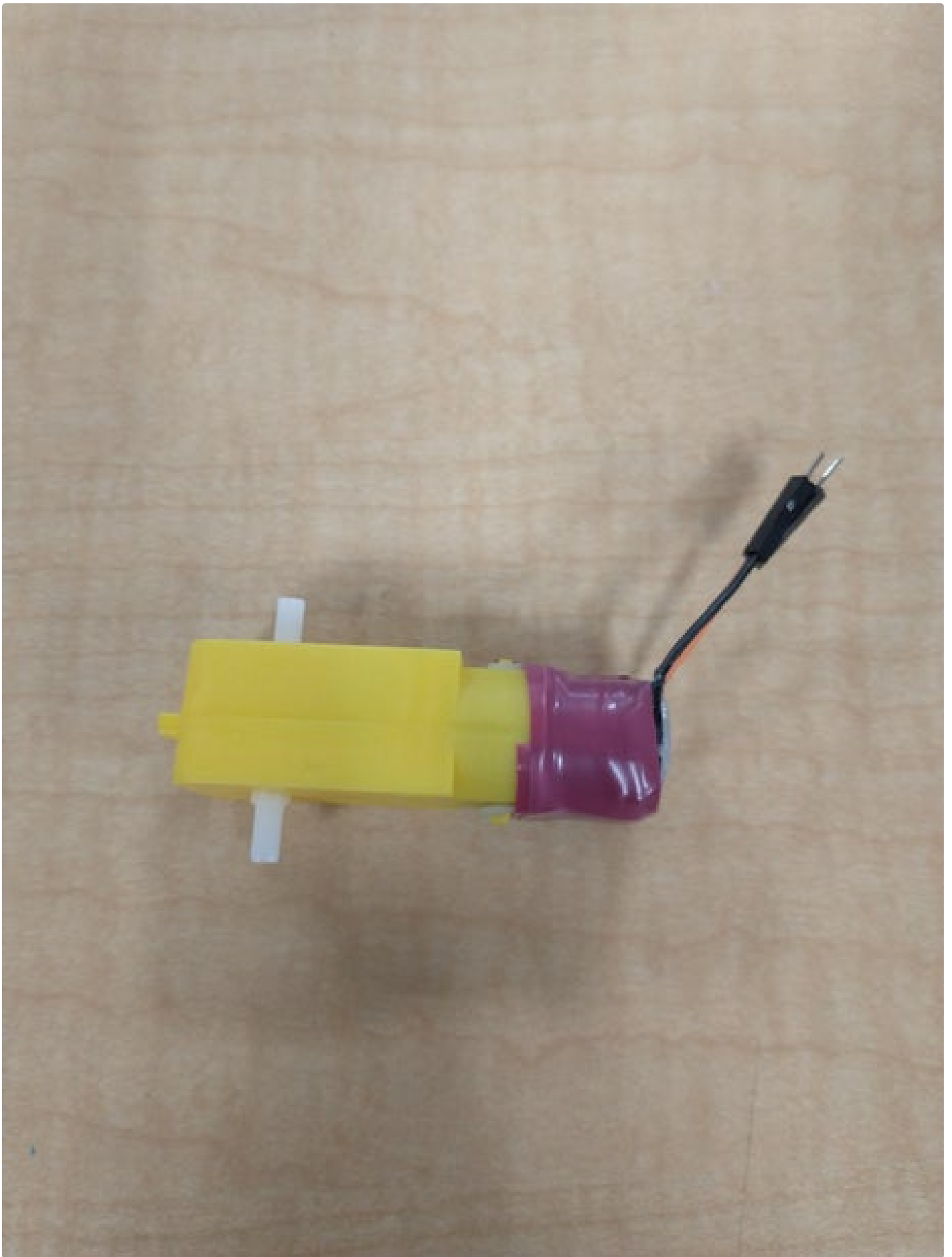


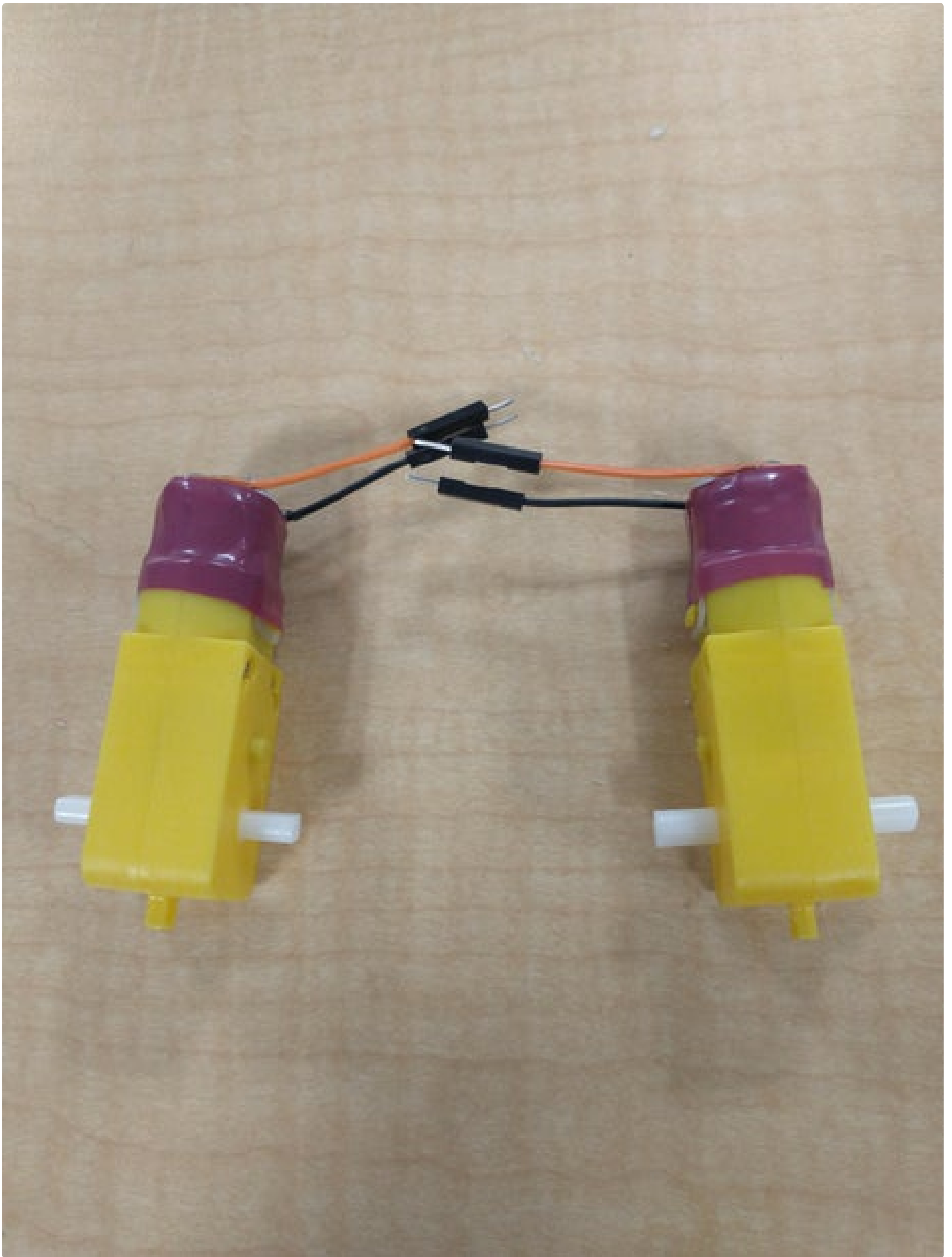


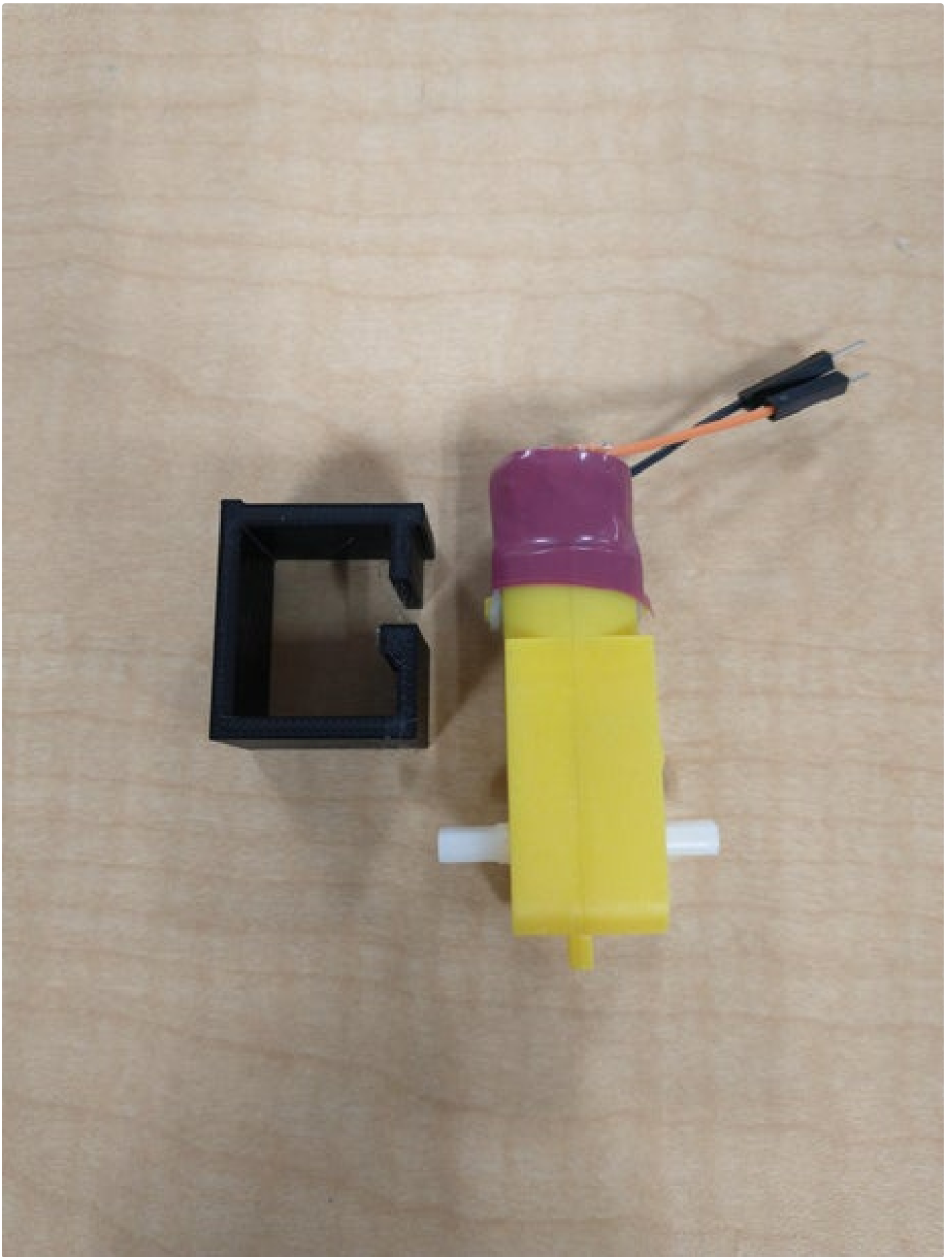


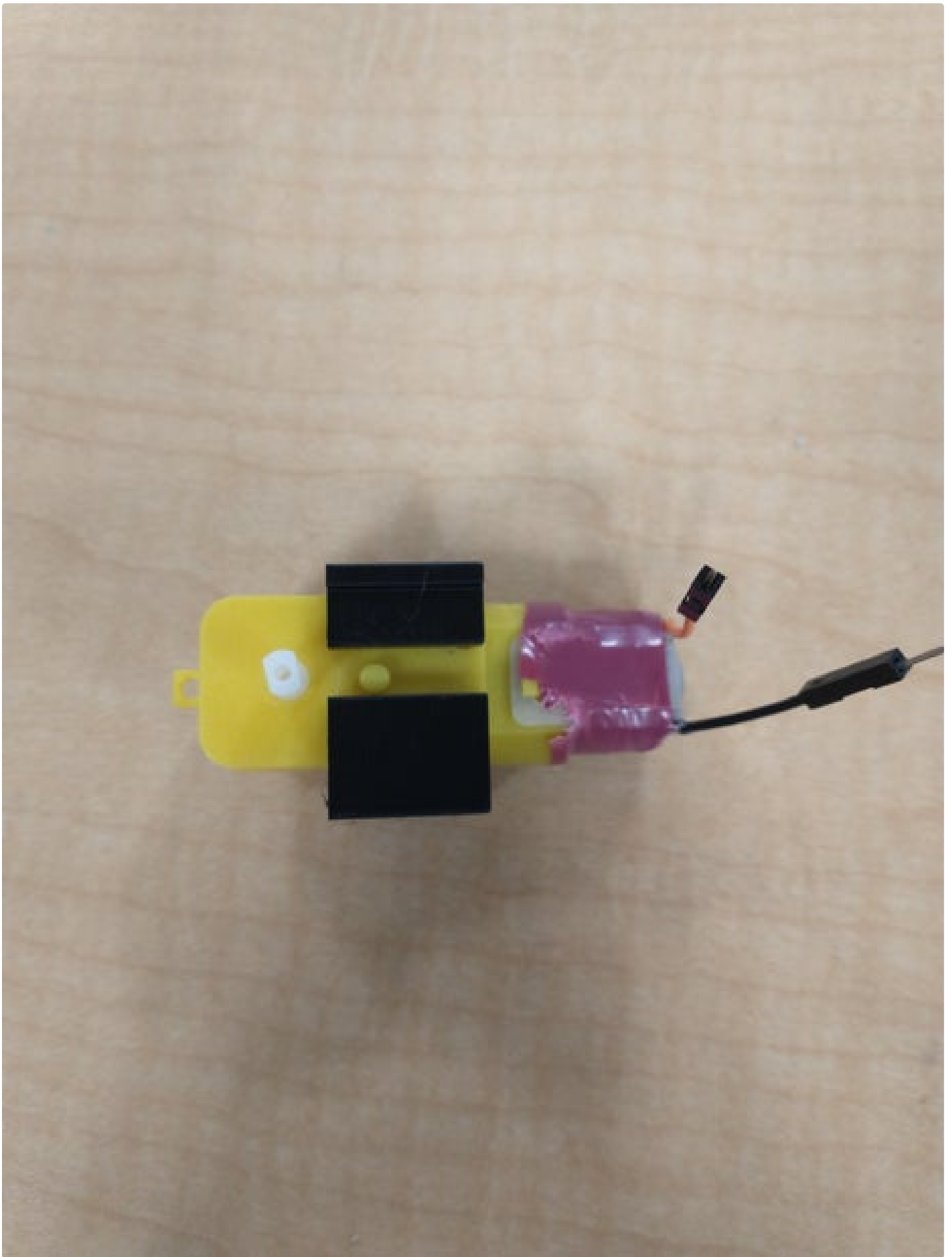


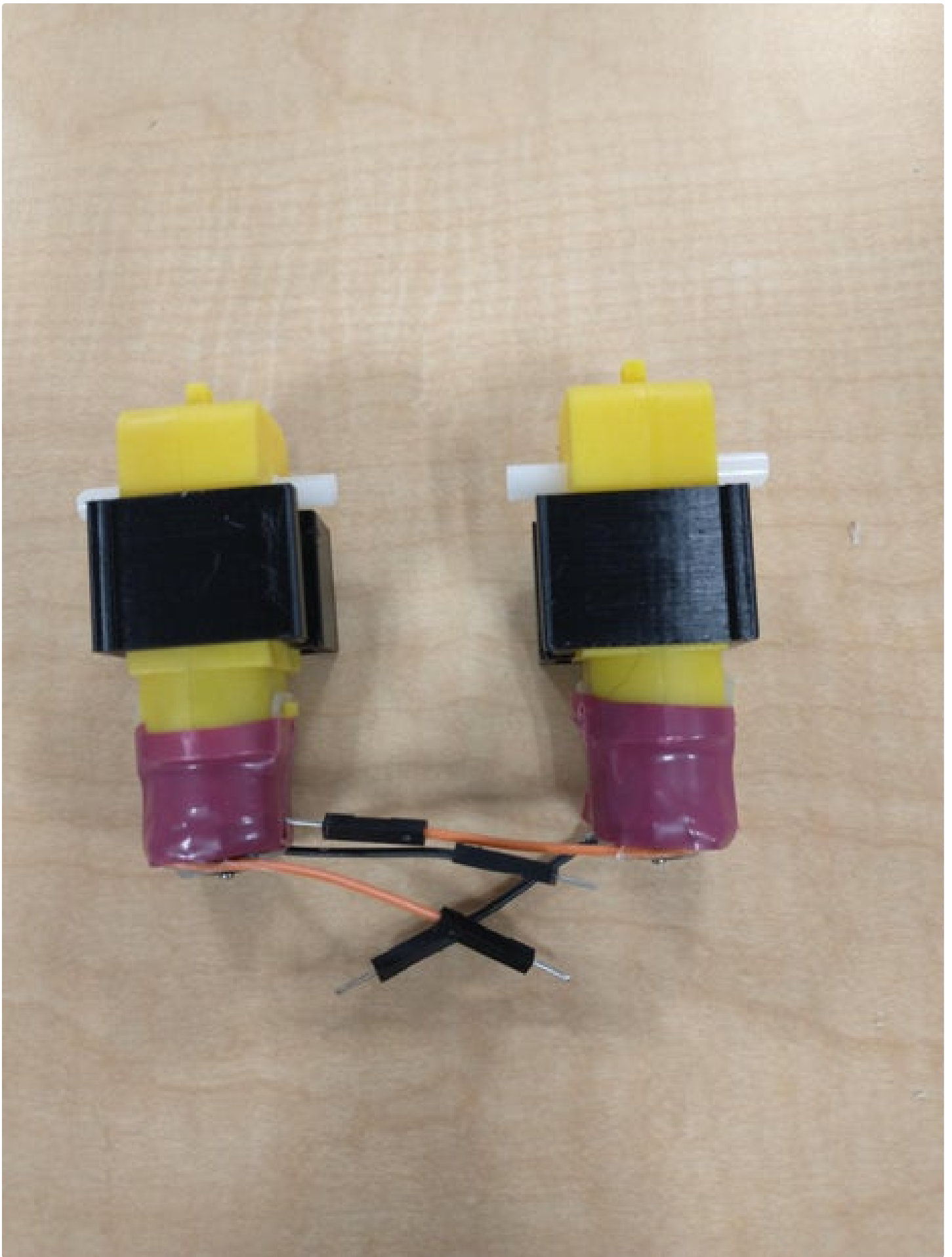






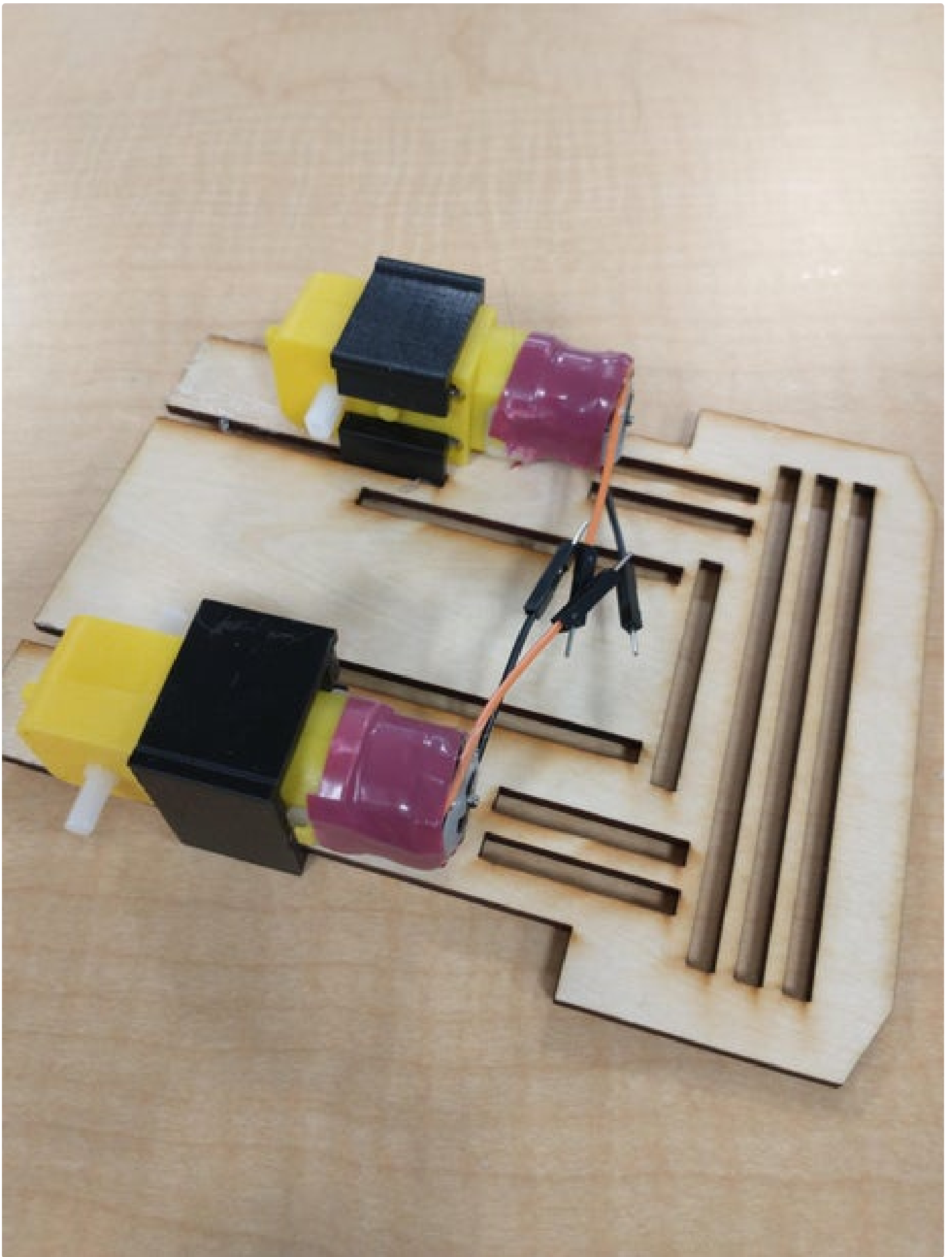


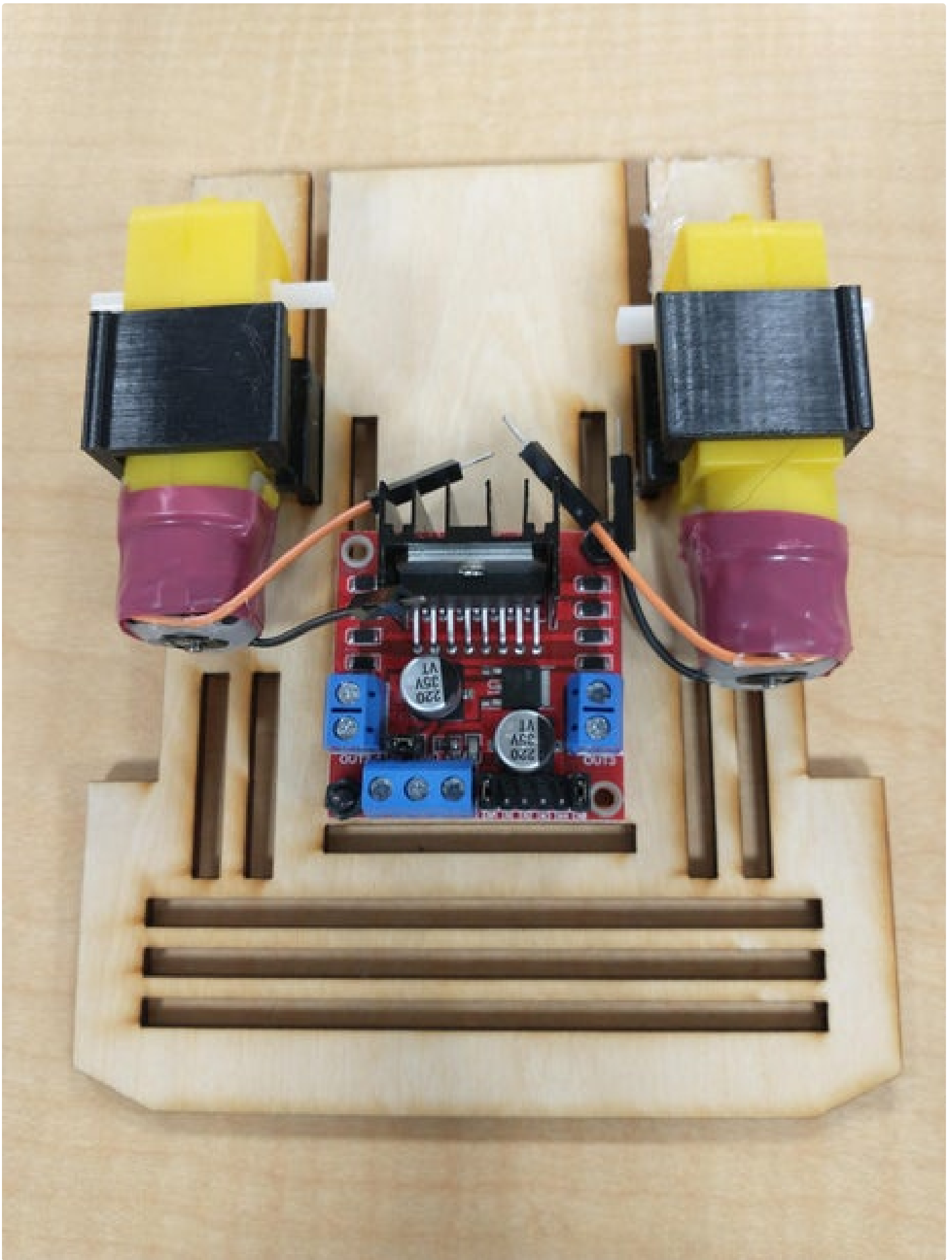


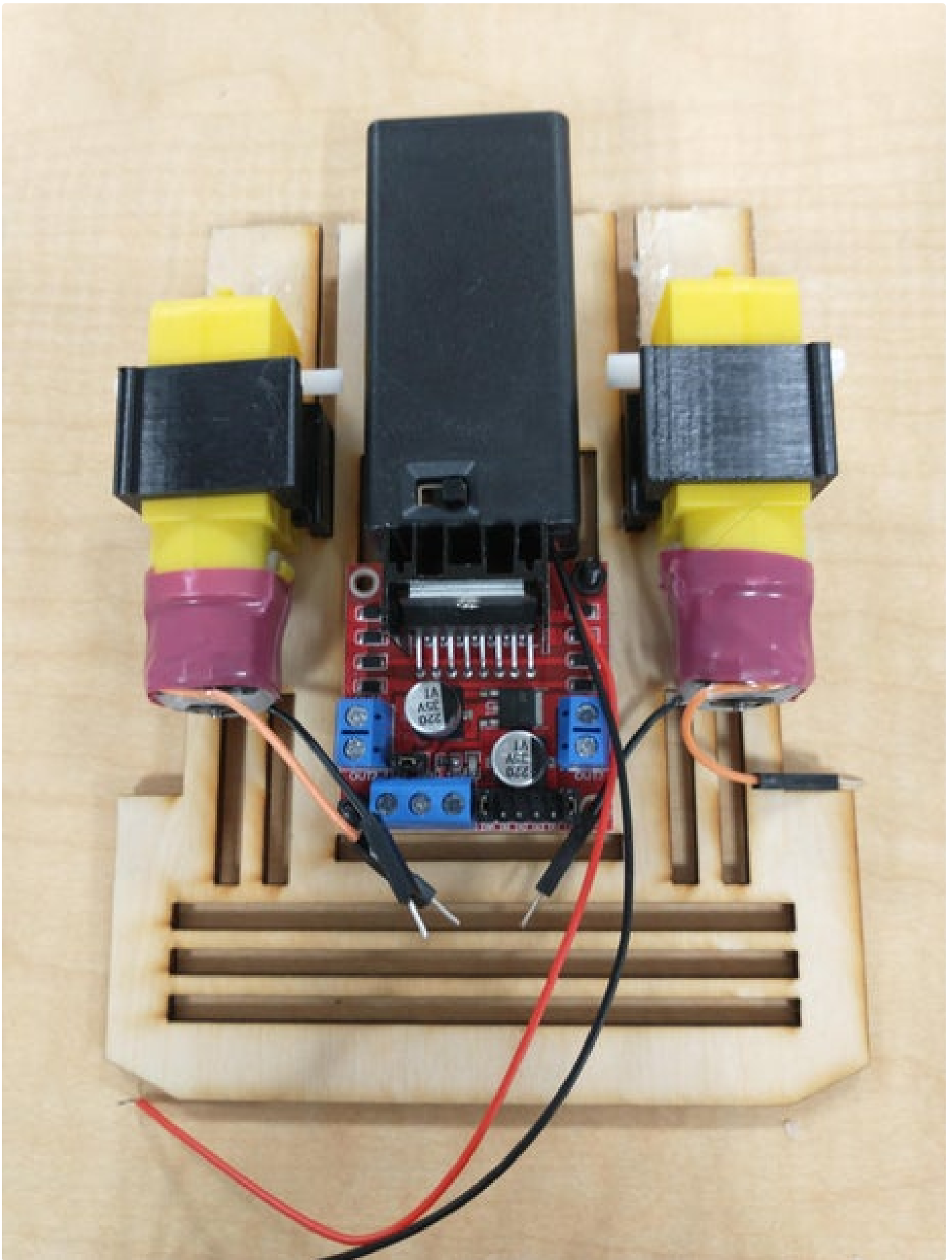


Step 3: Assemble Chassis

1. Slide motors onto the rear of the chassis. See the first picture for orientation.
2. Using two screws and nuts secure the L293N motor driver to the center of the chassis. See the second image for orientation.
3. Hot glue the 9v battery case to the chassis so the switch is adjacent to the motor driver.

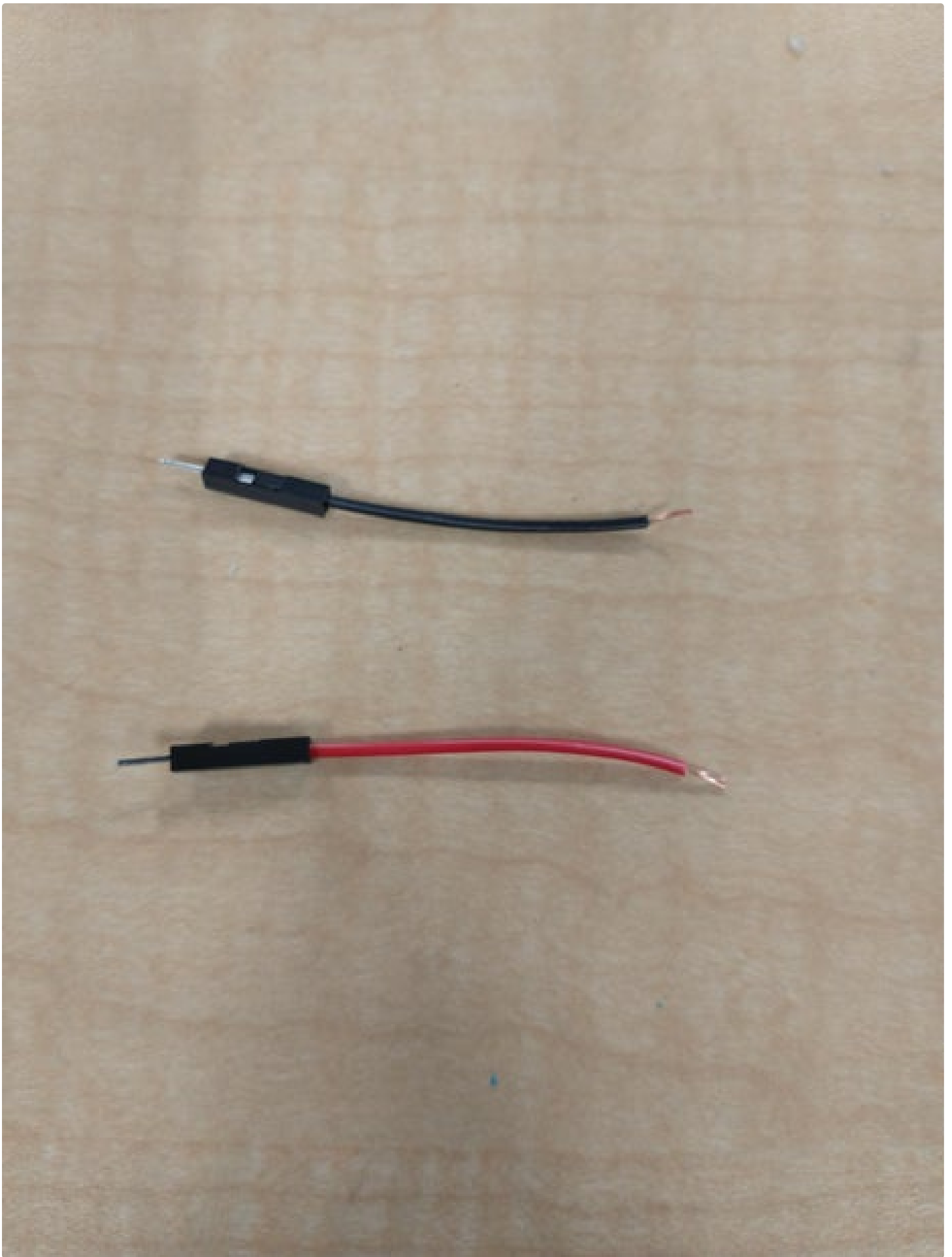


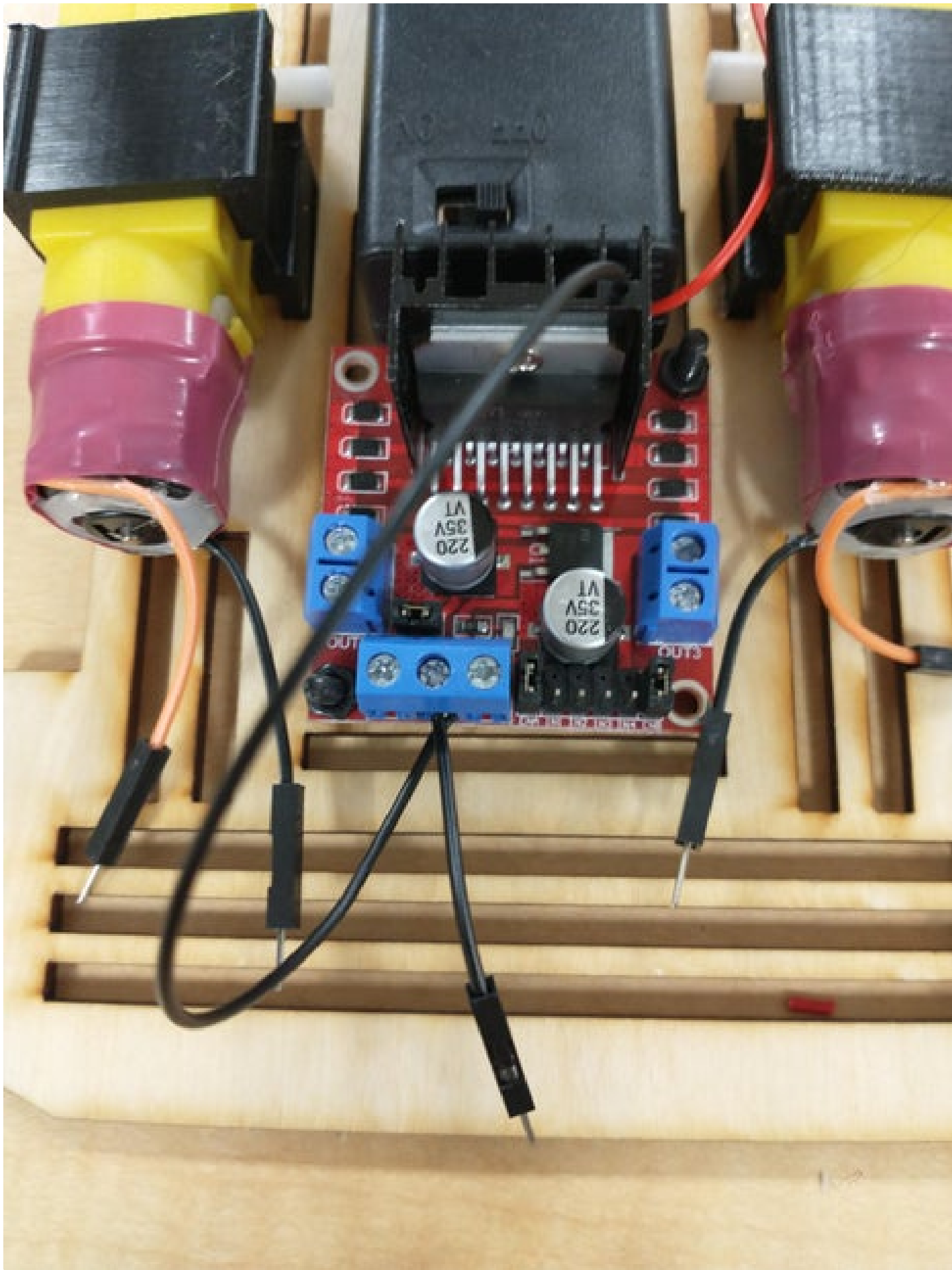


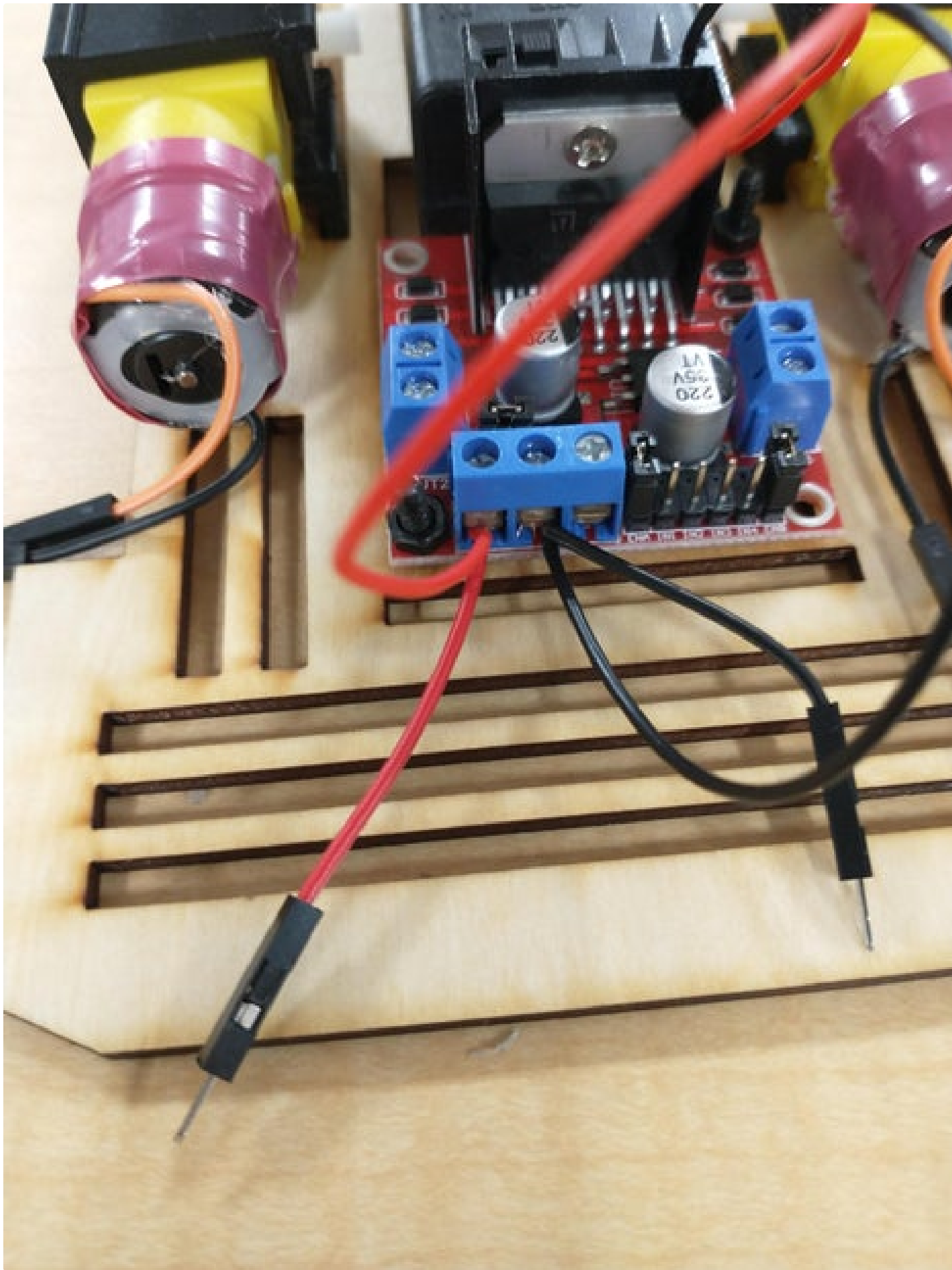


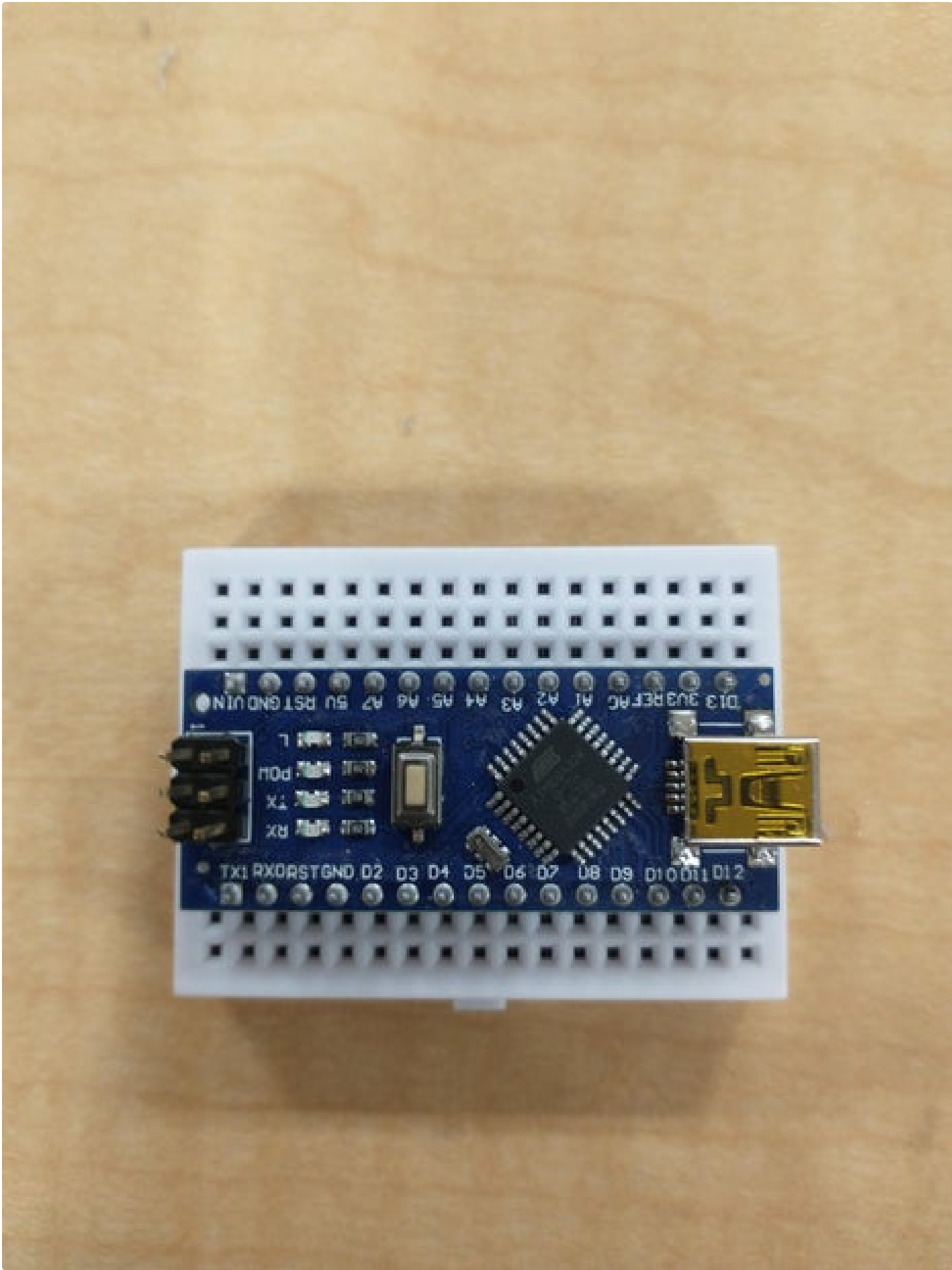
Step 4: Power Distribution Circuit

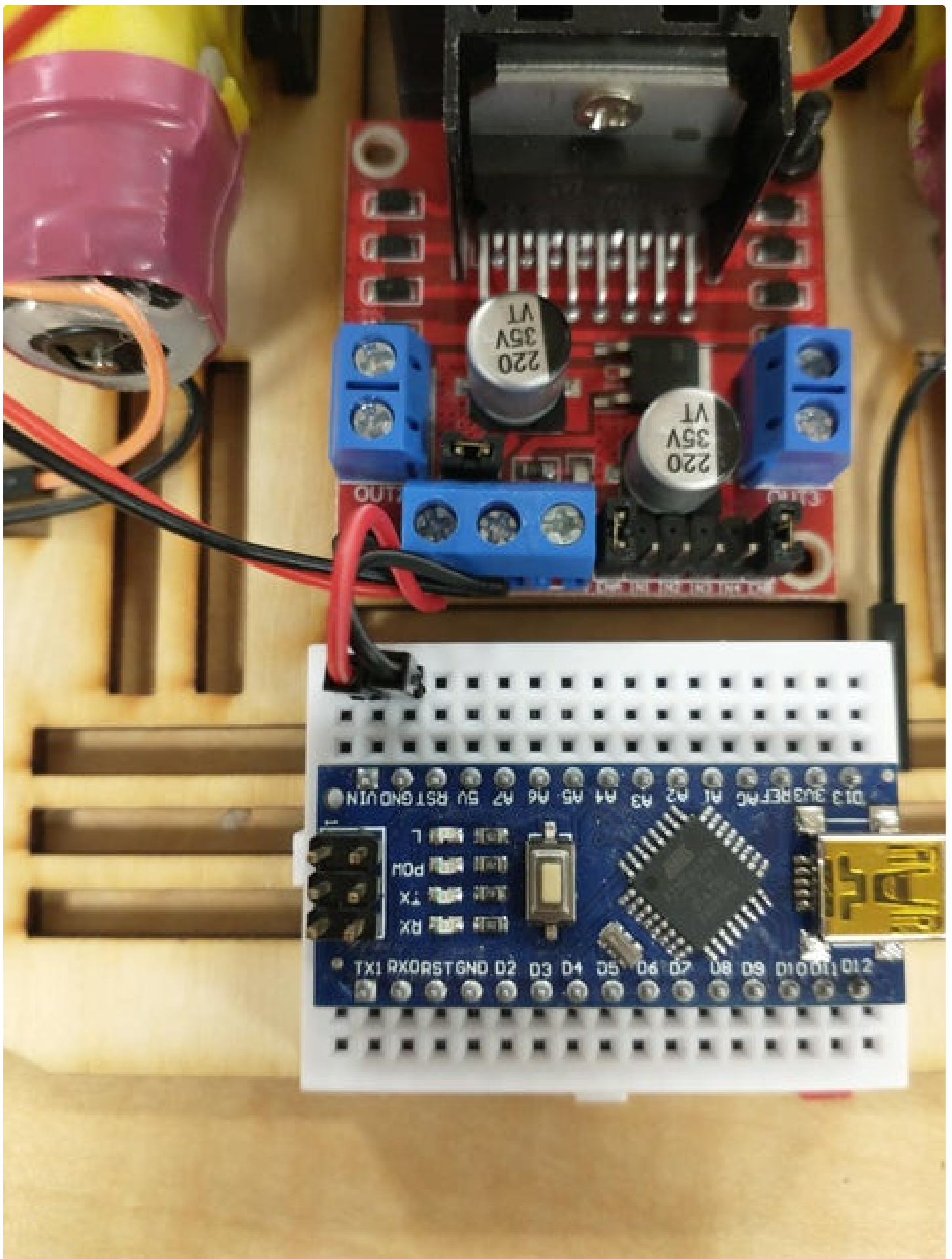
1. Take a red and black wire that has pins on both ends. Cut each in half. You will only use a half of each. You can save the other halves for backup.
2. Take the red wire and strip the end without a pin.
3. Take the black wire and strip the end without a pin.
4. Take the black wire coming from the 9v battery pack and strip it. Twist the black wire with a pin and the one from the battery pack together.
5. Input the exposed end of the joined black wires into the **GND** port on the L298N motor driver. Secure the GND port by tightening its screw.
6. Take the red wire coming from the 9v battery pack and strip it. Twist the red wire with a pin and the one from the battery pack together.
7. Input the exposed end of the joined red wires into the **12V** port on the L298N motor driver. Secure the **12V** port by tightening its screw.
8. Press the Arduino Nano into the mini breadboard. Make sure the side with **VIN** pin label has three rows of inputs and the other side two. See the fourth picture for example.
9. Input the black wire coming from the **GND** port on the L298N motor driver next to the Arduino Nano **GND** port on the breadboard. Note: any one of the Arduino Nano GND ports will work. In this Instructable I put the next to the one on the VIN side of the Arduino Nano.
10. Input the red wire coming from the **12V** port on the L298N motor driver next to the Arduino Nano **VIN** port on the breadboard.







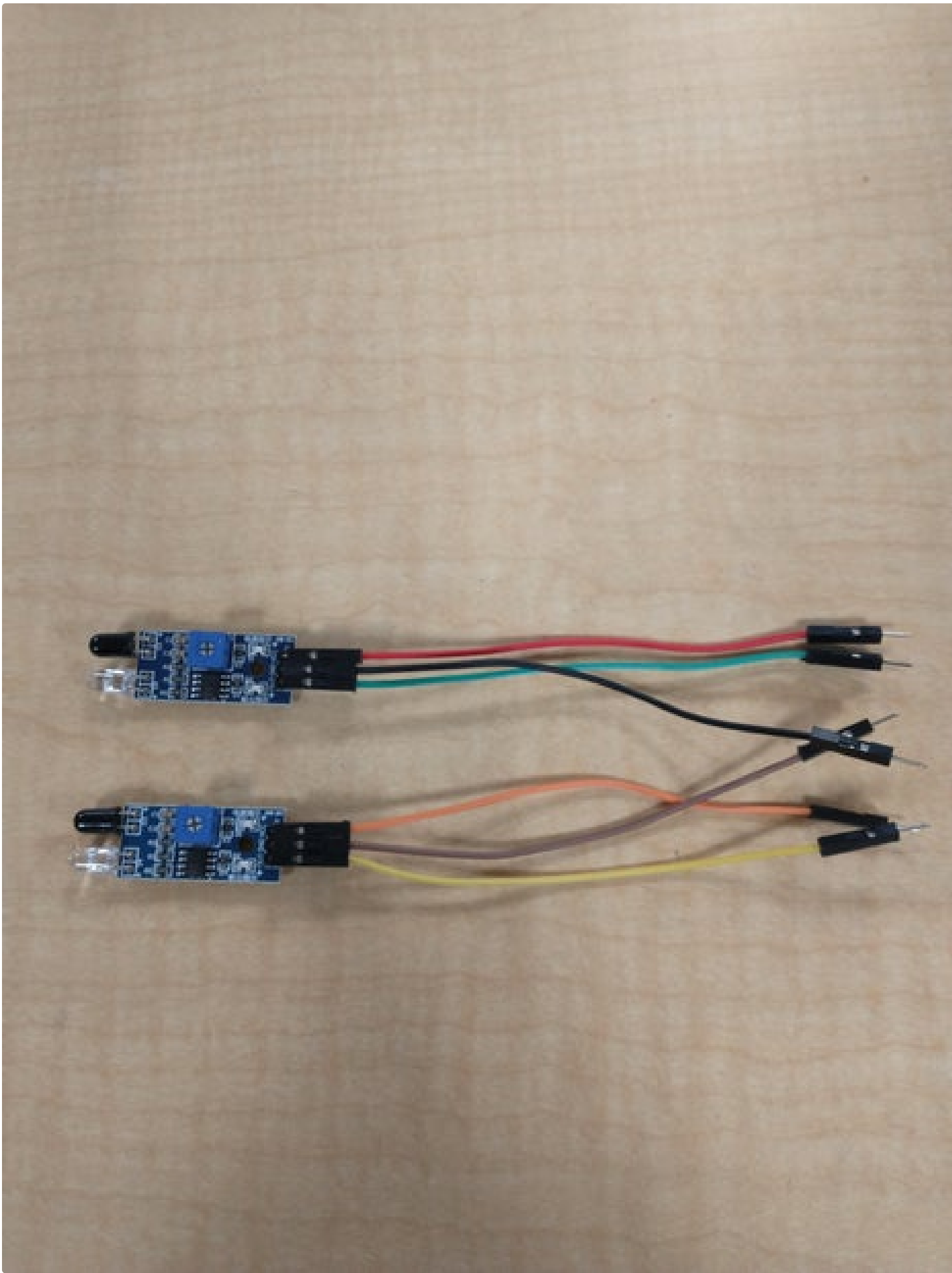


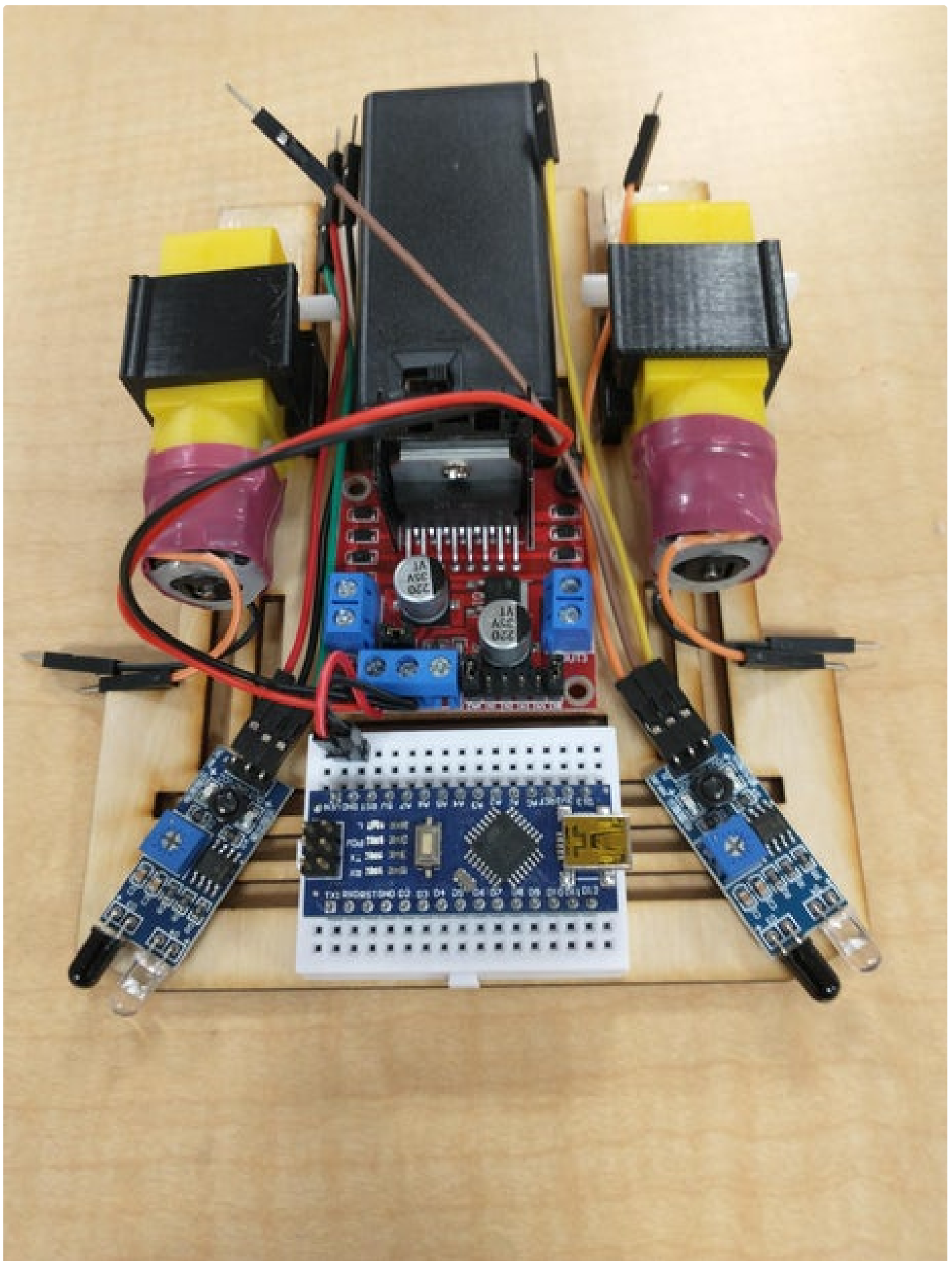


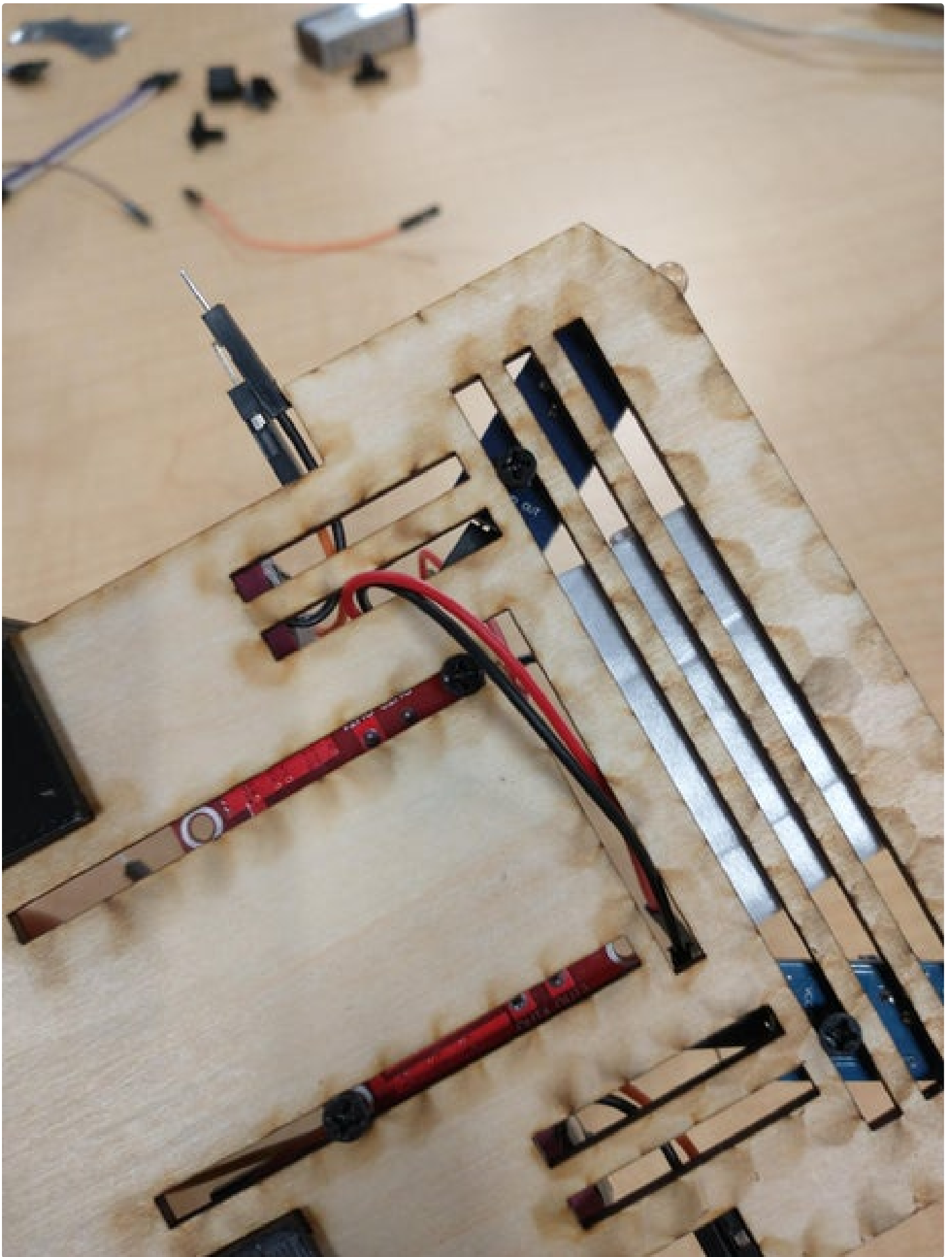
Step 5: Add Sensors

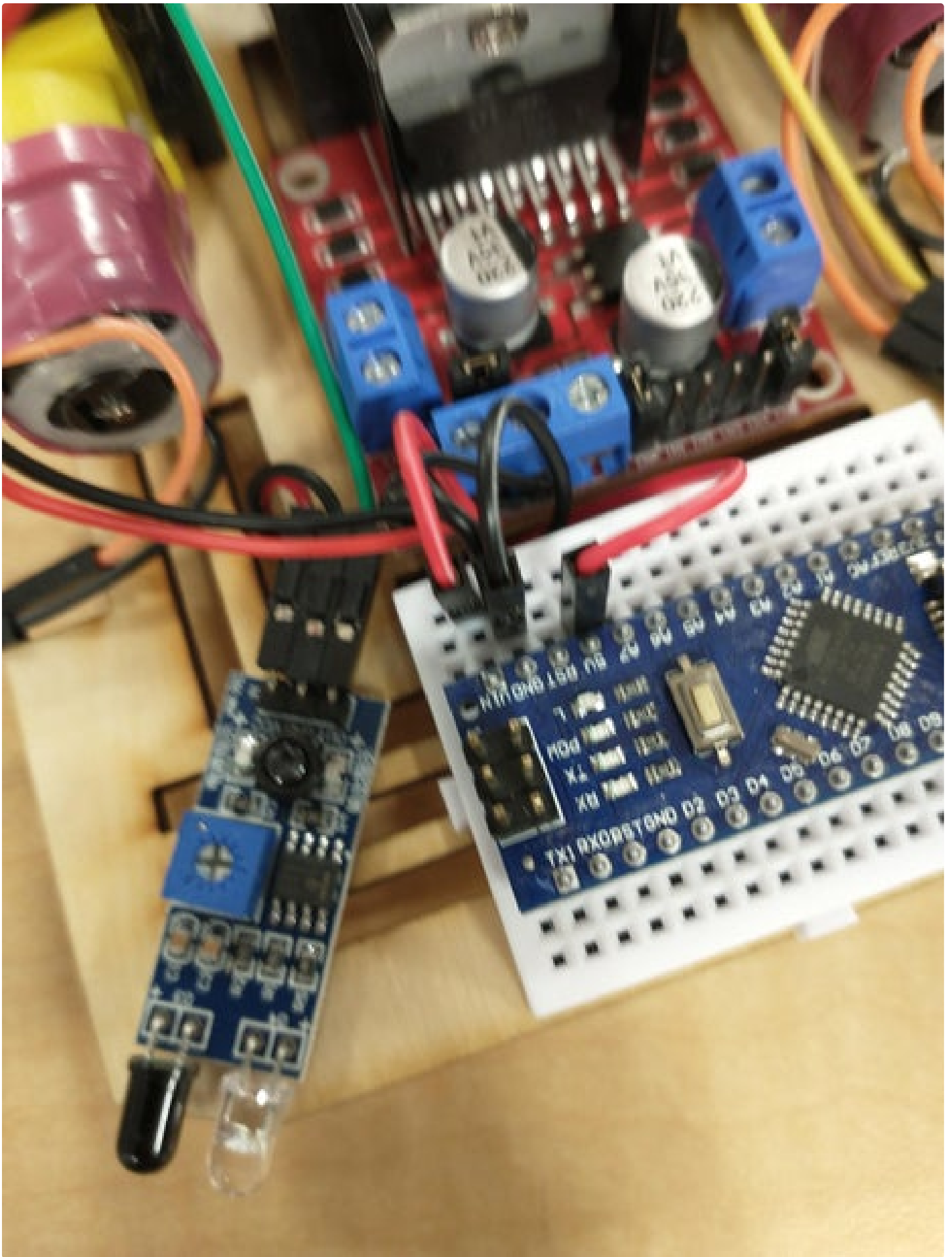
The orientation of the robot for this step is the front end facing you. See second picture for example.

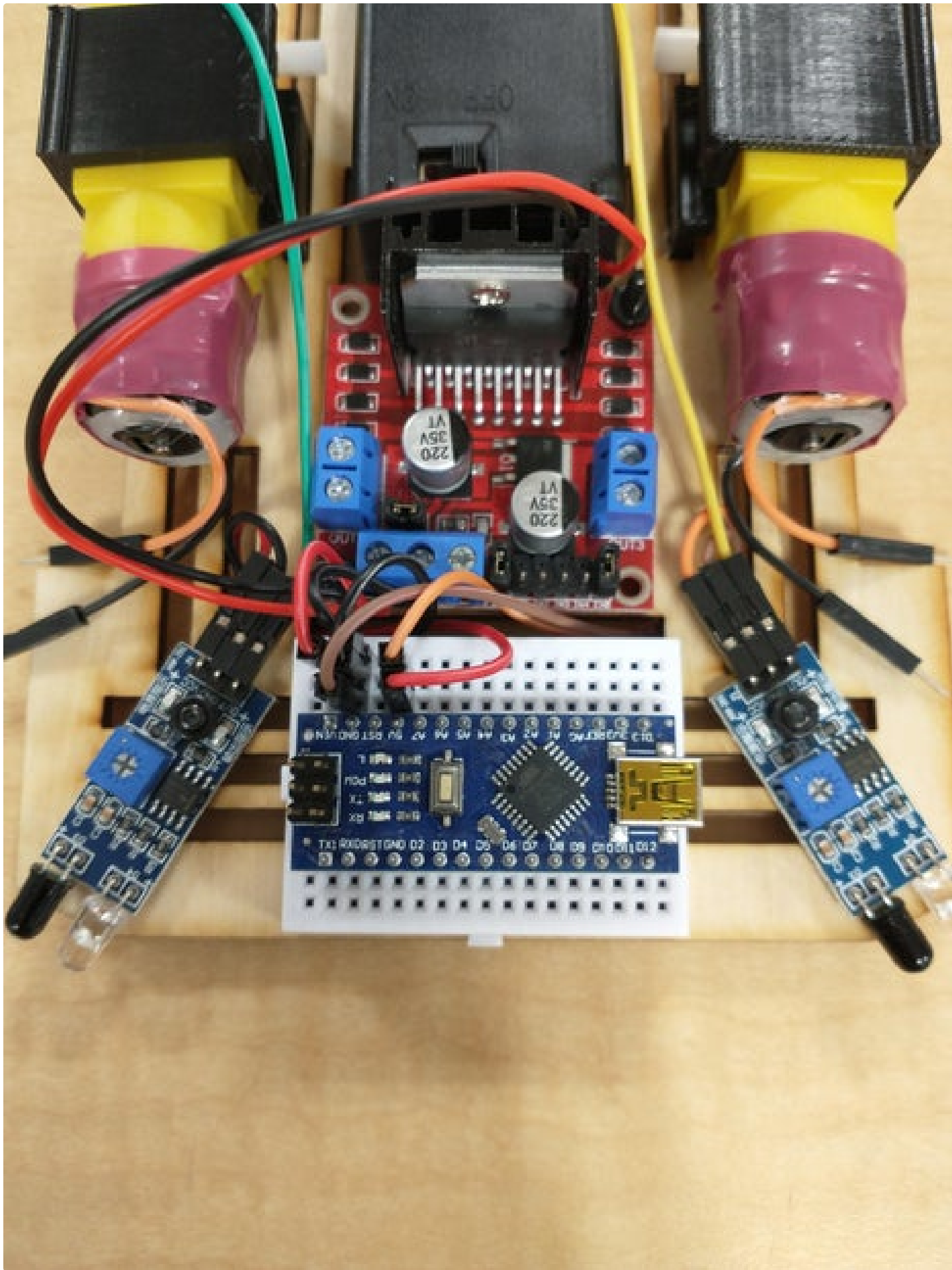
1. Take one of the IR Object Avoidance sensors and attach the following breadboard wire to its pins.
2. Attach a red wire with an input and pin to the **VCC** pin.
3. Attach a black wire with an input and pin to the **GND** pin.
4. Attach a green wire with an input and pin to the **OUT** pin.
5. Take the remaining IR Object Avoidance sensors and attach the following breadboard wire to its pins.
6. Attach an orange wire with an input and pin to the **VCC** pin.
7. Attach a brown wire with an input and pin to the **GND** pin.
8. Attach a yellow wire with an input and pin to the **OUT** pin.
9. Using an m3 screw and nut secure the sensor with the green wire to the left side of the car front end.
Make sure to angle the sensor out. You will see three lines cutout at the front of the car you can attach the sensor to any of these. See the second picture for example.
10. Using an m3 screw and nut secure the sensor with the yellow wire to the right side of the car front end.
Make sure to angle the sensor out. You will see three lines cutout at the front of the car you can attach the sensor to any of these. See the second picture for example.
11. Insert the red and orange wire pins coming from the sensors into the breadboard next the **5V** pin on the Arduino Nano. See picture 4 and 5 for examples.
12. **Note:** you can wrap the wire around any of the cut out areas on the chassis to keep the wires tidy. This is not required though. See the 3rd picture for an example.
13. Insert the brown and black wire pins coming from the sensors into the breadboard next the **GND** pin on the Arduino Nano. See picture 4 and 5 for examples.
14. Insert the yellow wire pin coming from the sensor into the breadboard next the **D12** pin on the Arduino Nano. See the 6th picture for an example.
15. Insert the green wire pin coming from the sensor into the breadboard next the **D10** pin on the Arduino Nano. See the 7th picture for an example.

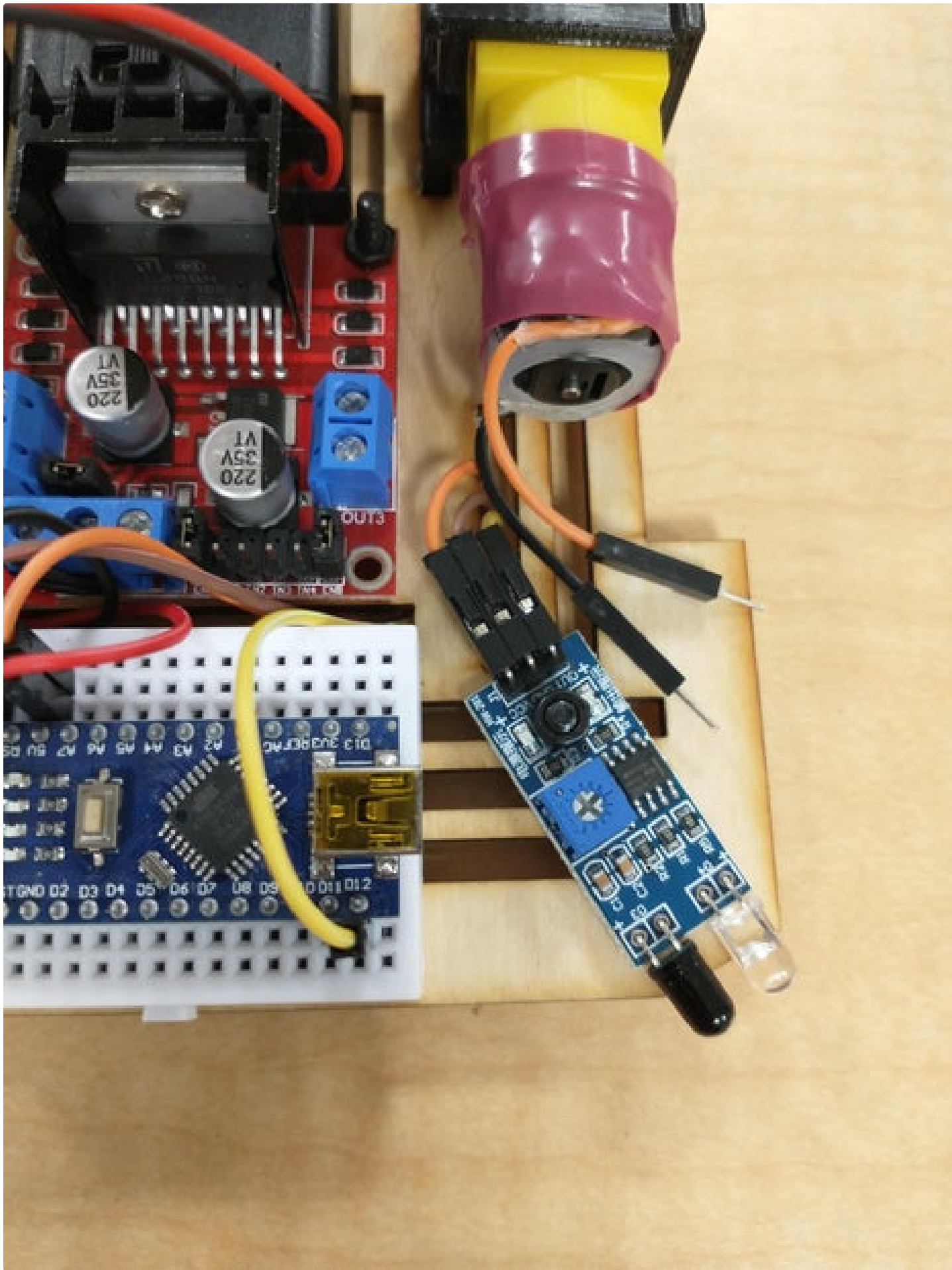


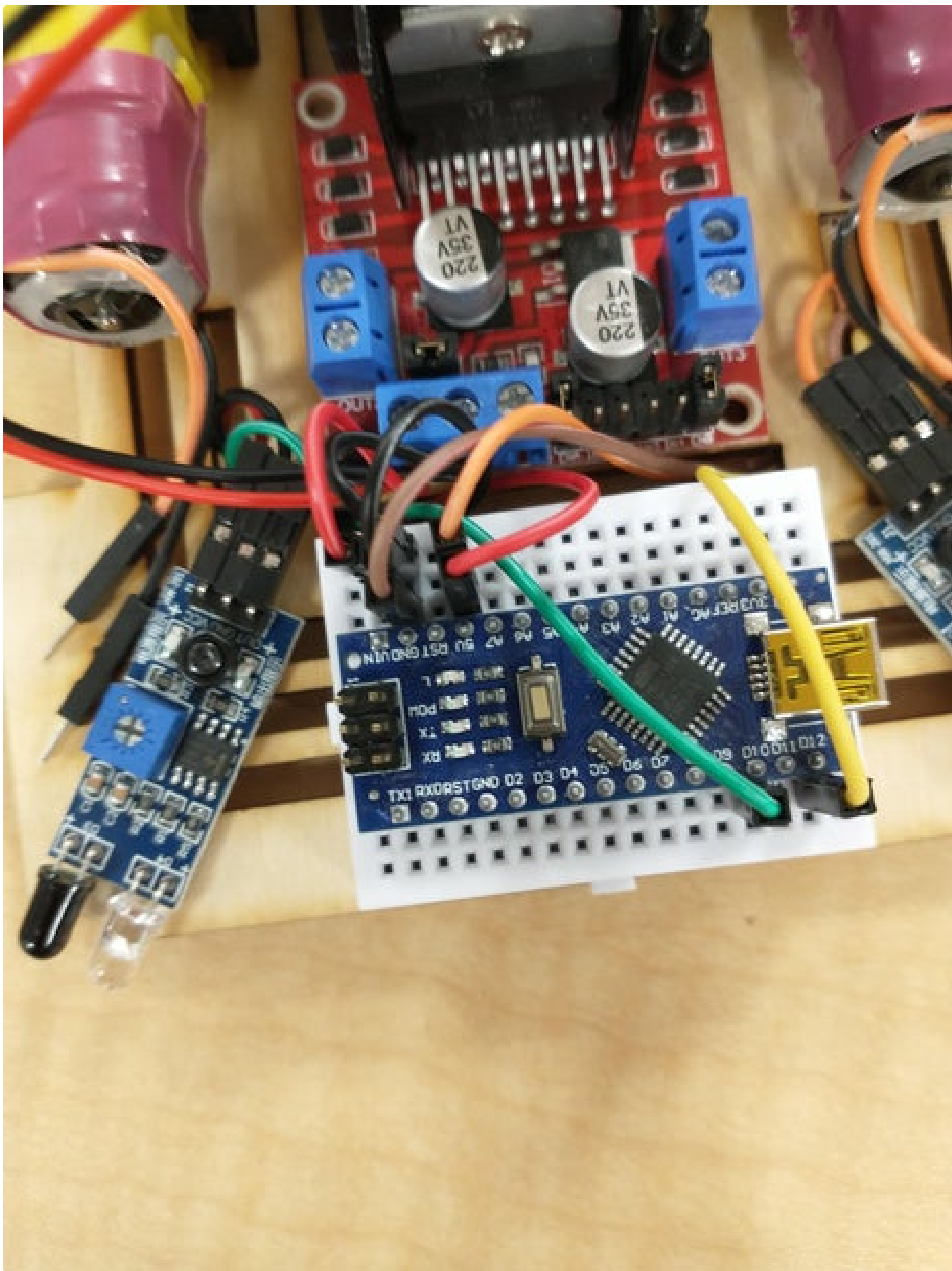


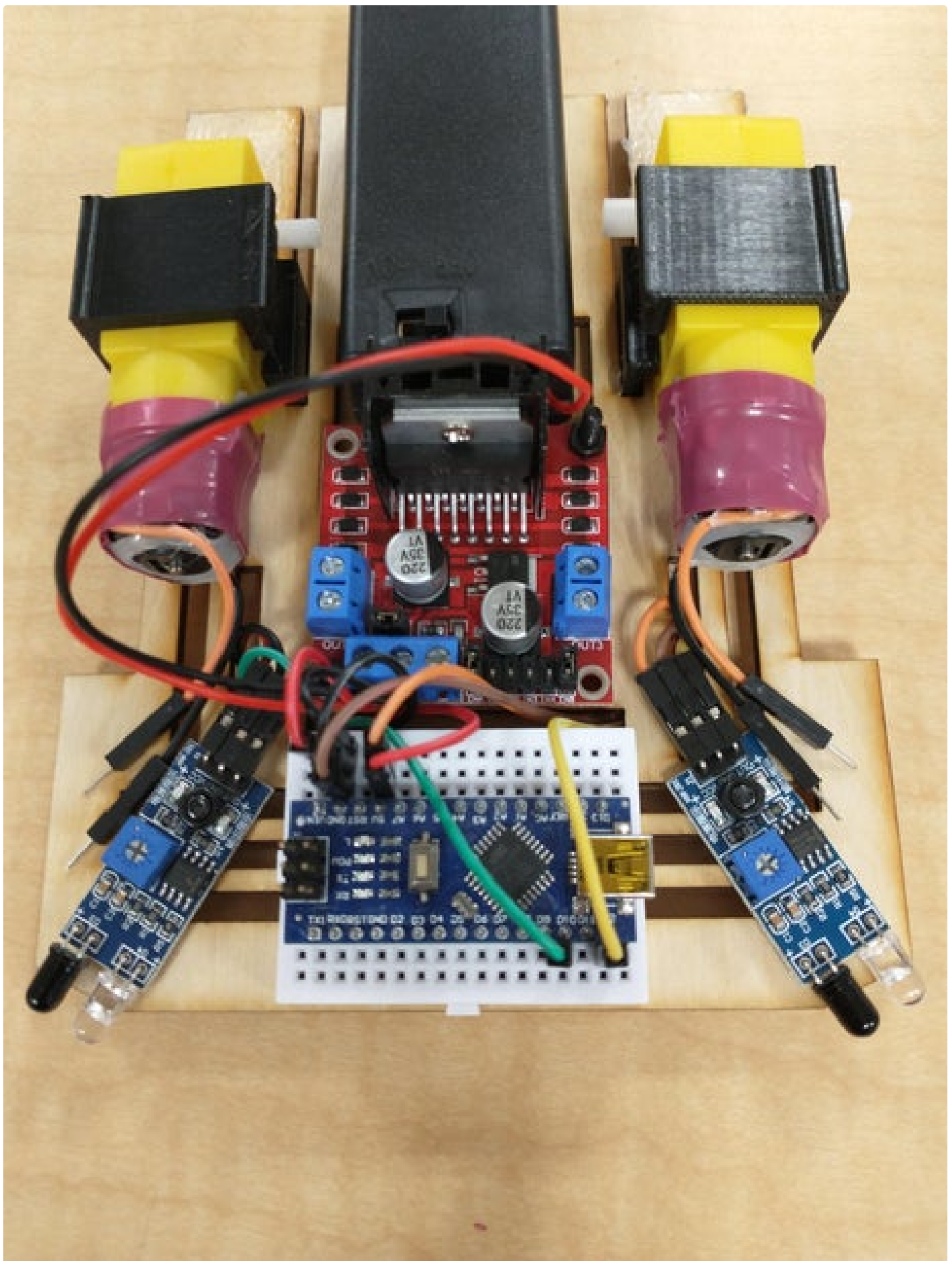










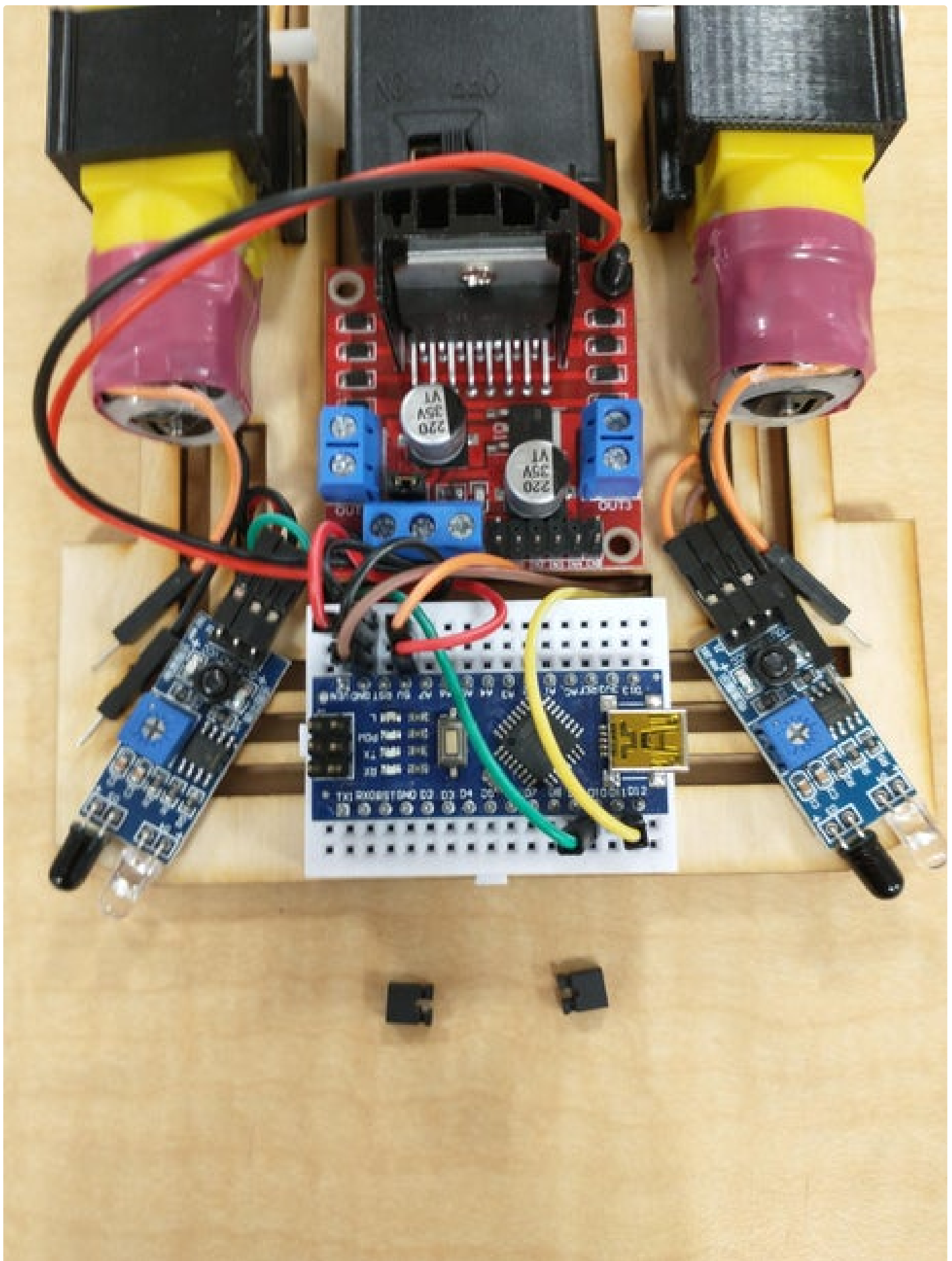


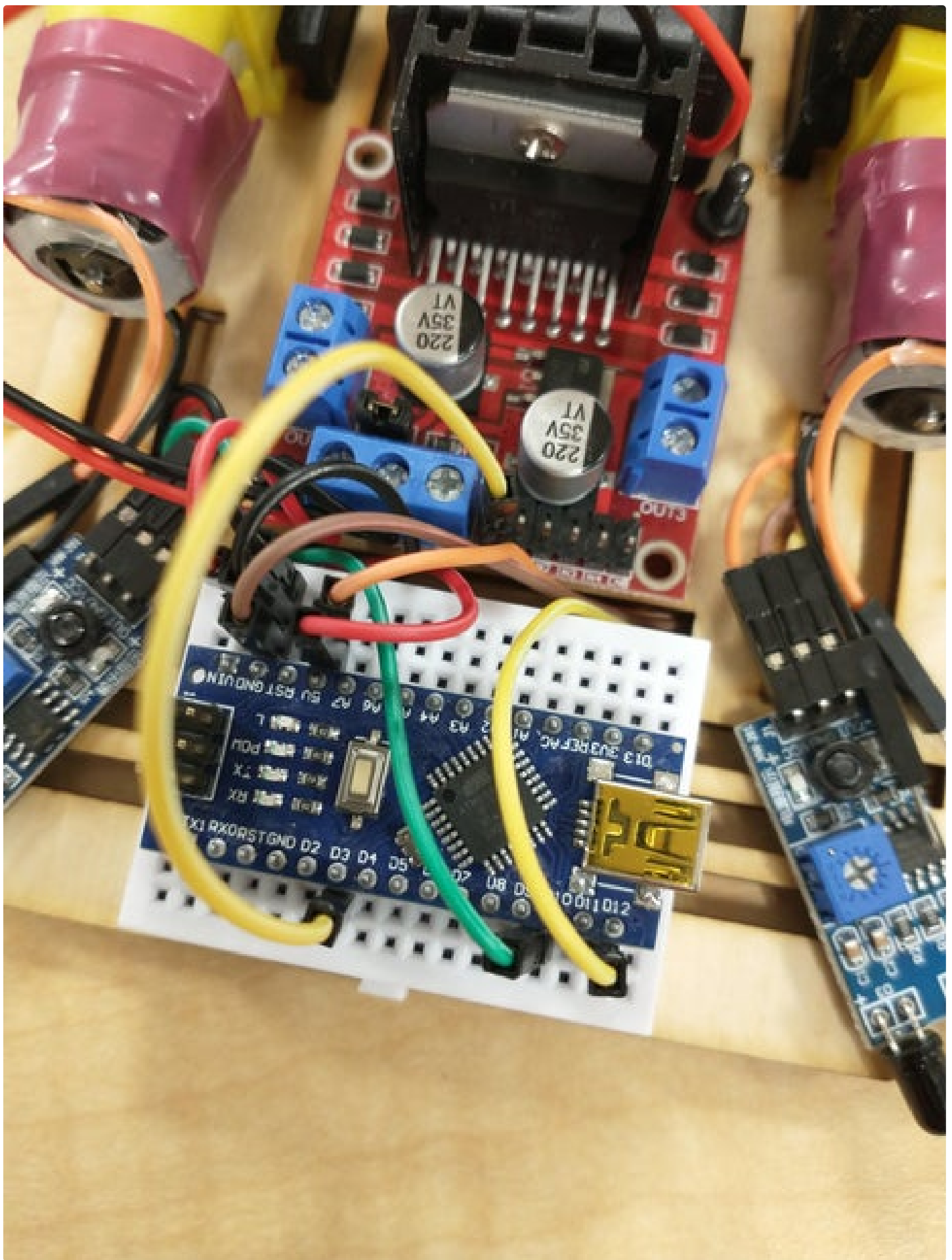
Step 6: Connecting Motor Driver to Arduino

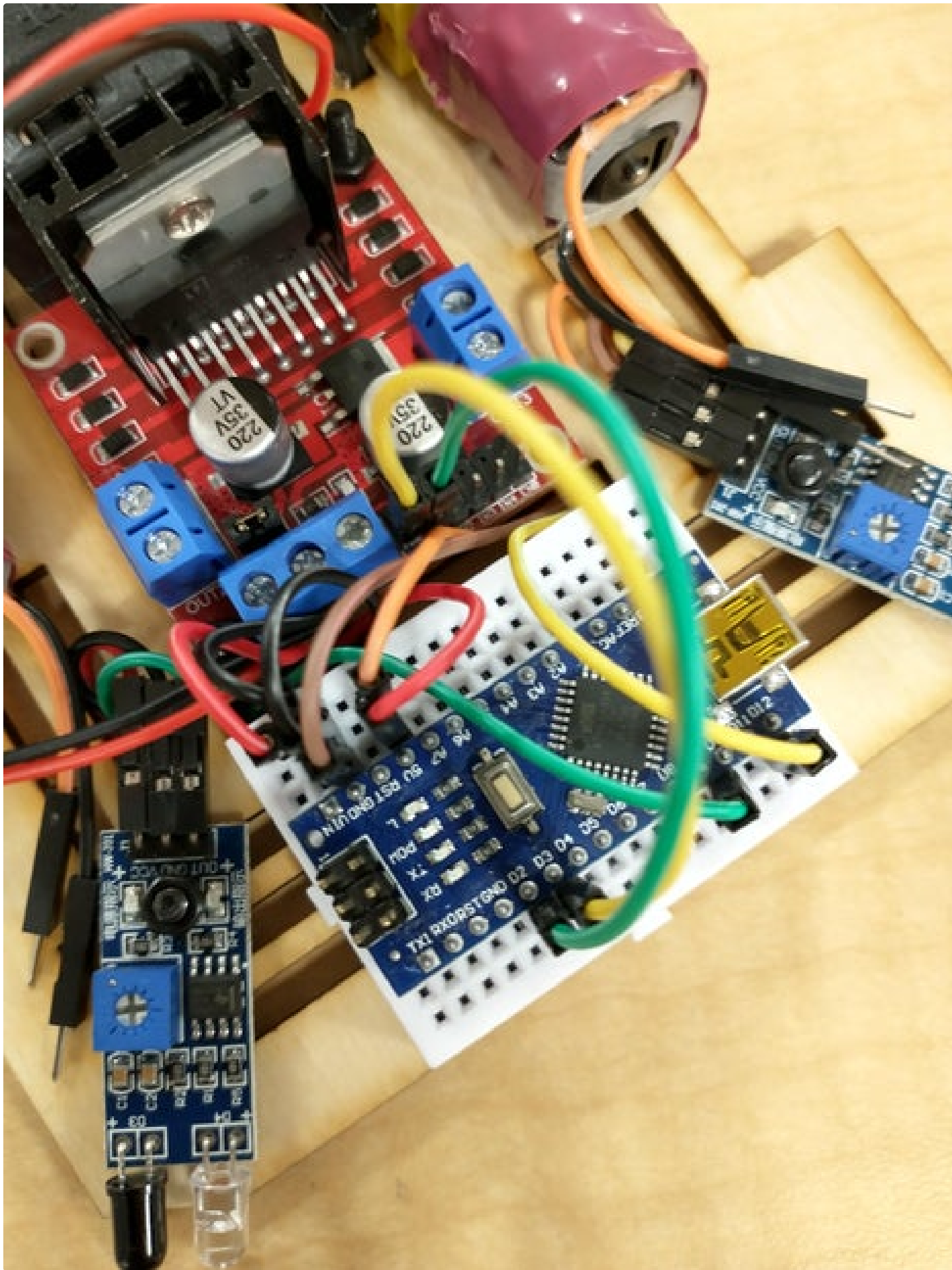
The orientation of the robot for this step is the front end facing you. See the last picture for example.

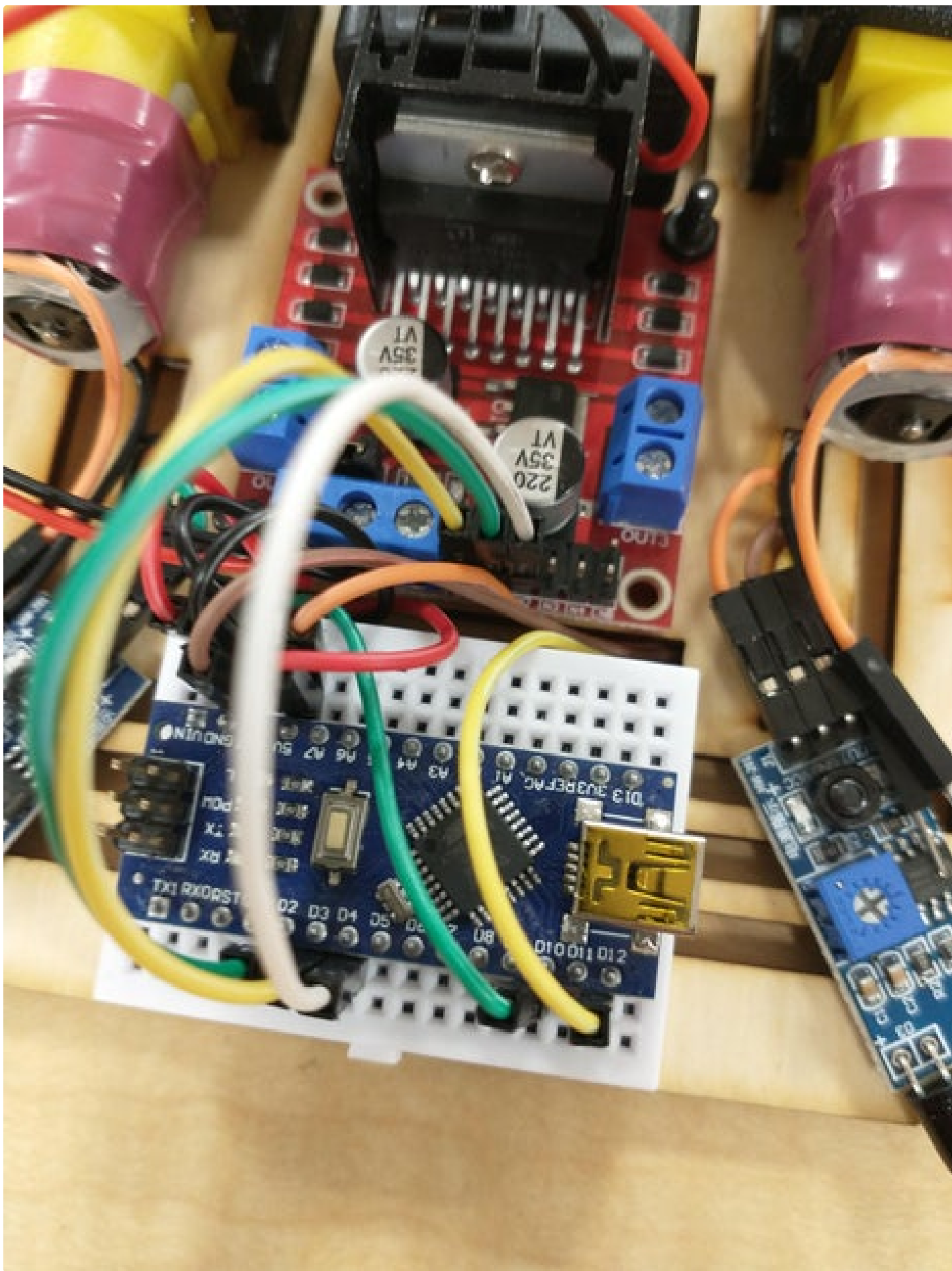
1. If your L298n motor driver has bridging clips attached to the **ENA** and **ENB** pins remove them.
2. Using wires with an input and pin make the following connection to the motor driver and then to the breadboard inputs adjacent to the Arduino Nano pins.
3. Attach yellow wire to **ENA** on the driver and insert it next to the Arduino Nano **D3** pin.
4. Attach green wire to **IN1** on the driver and insert it next to the Arduino Nano **D2** pin.
5. Attach white wire to **IN2** on the driver and insert it next to the Arduino Nano **D4** pin.
6. Attach purple wire to **IN3** on the driver and insert it next to the Arduino Nano **D5** pin.
7. Attach grey wire to **IN4** on the driver and insert it next to the Arduino Nano **D8** pin.
8. Attach blue wire to **ENB** on the driver and insert it next to the Arduino Nano **D9** pin.
9. Input each wire from the left motor into the **OUT1** and **OUT2** ports on the motor driver. Do not tighten screws, you may need to swap the wires if the motor is driving in reverse. This is called reversing polarity.
10. Input each wire from the right motor into the **OUT3** and **OUT4** ports on the motor driver. Do not tighten screws, you may need to swap the wires if the motor is driving in reverse. This is called reversing polarity.

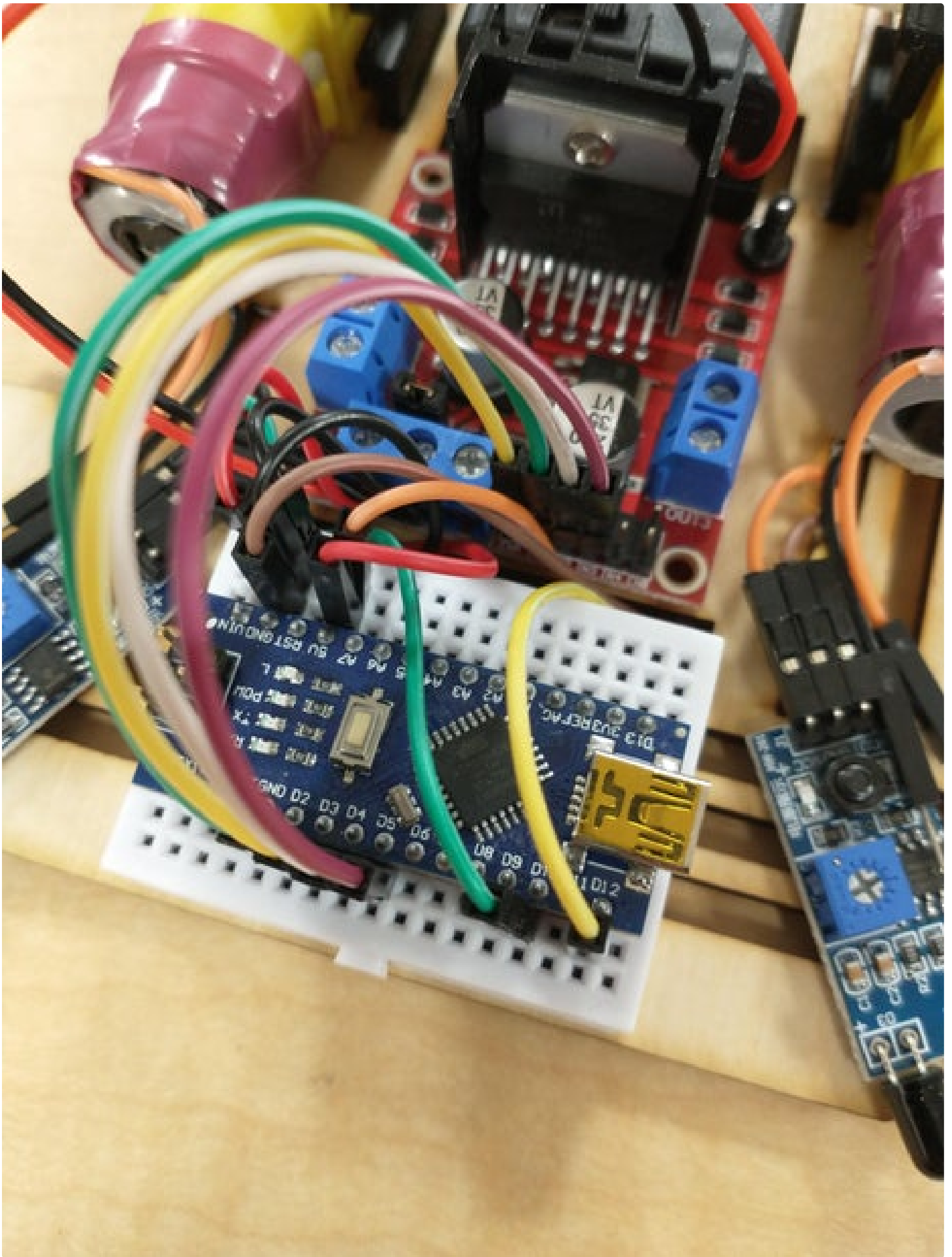
Note: you will secure the motor wires in **Step 9: Edit and Upload Code**.

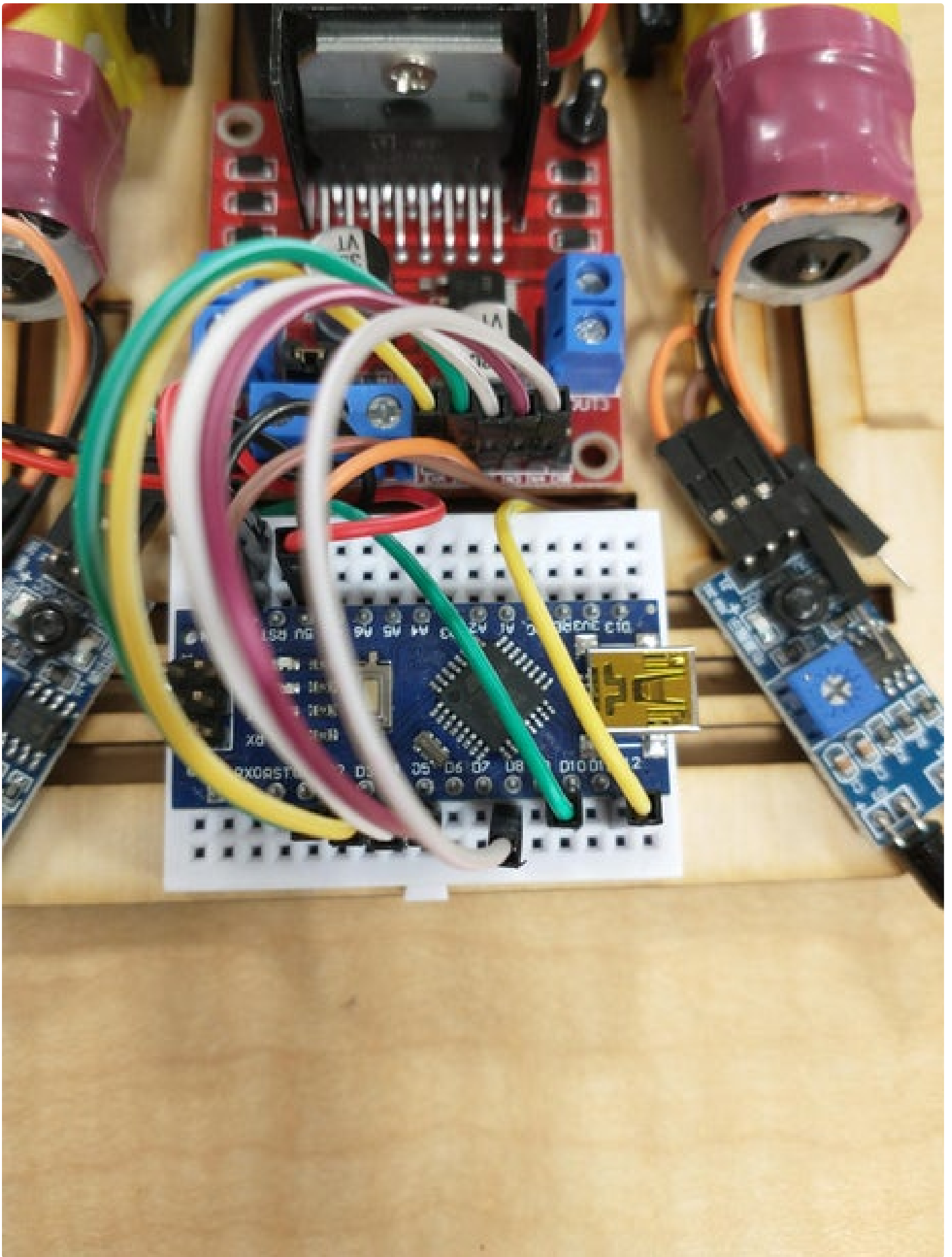


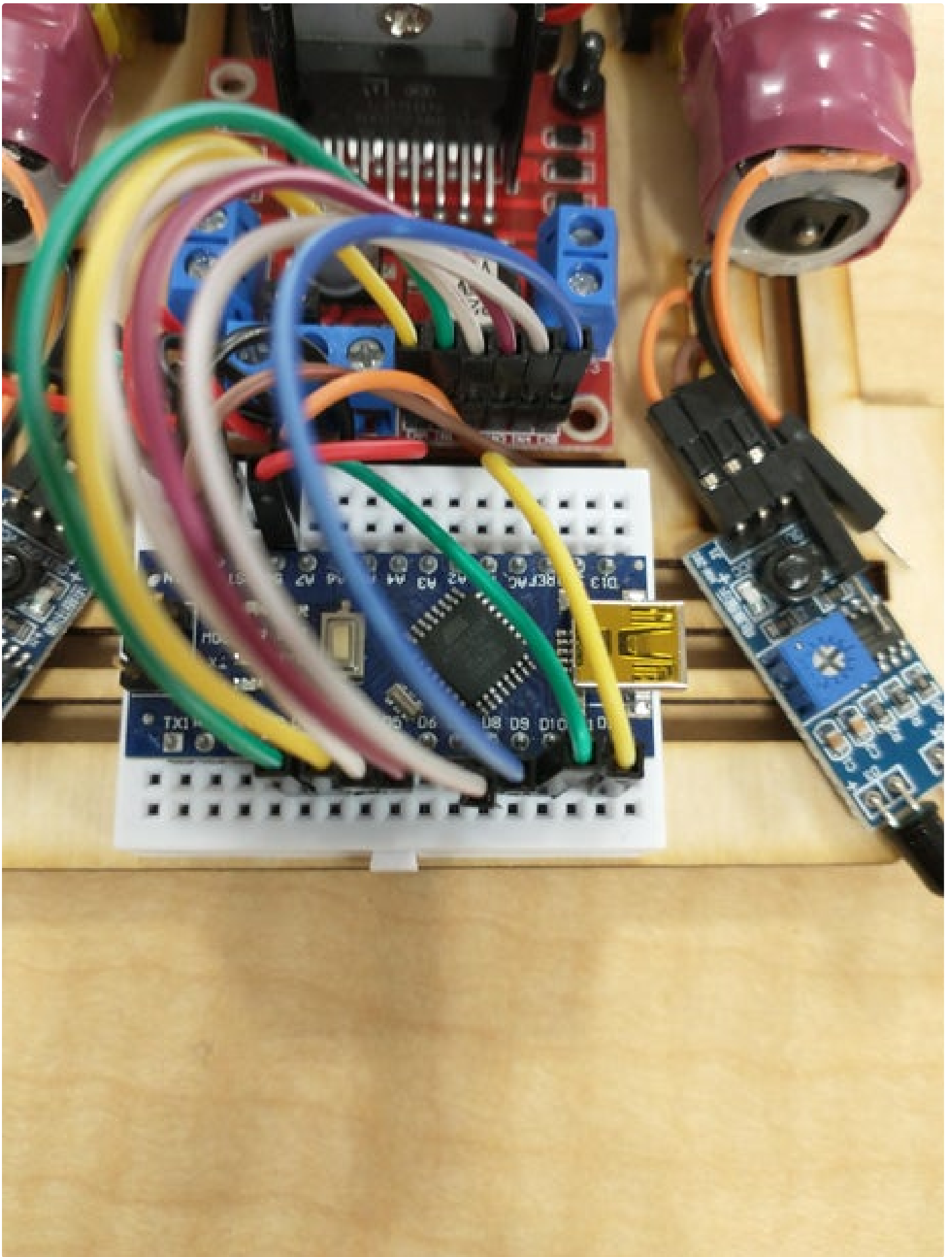


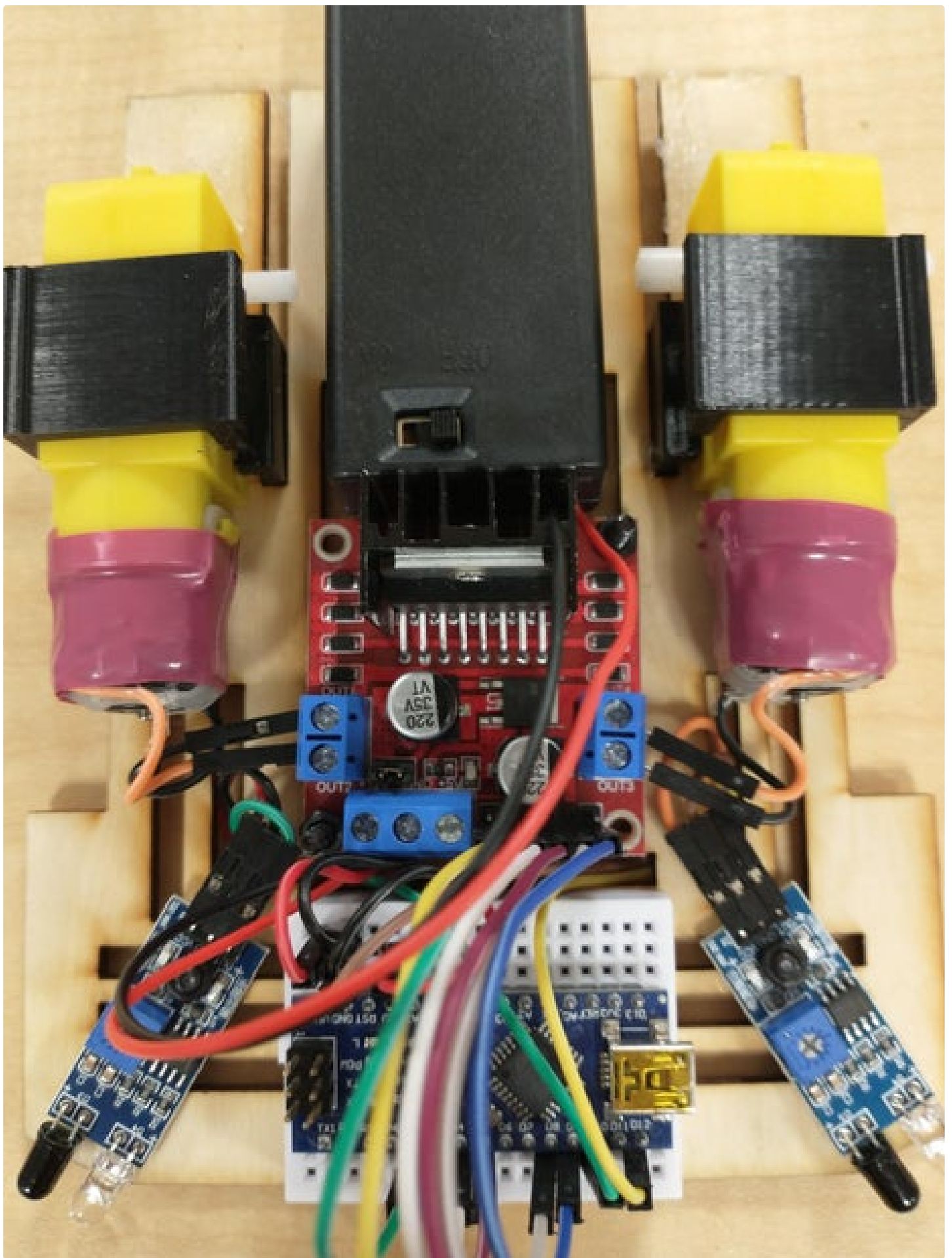






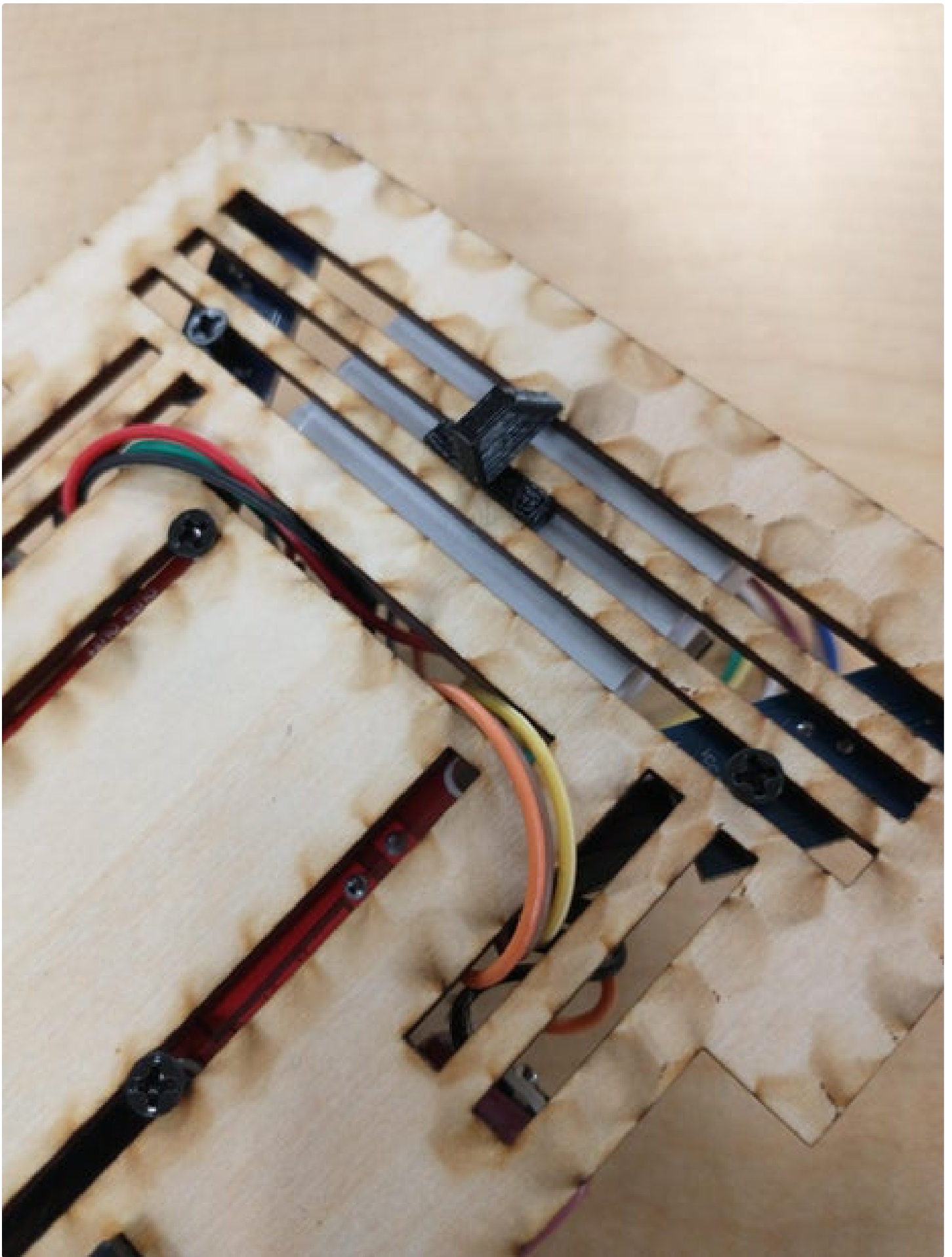


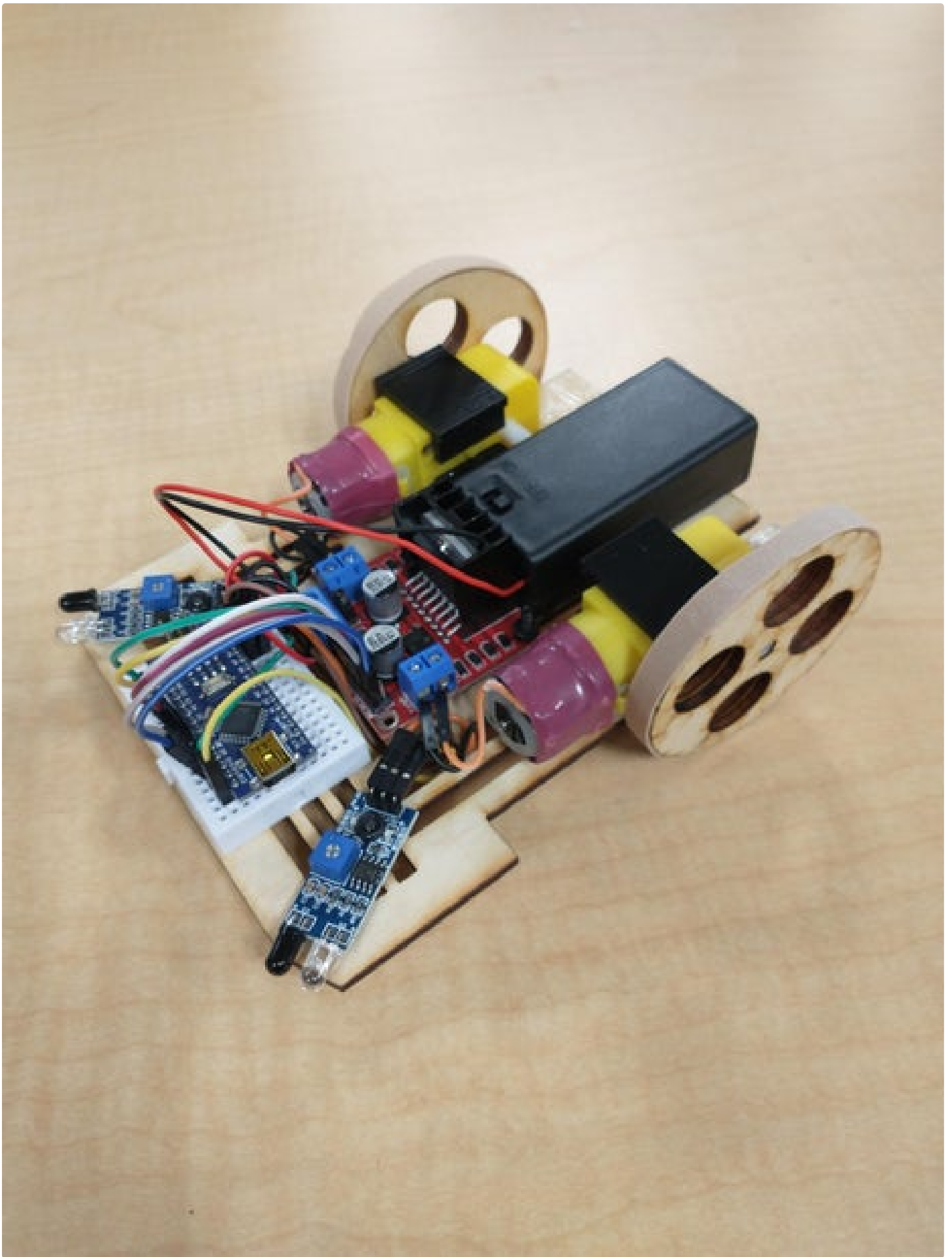


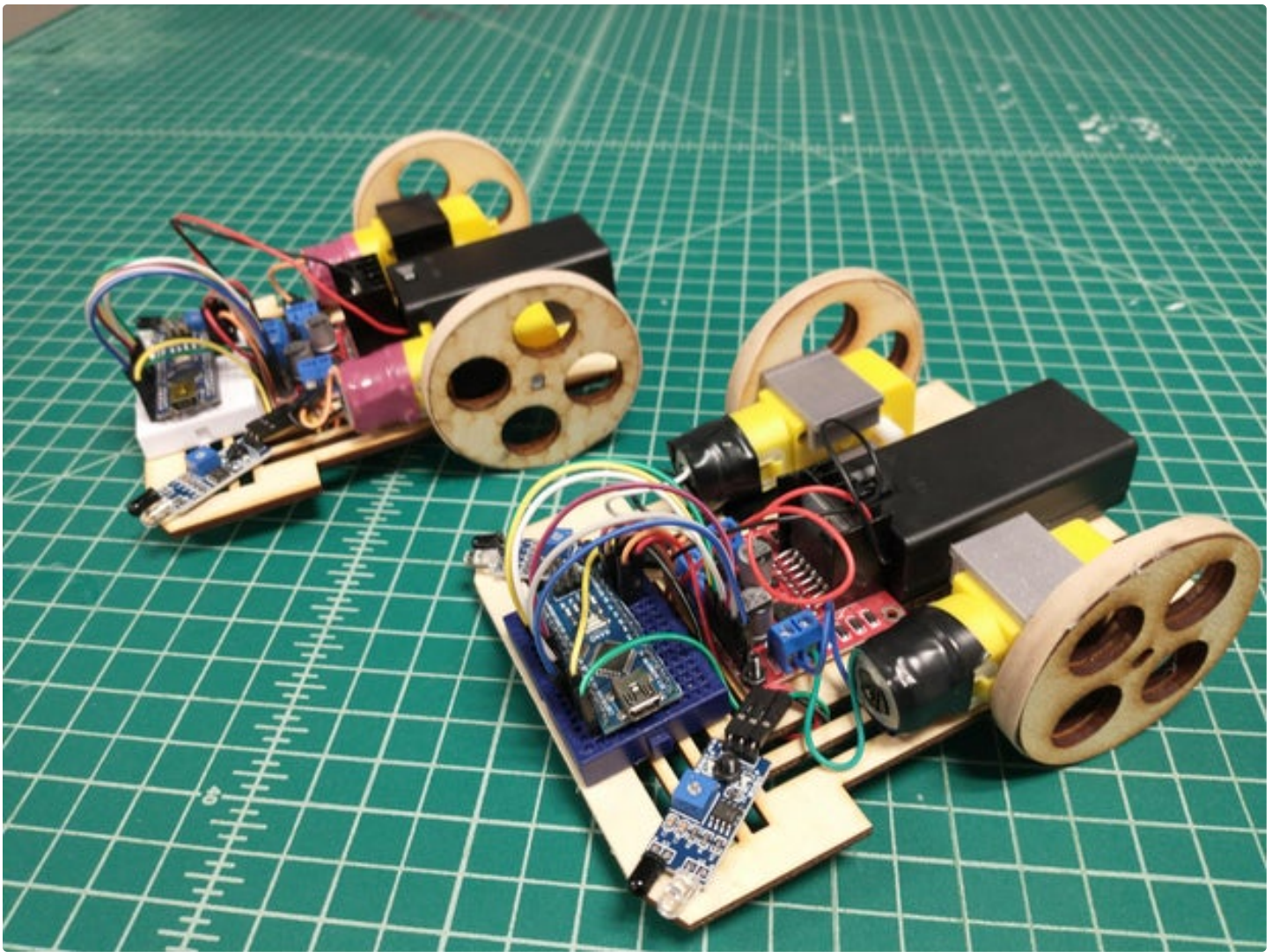


Step 7: Wheels and Break

1. Attach the 3D printed break pad to the underside of the robot. You can change the position of this to alter the robots performance. The break will slide into any of the three cutouts on the front of the car. See the first picture for details.
2. Slide a wheel on each motor.







Step 8: MBlock

In order to write code for this robot in Scratch we will need to use the mBlock software. It is entirely free and you do not need to create an account to use it. You can download and install it at this link: www.mblock.cc/en/download

1. Download and install mBlock from the link provided above.
2. Download the TCPL's prewritten code here: [Google Drive](#).
3. Open mBlock once it's fully installed.
4. On the bottom left **Devices** tab close the CyberPi device by selecting the blue x. A message box will appear asking if you want to delete this sprite. Select delete.
5. Navigate back to the **Devices** tab and select **add**. A new page full of devices will appear.
6. Locate the **Arduino Nano** and install it by selecting the blue plus symbol. Then select **ok**.
7. Locate the **Arduino Nano(old)** and install it by selecting the blue plus symbol. Then select **ok**. We add the Arduino Nano(old) in case we cannot upload the code to the regular one.
8. Now that you have both Arduino Nanos installed you can open TCPL's code. Select from the top ribbon **File, Open from your computer**, select **Don't Save**, and then navigate to where you downloaded the code and open it.
9. This will open the full code.

Download mBlock

One-stop coding platform tailored to coding education,
trusted by 15 million educators, and learners



mBlock web version

Chrome browser recommended >>

Support Windows/Mac/Linux/Chromebook

[Code with blocks](#)[Code with Python](#)

mBlock PC version

Version: V5.4.0

Released: 2021.11.30

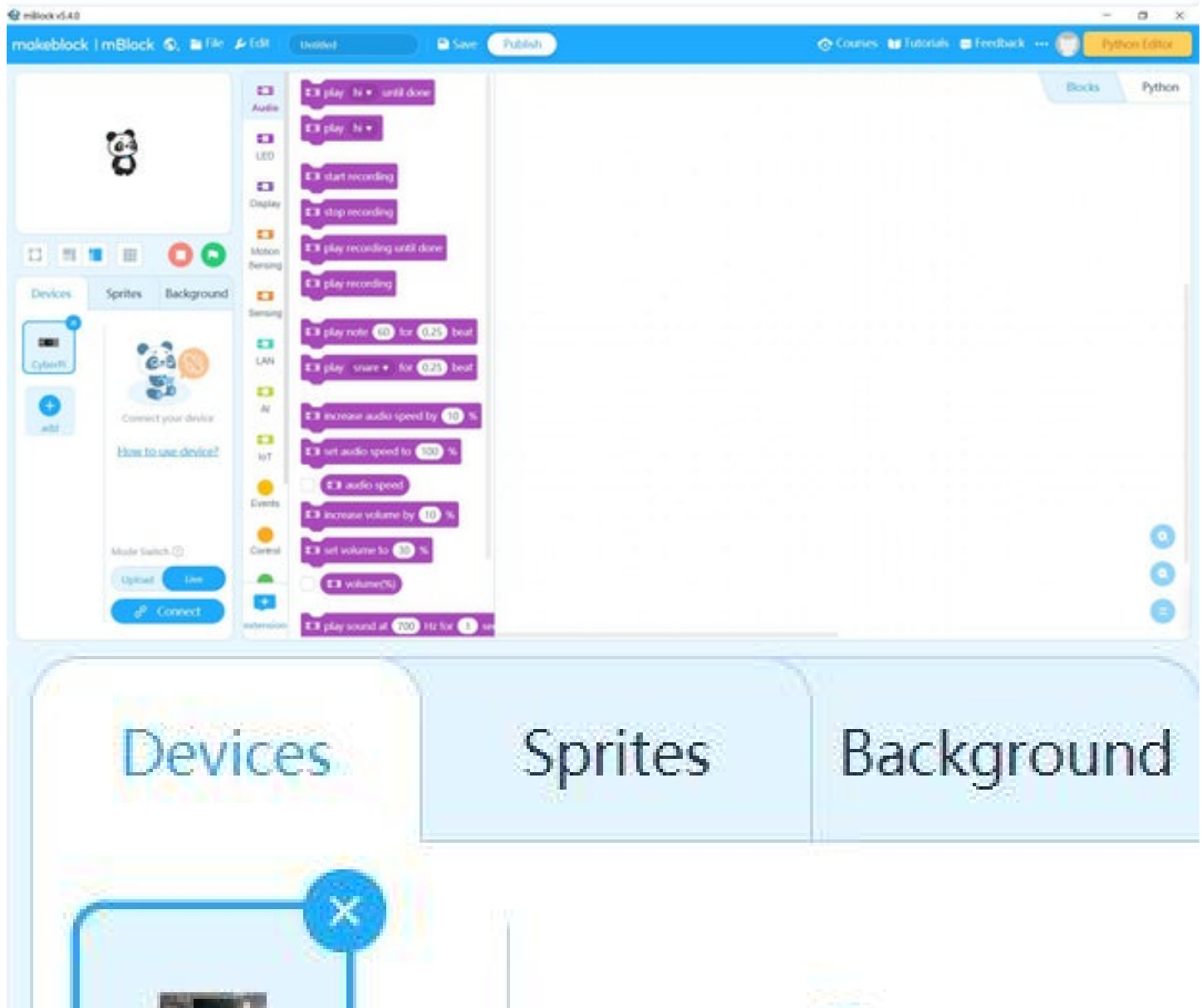
[Released log >>](#) [Previous version >>](#)

[Download for Windows](#)

Win7 or Win10 (64-bit recommended)

[Download for Mac](#)

macOS 10.12+



CyberPi



add



Connect your device

[How to use device?](#)

Mode Switch (?)

Upload

Live



Connect

Delete the sprite



Are you sure to delete this sprite?

Delete

Cancel

Devices

Sprites

Background



add

Device Library



CyberPi

Developers: mBlock



xLight

Developers: mBlock



mBot Mega

Developers: mBlock



Codev

Developers: mBlock



Neuron

Developers: mBlock



mBot

Developers: mBlock



mBot Ranger

Developers: mBlock



Bluetooth Controller

Developers: mBlock



MotionBlock

Developers: mBlock



Halocode

Developers: mBlock




Become a developer of mBlock to unlock more potential.


Cancel

OK


Device Library




microbit
Developers: mBlock




Arduino Uno
Developers: Ablock




Arduino Mega2560
Developers: Ablock




Arduino Micro
Developers: Ablock




Arduino Yun
Developers: Ablock




Arduino Leonardo
Developers: Ablock




OrangeBoard
Developers: woojh




Arduino Nano
Developers: Ablock




cerebro
Developers: epsilon11101




Robot BrickOne
Developers: duongtheluongnv



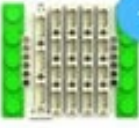
Vietduino Uno
Developers: xuanngoc1992




Tello
Developers: callblueday




Arduino Mega
Developers: qbabyboy520





Arduino Uno
Developers: kondal2000





Brain GO
Developers: smathuang











[Become a developer of mBlock to unlock more potential.](#)

Cancel

OK

Device Library

PikaBot

Developers: bhavithiran97

Eduino Board

Developers: lab767

Fingertip Piano

Developers: mBlock

MONCO

Developers: lab767

MONCO

Developers: lab767

Maker Uno

Developers: bhavithiran97

leqo mindsorms ev3

Developers: pcc404

T-Block

Developers: arad.roq

light lesson

Developers: 13911082190

Arduino Nano (old)

Developers: Ablock

vrbot2

Developers: koha2006

CakeSensor

Developers: semibank

Real STEAM

Developers: techms5499

OTUS

Developers: Parconier

Maker Mini Sumo

Developers: bhavithiran97

Arduino Uno

Developers: bhavithiran97

Arduino Uno

Developers: bhavithiran97

Arduino Uno

Developers: bhavithiran97

Arduino Uno

Developers: bhavithiran97

Arduino Uno

Developers: bhavithiran97

Become a developer of mBlock to unlock more potential.

Cancel

OK

Devices

Sprites

Background

Arduino ...

Arduino ...

Connect your device

Autonomous Robot Racer: Scratch Mode: Page 56

Arduino ...



add

How to use device?

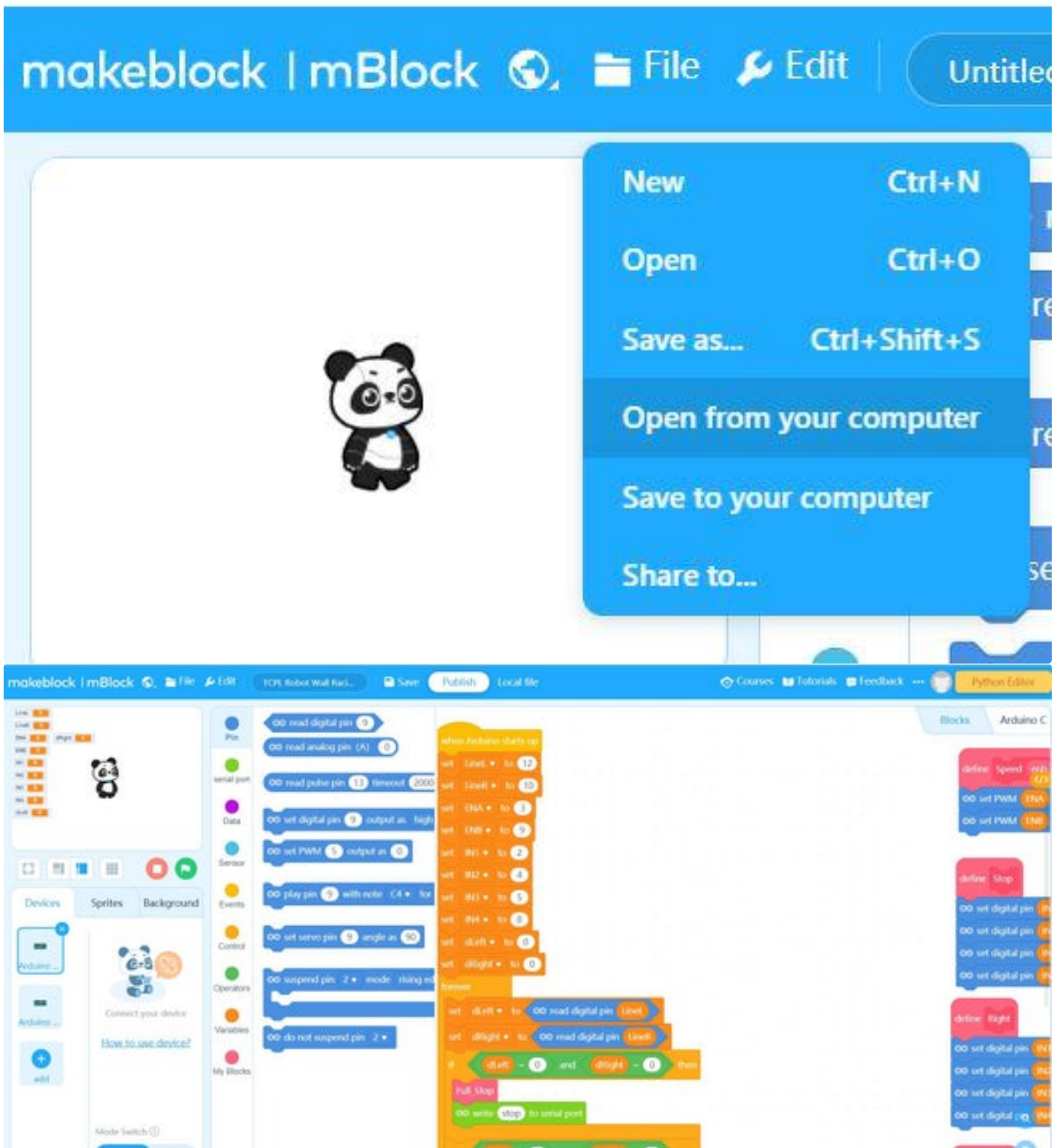
Mode Switch (?)

Upload

Live



Connect



Step 9: Edit and Upload Code

In this Instructable, I will only go into details on how to edit the premade code from the previous step. In a later Instructable I will share an overview and lesson plan on how to write this code from the beginning.

Depending on the microcontroller of your Nano you will need to use either the Arduino Nano or Arduino Nano(old) device. To figure this out we will attempt to upload code from each device in mBlock.

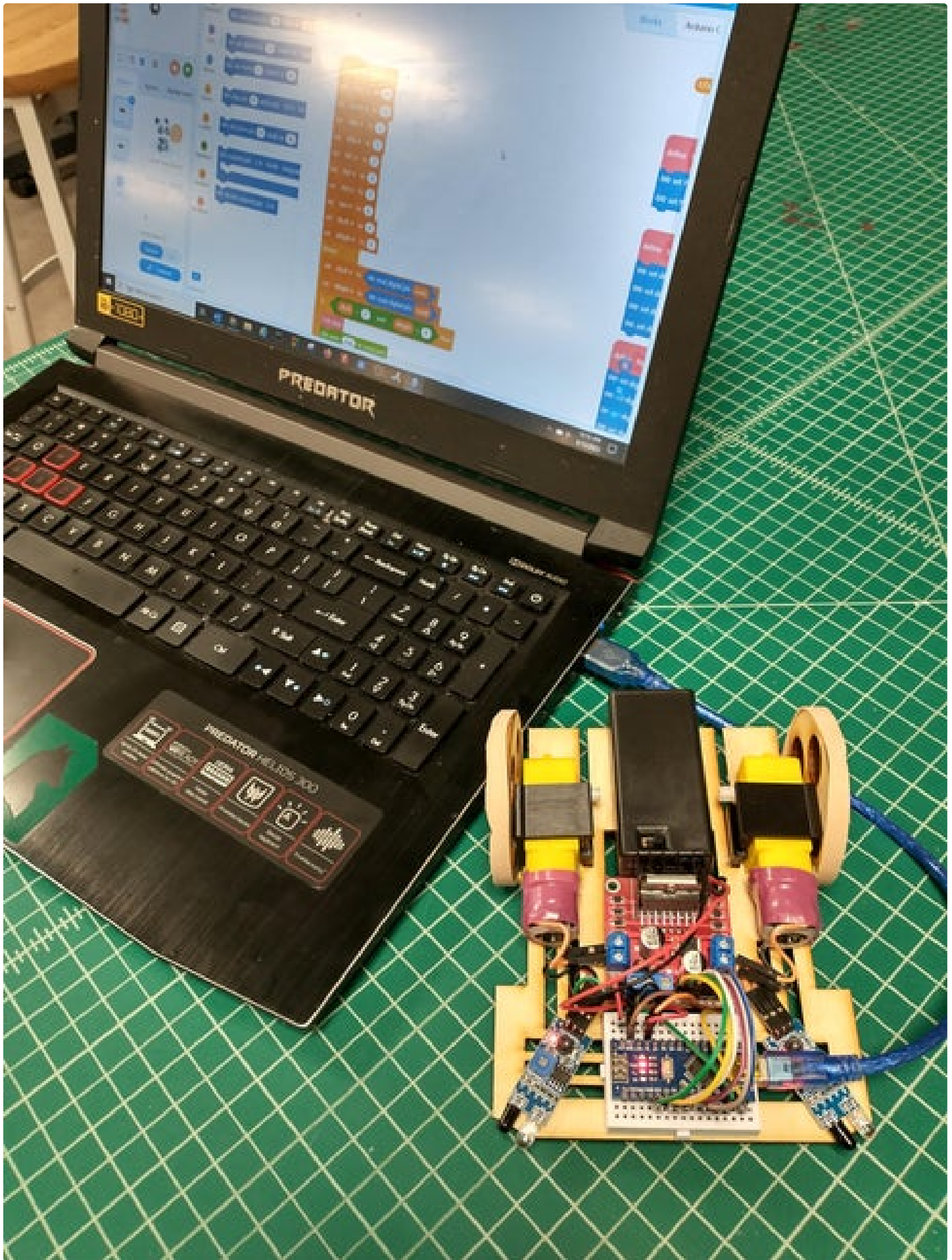
1. Make sure the battery case on the robot is switched off.
2. Insert a 9v battery into the case and lock it into place.
3. Attach the USB cable to the robot and connect it to your computer.
4. Select the top device. If you followed along in the last step this is the Arduino Nano.
5. Select **Connect** under the COM. The number after COM may differ on your device and can be ignored.
6. Navigate back to the **Devices** menu on the bottom left of the screen and select the larger **Upload** button to begin writing the script to the robot.

Note: If the code fails to upload you will need to navigate to the **Devices** menu select the bottom option, Arduino Nano (old), and repeat steps 5 and 6 above.

Once you have the code uploaded make sure the motor wires are placed into the motor driver ports, see **Step 6 Connecting Motor Driver To Arduino**. Turn on the robot making sure there is nothing in front of the sensors. If both wheels are driving forward you can tighten the screws above each port. If any of the motors are driving in the wrong direction swap the wires to opposite ports. If the motor continues to drive in the wrong direction make sure both sensors are not detecting any object. You may need to turn the sensitivity of the sensor(s) down if there is nothing in front of it and both indicator lights are on, see the 8th picture for an example. Take a screwdriver and gently rotate the potentiometer clockwise until only one light is on, see the 9th picture. By turning the potentiometer clockwise the sensor's sensitivity is being reduced; turning the other way increases it. Turning the dial too far in any direction will make it so the sensor either turns off or makes it always detect. If the solutions above do not work you will need to go back to **Step 6 Connecting Motor Driver To Arduino** and double-check that everything is connected properly.

Editing the code:

1. Navigate to the **Blocks** area in mBlock and find the stack of Scratch code where the topmost block reads, *"when Arduino starts up"*.
2. Click, hold, and drag until you see the **Sp_Forward** block. You will notice comments on what input bubble controls which motor. The orientation of the robot for these comments is the rear of the robot facing you.
3. Click on the left input bubble and insert any whole number between 0 and 255. Input the same number into the right bubble. We use what is known as pulse width modulation, or PWM for short, to control the speed of the robot. For more info on this click [here](#). Basically, 0 is off and 255 is the maximum speed. You can adjust the speed of how fast you want your car to go in a straight line by inserting numbers less than 255 for example, the value of 180 will drive the motors slowly. On some tracks going at full speed will cause the car to crash through a wall. experiment with different values to get the best results.
4. Locate the **Sp_Left** block. The first bubble controls how fast the one wheel turns forward and the other controls how fast the other wheel travels backwards. Reversing one of the wheels during a turn will cause the robot to make a sharper turn. Experiment with different turning and breaking speeds to find the optimal turn around a corner.
5. Locate the **Sp_Right** block. The first bubble controls how fast the one wheel turns backwards and the other controls how fast the other wheel travels forwards. Reversing one of the wheels during a turn will cause the robot to make a sharper turn. Experiment with different turning and breaking speeds to find the optimal turn around a corner.
6. Connect your car to your computer and upload the code to the computer. You can refer to the beginning of this section for a refresher on how to upload code to an Arduino Nano.



Devices

Sprites

Background



Arduino ...



Arduino ...



add



Connect your device

[How to use device?](#)

Mode Switch (?)

Upload

Live



Connect

USB



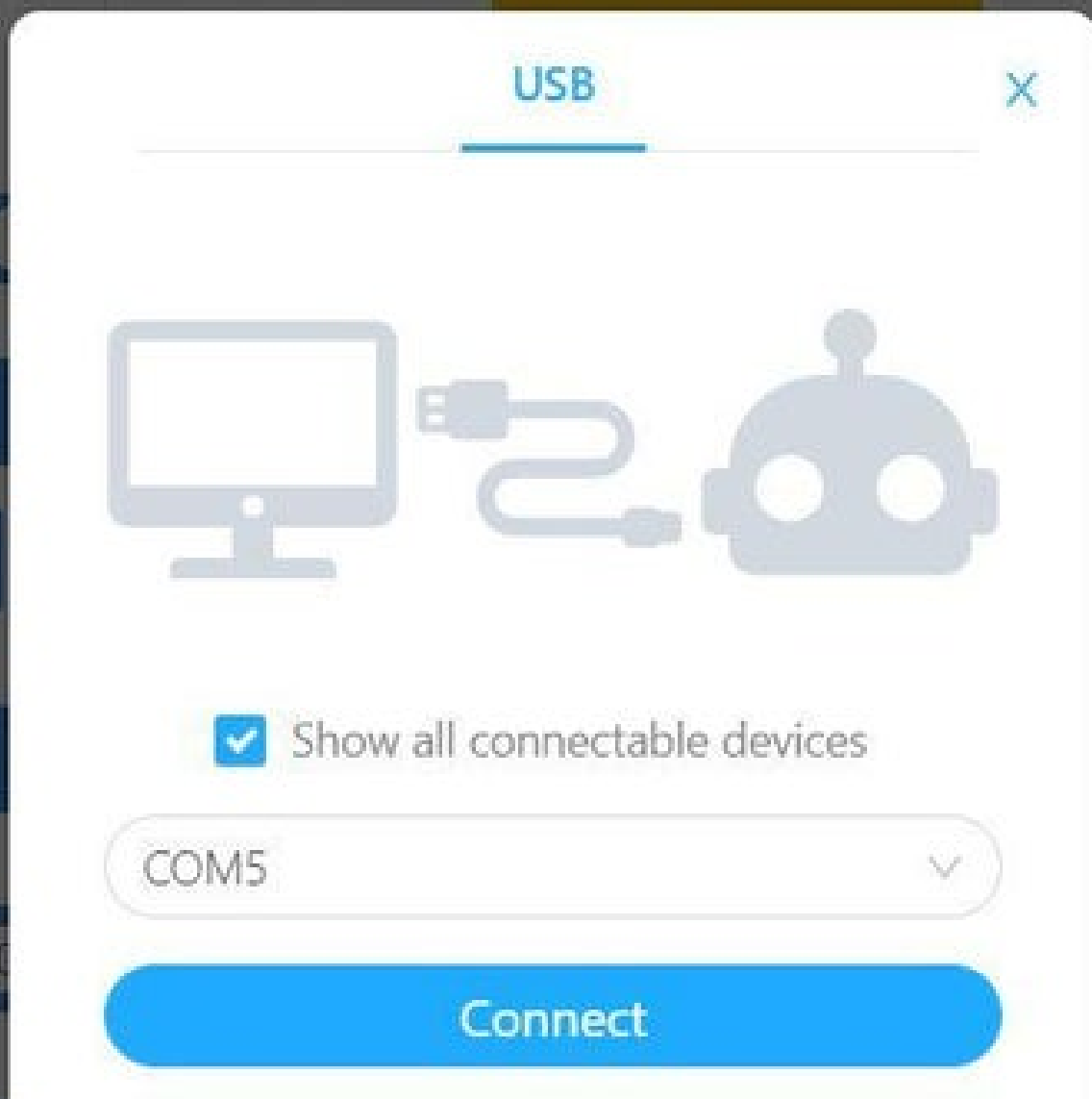
Show all connectable devices

No device can be detected to connect... ▾

Connect

- Please make sure the USB cable is properly connected to the device.
- Please make sure the device to be connected is turned on.
- Only one device can be connected at a time in this version. So connecting this

time in this version, so connecting this device will lead to the disconnection of the previous one.



- Please make sure the USB cable is properly connected to the device.
- Please make sure the device to be

connected is turned on.

- Only one device can be connected at a time in this version. So connecting this device will lead to the disconnection of the previous one.

Devices

Sprites

Background



Upload

[How to use device?](#)

Mode Switch (?)

add

Upload

Live



Upload

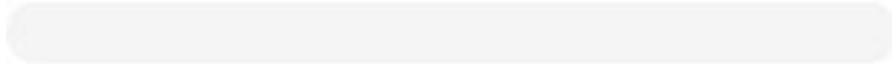


Disconnect



Setting

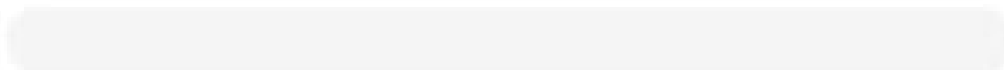
Upload Progress



0%

```
libraries/arduino/Mouse/src/" -I"arduino-  
libraries/arduino/Robot_Control/src/" -I"arduino-  
libraries/arduino/Robot_Motor/src/" -I"arduino-  
libraries/arduino/RobotIRremote/src/" -I"arduino-  
libraries/arduino/Stepper/src/" "C:\Users\jtang\mblock-  
avr\temp\build\code.cpp" -o "C:\Users\jtang\mblock-  
avr\temp\build\code.o"  
avr-toolchain\bin\avr-gcc -w -Os -Wl,--gc-sections -  
mmcu=atmega328p -o "C:\Users\jtang\mblock-  
avr\temp\build\out.elf" "C:\Users\jtang\mblock-  
avr\temp\build\code.o"  
"C:\Users\Public\Programs\mblock\resources\app\mlink-  
v1\external\arduino\nano\libnano.a" -lm
```

Upload Progress



- Please review the coding blocks or the code for this

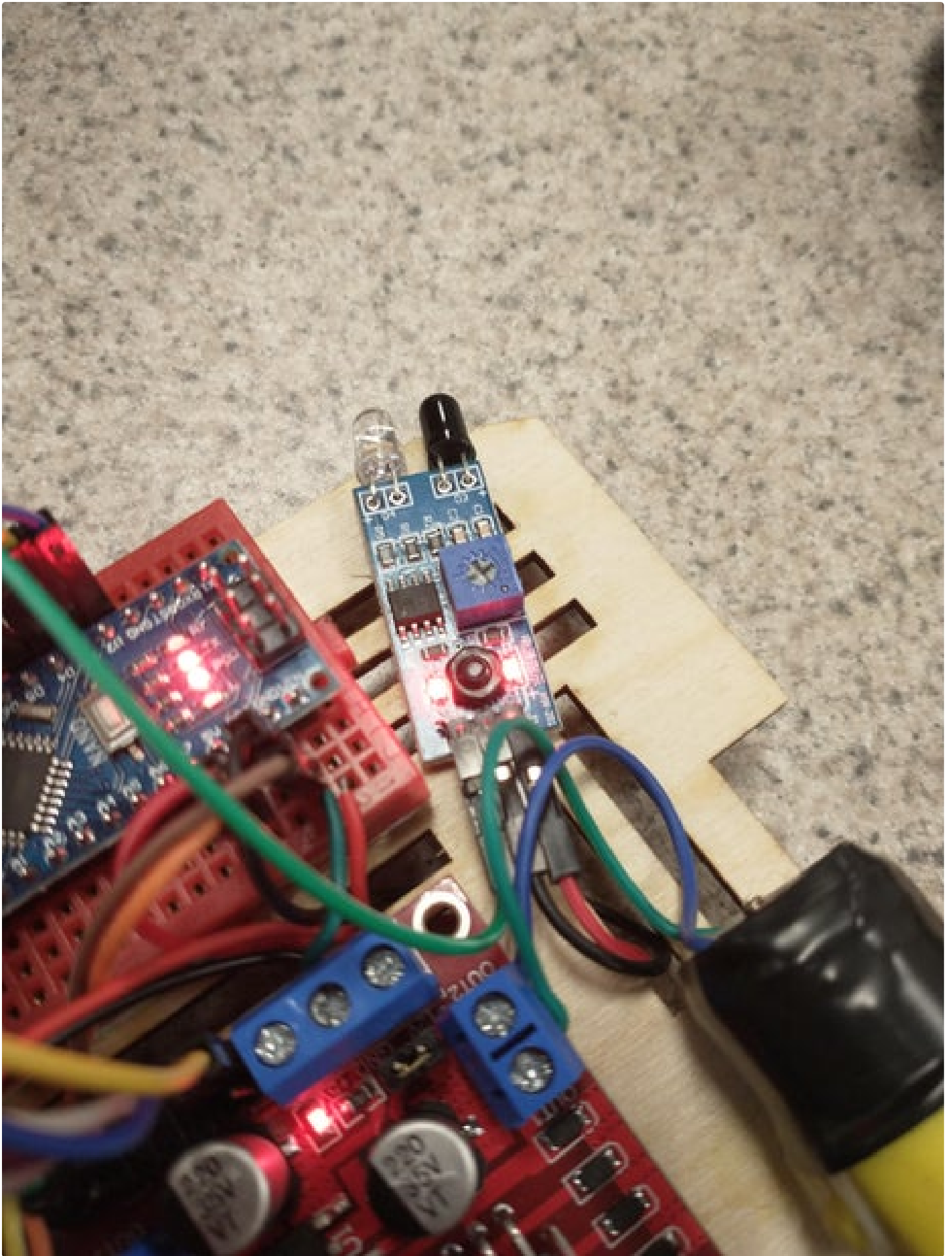
device. Make sure all the data, parameters and programming logic are correct.

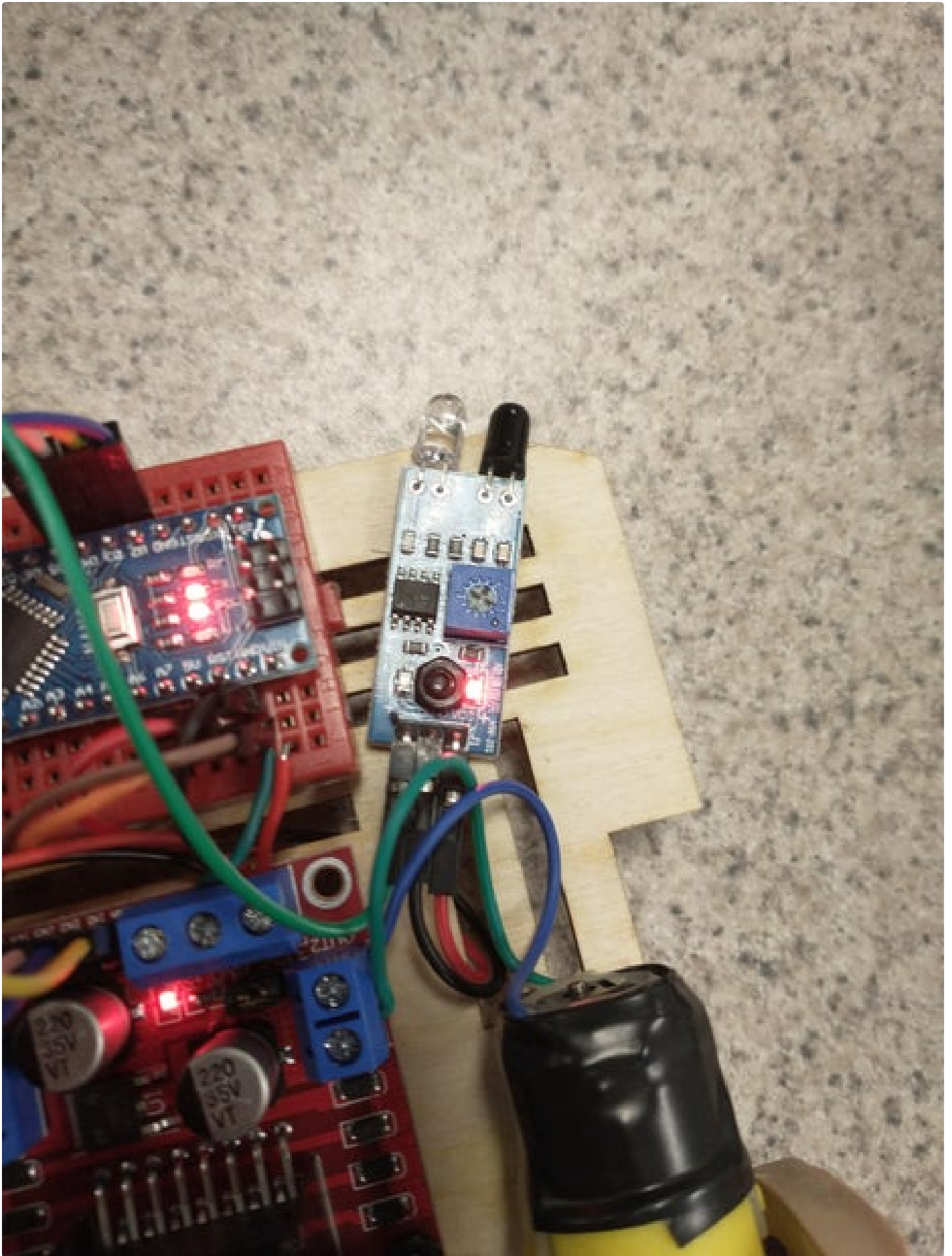
- Please make sure if your device is properly connected. You can restart the device and connect it again.

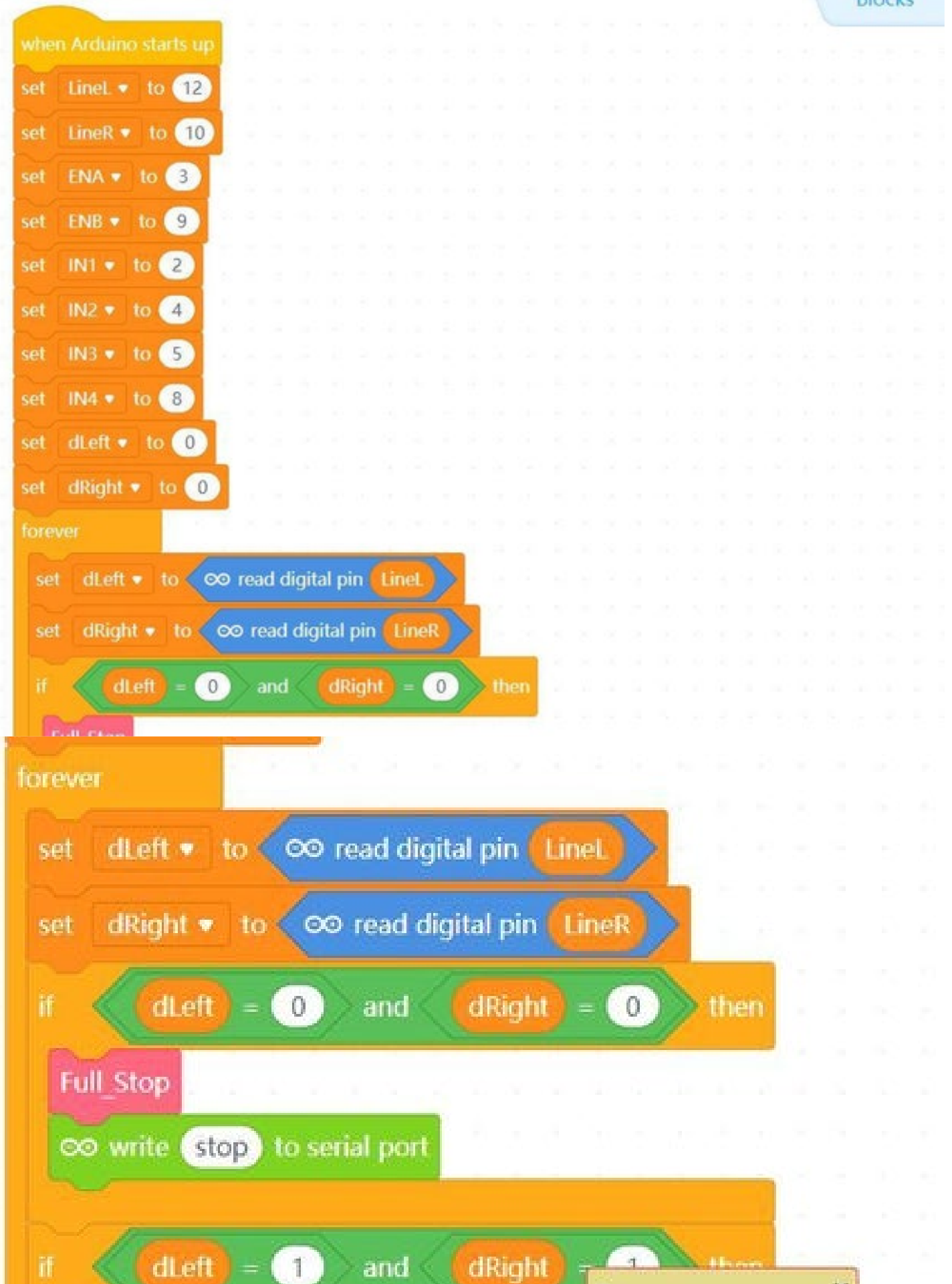
```
mmcu=atmega328p -o "C:\Users\jtang\mblock-  
avr\temp\build\out.elf" "C:\Users\jtang\mblock-  
avr\temp\build\code.o"  
"C:\Users\Public\Programs\mblock\resources\app\mlink-  
v1\external\arduino\nano\libnano.a" -lm  
avr-toolchain\bin\avr-objcopy -O ihex -R .eeprom  
"C:\Users\jtang\mblock-avr\temp\build\out.elf"  
"C:\Users\jtang\mblock-avr\temp\build\out.hex"  
arduino build finish  
processing code by middleware of arduino  
processing code completed  
get code upload driver by stk500v1  
start uploading.....
```

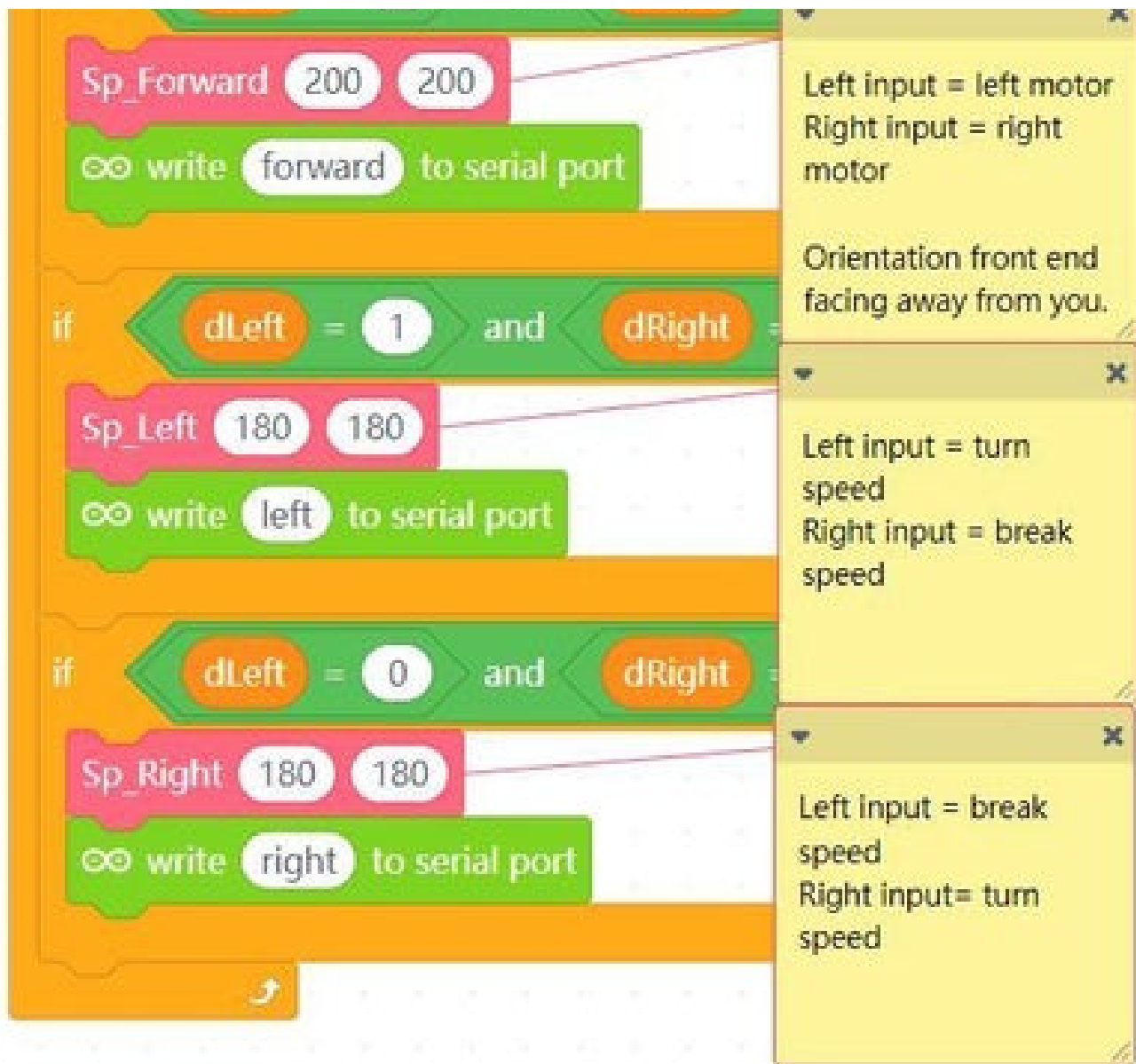
Cancel

Retry









Step 10: Make a Track

You can make track out of almost anything as long as it's a surface that doesn't fully absorb light. The sensors detect the walls by receiving IR signals reflected off a surface. At TCPL we use a combination of recycled cardboard and foam to make the track. There is no official measurement for each track piece.

For straight pieces, I cut a length of material between 6 - 15 inches long and about 2.5 inches tall. I then cut a roughly 2.5x2.5 square and hot glue it to the back of the piece to make a stand. See the fourth picture for an example.

Note: If you would rather not use hot glue you can slightly bend straight pieces and they will stand up. You can see an example of this kind of track in the video.

When it comes to curves I will bend the material into shape and that causes it to stand without needing anything glued to it. See the last two pictures for examples.

The distance between the outer and inner walls should always be 15 inches or more. This is due to the sensor's max range of 30cm. If both sensors see a wall the robot will stop.

Once the track is set up I will place some painter's tape to mark the finish/starting line. You can find the app TCPL uses to capture lap times here: [Apple](#) or [Android](#).

TCPL runs a very simple time attack event where participants and/or teams get 3 laps to set a personal best time. The participant and/or team with the fastest time wins for that week. TCPL will change the track layout weekly to keep things interesting.

<https://www.youtube.com/watch?v=vxfp70L4fLA>



