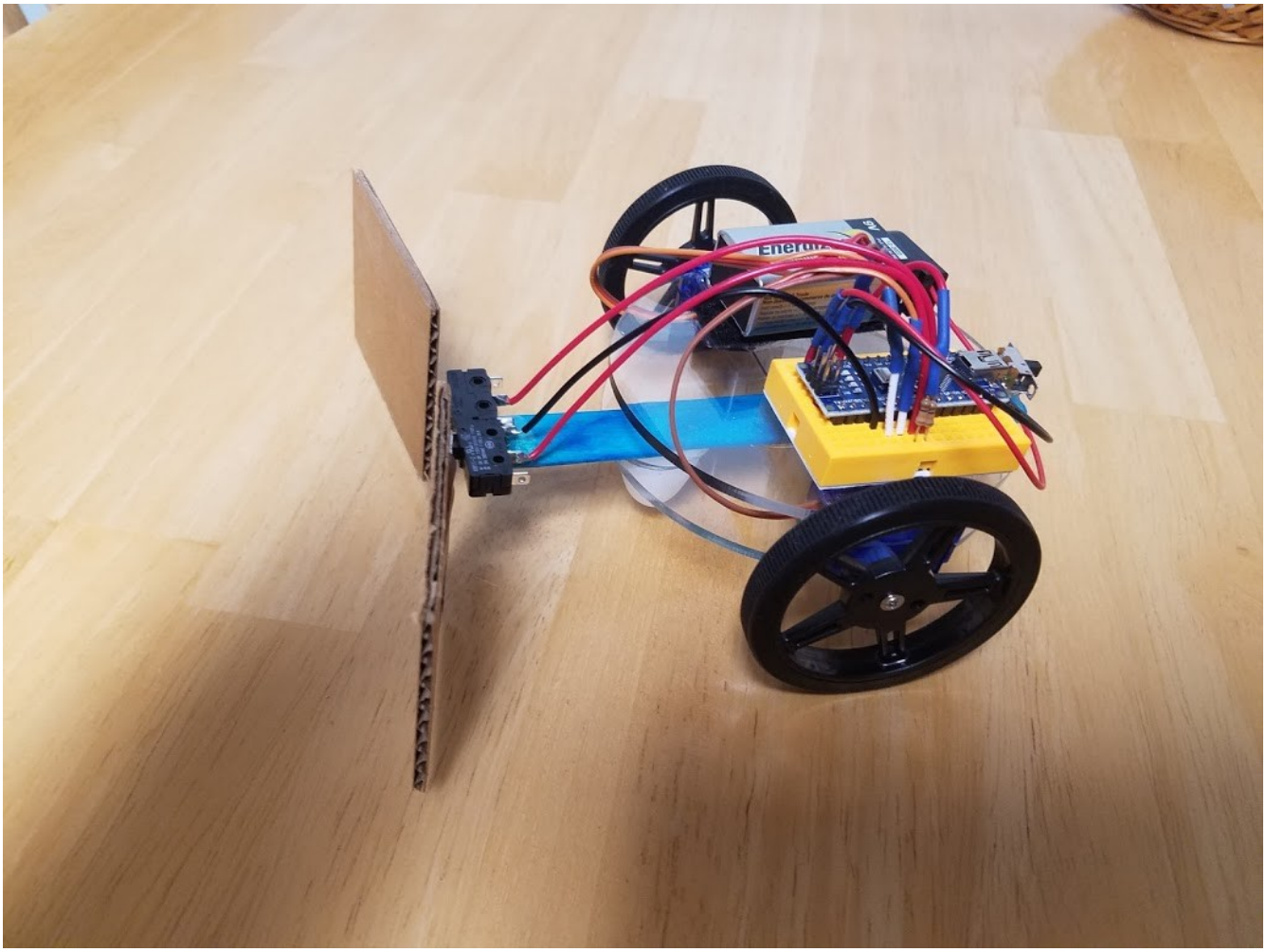
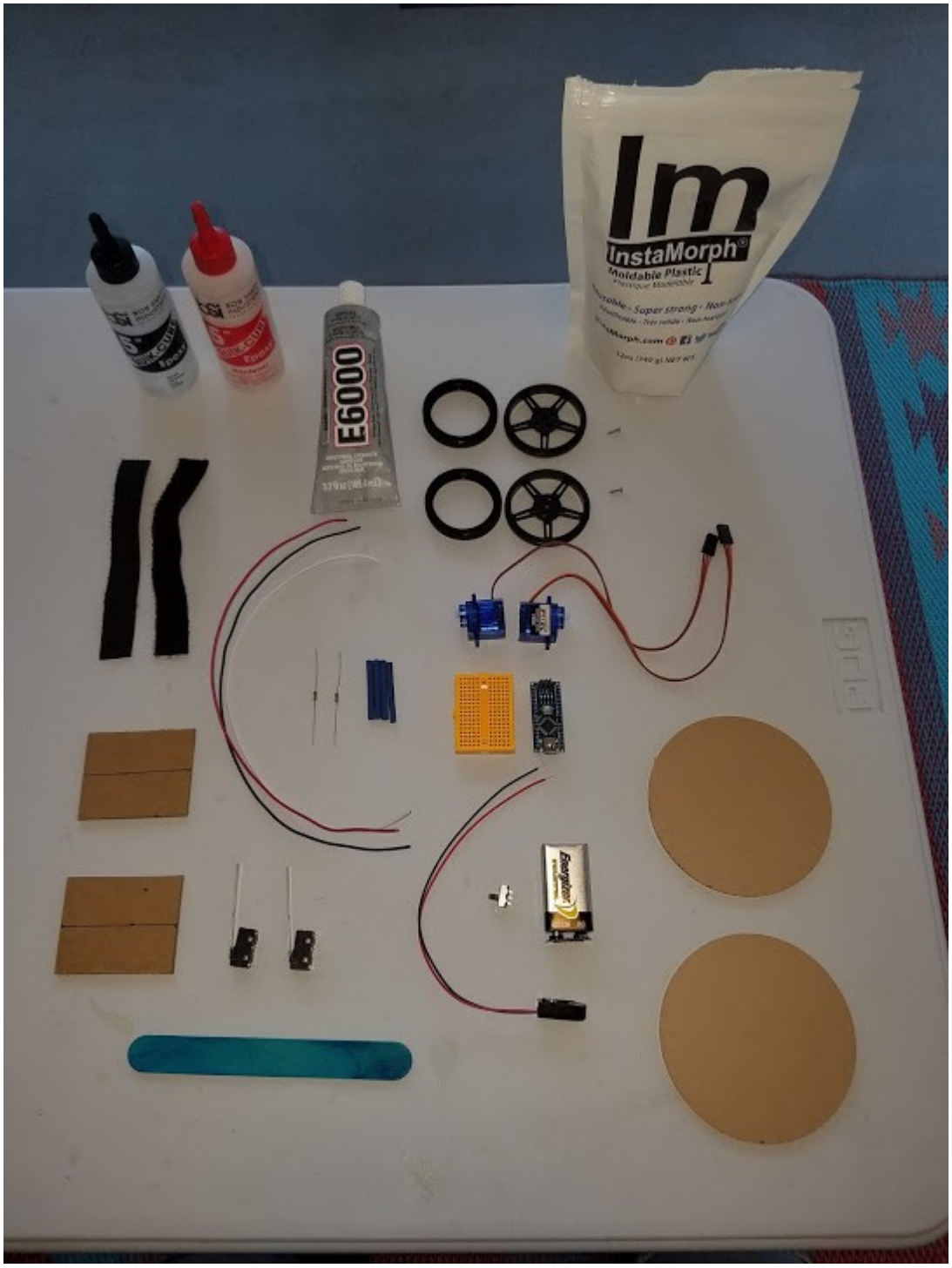
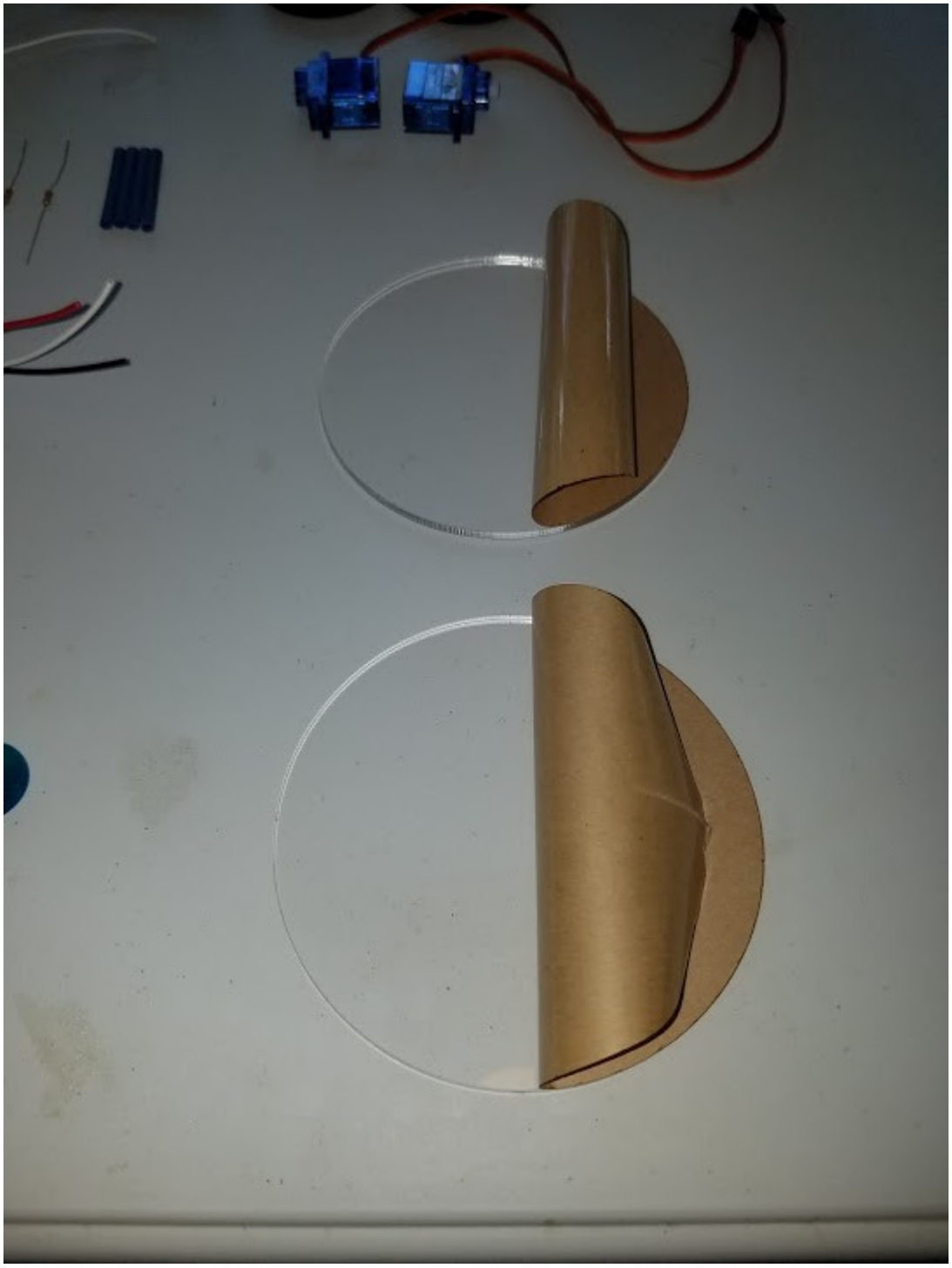


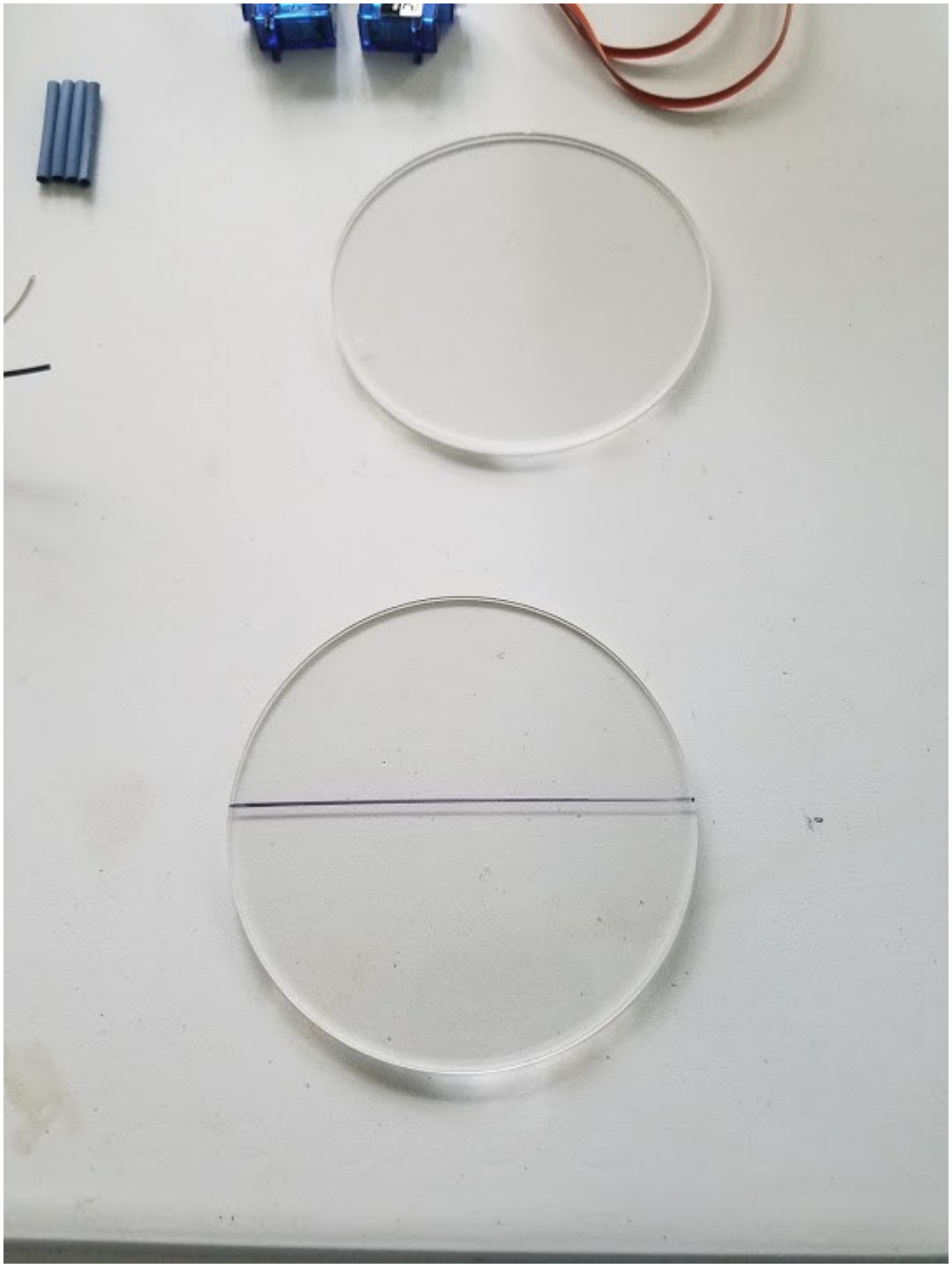
This is called the "Nano Turtle" because it's built with the Arduino Nano and it looks like a turtle. Turtles have a long history in mobile robotics from William Grey Walter's "Machina speculatrix" to the Logo programming language "turtle" to Willow Garage's ROS (Robot Operating System) an open source robotics framework which names it's distributions after various turtle themes.



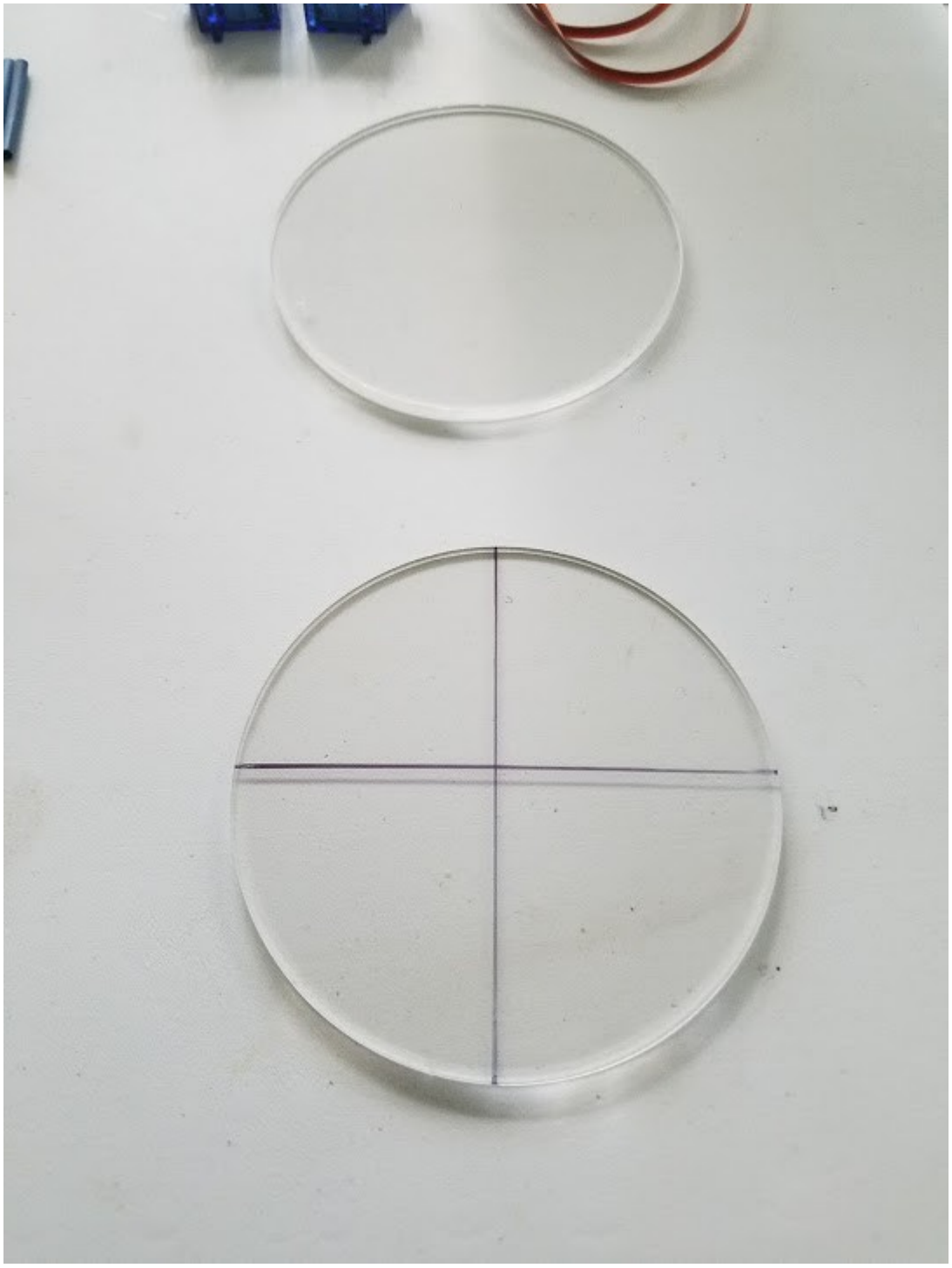
Here are all the parts and materials necessary for this build. If you have a couple of continuous-rotation servos and an Arduino DO NO LET ANYTHING KEEP YOU FROM BUILDING THIS PROJECT! For example if you do not have servo wheels... glue the wheels you do have to the servo horns that came with your servos. If you don't have acrylic rounds use something else for the body. This is homebrewed robotics... be creative. Have some fun but get it done!



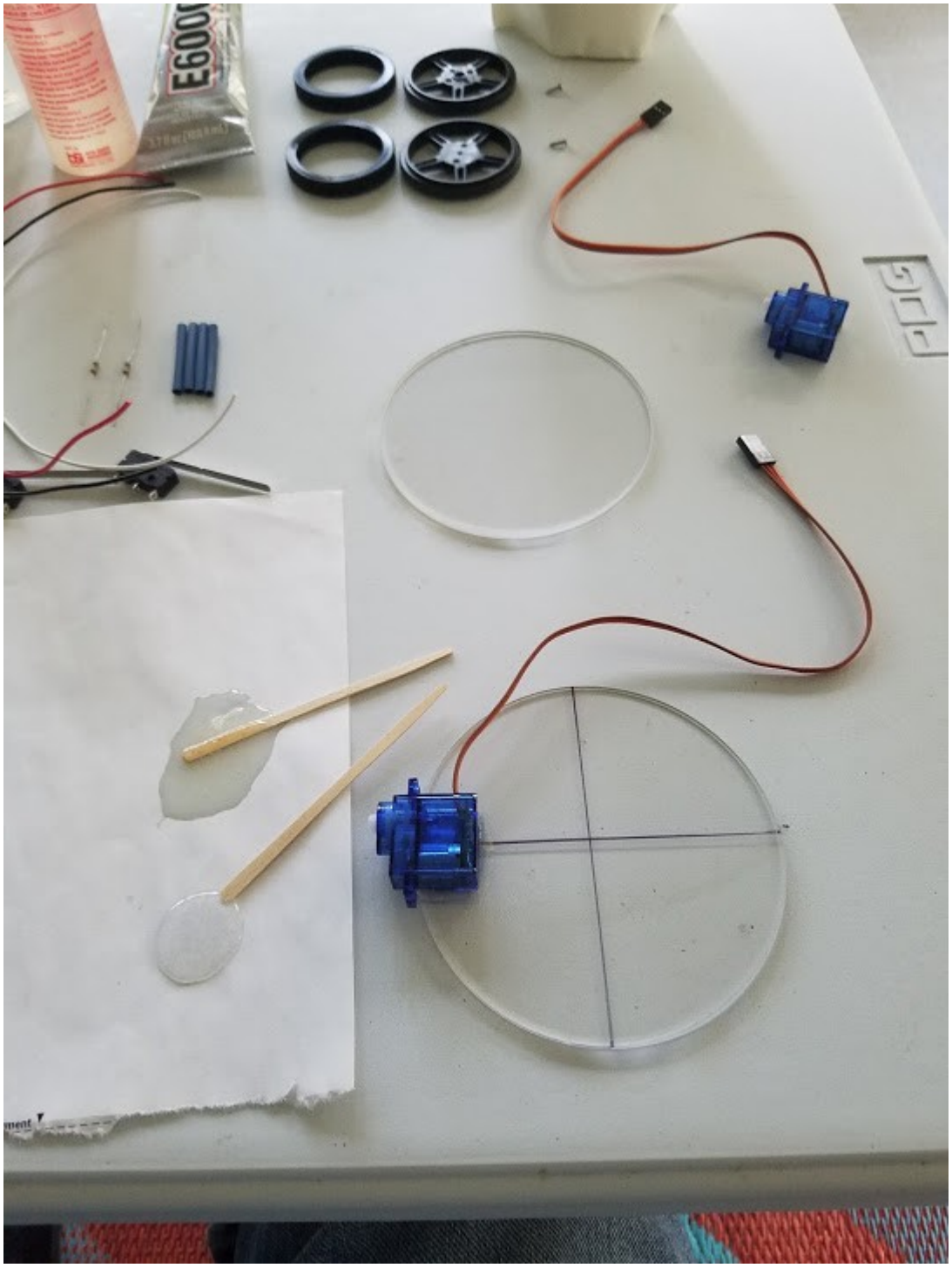
1. Peel covering off both 4” acrylic rounds.



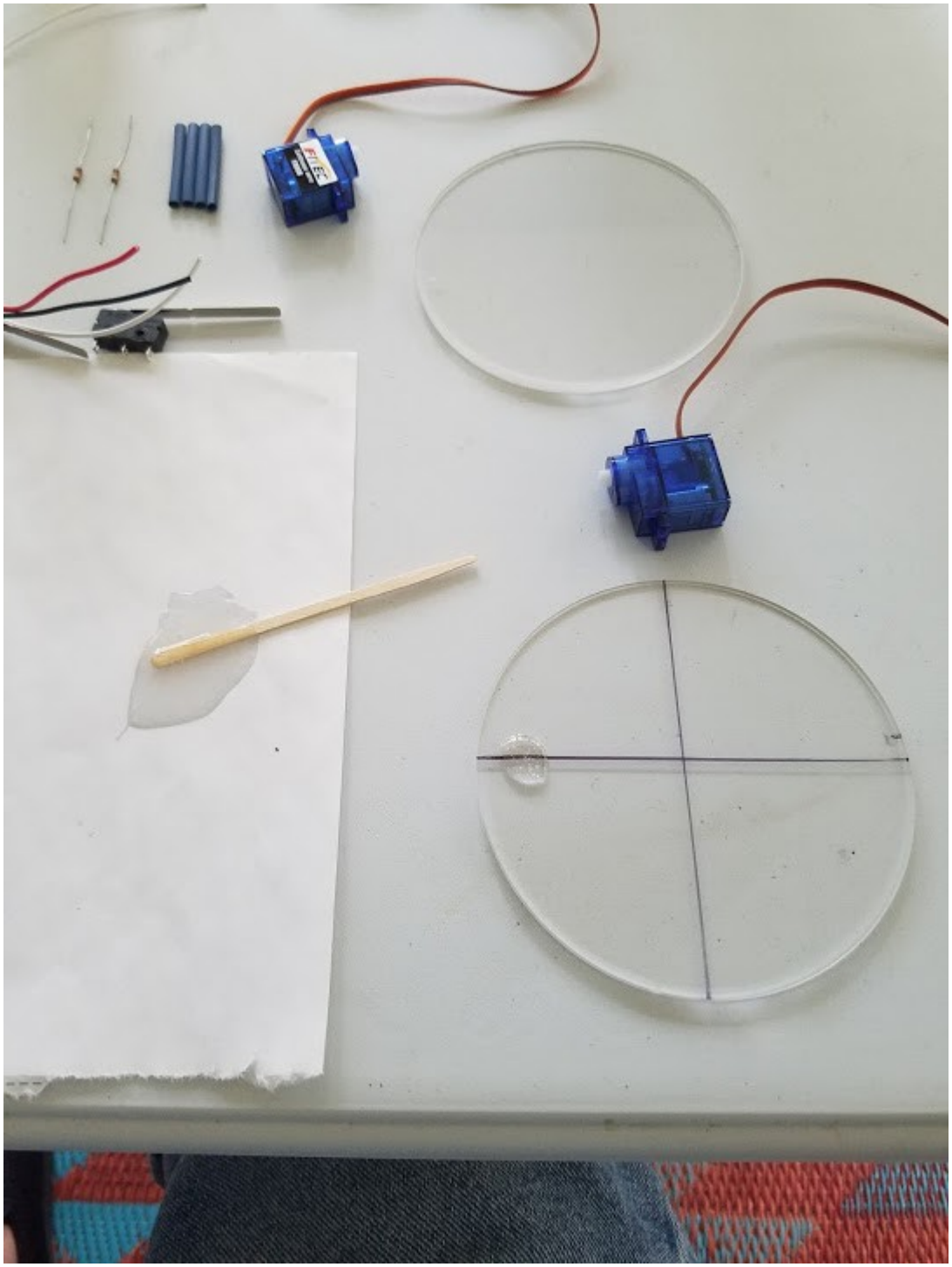
2. Draw a line through the center of one 4” acrylic round using an Ultra Fine Point Sharpie.



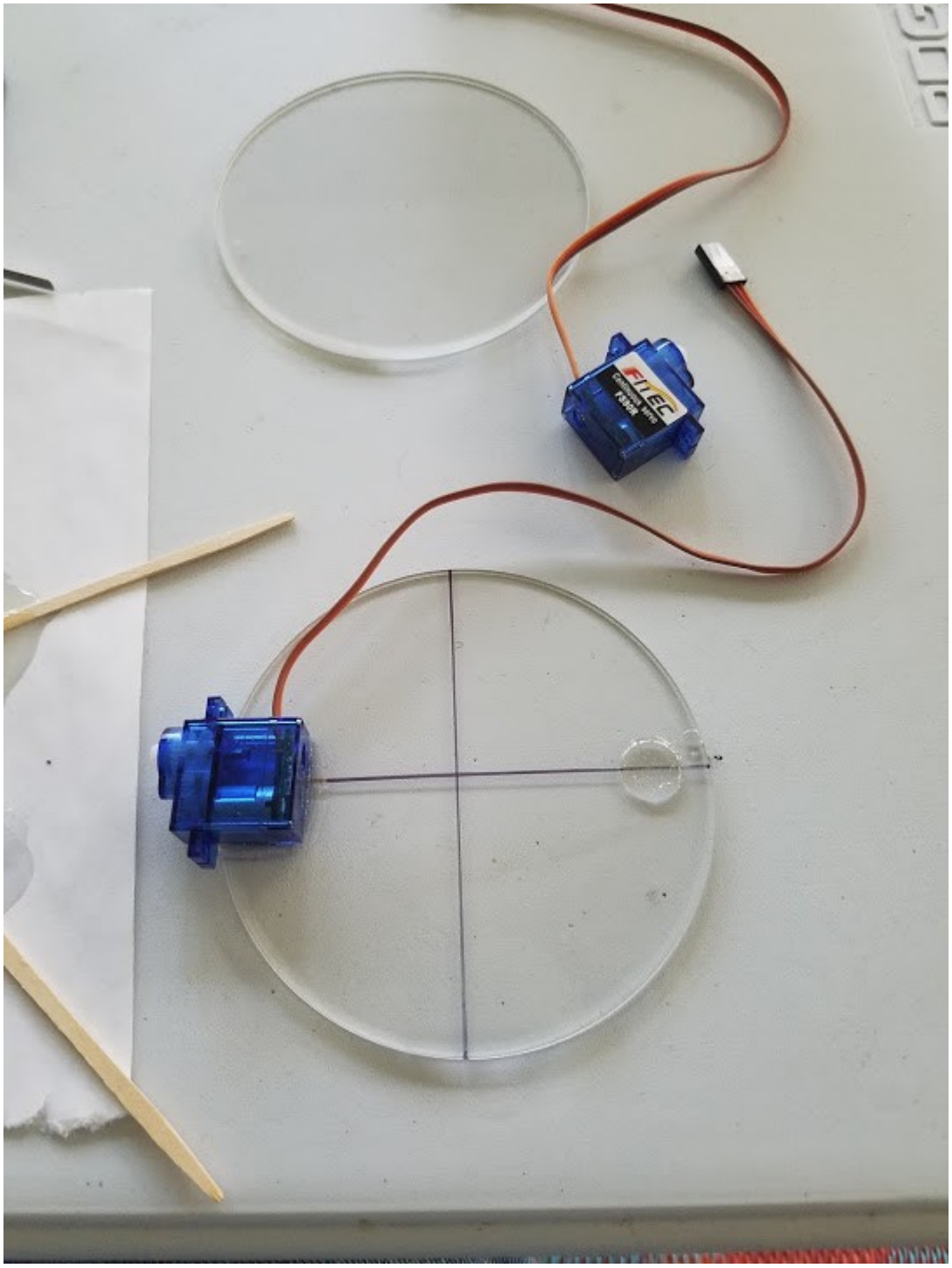
3. Draw another line through the center of the 4” acrylic round perpendicular to the first.



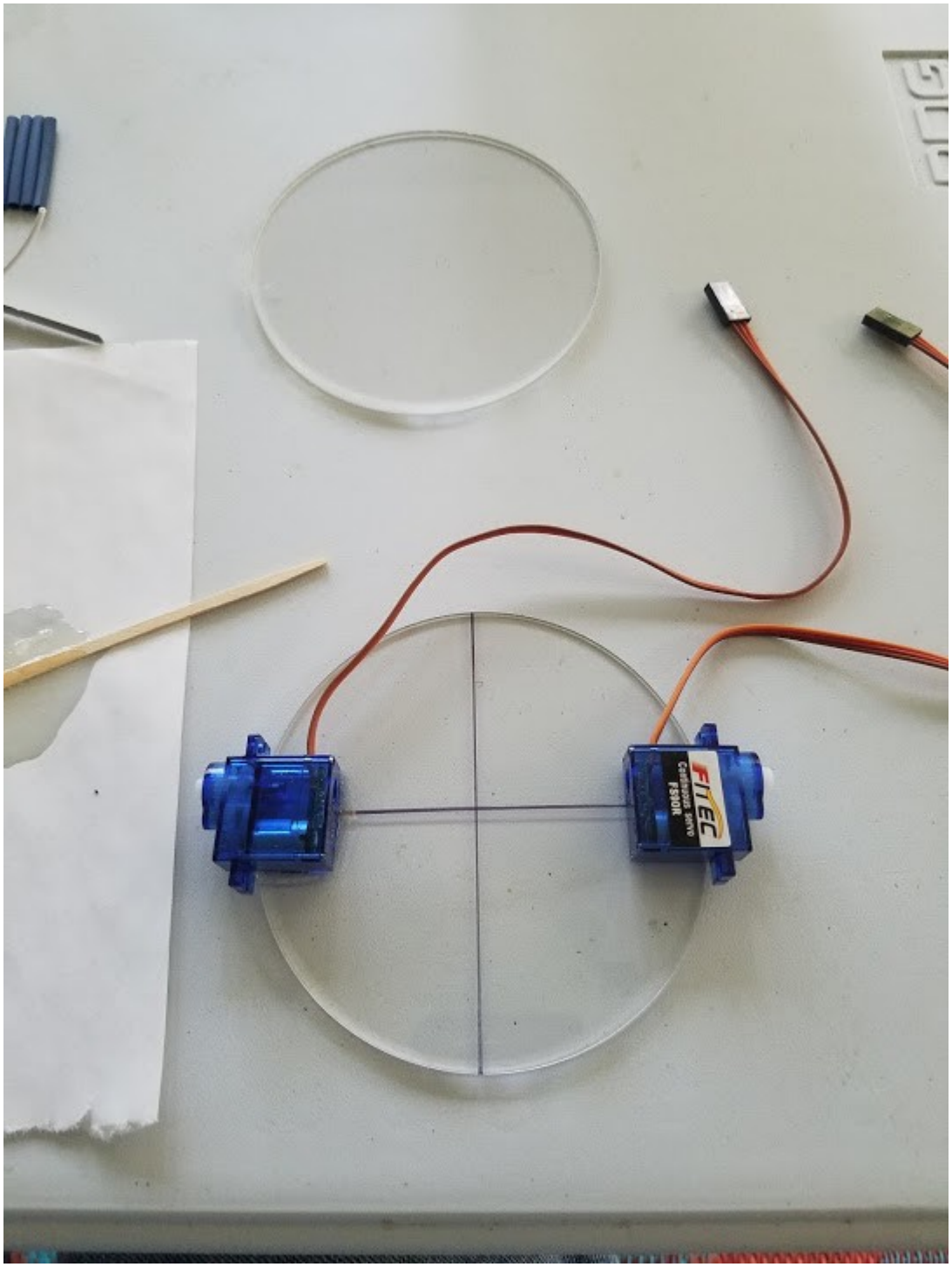
4. Place one servo on center of horizontal line. Position it such that the gear-shaft is facing forward, centered and the “ear” attachments overhang the acrylic round.



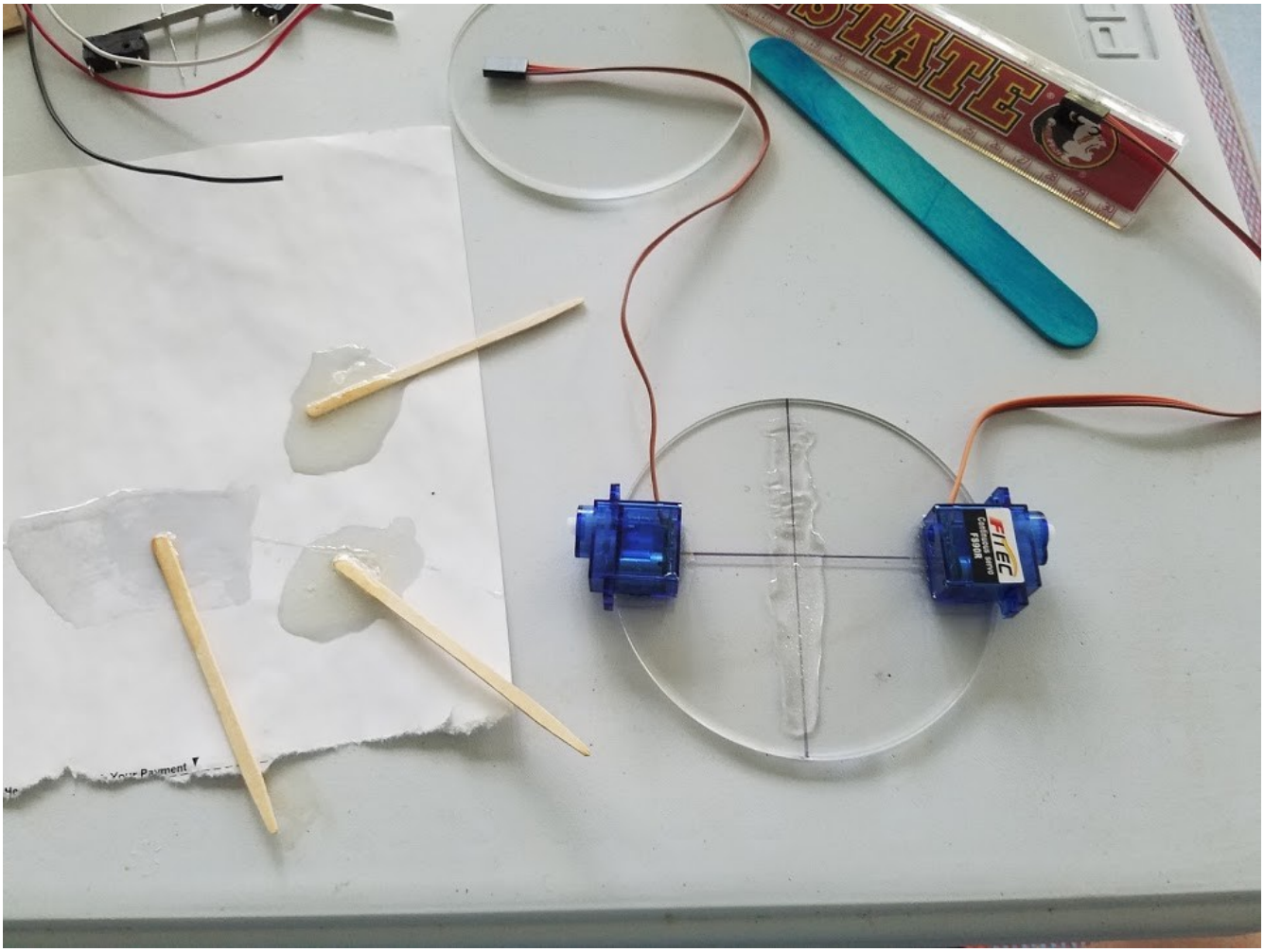
Epoxy in place. Mix ½ epoxy and ½ hardener. Mix with craft picks. Use a timer… you can get an app for your smartphone. I generally set it for 7 minutes and spend 1 minute mixing the epoxy. Use the craft pick to place it and finally hold the servo in place until the glue hardens. I recommend Bob Smith's 5-minute epoxy but there are other good brands including LOCTITE.



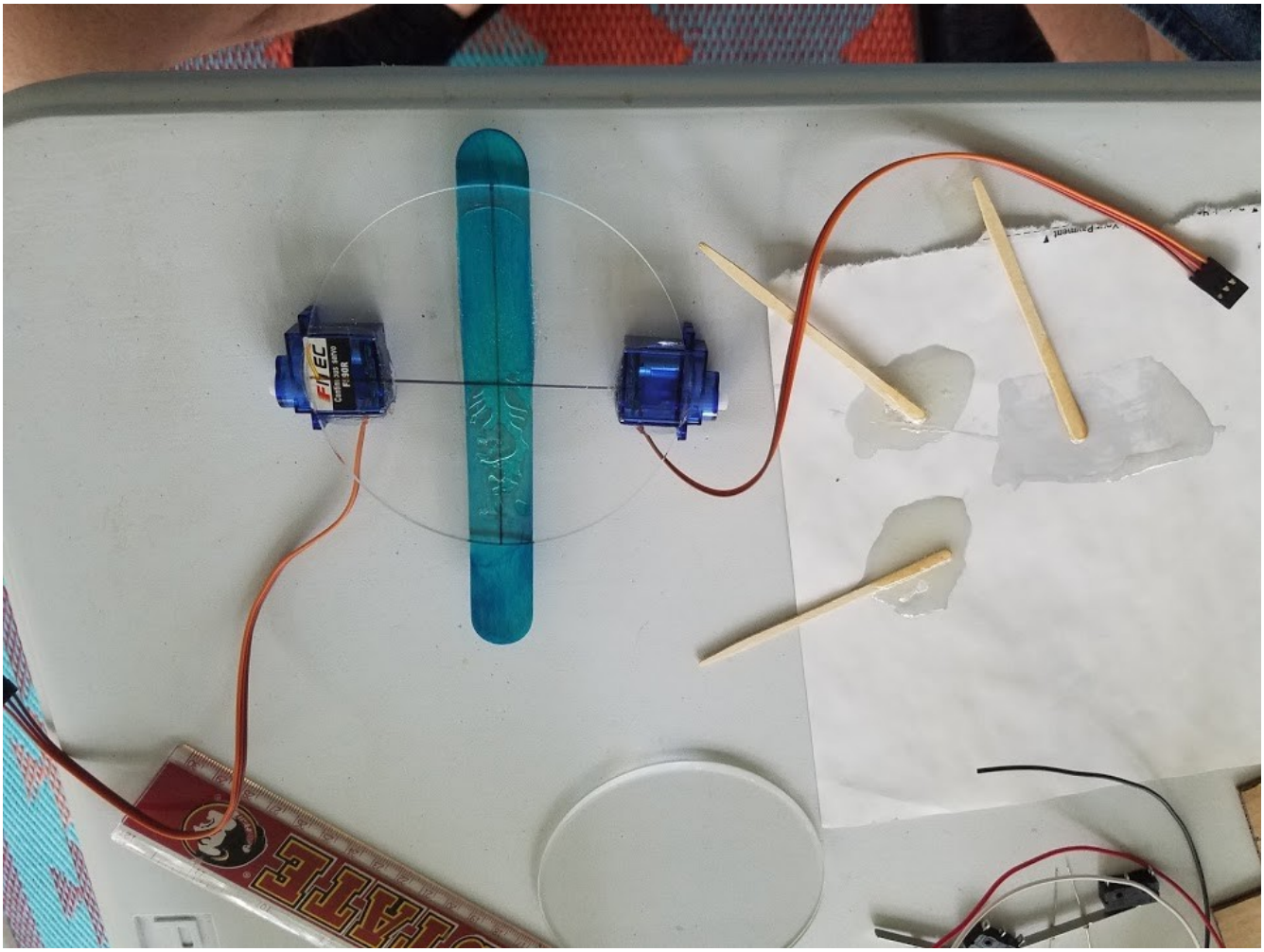
5. Do the same thing for the other side. First visualize it then make a dab of epoxy and adhere.



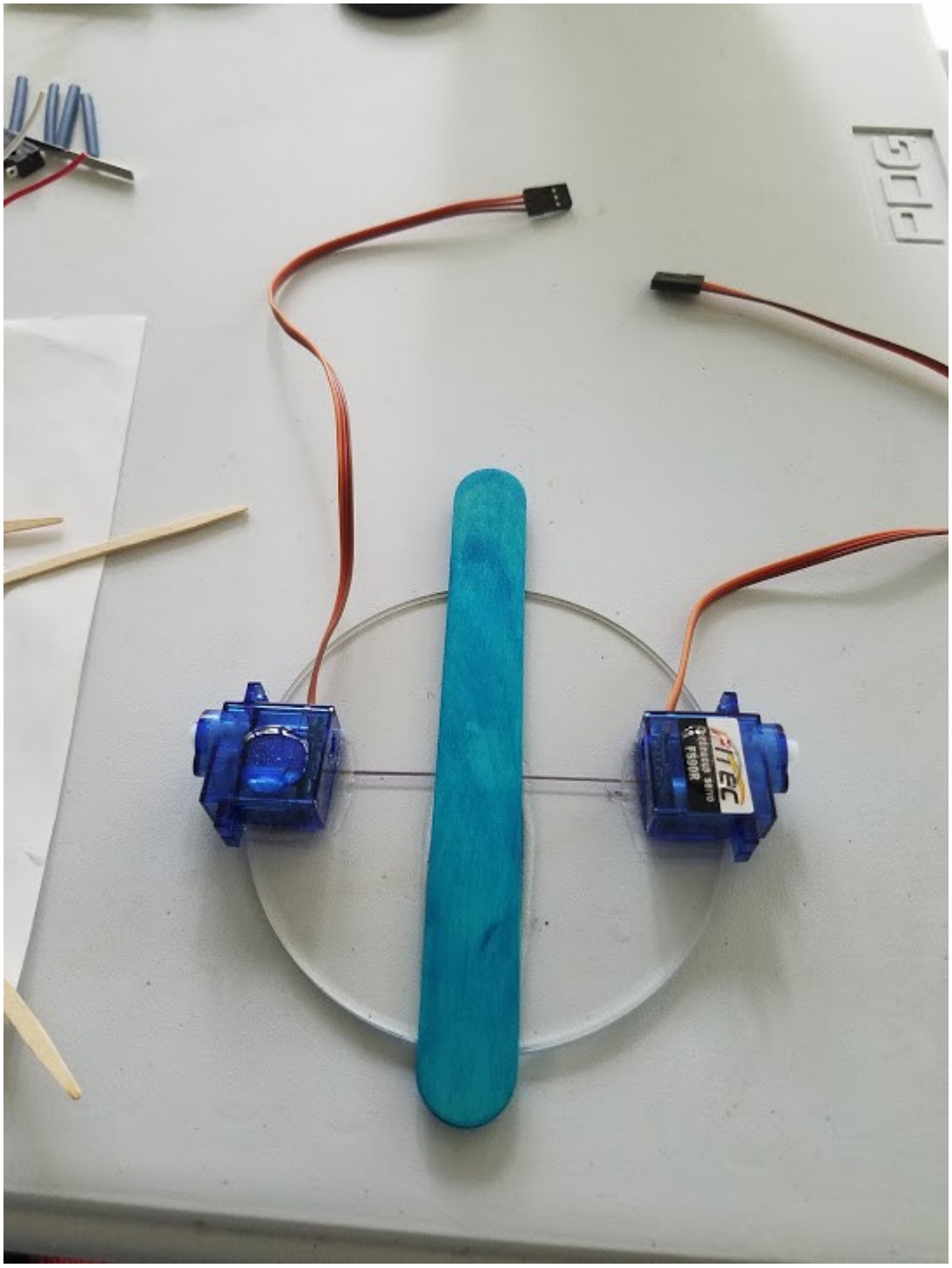
You should hold the servos in place for at least 5 minutes. Consider it an exercise in patience.



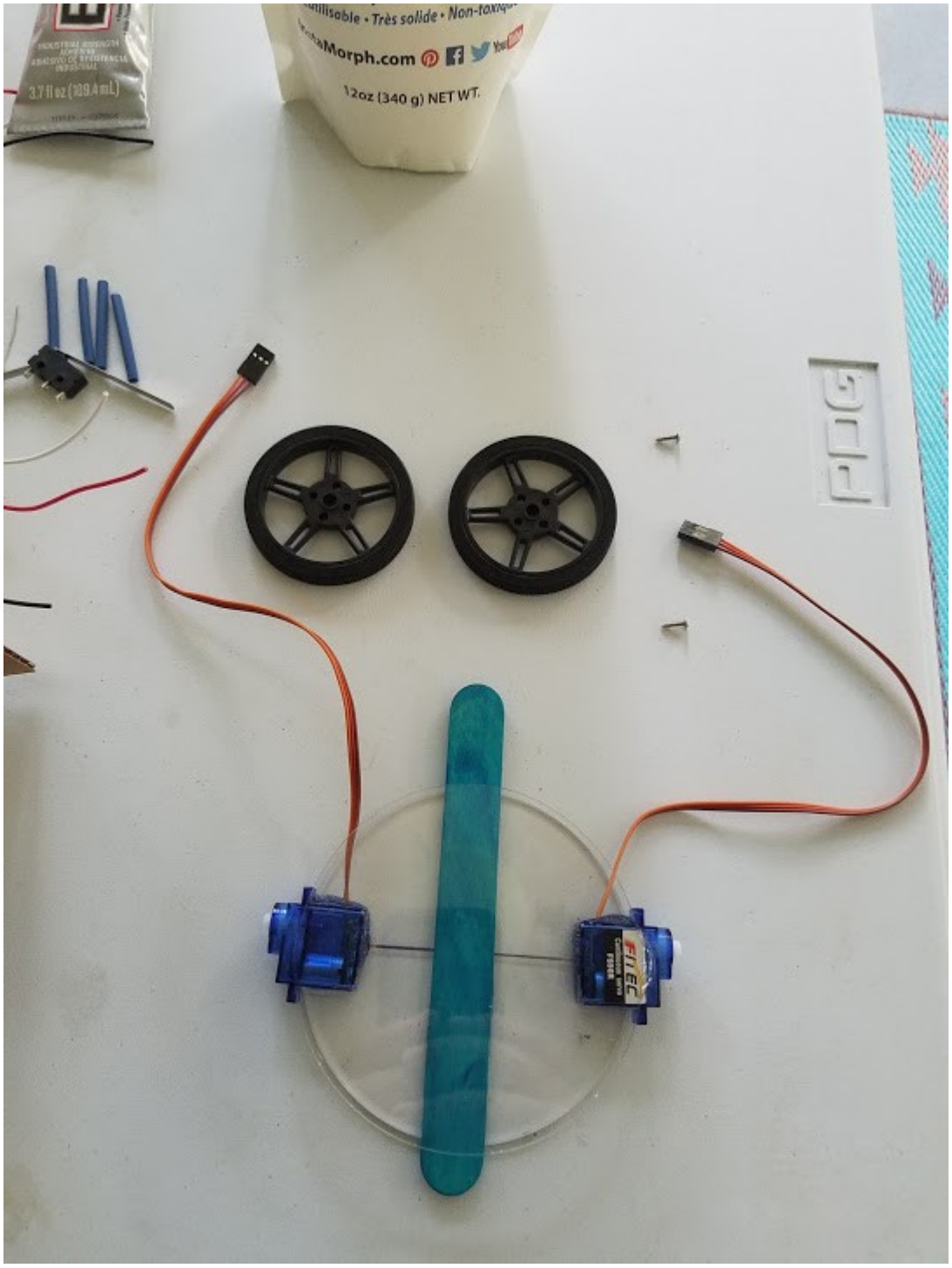
6. Take a 6” Popsicle stick and glue it to the perpendicular line. This will be for the snap-action switch sensors. Have approximately 1.5” protruding out the front and a .5” tail. Again hold in place for at least 5 minutes. Consider it an exercise in patience.



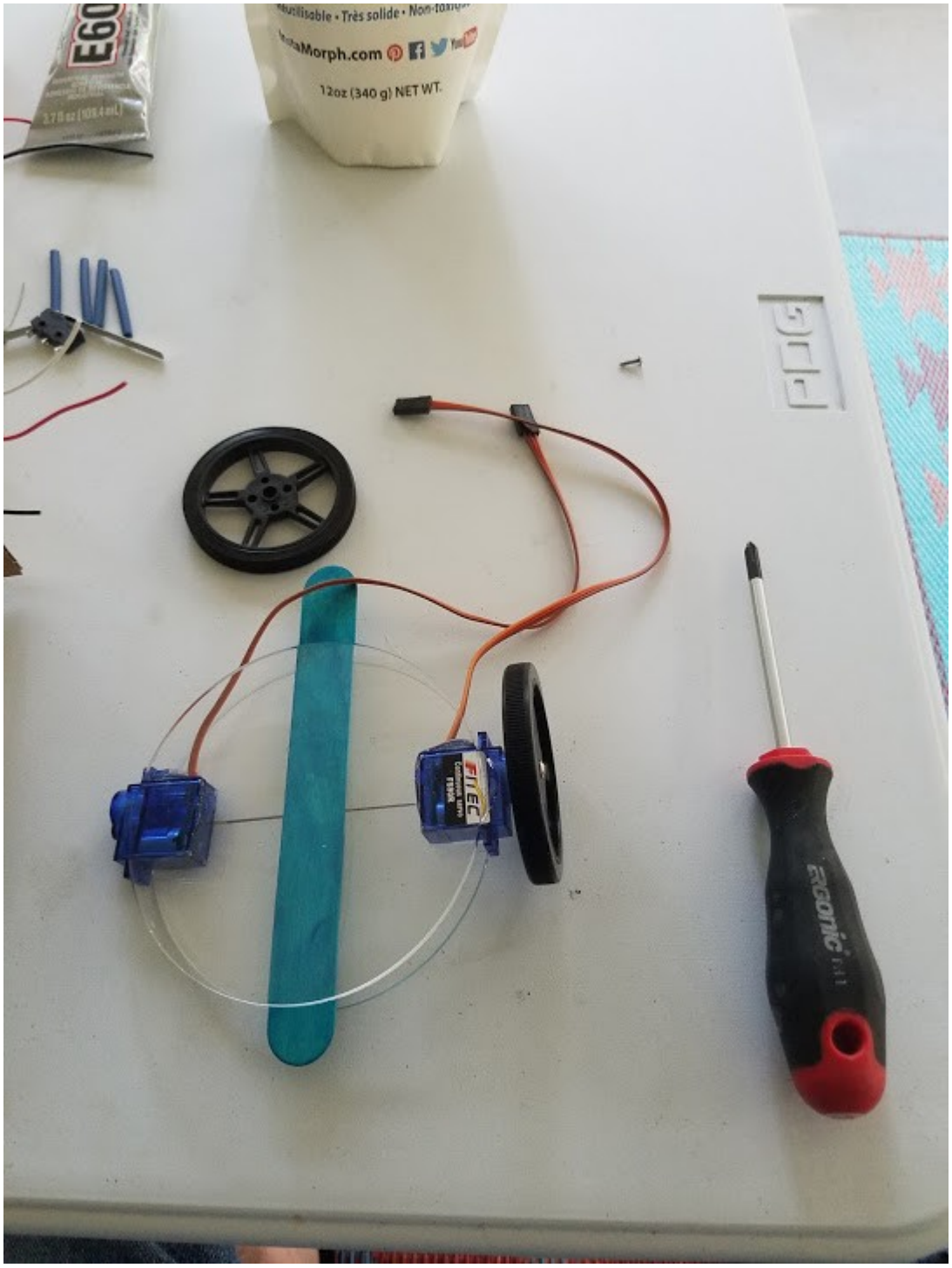
You can center the stick looking at the bottom.



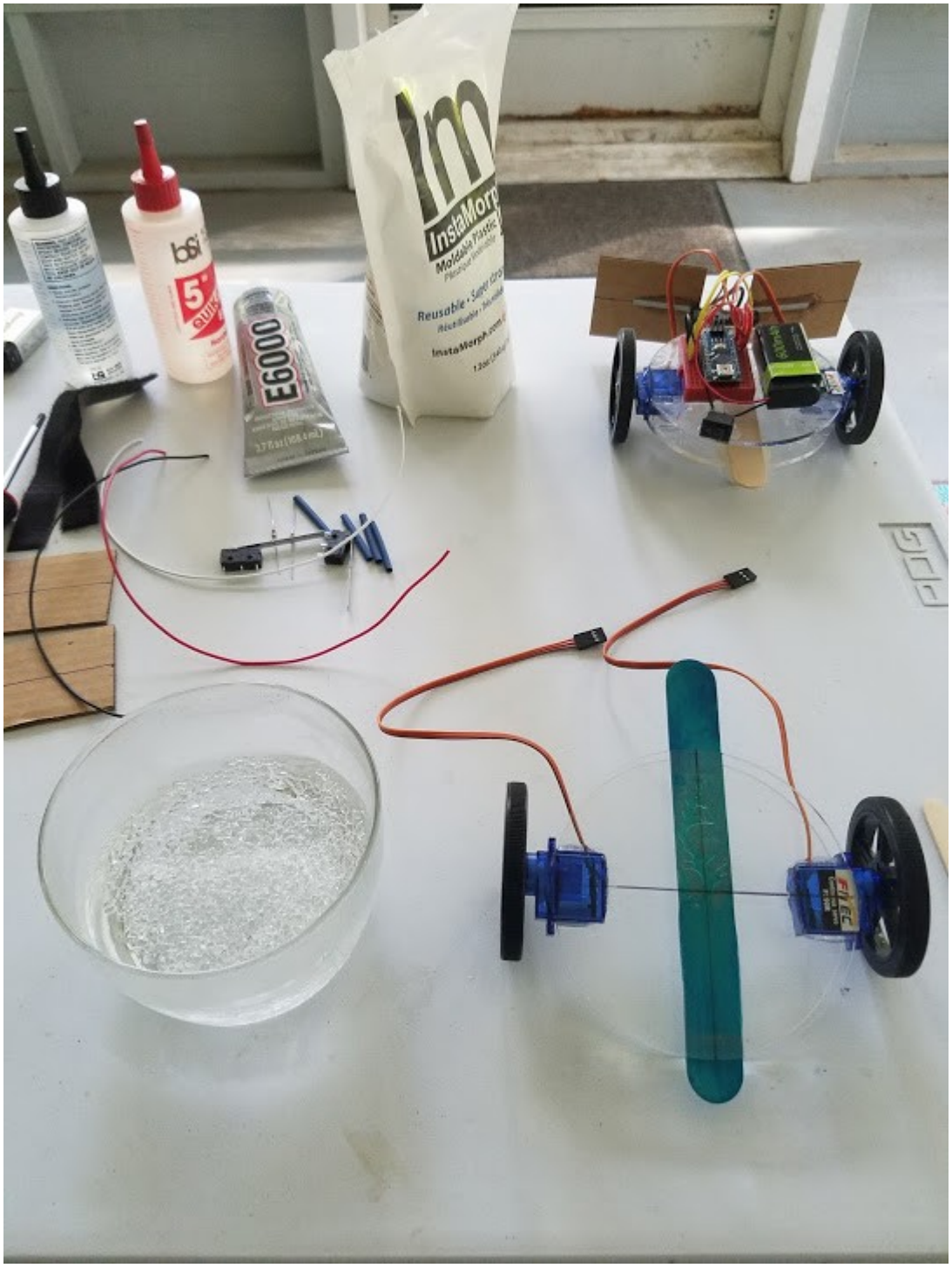
7. Now finish the sandwich by gluing the top 4” acrylic round to the two servos.



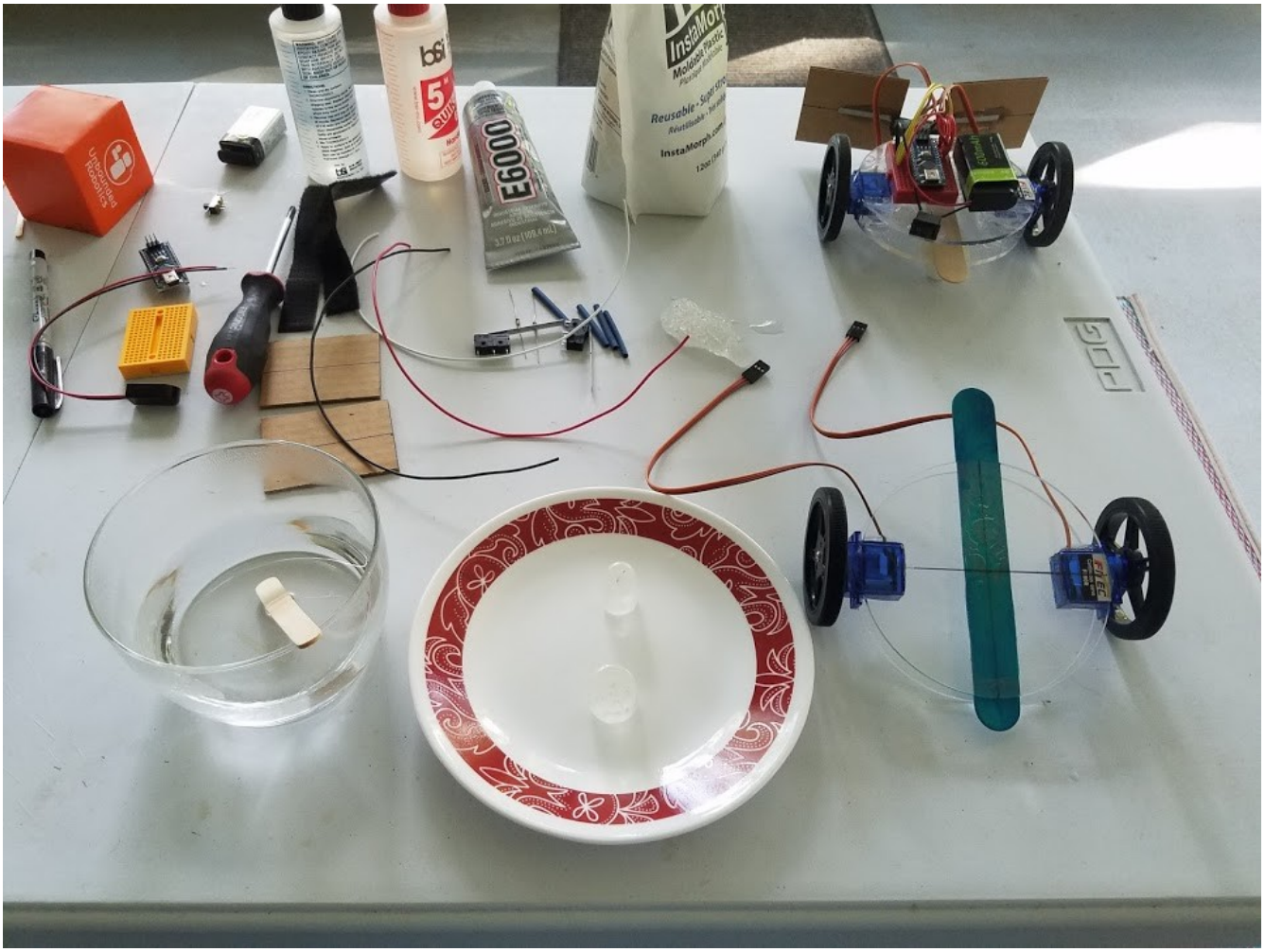
Make sure the 4” acrylic round is centered! Again hold in place for at least 5 minutes. Consider it an exercise in patience.



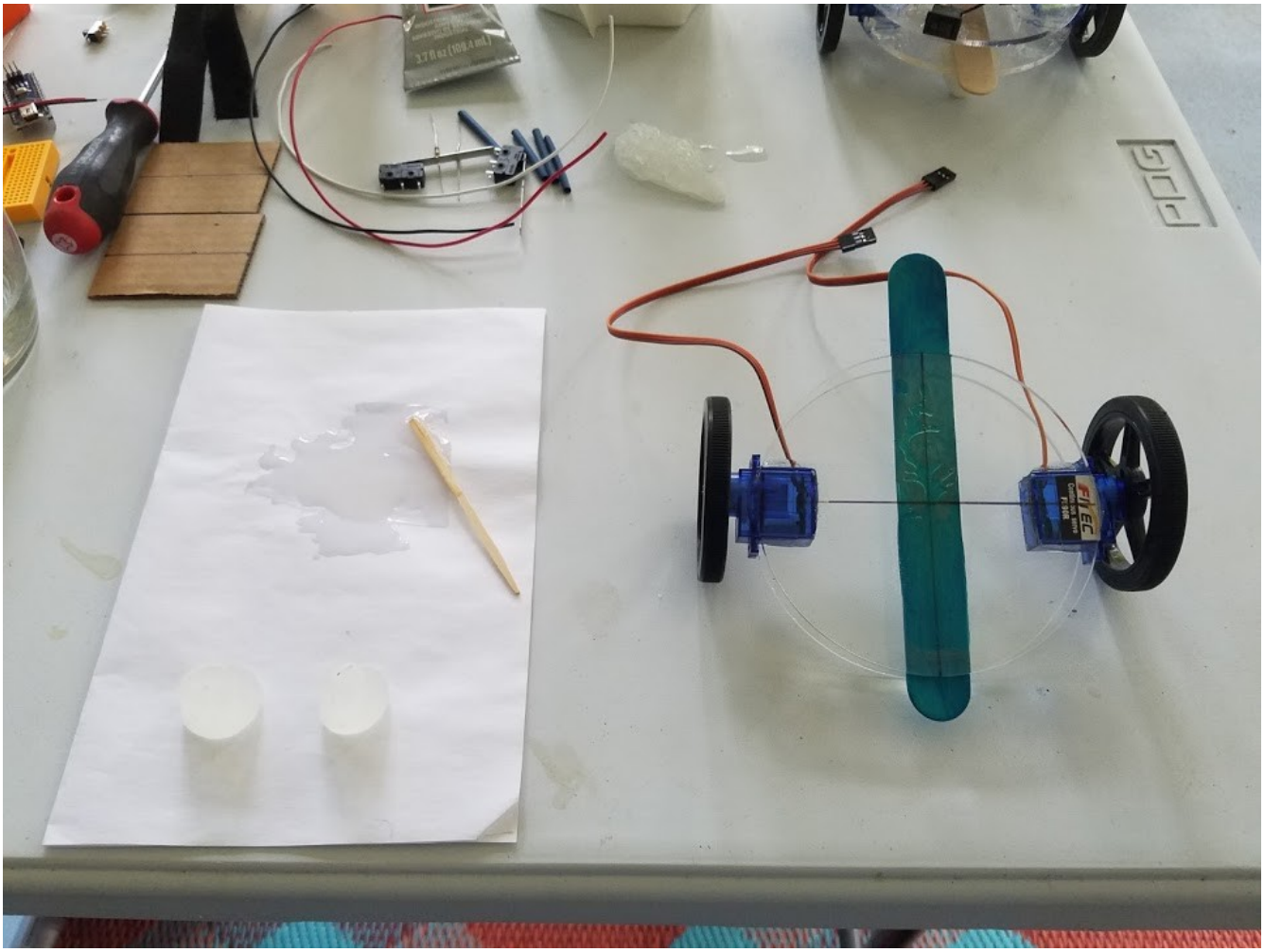
8. Next thing is to attach the servo wheels. Put the tires on the wheels and attach to servos with screws.



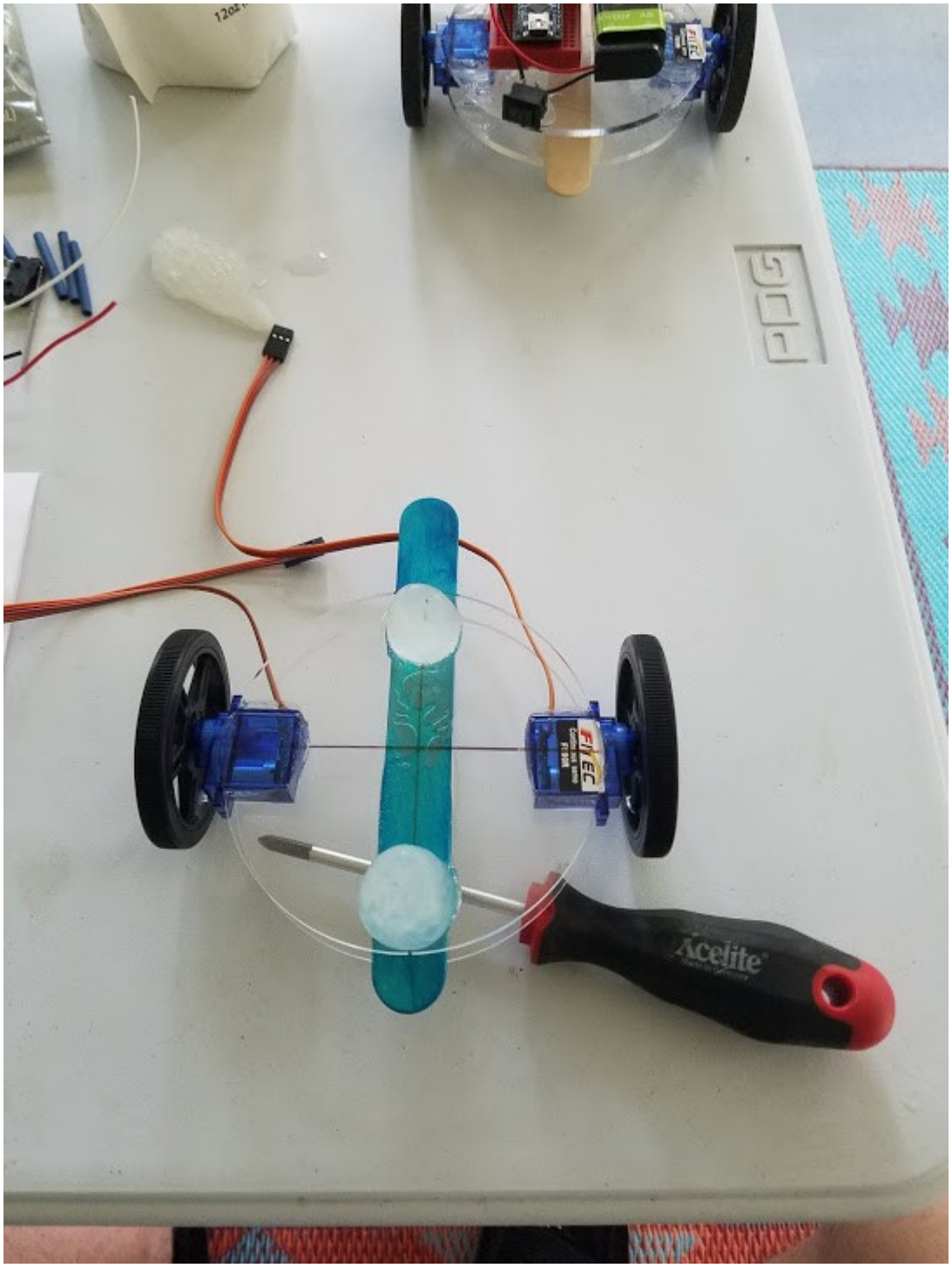
9. Next you'll need to create some standoffs to balance the robot while it drives. I use "Instamorph" which is a thermal polymer. You put the plastic beads in a glass bowl and heat in a microwave for 3 minutes and shape into whatever shape you want.



Here I've made two pegs to balance the chassis. When the plastic cools it retains its shape. Do not let it touch plastic (like the acrylic rounds) until it is cooled otherwise it will stick. Heat in a glass container and handle with a wood like a Popsicle stick.

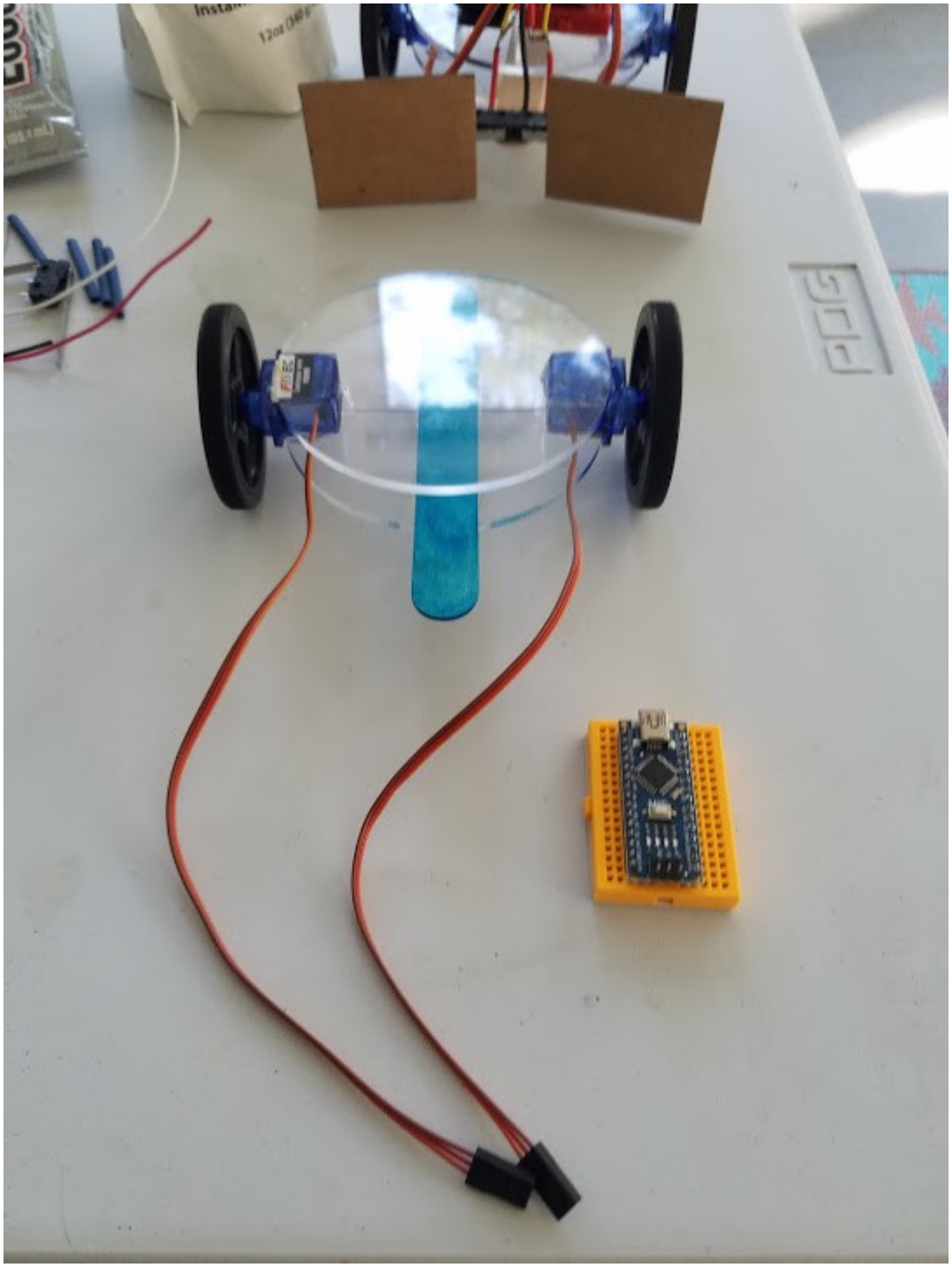


10. Don’t sweat it if you don’t get the shape or height of the pegs the first time. You can reheat the Instamorph and re-do it until you get it right. The front and rear pegs should be short enough the servo wheel tires touch the ground and are able to drive the robot. The robot should rest on the rear peg.

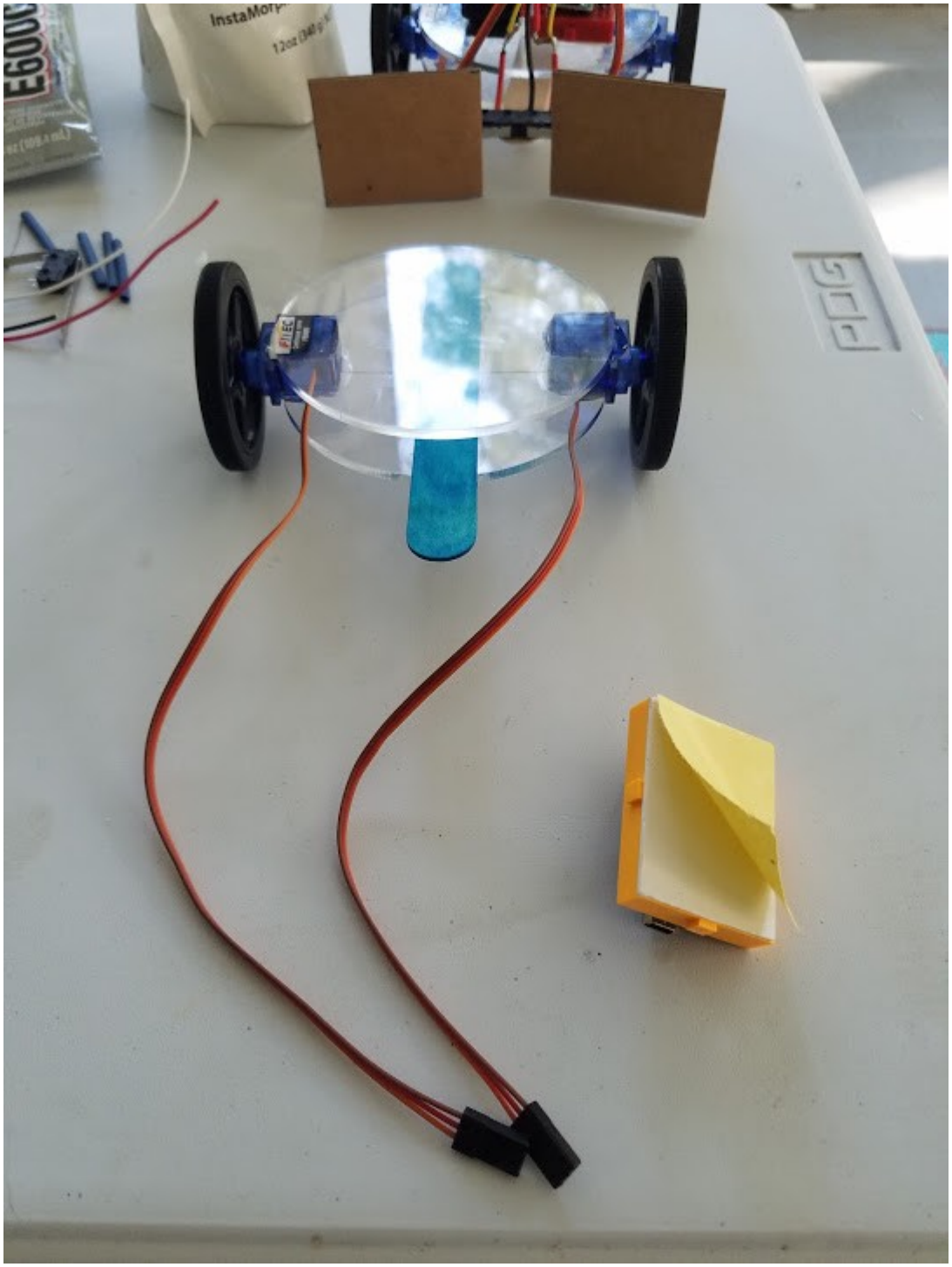


11. Glue the two pegs in place with 5-minute epoxy. Hold in place until epoxy glue cures.

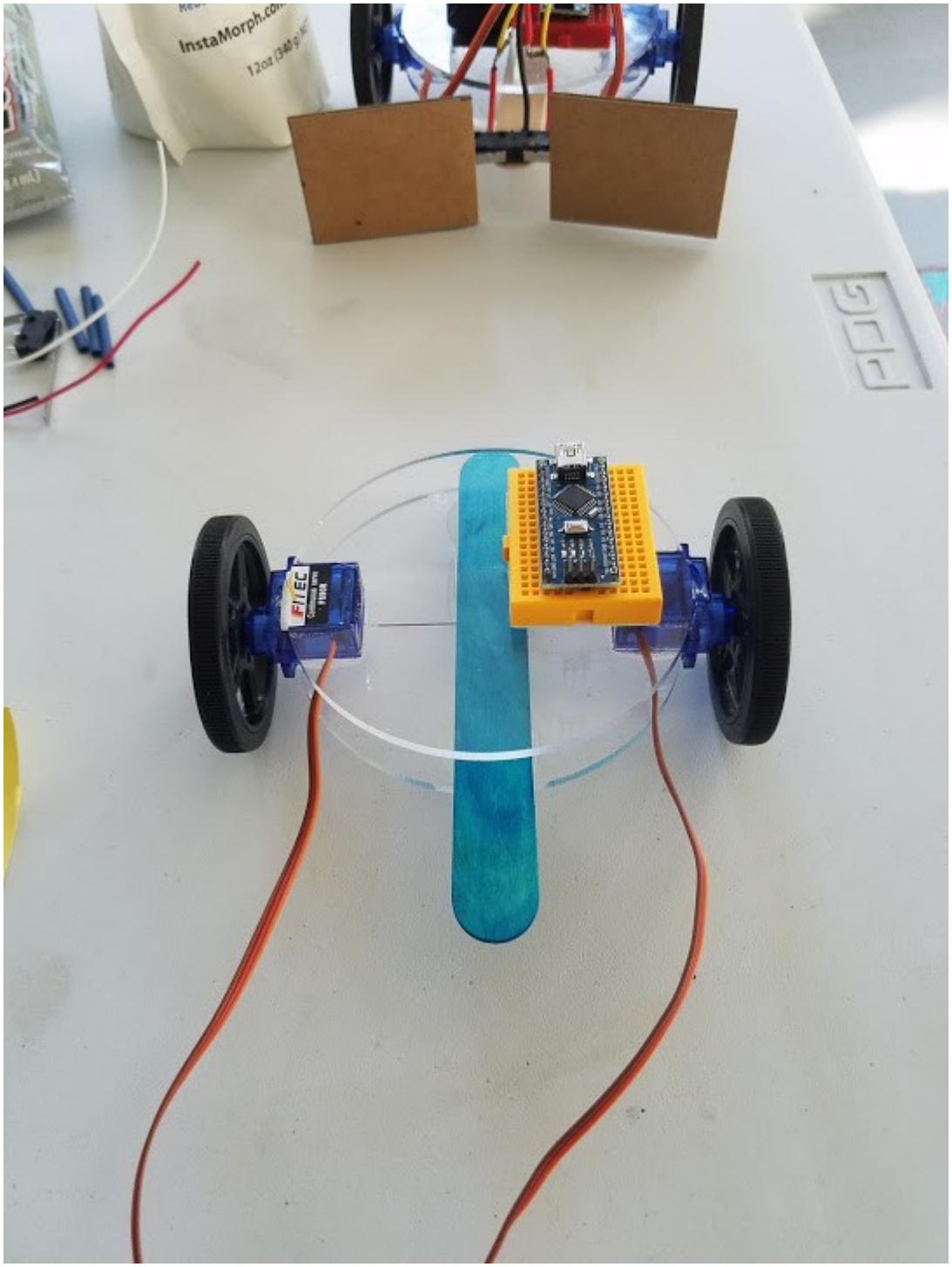
12. Next step is to carefully push the Arduino Nano into the solderless breadboard leaving one row in the top and bottom and columns on the side with VIN (Voltage In).



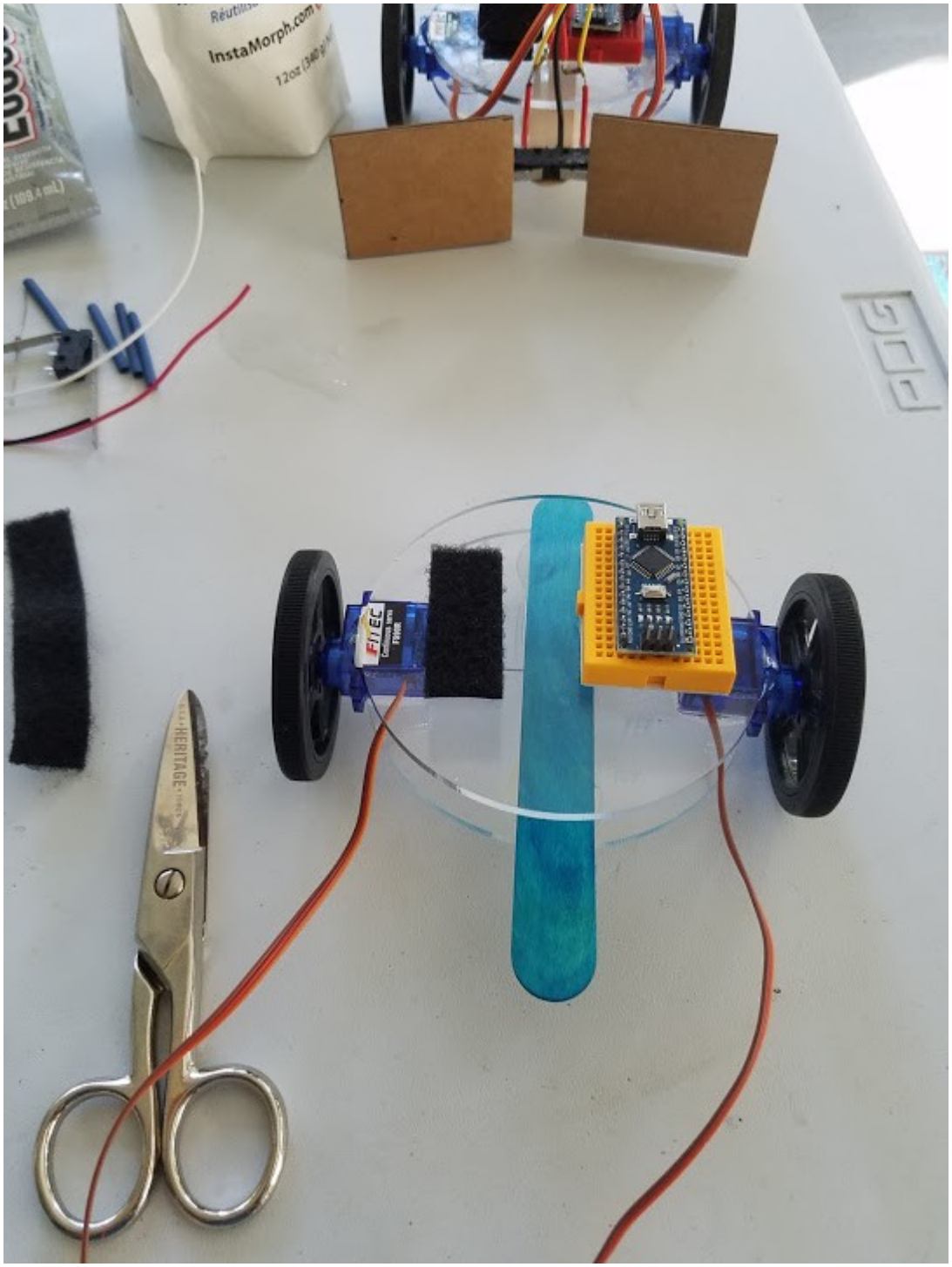
Note: the microUSB connector should go towards the rear.



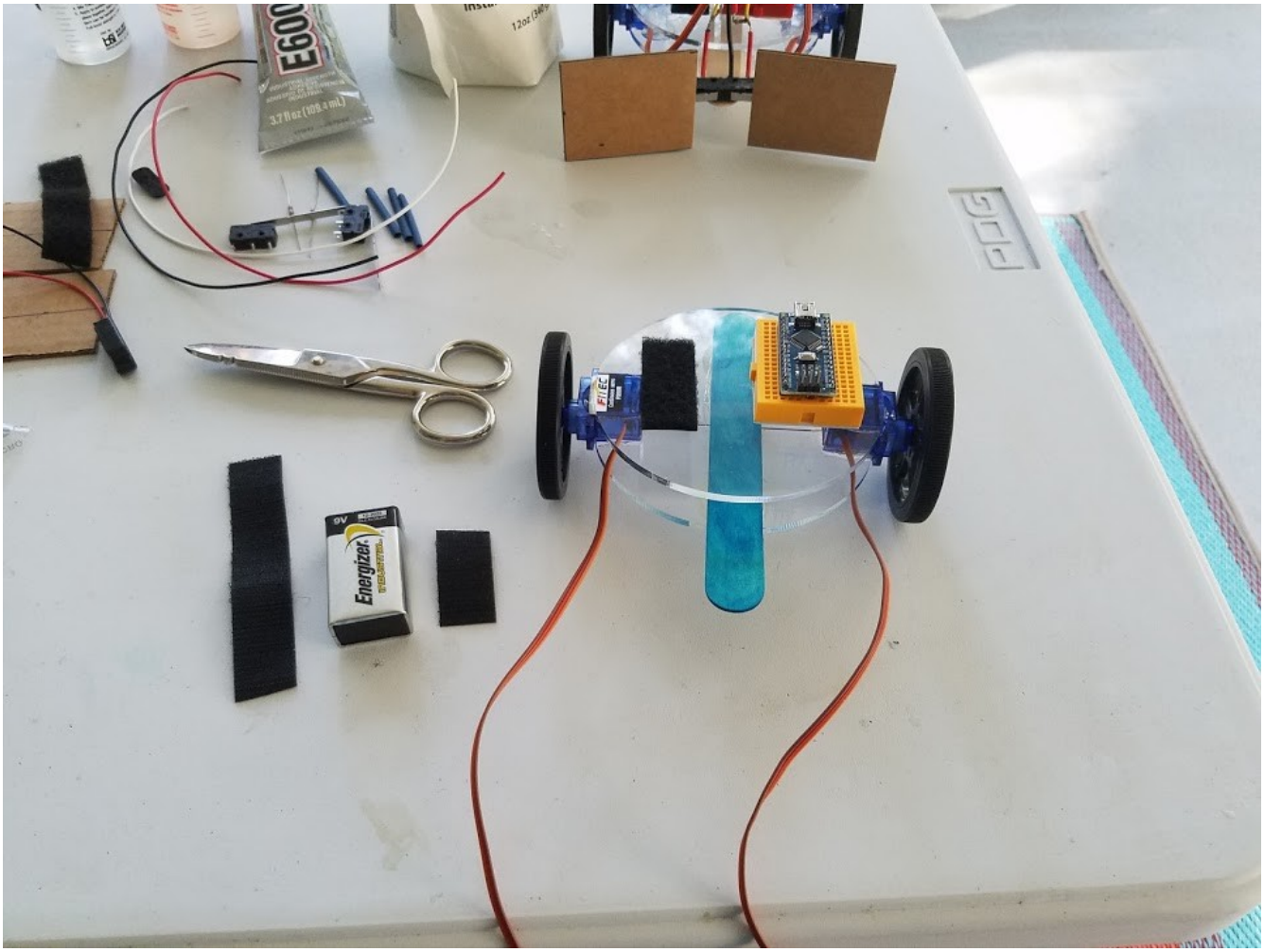
13. Remove covering from bottom of solderless breadboard.



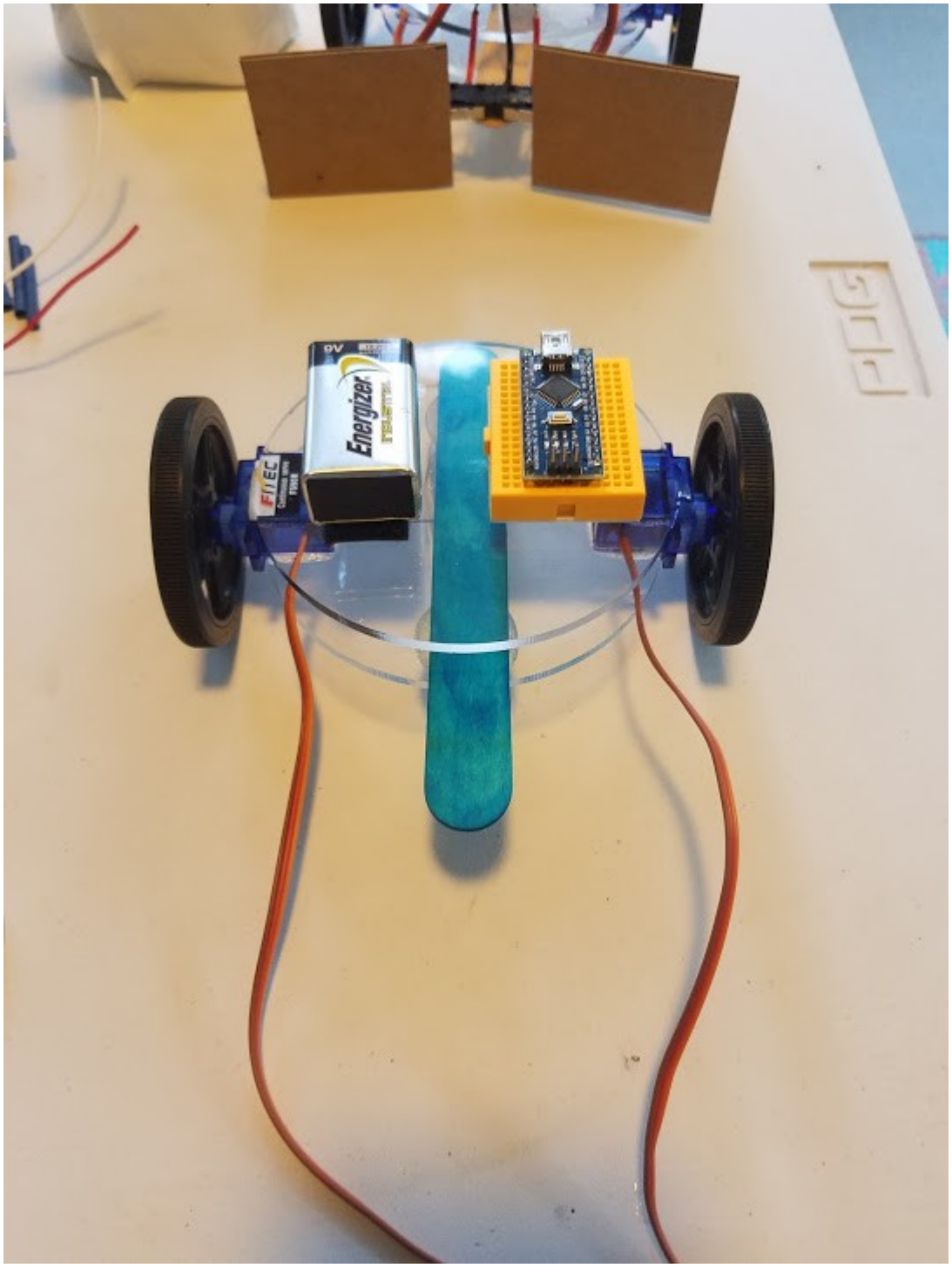
Carefully stick the solderless breadboard with Arduino Nano on the left side (robot's left) of the top acrylic round.



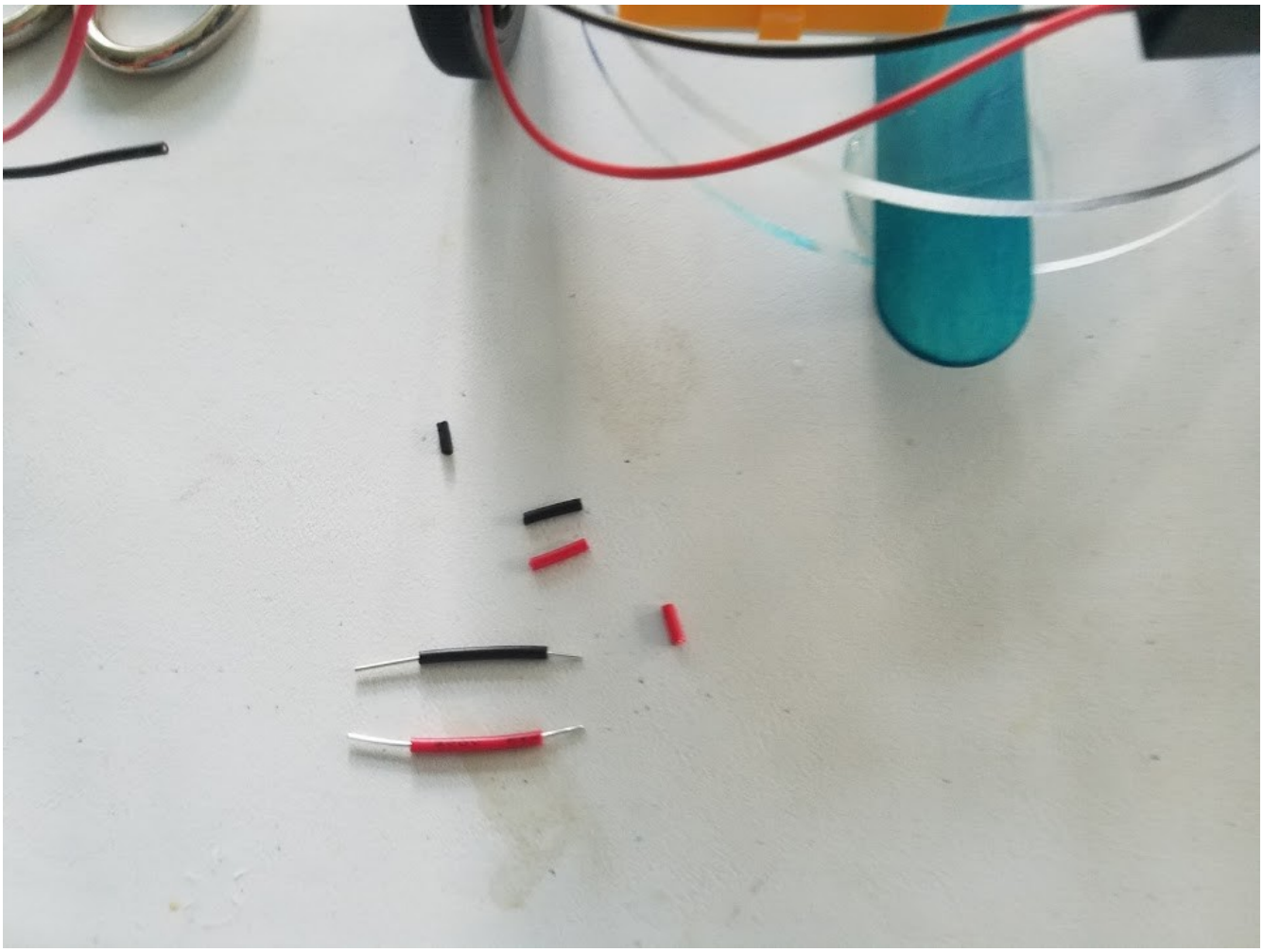
14. Cut ~2” of loop Velcro and place on right side of robot (robot’s right) top circle.



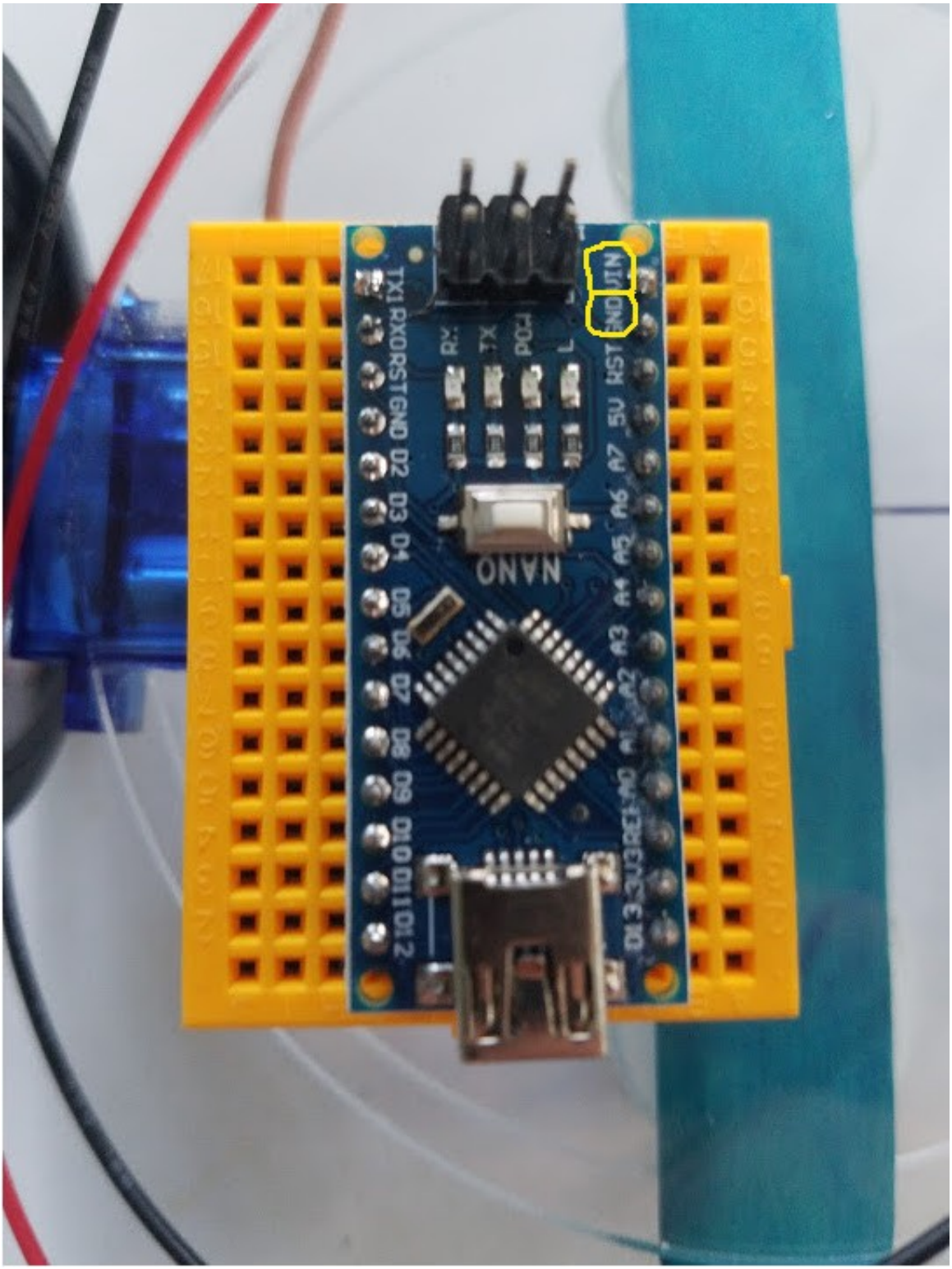
15. Cut ~1.5” of hook Velcro and put on 9 Volt battery.



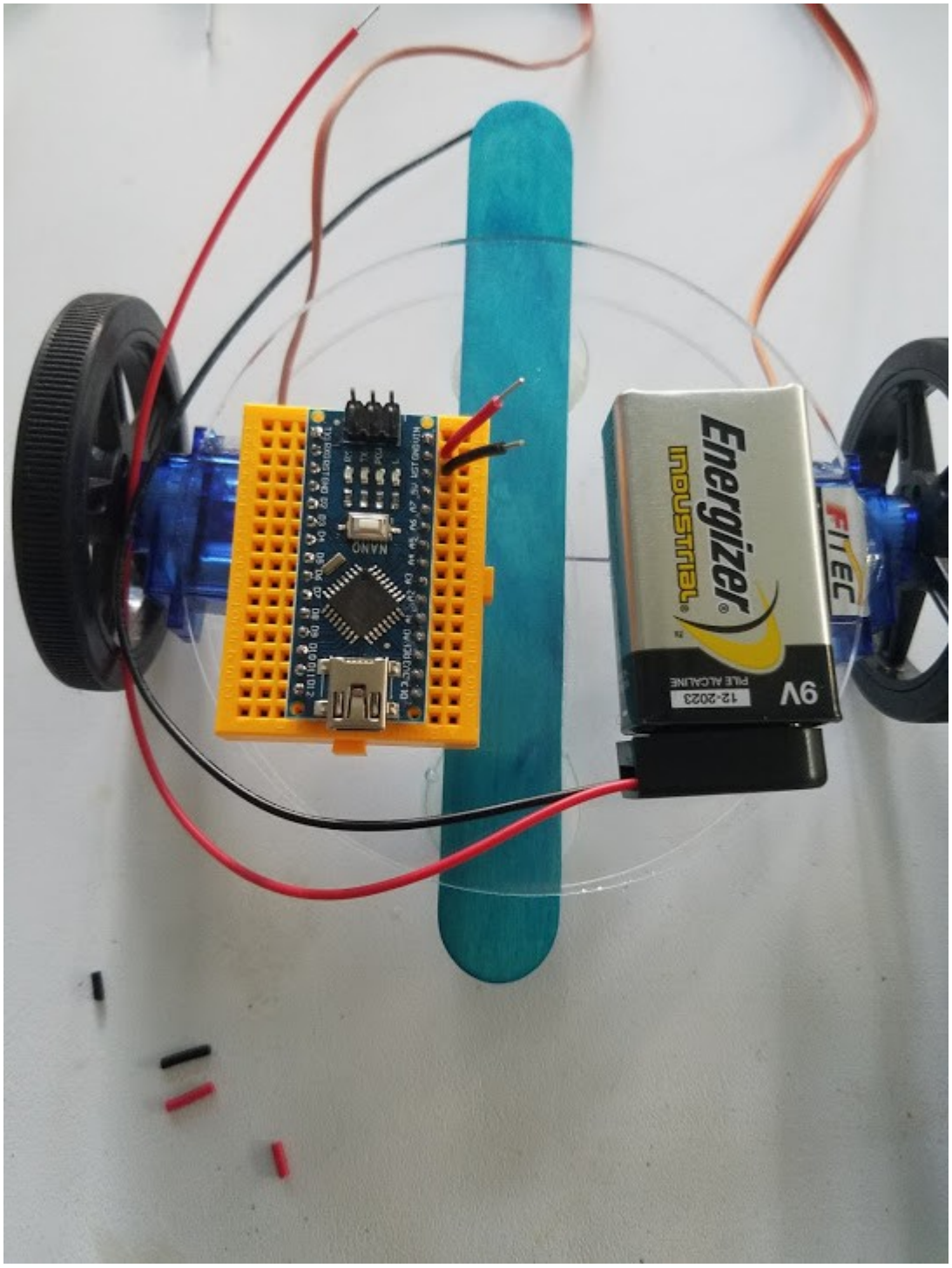
Put battery on loop Velcro on top acrylic round.



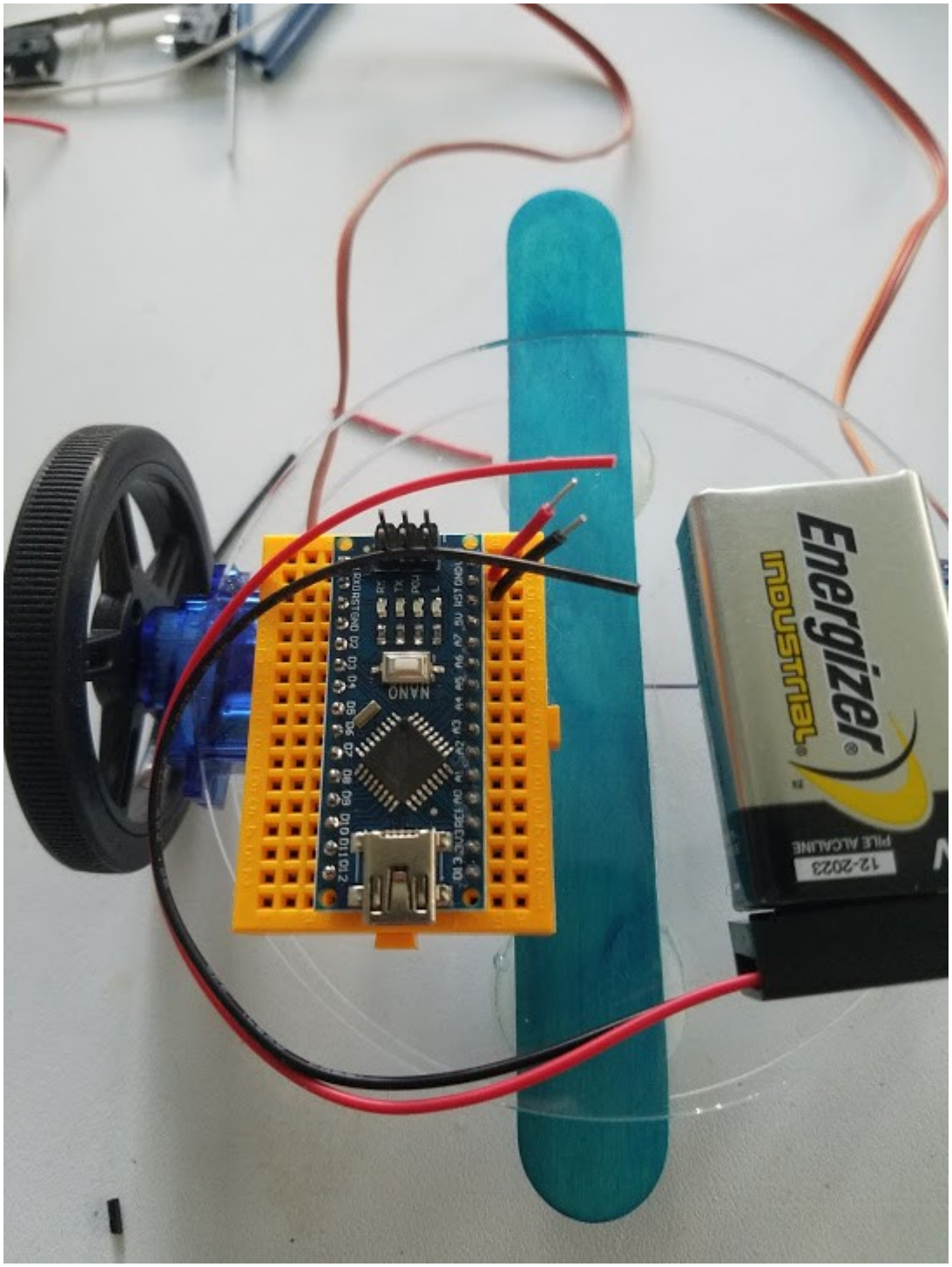
16. Cut two ~1” length of 22-gauge solid wire; one red, one black. Strip ~.25” insulation off one end and ~.125” off the other. They will be used as plugs for the battery-snap wires to plug into the solderless breadboards.



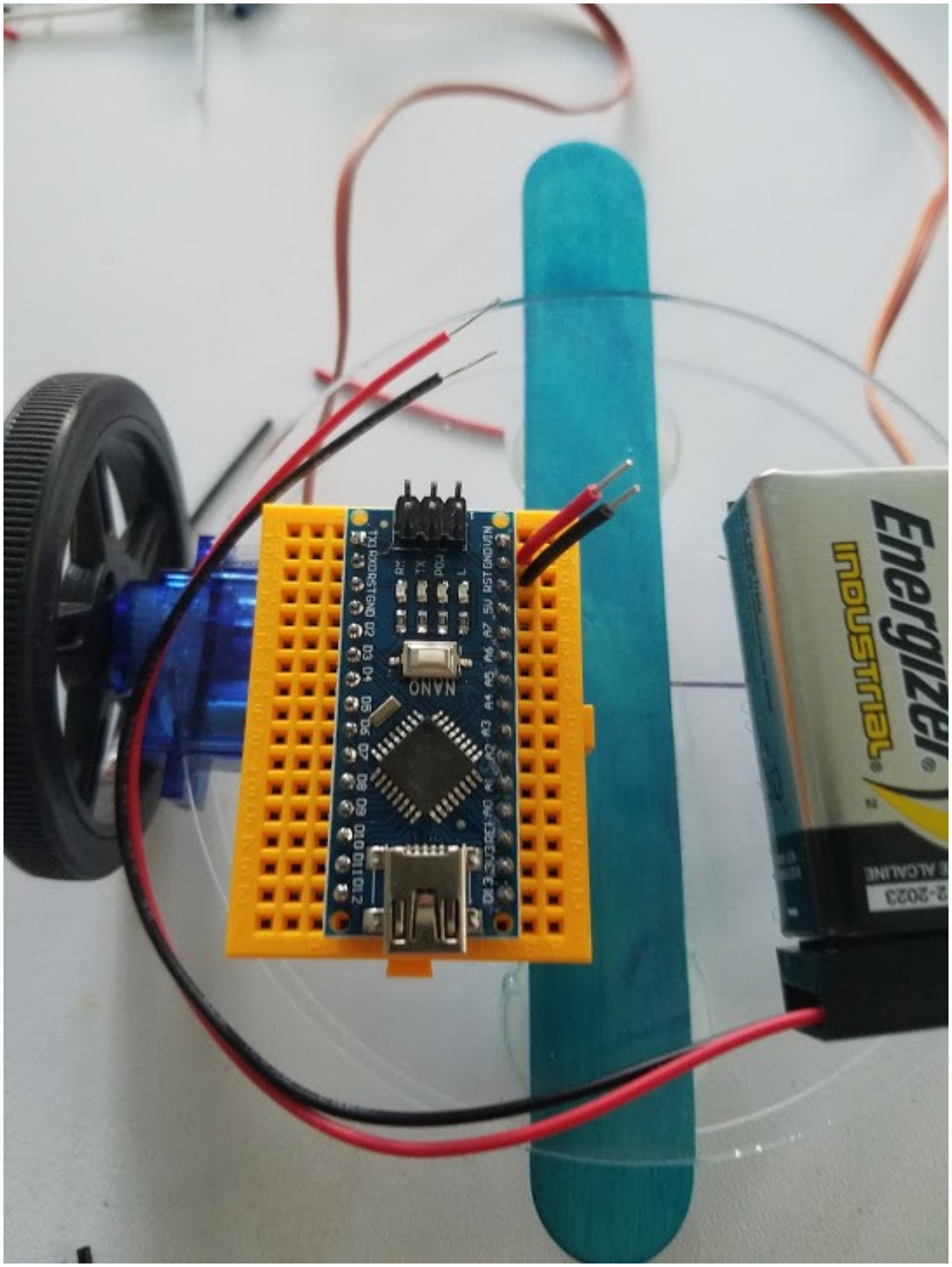
Locate the VIN and GND pins on the Arduino Nano and the solderless breadboard sockets alongside (marked with yellow circles above).



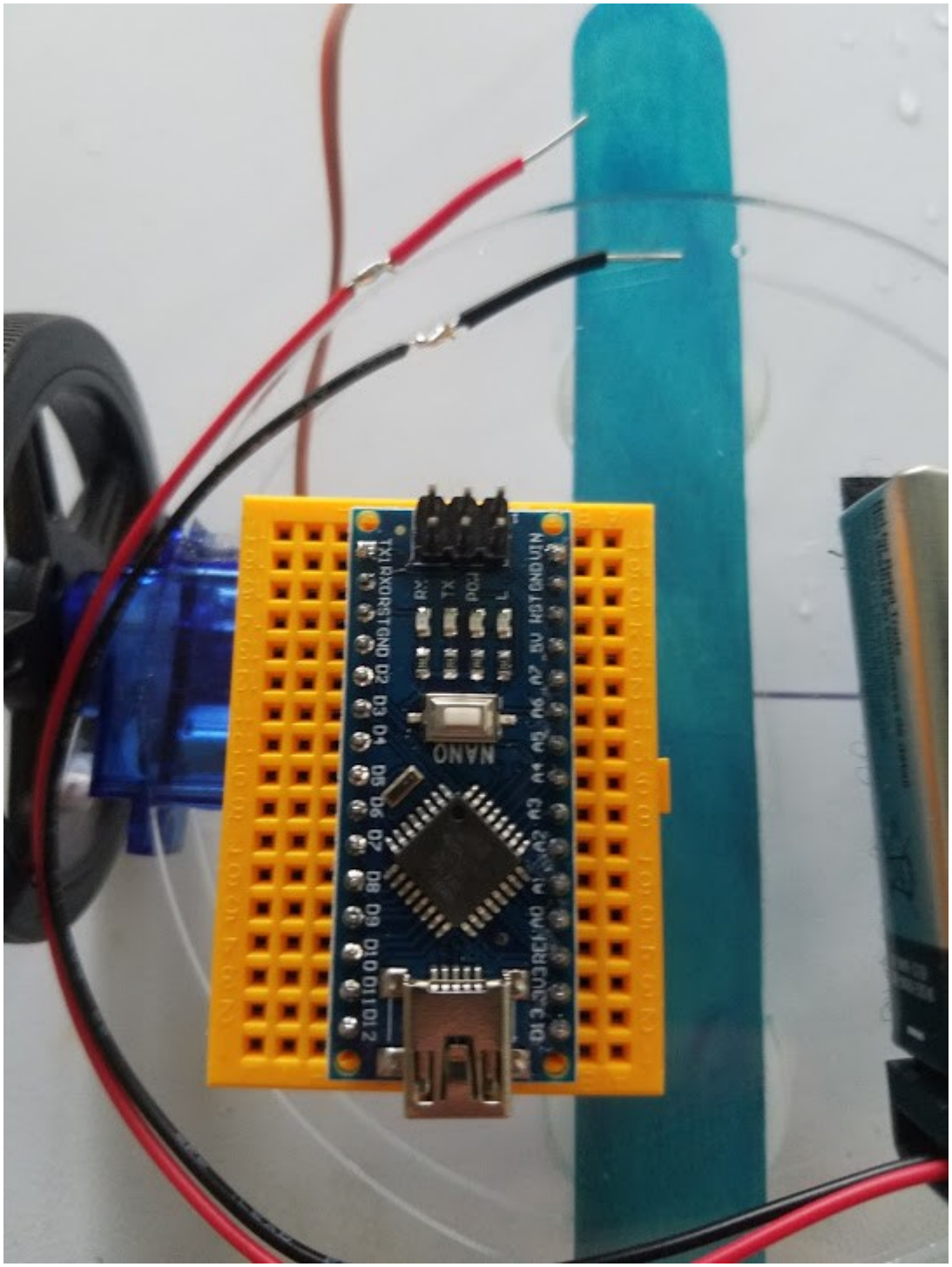
Plug the red 1/2” 22-gauge wire into the solderless breadboard socket alongside VIN (Voltage In) on the Arduino Nano and the black 1/2” 22-gauge wire into the solderless breadboard socket alongside GND (Ground) on the Arduino (see above).



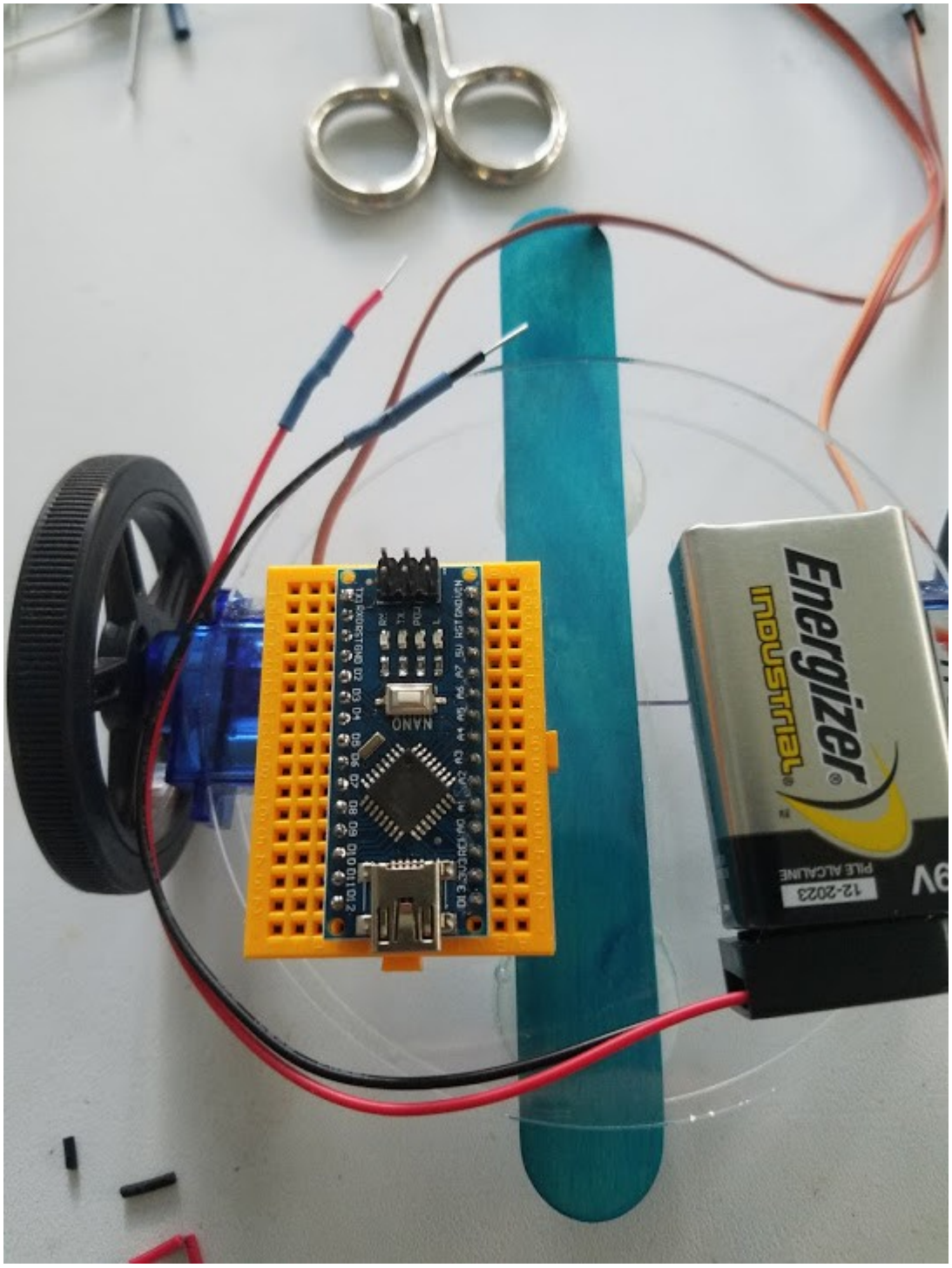
Cut the battery snap wires to length (see above). If it is connected to the 9 Volt battery DO NOT SHORT OR CONNECT THE WIRES TOGETHER!



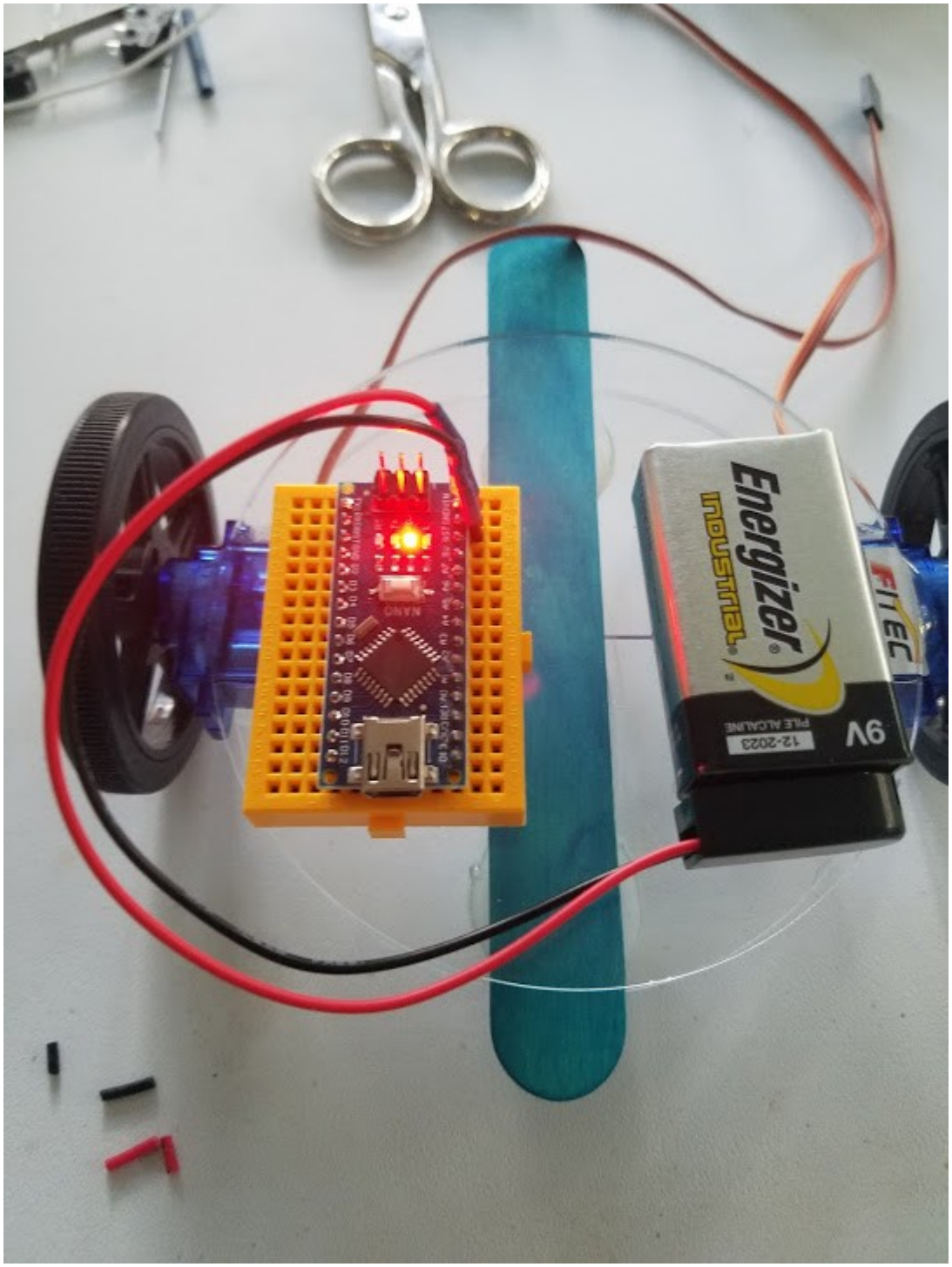
Strip the insulation off the red and black wires (~1/2”). Again if it is connected to the battery DO NOT SHORT OR CONNECT THE WIRES!



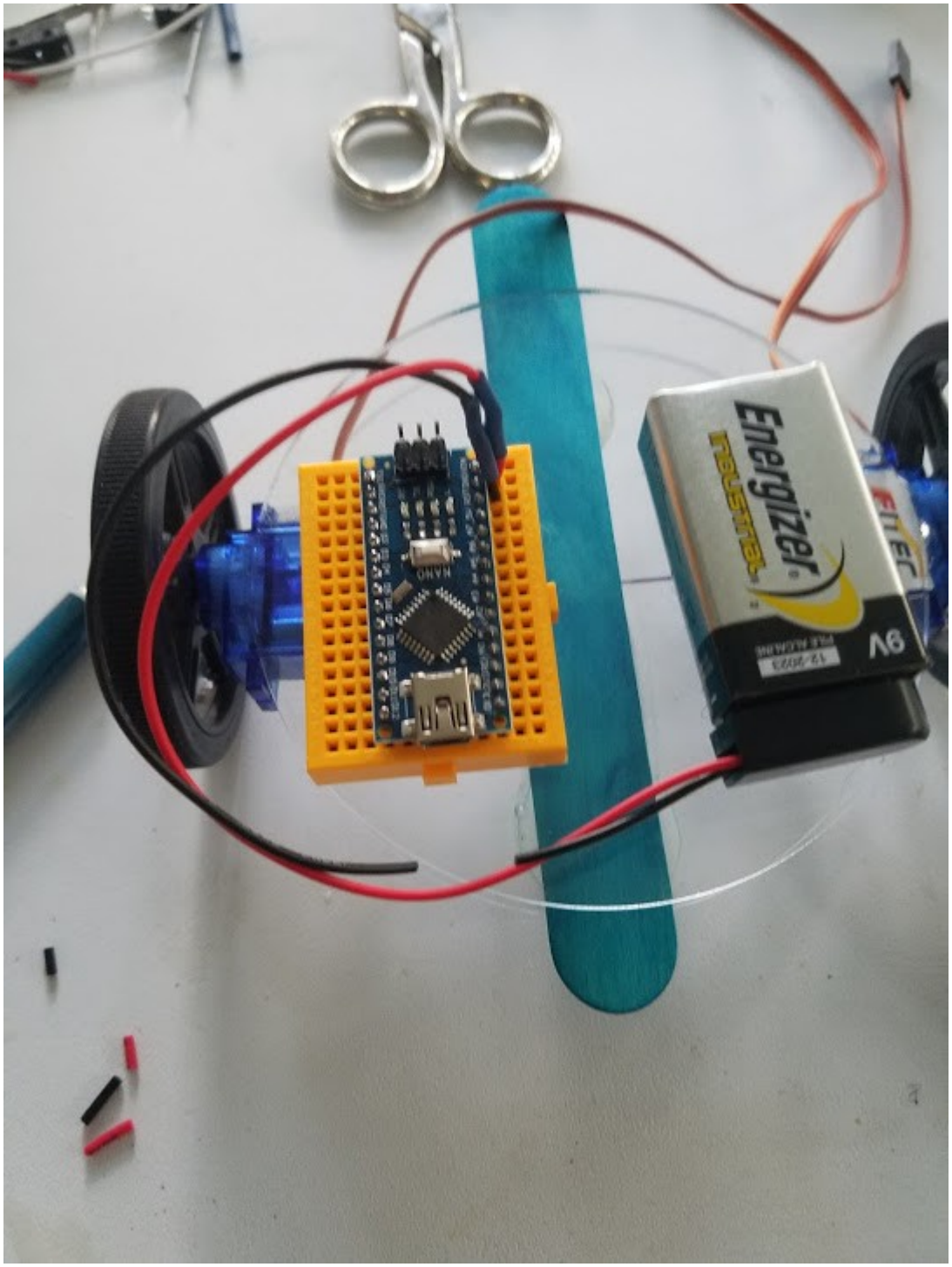
Wrap the red battery-snap wire around the 1/4” end of the red solderless breadboard plug and solder. BE CAREFUL WHEN SOLDERING! ALWAYS WEAR SAFETY GLASSES! Wrap the black battery-snap wire around the 1/4” end of the black solderless breadboard plug and solder.



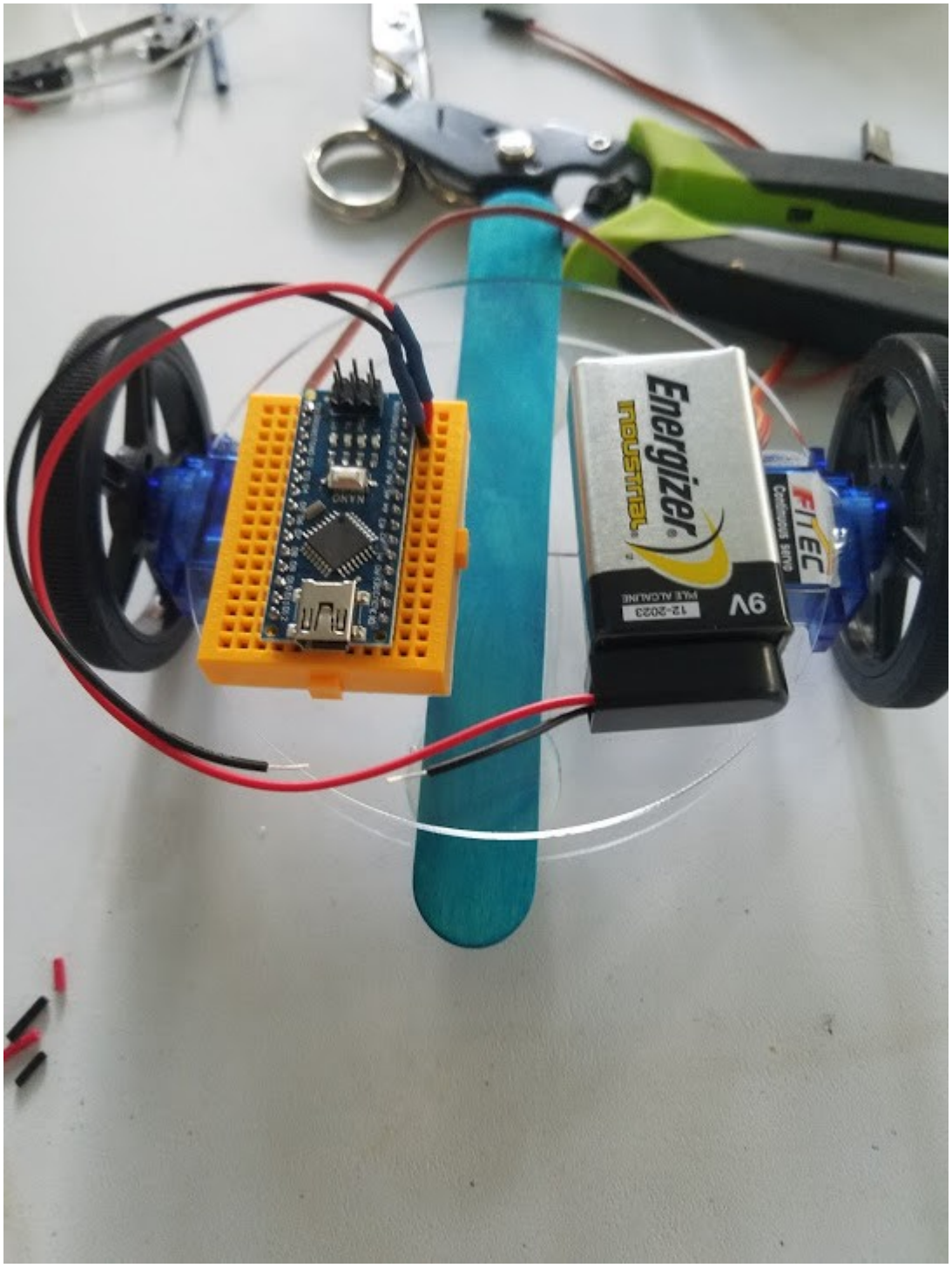
Put heat-shrink tubing or electrical tape on or over solder joints to prevent short circuits.



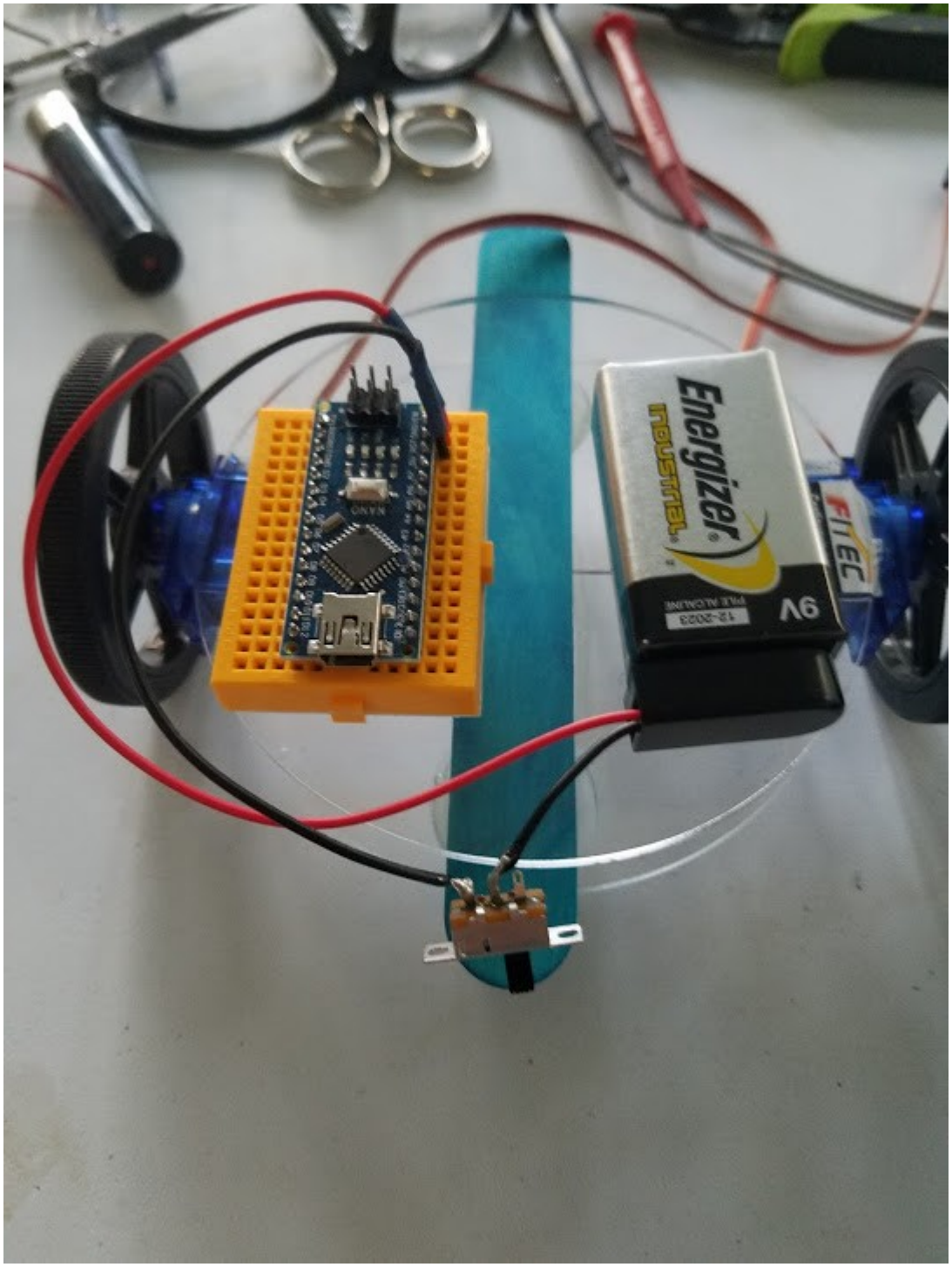
Plug black and red wire into the VIN and GND sockets on the solderless breadboard. The Arduino should light up!



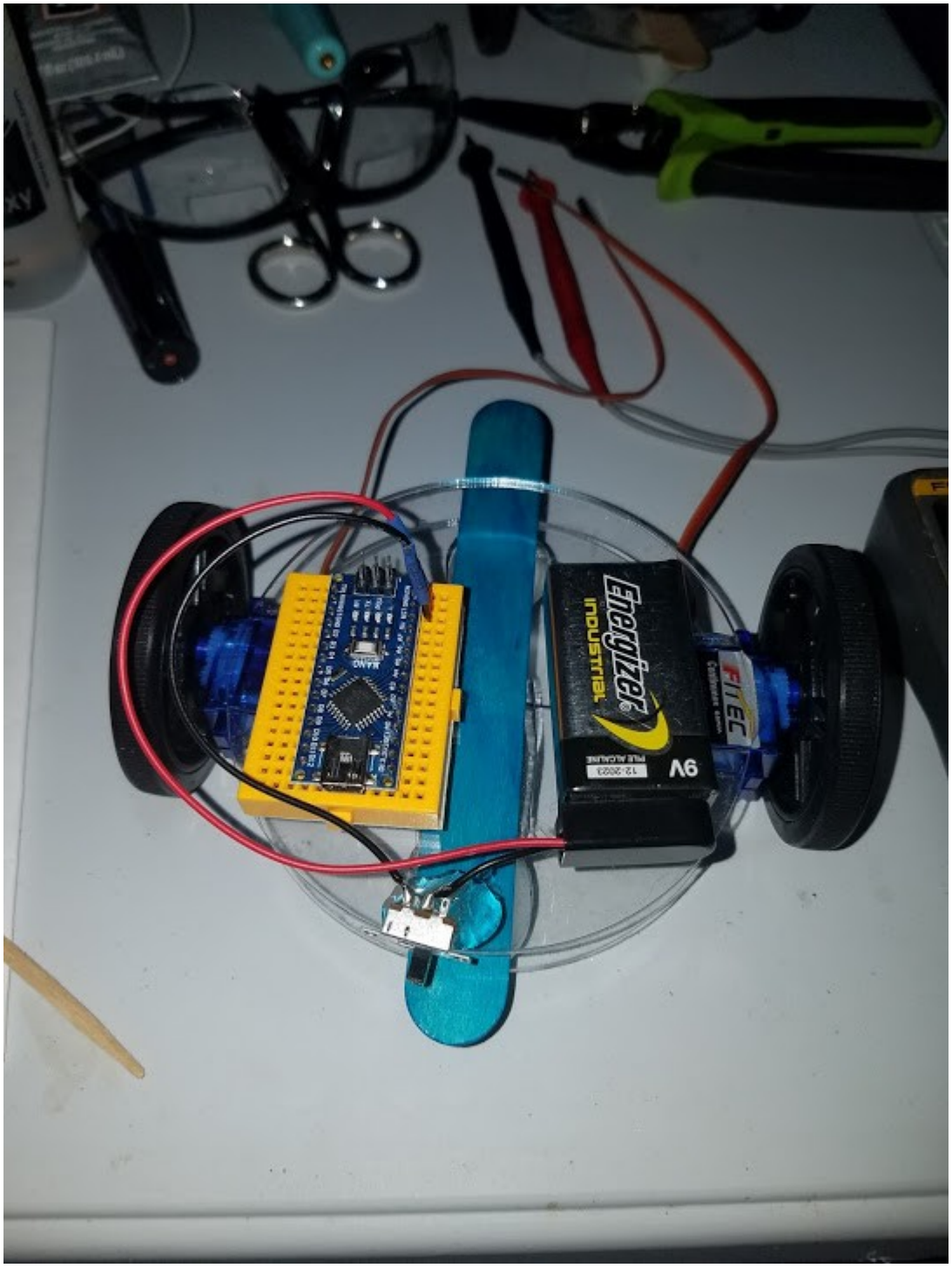
17. Next we're going to install the On/Off switch. Cut the black wire near the tail of the robot turtle. If the battery is connected the light on the Arduino should go "Off".



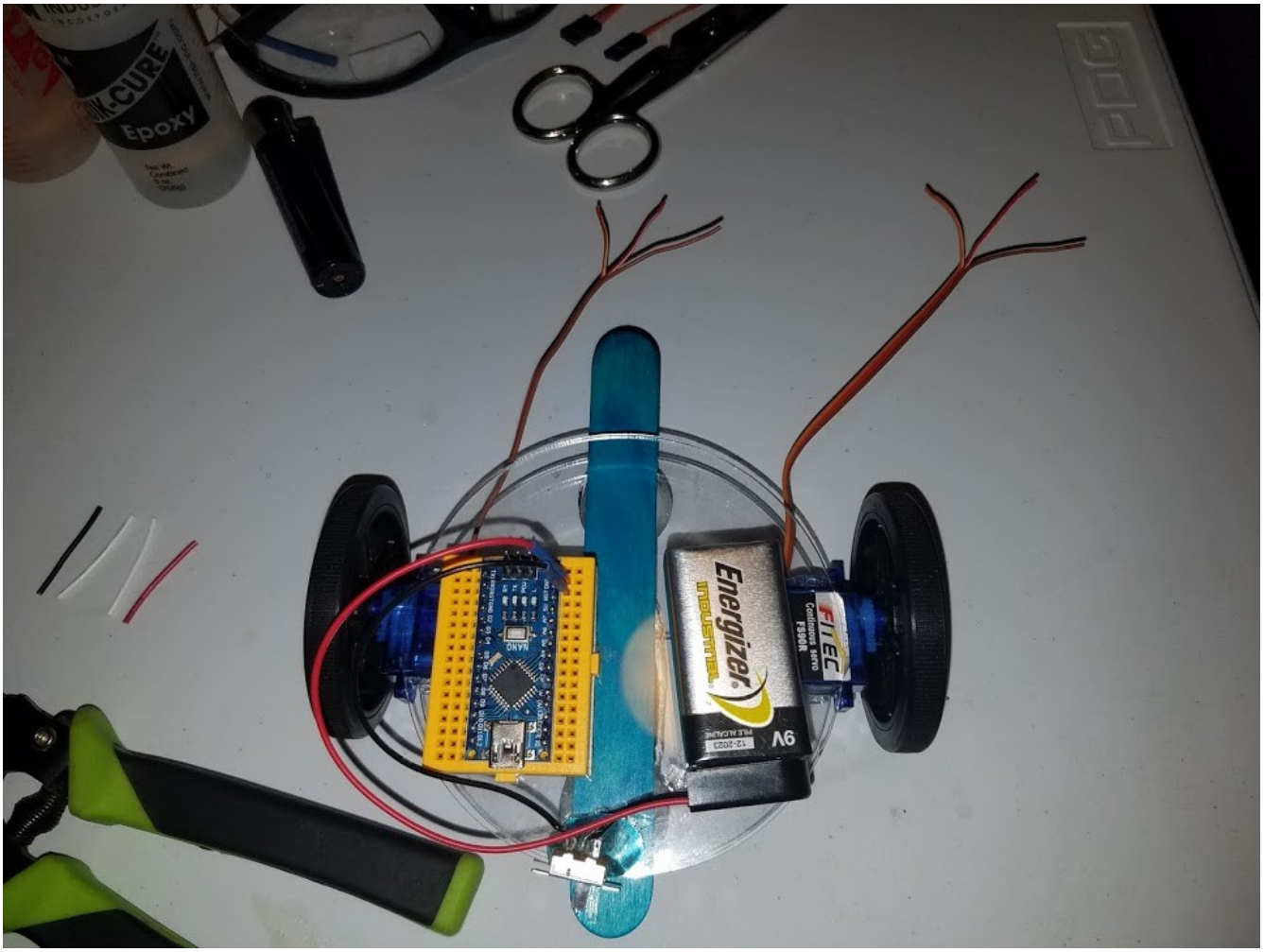
Strip ~1/4 off each end of the black wire.



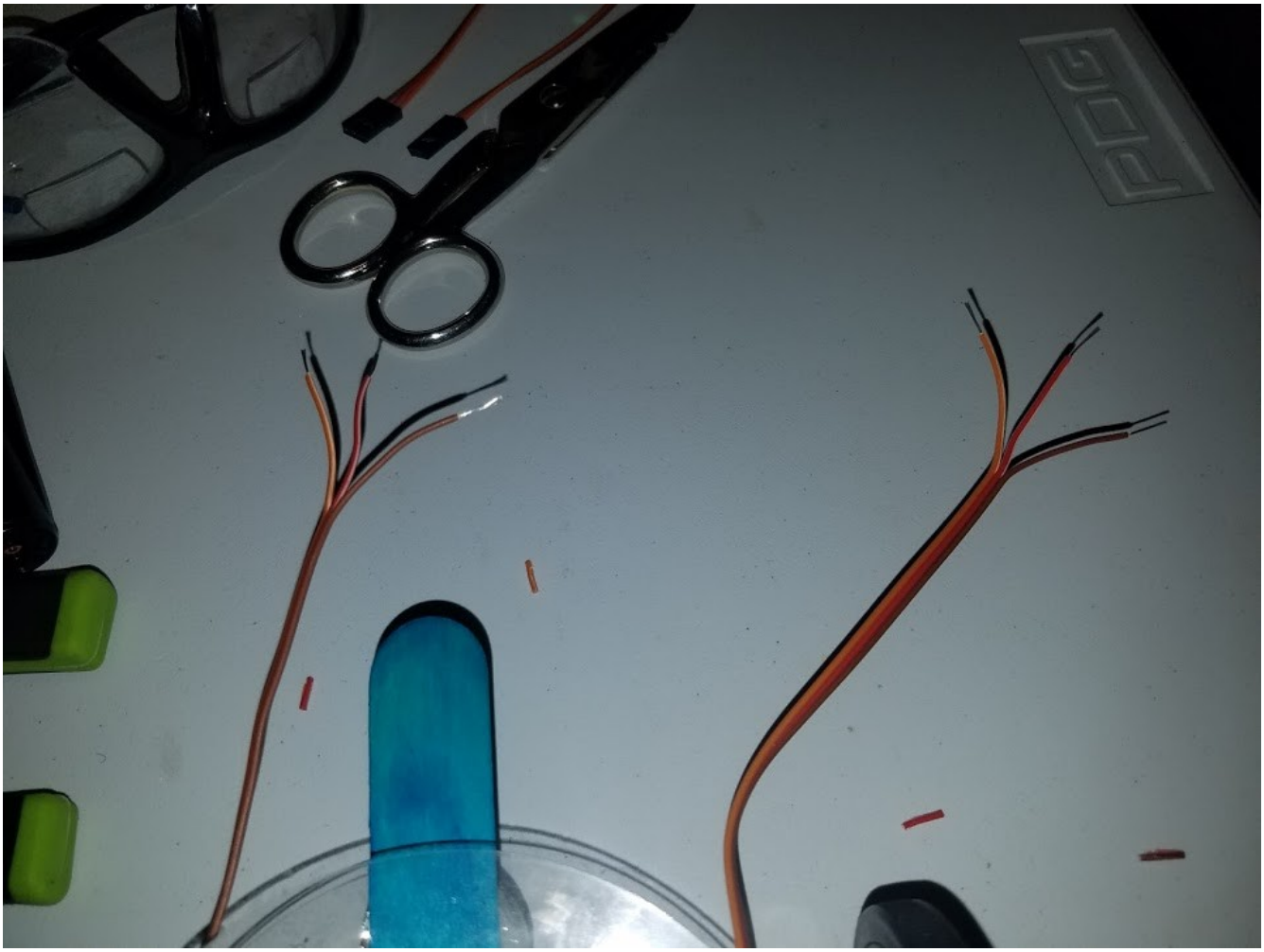
Solder each wire to terminal of normally open switch. The Arduino should turn off and on (LED) when you open and close the switch.



Glue with 5-minute epoxy the switch in place.

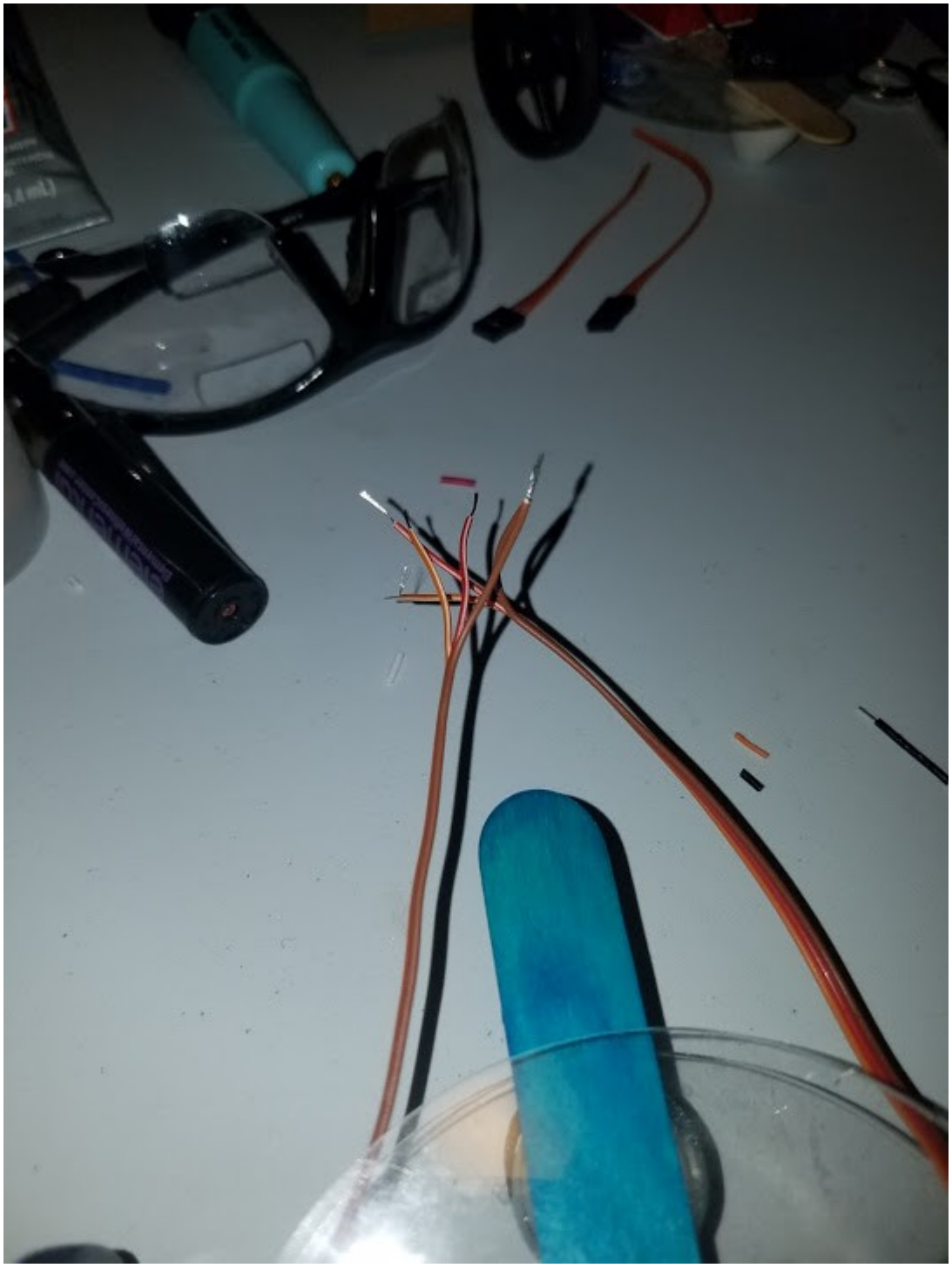


18. The next step is to cut the servo wires to length. Each servo has 3 wires: Ground (black or brown), power (red) and signal (white, yellow or orange). Cut and peel the wires apart so you have 3 distinct wires (instead of a 3 conductor ribbon cable).

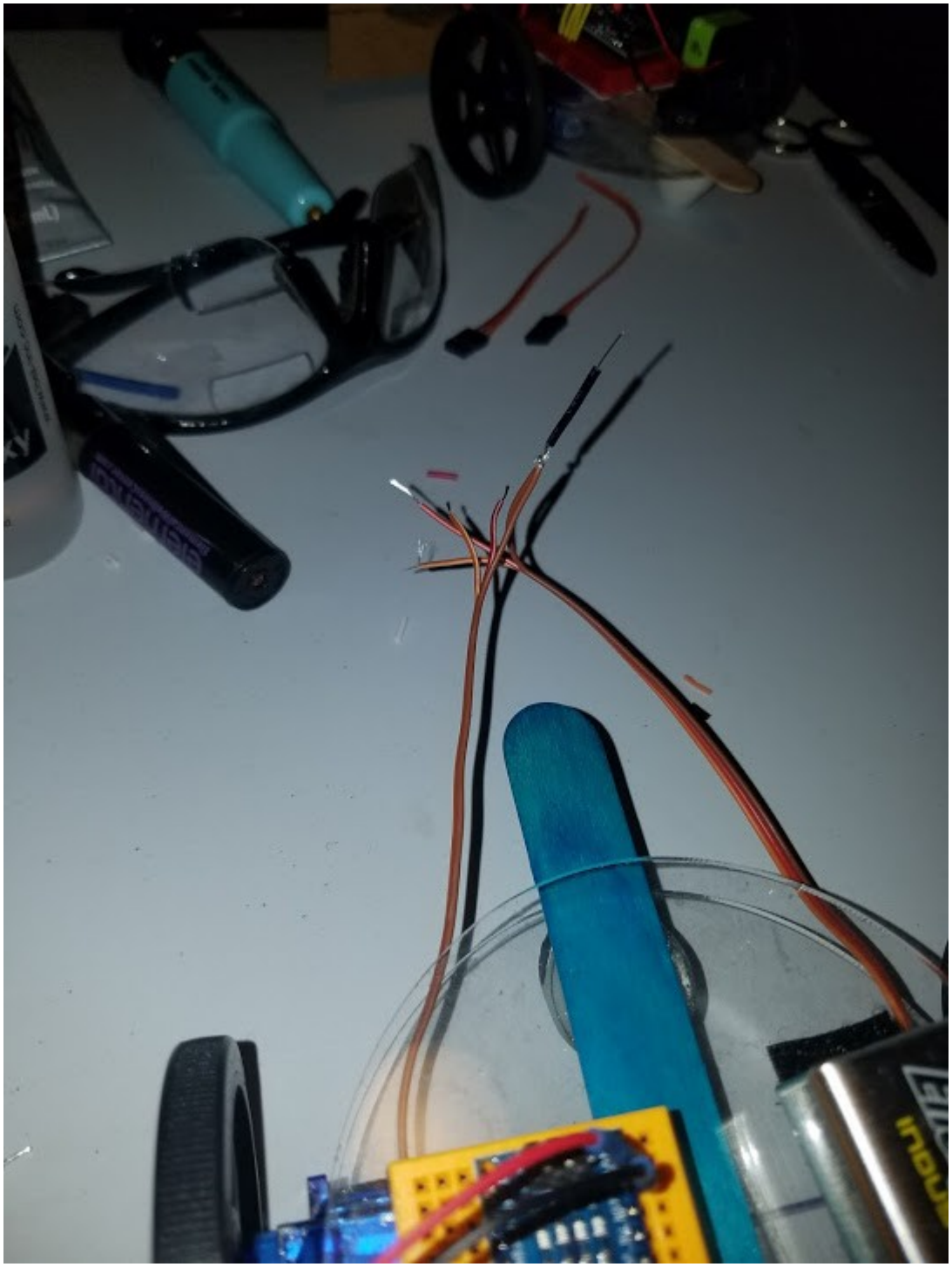


Strip ~1/4” insulation off each individual wire.

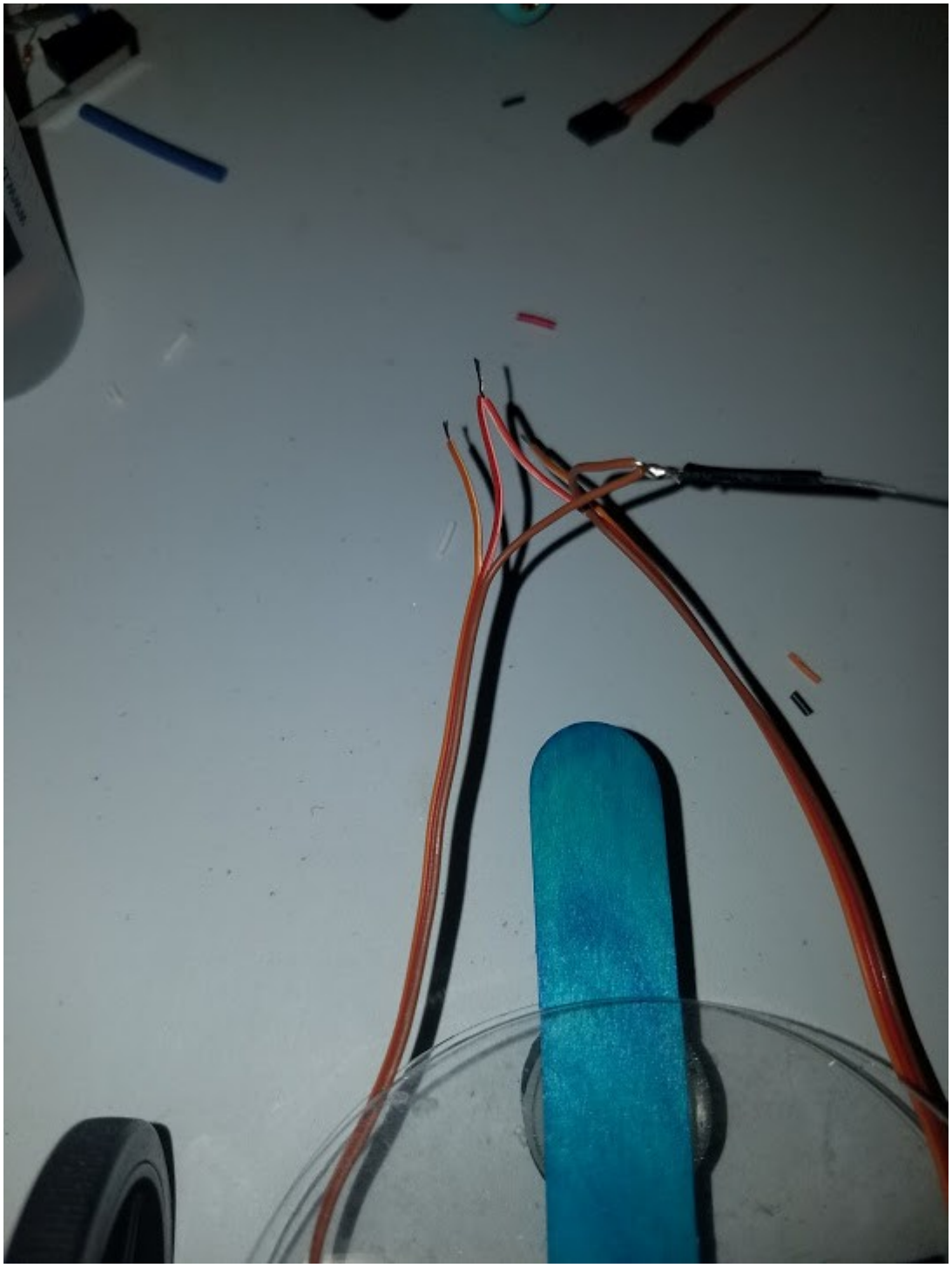
Make 4 solderless breadboard plugs like you did for the battery snap harness wires. One black, one red and two a third color (white, yellow or orange).



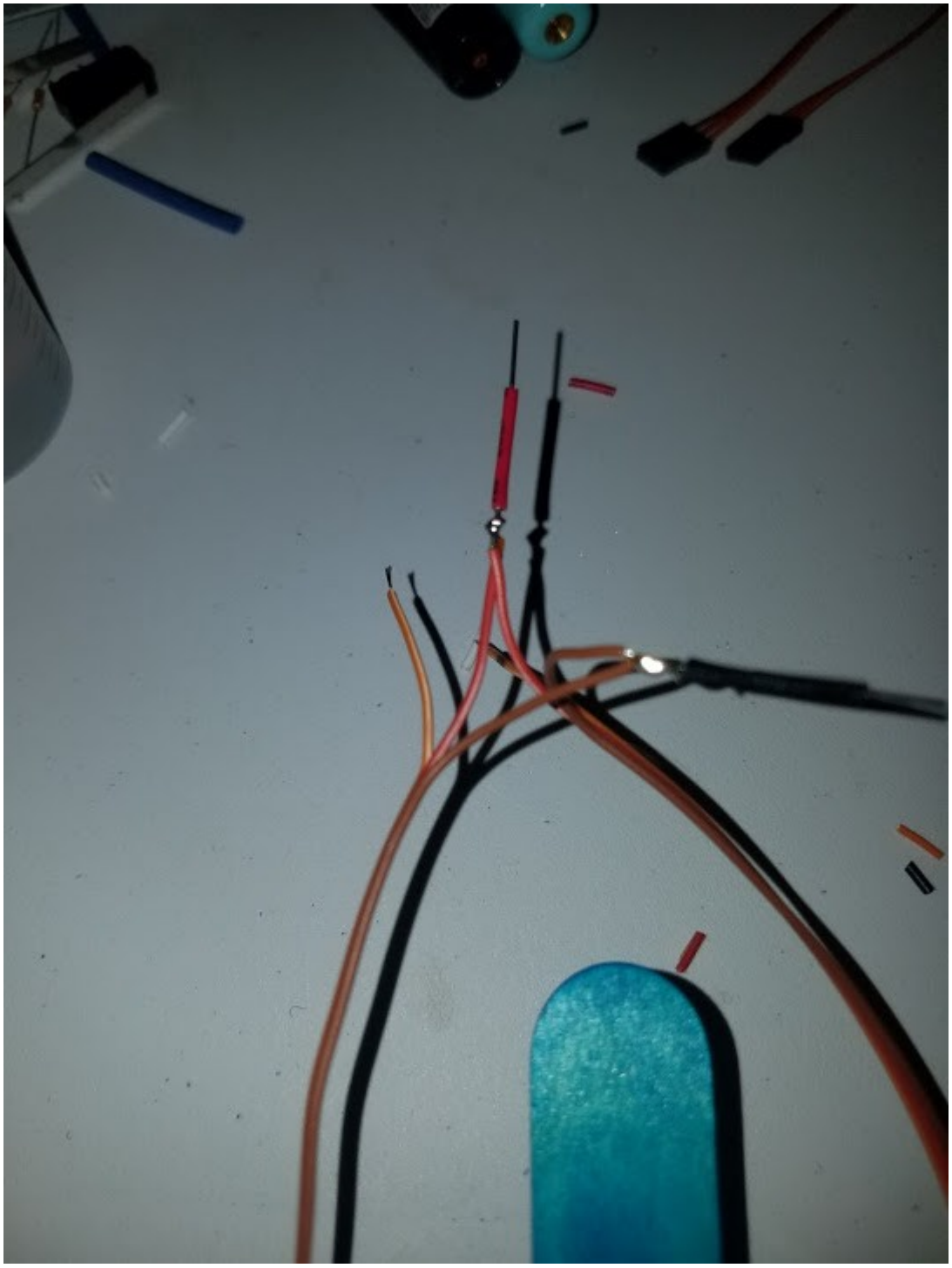
Common the grounds (twist the two brown wires together).



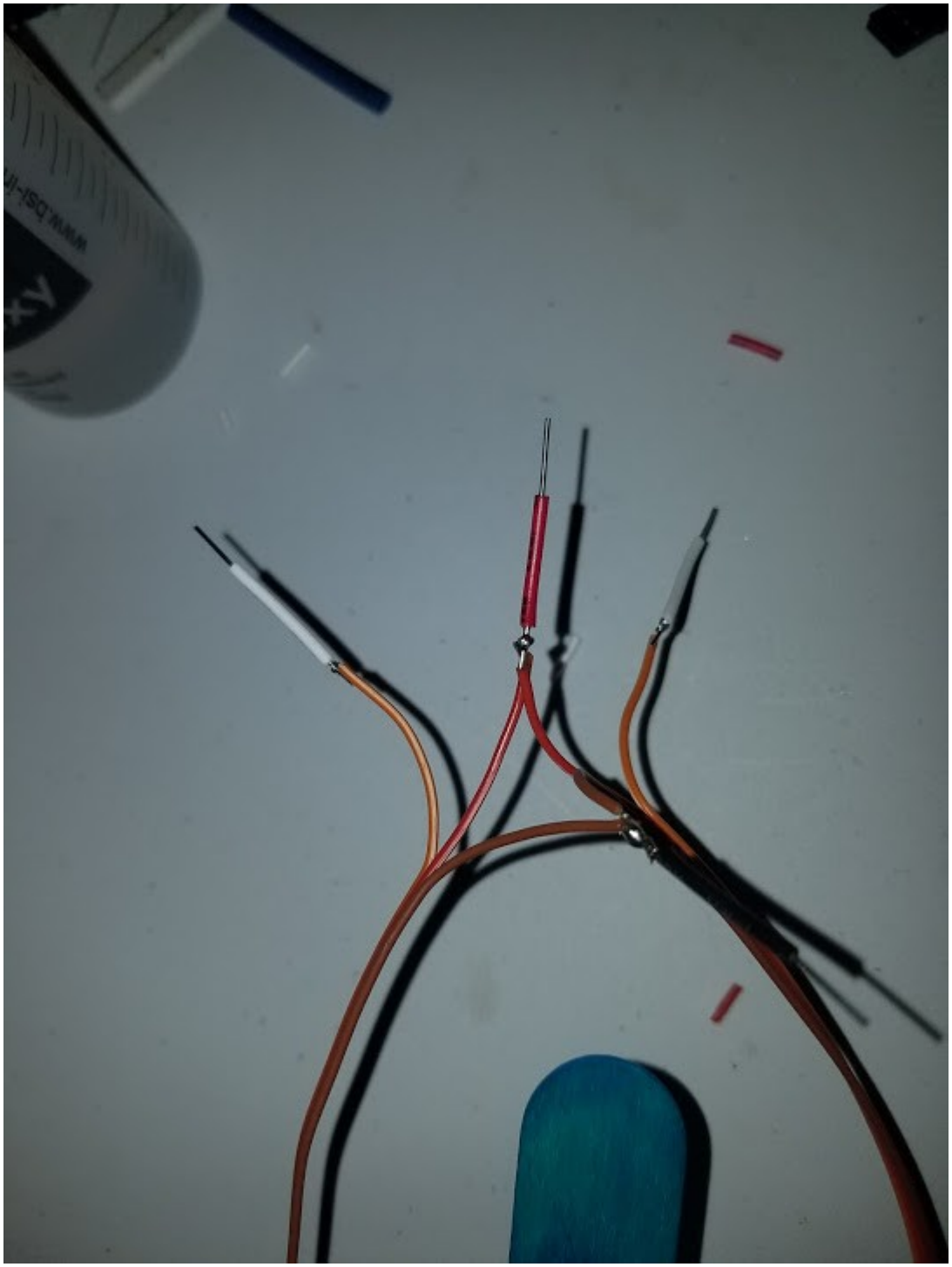
Then wrap the wires around the 1/4” 22-gauge black solderless breadboard plug. Solder.



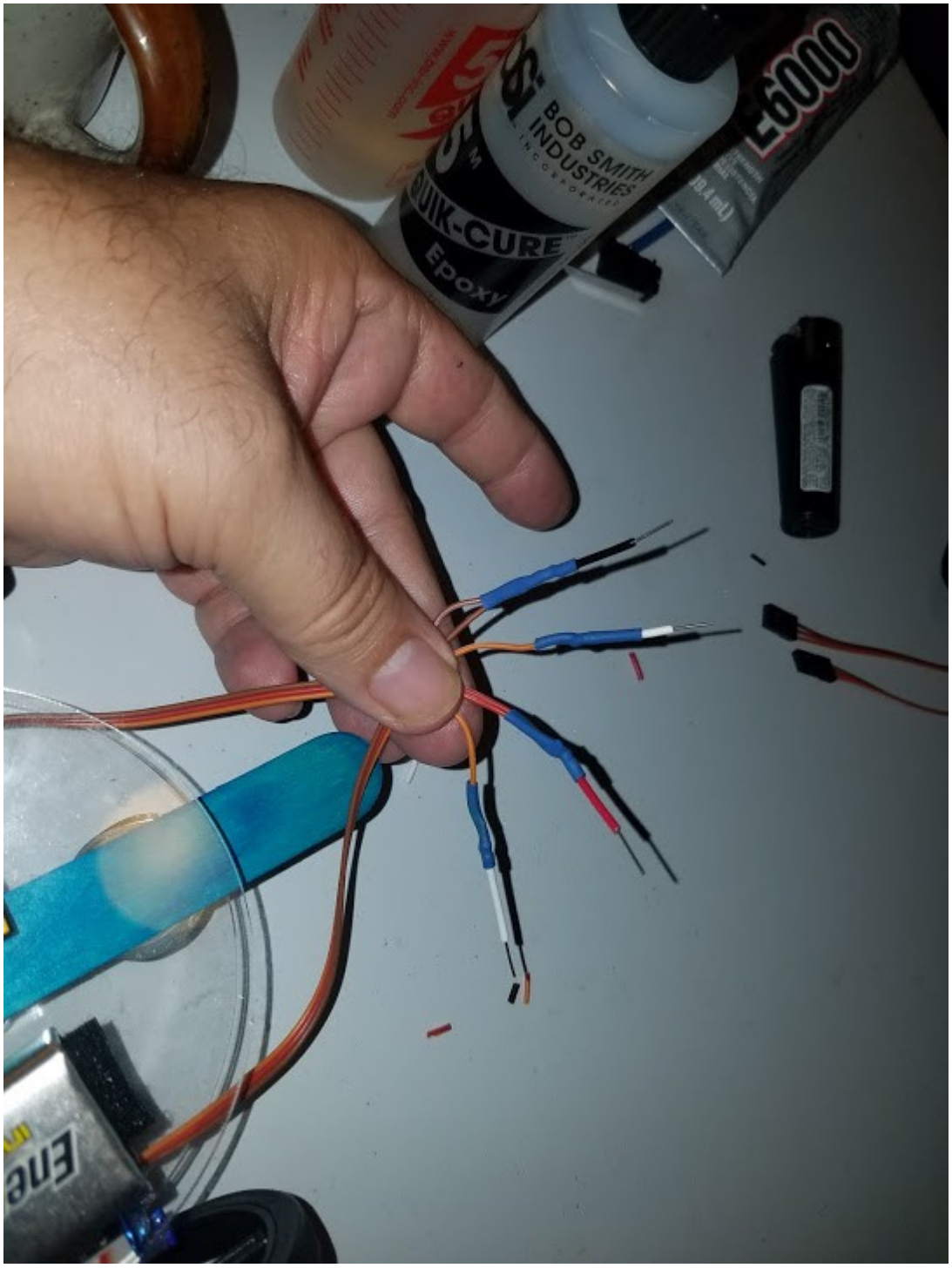
Wrap the two power (red) wires together.



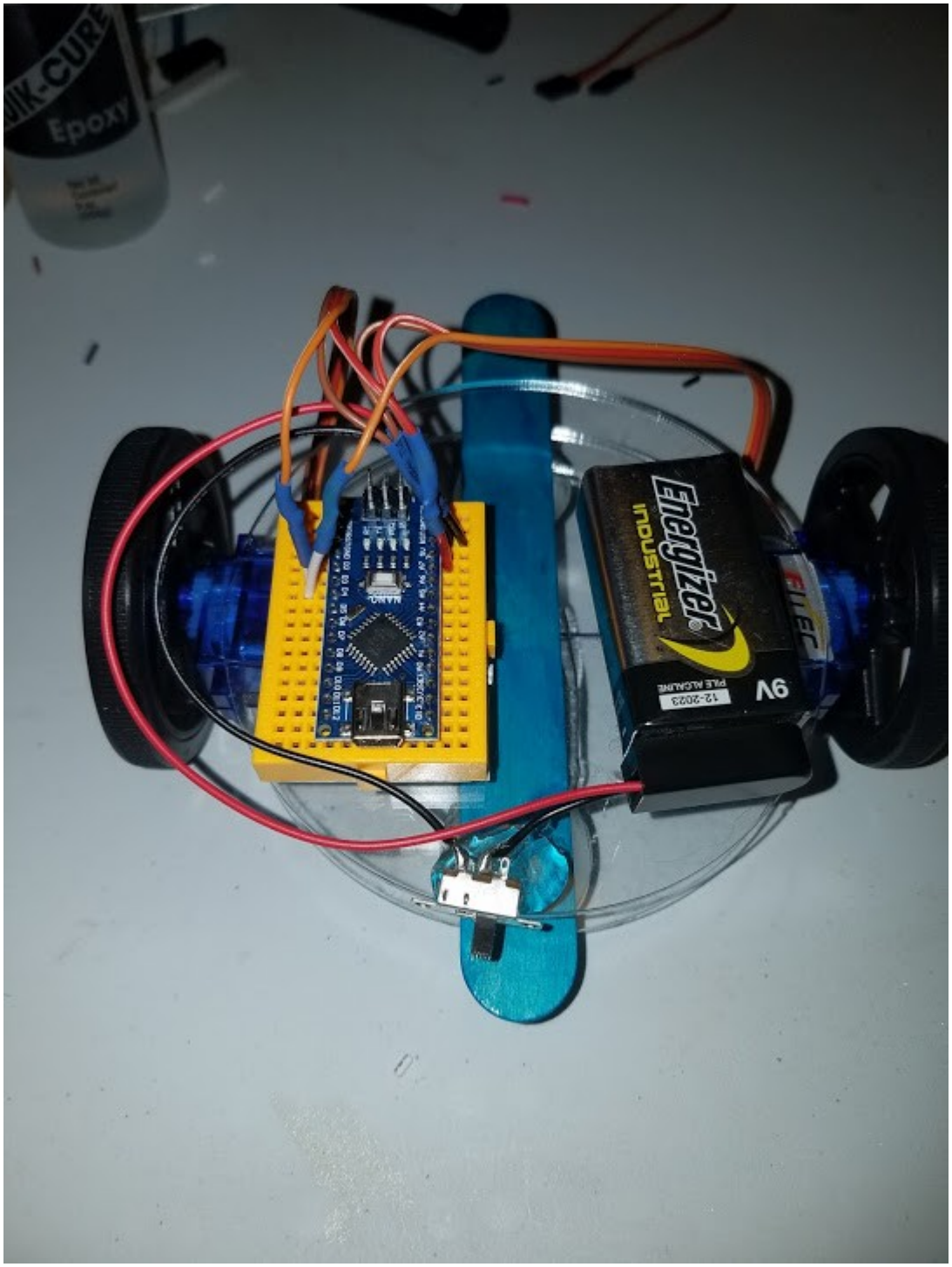
Then wrap the wires around the 1/4” 22-gauge red solderless breadboard plug and solder.



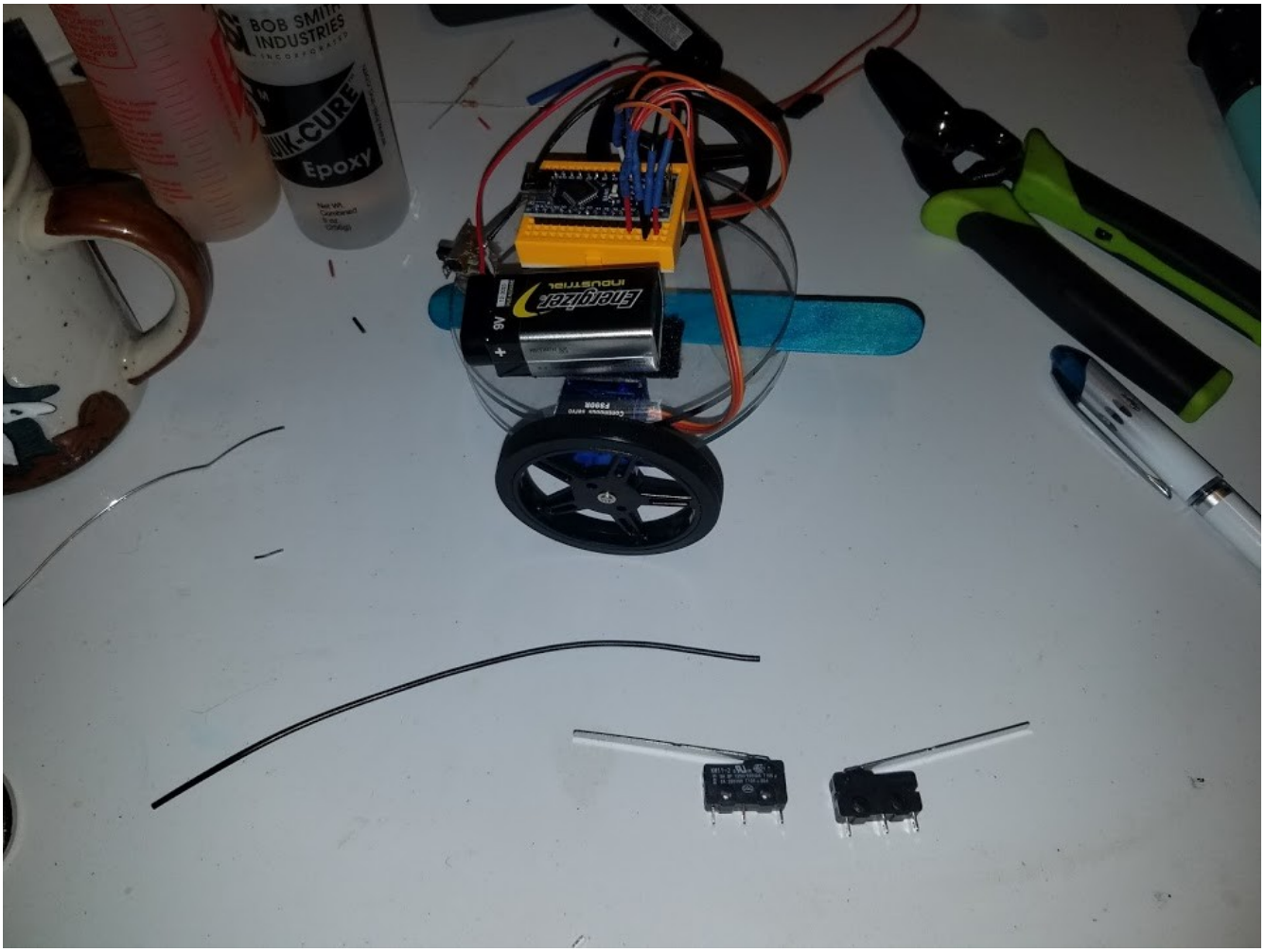
Then wrap the wires around the 1/4” 22-gauge signal plug (white, yellow or orange) and solder.



Cover the solder joints with heat-shrink tubing or electrical tape.

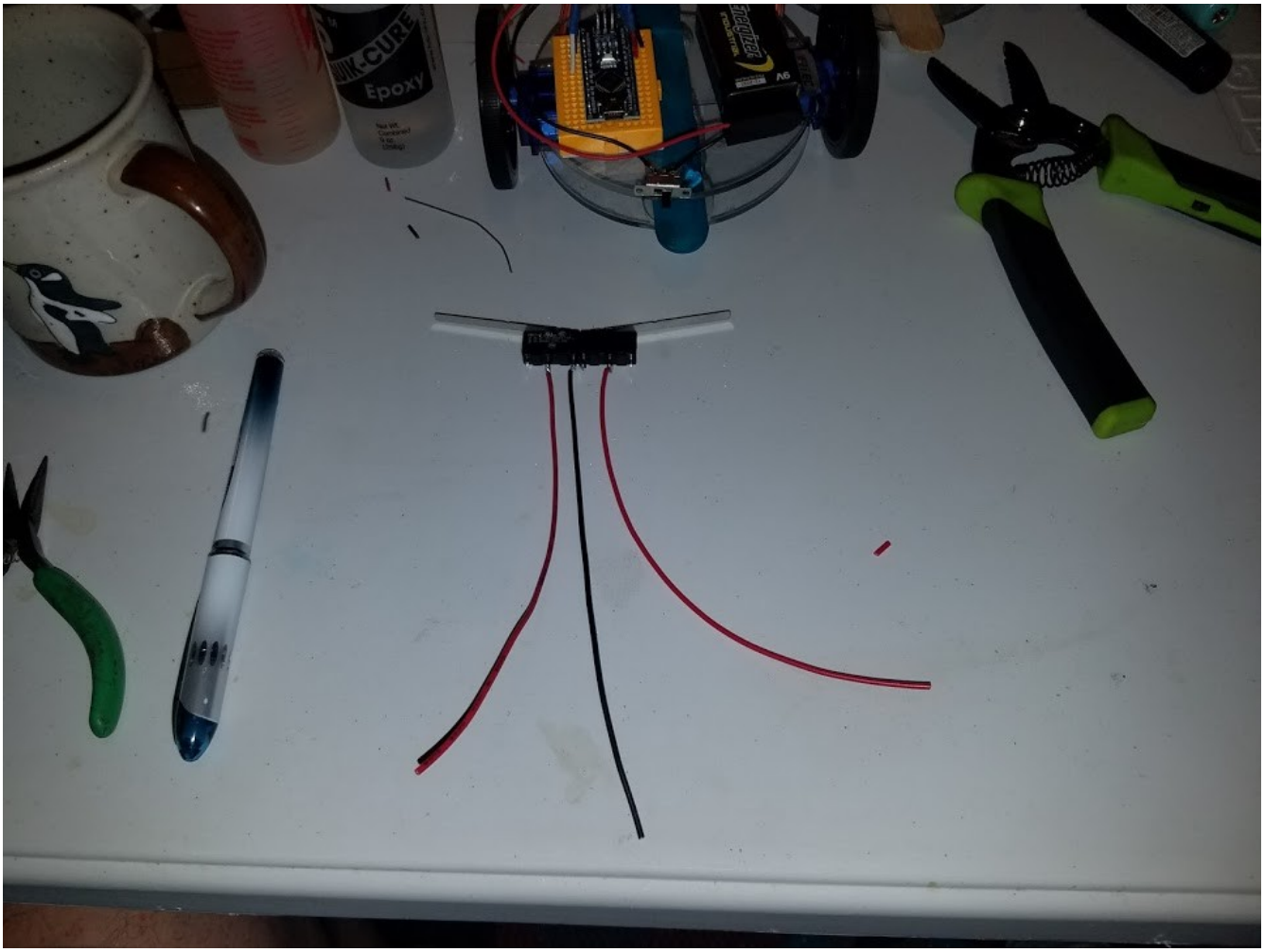


19. Plug the black 22-gauge solid wire into GND. Plug the red 22-gauge solid wire into 5V. Plug the left signal wire 22-gauge solid wire into D2 and plug the right signal 22-gauge wire into D3.

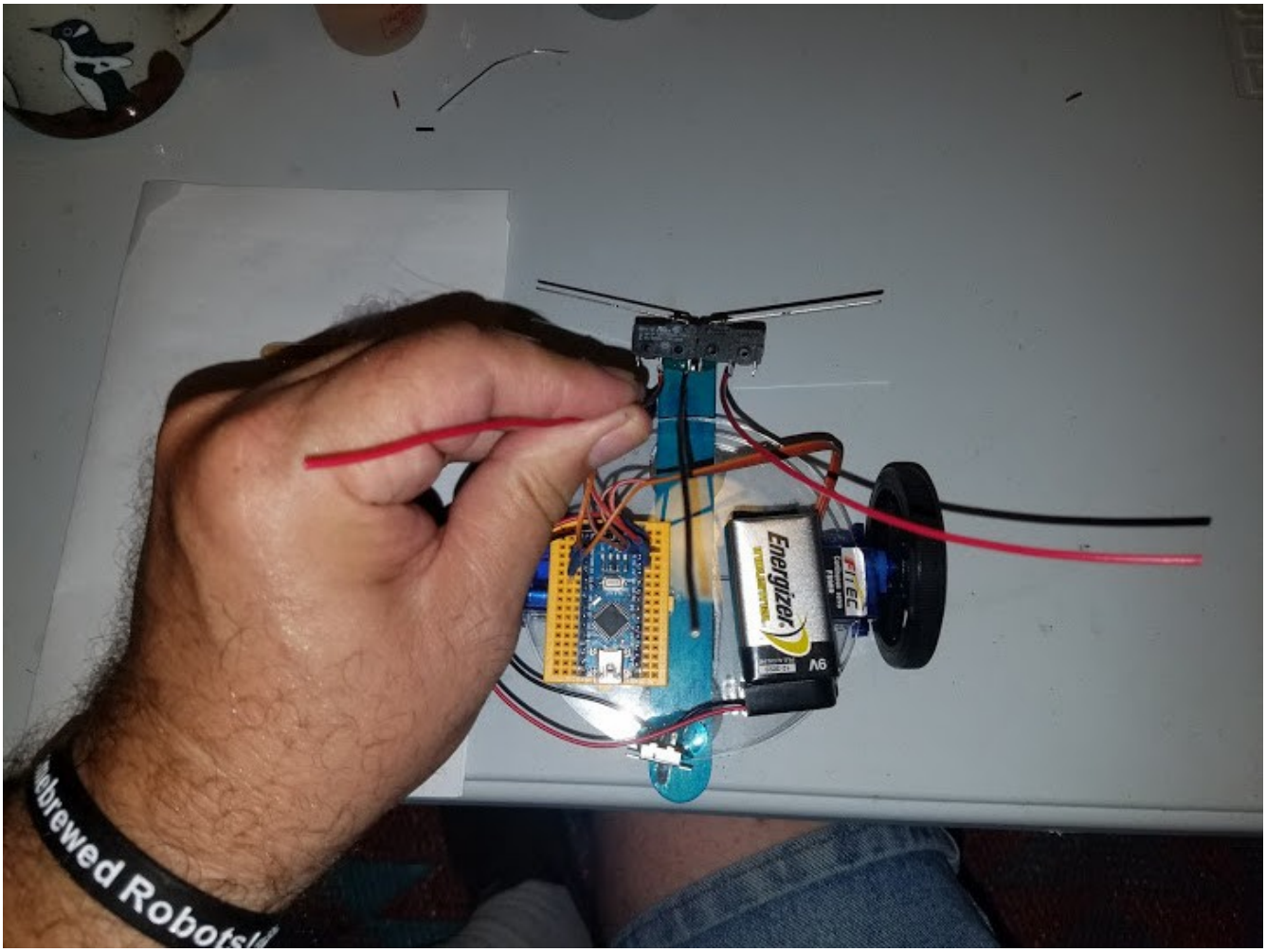


20. Now let's connect the two snap-action switches. Ohm or tone out the your switch to verify which terminals are normally open.

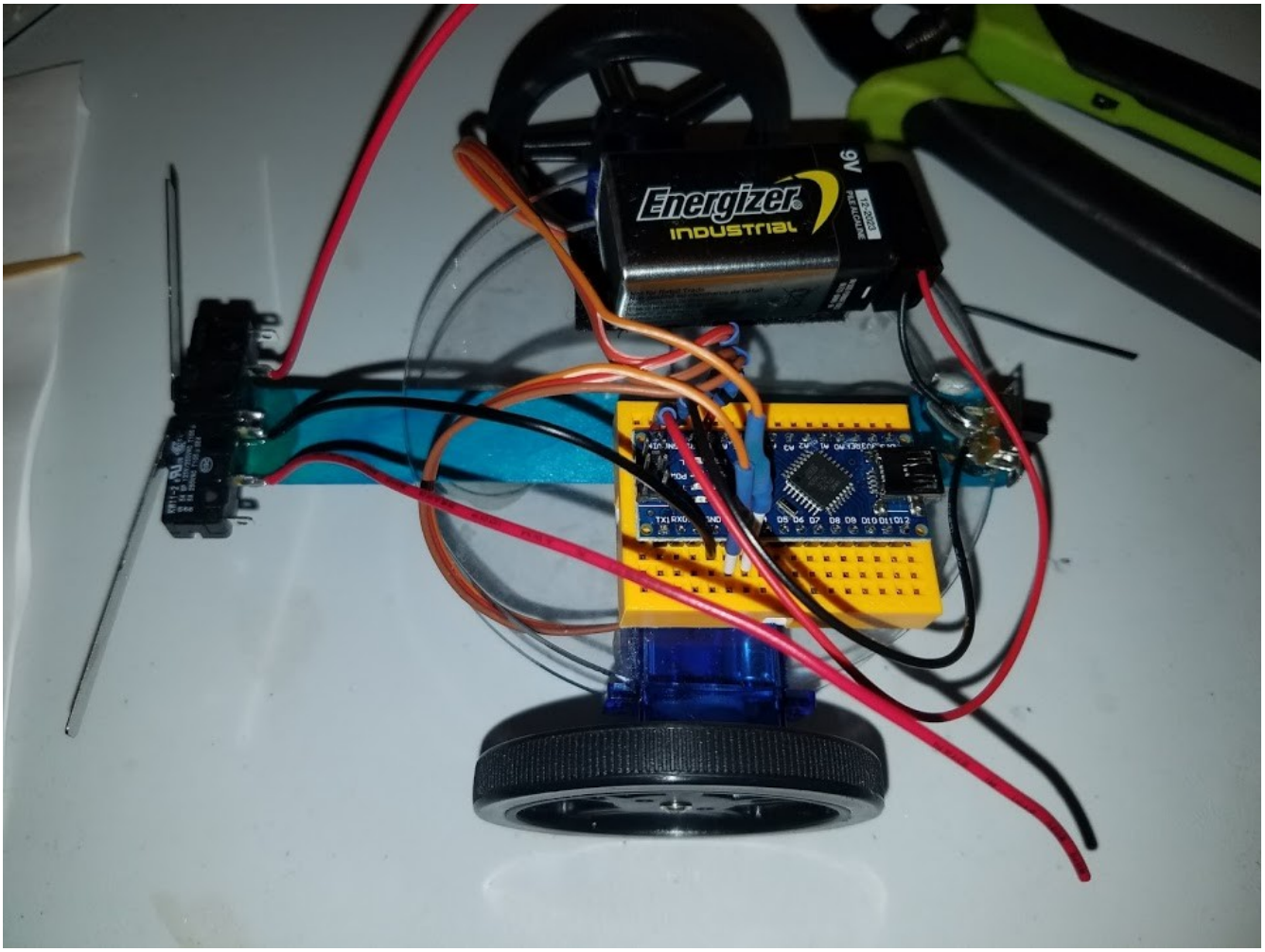
Connect an ~6” piece of 22-gauge solid black wire to the common terminal of each snap-action switch.



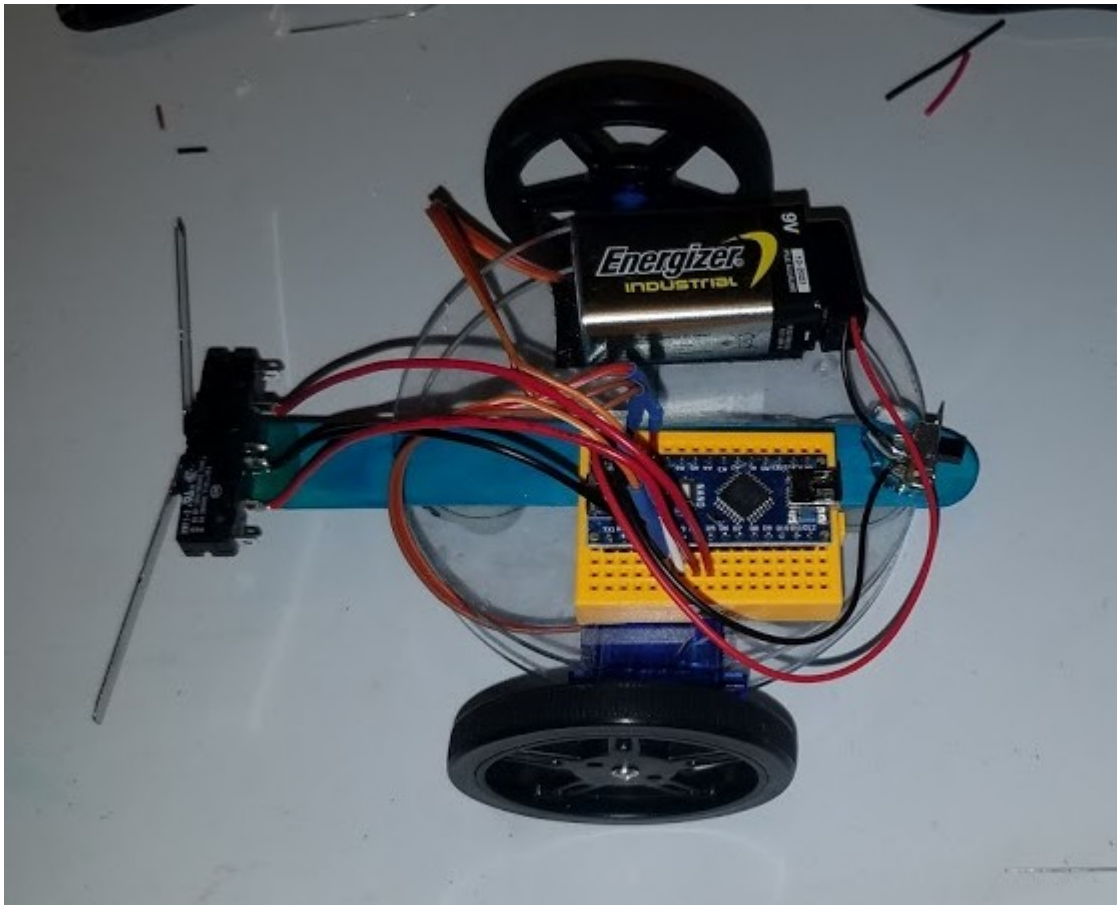
Connect a piece of ~6” red solid 22-gauge wire to each normally open terminal of the snap-action switches. Solder wire to each terminal.



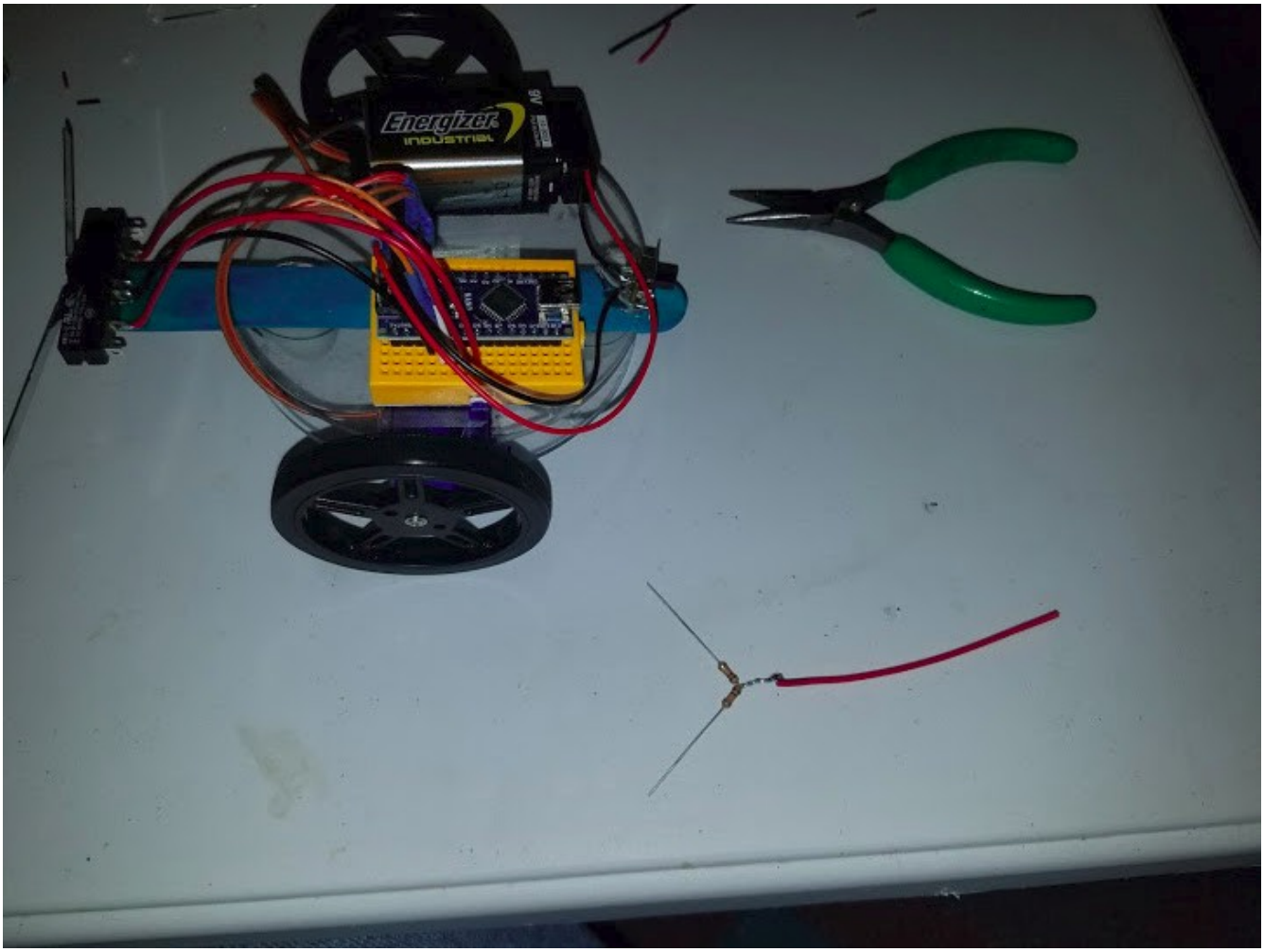
Bend the whisker switch wires up in preparation to glue the whiskers to the front of the Popsicle stick with 5-minute epoxy. Be sure and not get glue on the moving parts of the switch.



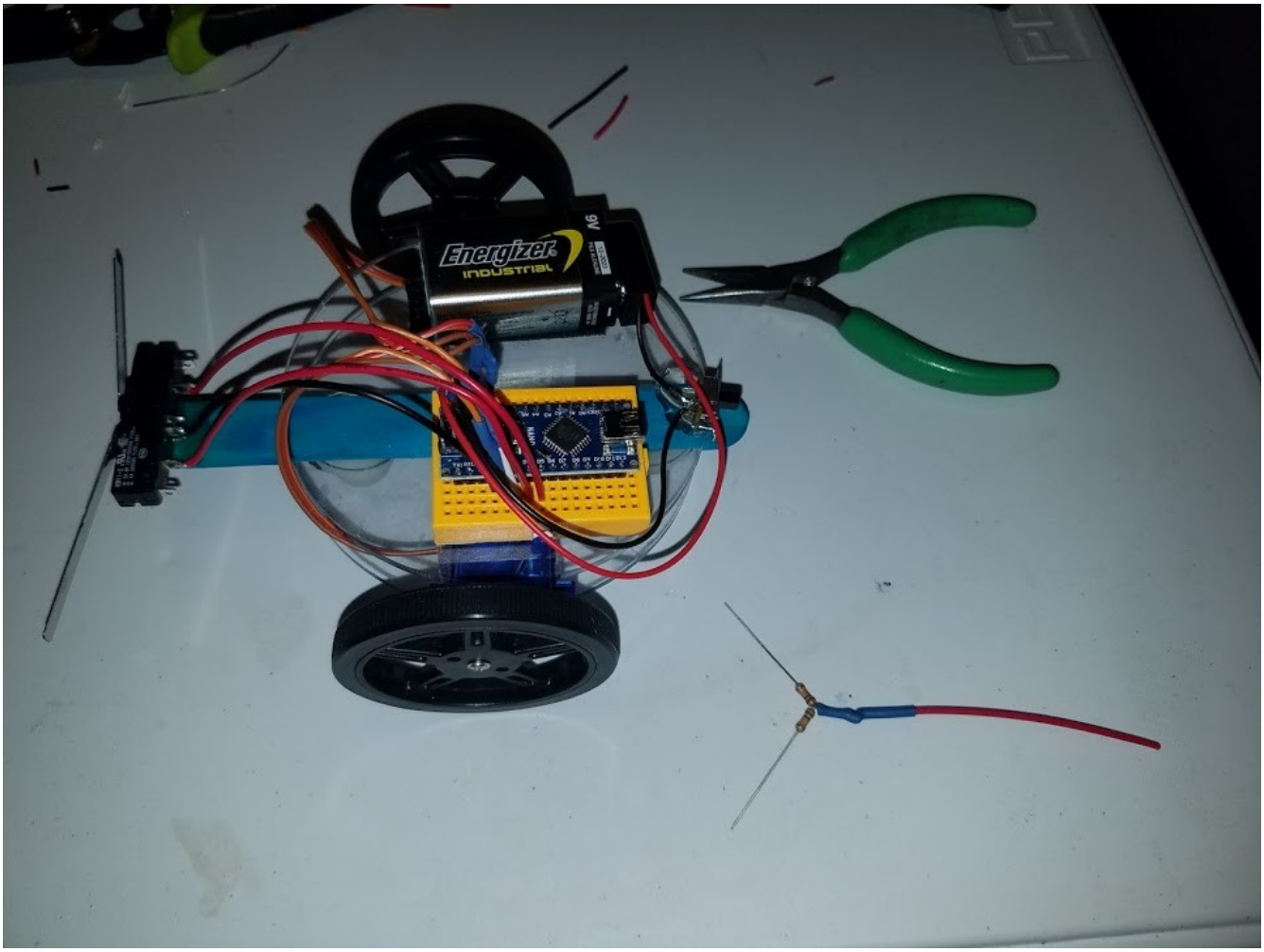
Cut the black 22-gauge wire to length and strip ~1/2” insulation off the end. Plug 22-gauge plug into GND on the left side off the Arduino.



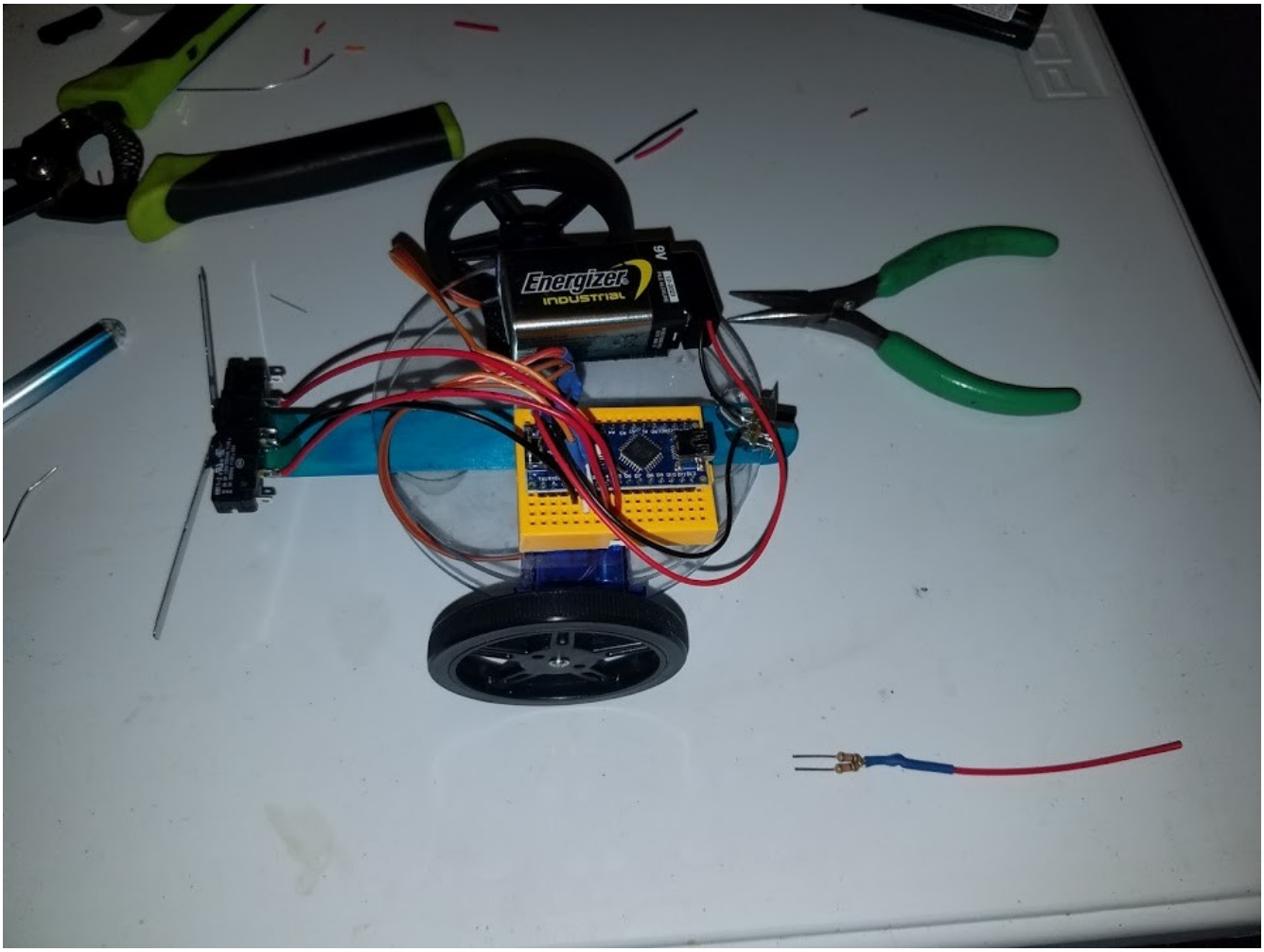
Cut the two red whisker 22-gauge solid wires to length. Strip ~1/2” off the end of each wire and plug the left switch into D4 and the right switch into D5.



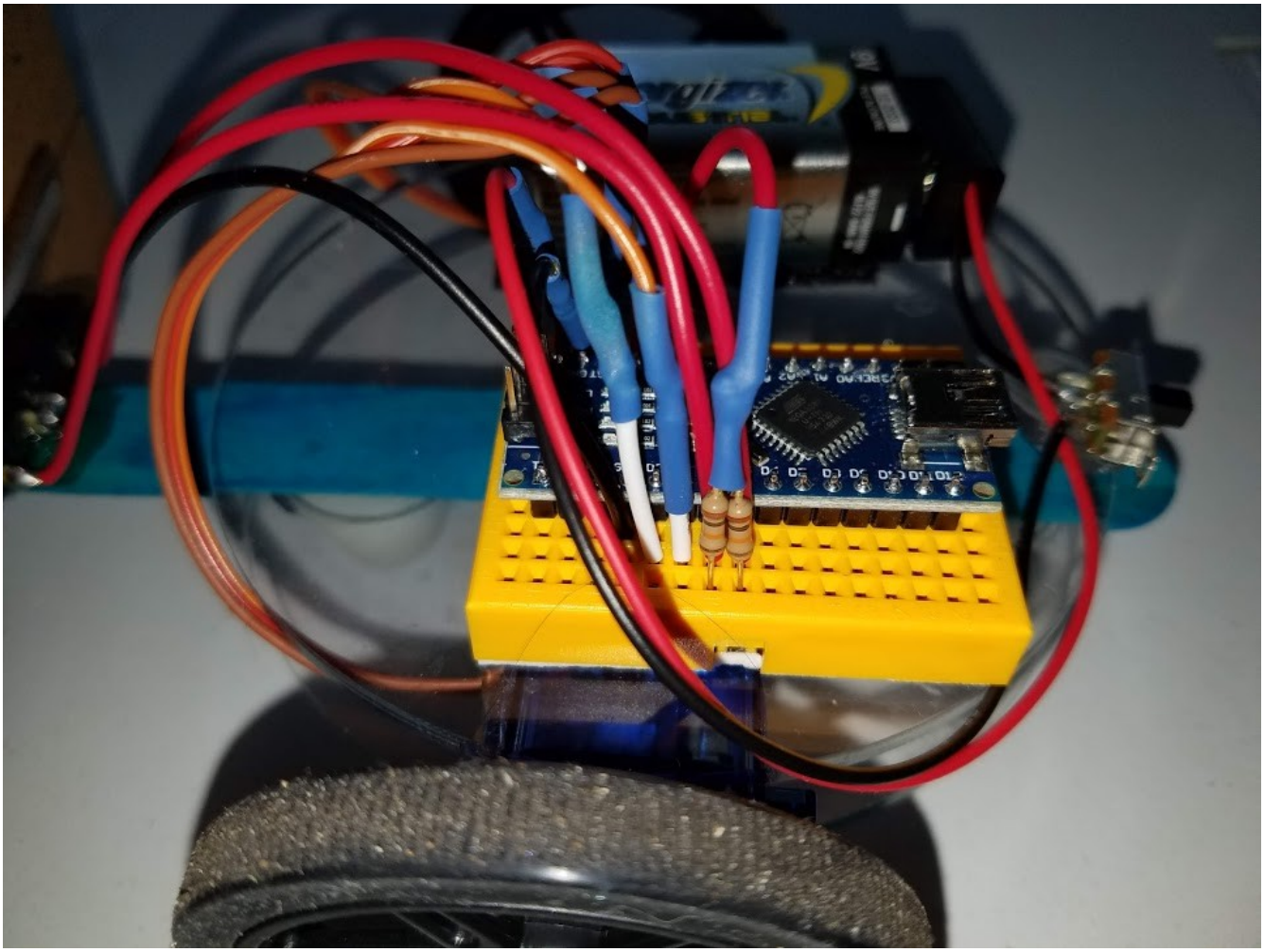
Take two 10K Ohm resistors and wrap two ends together. Take ~3” piece of 22-gauge solid wire and strip ~1/2” insulation off one end. Wrap the two resistor ends around the 22-gauge solid wire and solder. These resistors are very important! They limit the current into your Arduino otherwise it could be damaged.



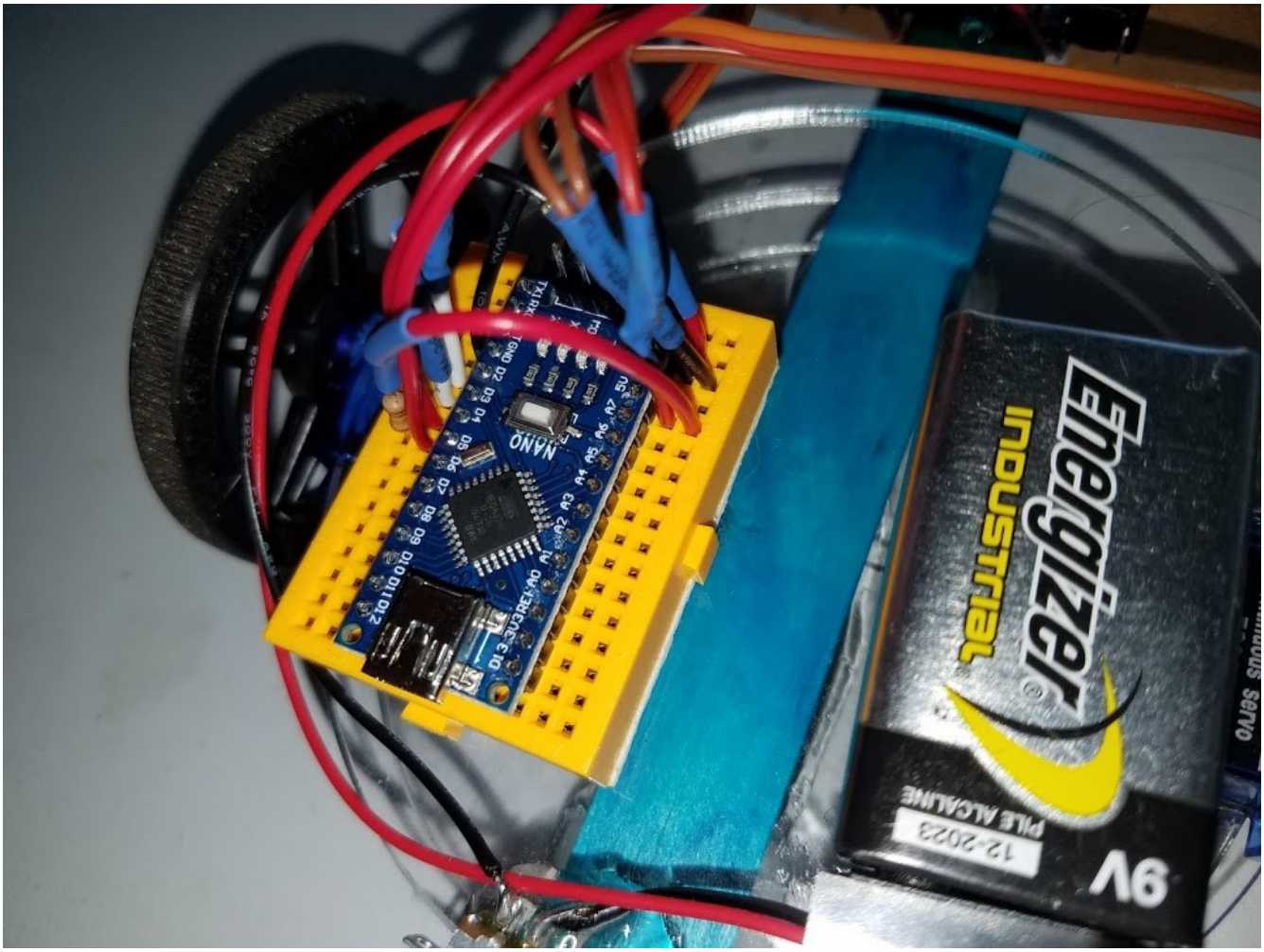
Heat shrink or wrap electrical tape around solder joint.



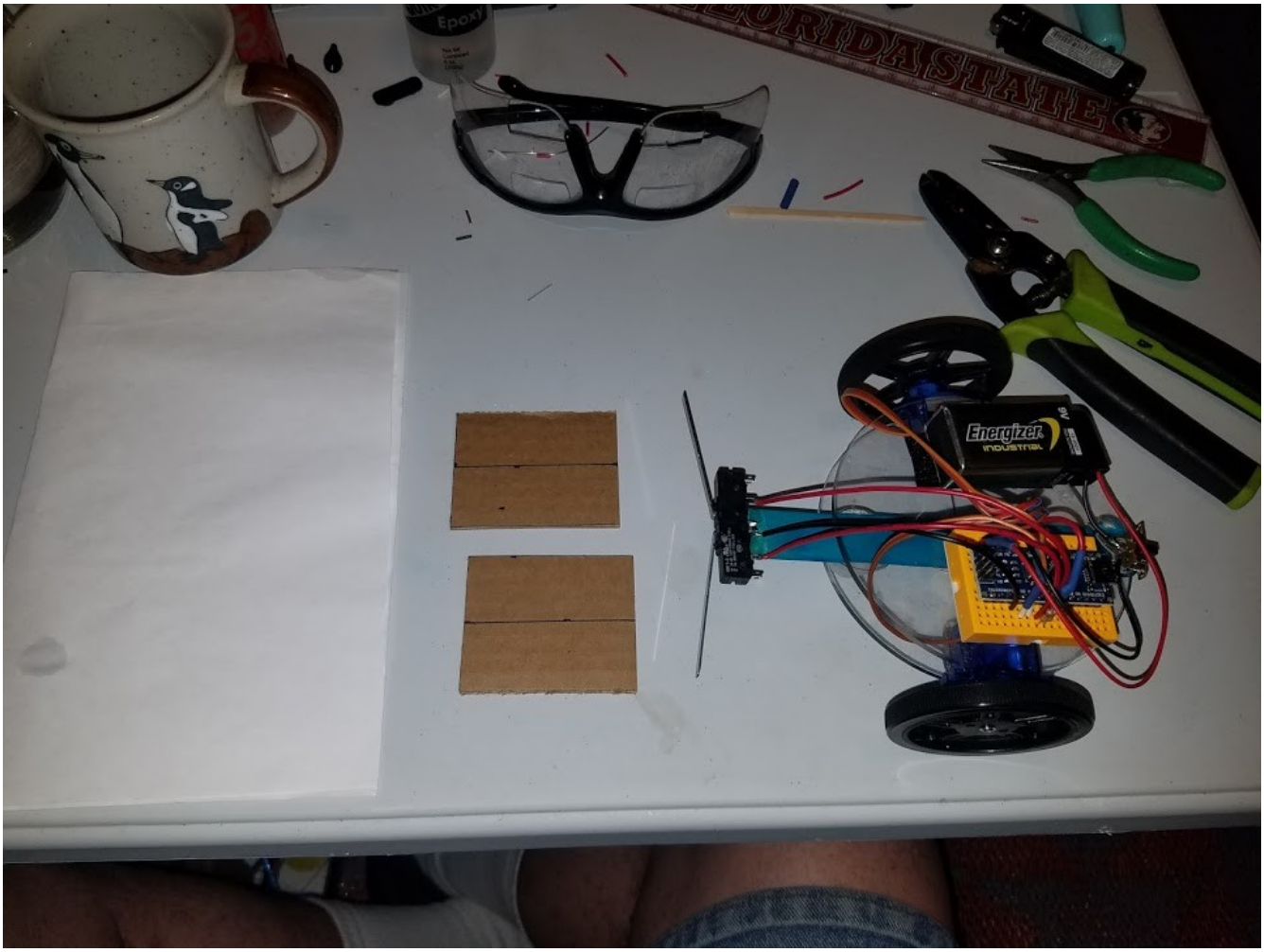
Cut the resistor leads to ~1/2”.



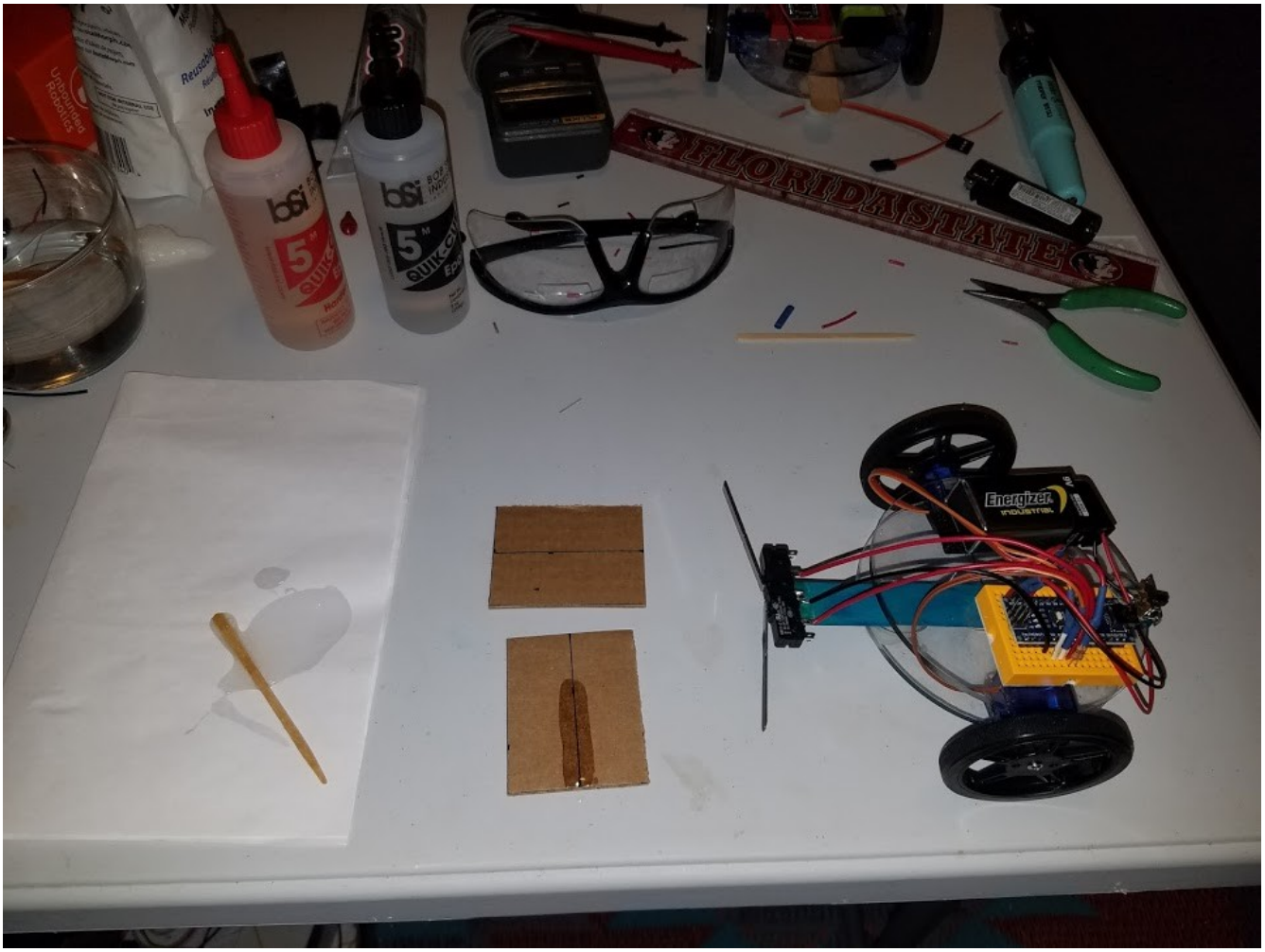
Plug resistor leads into D4 and D5.



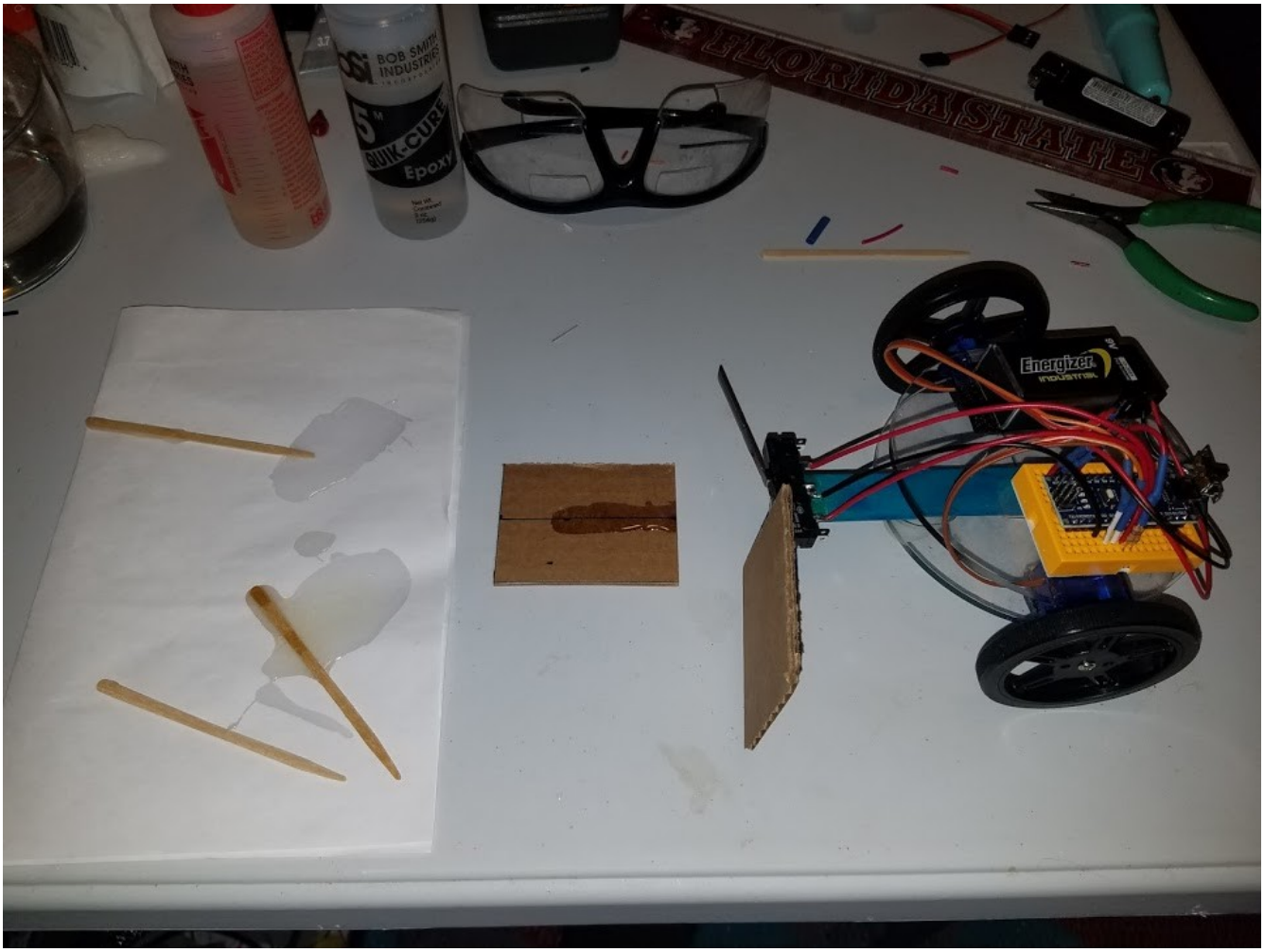
Cut red 22-gauge solid wire to size and strip off ~1/2". Plug into 5V on right side of Arduino Nano.



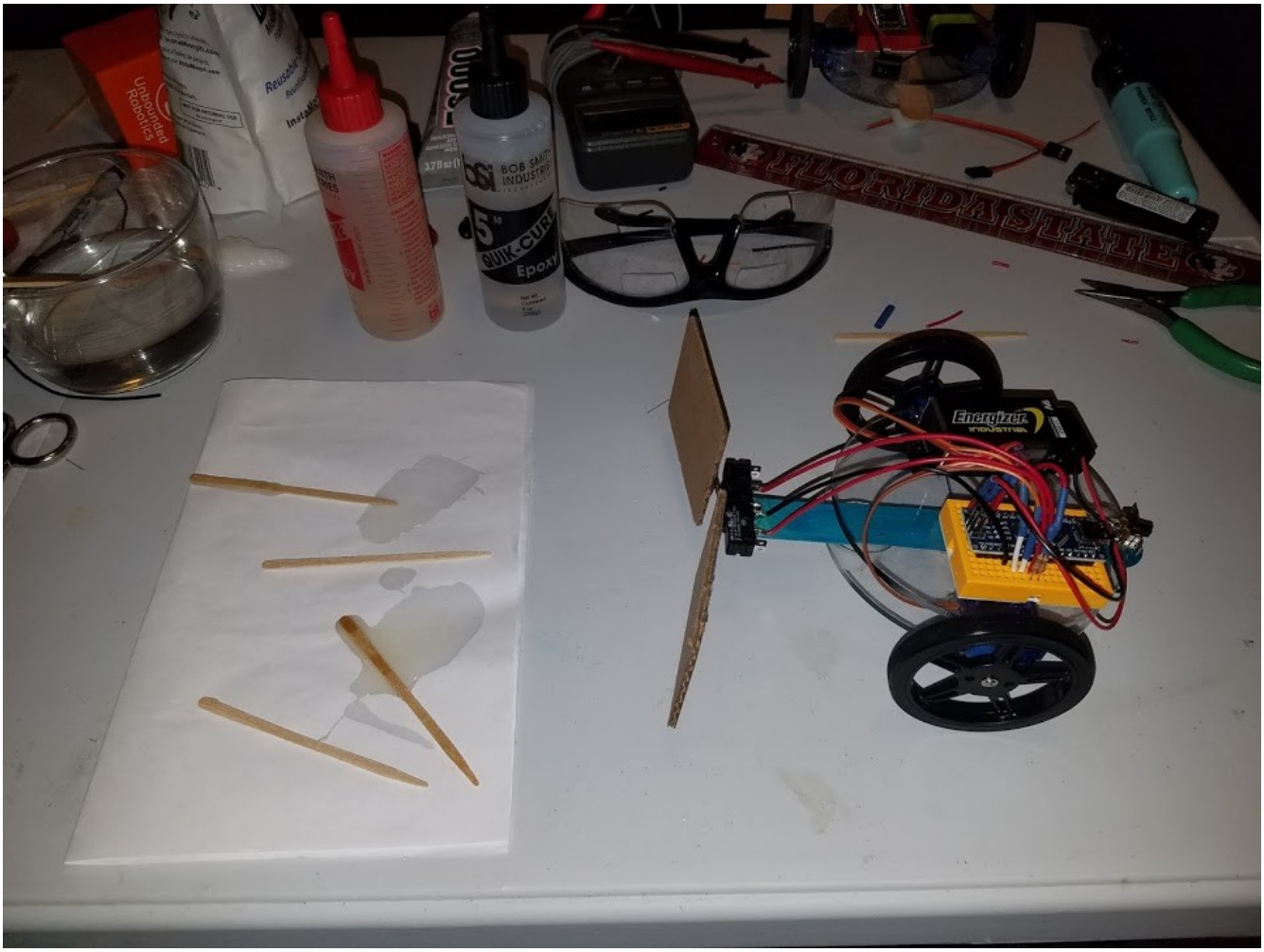
21. Lastly let's make the whisker extensions. This is to widen the whiskers so they actuate on low and high obstacles. Cut two pieces of cardboard ~2.5" x 2". Make a line through the center.



Put glue on line and glue cardboard to snap_action switch lever one at a time. Again, be careful not to glue the moving parts.



Glue the other whisker extension to the snap-action lever.



This concludes the electrical and mechanical build. Below find the Arduino code to make the robot go forward and sense and bump around obstacles as it moves through the environment.

```
//  Nano Turtle code:

#include <Servo.h>              // Load "Servo" library

Servo servoLeft;               // Left drive servo

Servo servoRight;              // Right drive servo

const int BumpLeft = 4;        // Left bumper Pin 4

const int BumpRight = 5;       // Right bumper Pin 5

int BumpStateLeft = 0;         // Set Pin value

int BumpStateRight = 0;        // Set Pin value
```

```
void setup()

{
  Serial.begin(9600);                 // Setup serial monitor for debug
  servoLeft.attach(2);                // Set left servo to pin 2
  servoRight.attach(3);               // Set right servo to pin 3
  pinMode(BumpLeft, INPUT);           // Set BumperLeft to input
  pinMode(BumpRight, INPUT);          // Set BumperRight to input
}

void loop(){

   BumpStateLeft = digitalRead(BumpLeft);
   Serial.println (BumpStateLeft, DEC);
   if (BumpStateLeft == 0){
   Stop();
   Reverse();
   Stop();
   Counterclockwise();
   Stop();
   Forward();
}

    BumpStateRight = digitalRead(BumpRight);
   Serial.println (BumpStateRight, DEC);
```

```
  if (BumpStateRight == 0){
  Stop();
  Reverse();
  Stop();
  Clockwise();
  Stop();
  Forward();
}

{
  Forward();
}
}

void Forward()
{
 servoLeft.write(75);  // 79
 servoRight.write(104);  // 104
 delay(5);
}

void Counterclockwise()
{
 servoLeft.write(104);
 servoRight.write(104);
 delay(600);
```

```
}

void Clockwise()
{
  servoLeft.write(79);
  servoRight.write(79);
  delay(400);
}

void Reverse()
{
  servoLeft.write(104);
  servoRight.write(79);
  delay(400);
}

void Stop()
{
  servoLeft.write(90);
  servoRight.write(90);
  delay (100);
}

void Left()
{
  servoLeft.write(0);
```

```
  servoRight.write(105);
  delay (50);
}

void Right()
{
  servoLeft.write(90);
  servoRight.write(79);
  delay (50);
}
```