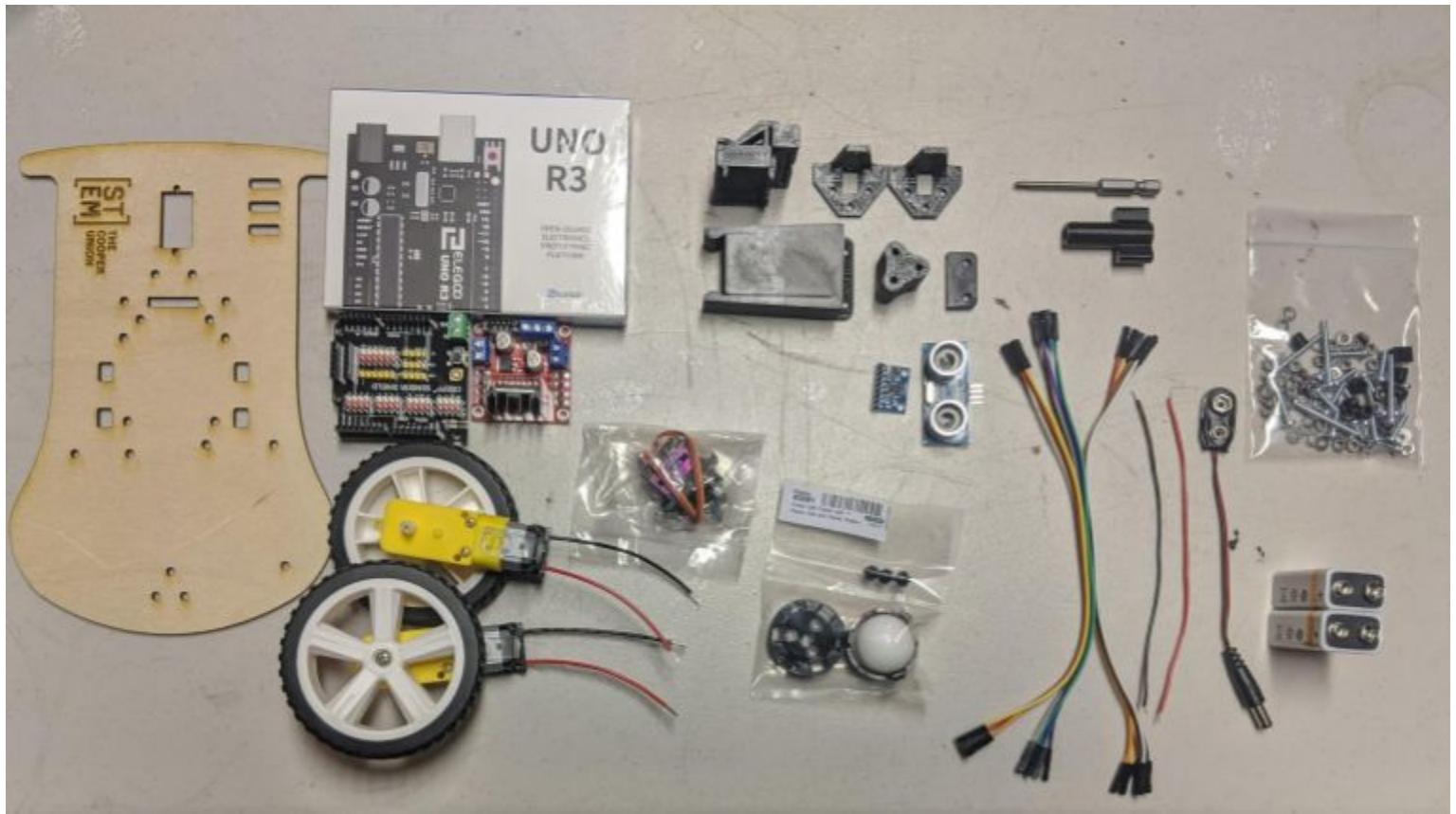


Assembly Procedure

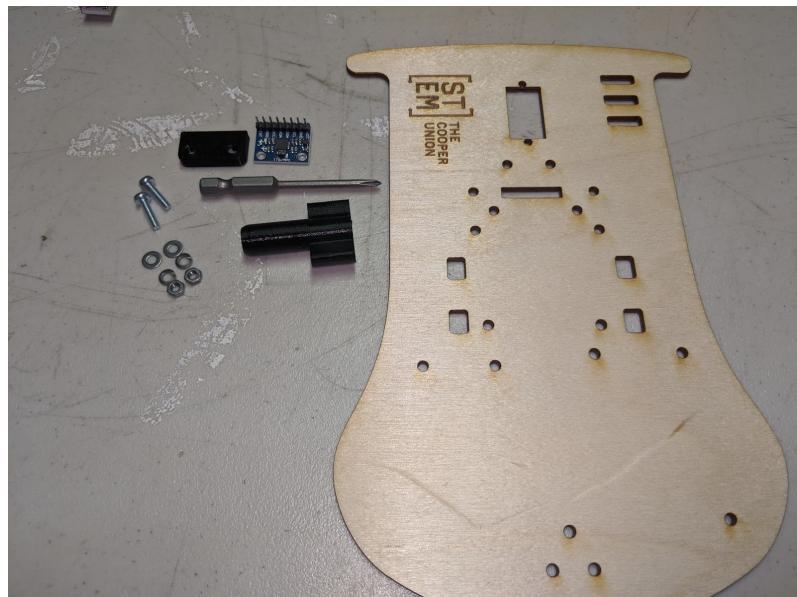


1. MPU6050 attachment

a. Retrieve:

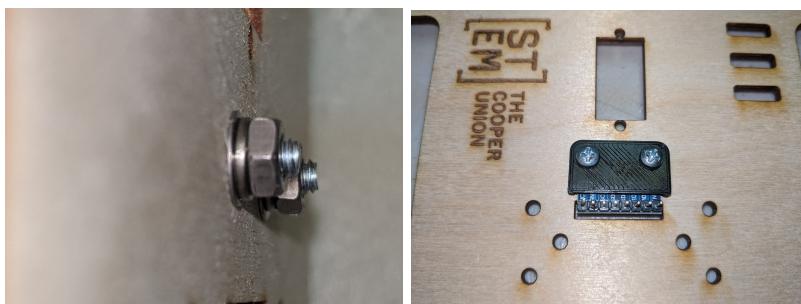
- i. MPU6050
- ii. MPU6050 Cover
- iii. 2x M3x12mm
- iv. 2x M3 Nut
- v. 2x M3 Lock Washer
- vi. 2x M3 Washer
- vii. Frame
- viii. 3mm Hex Nut Key

1.



b. Fasten MPU6050 and Cover onto frame as shown

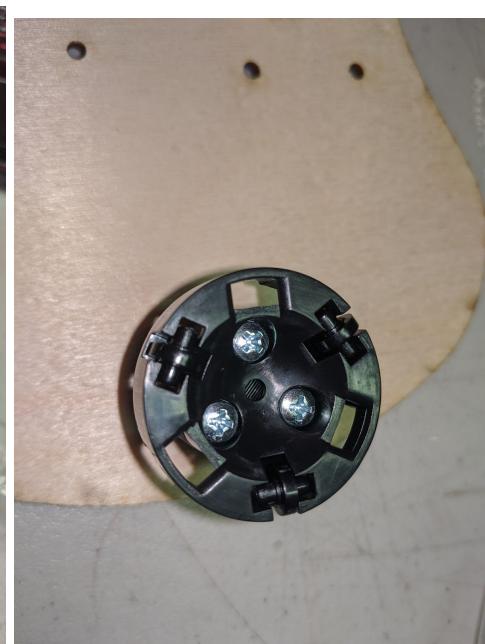
- i. Note about fastening, the order of washers goes Washer, Lock Washer, then Nut
(THIS GOES FOR ALL FASTENERS, remember this for later)
- ii. Second Note: Finger tighten nut first, then use 3mm Key to tighten. Remember that this key is plastic, so do not use too much force as you may strip the plastic key.



iii.

2. Ball Caster Attachment

- a. Retrieve:
 - i. Ball Caster bag
 - ii. Ball Caster Spacer
 - iii. 3mm Hex key
 - iv. Phillips Head driver
 - v. 3x M3x30mm
 - vi. 3x M3 Nut
 - vii. 3x M3 Washer
- b. Remove from ball Caster bag the part with 3 holes in it
 - i. Attach to frame with spacer as shown
 - ii. Note: NO LOCK WASHERS HERE
 - iii. Note: the side the logo is on in the picture



- c. Remove 3 mini rollers from bag and place into ball caster housing.
- d. Place ball on top of rollers
- e. Snap final part of housing around ball



3. Motor Mounting

a. Retrieve:

- i. 2x Motors
- ii. 2x Motor Mounts
- iii. 4x M3-30mm
- iv. 6x M3-12mm
- v. 10x M3 Nuts
- vi. 10x M3 Lock Washers
- vii. 10x M3 Washers

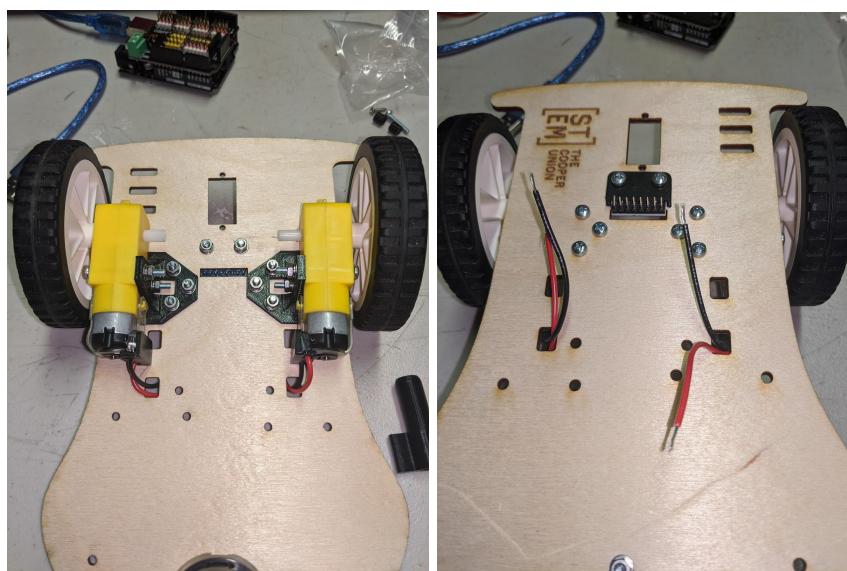
b. Mount Motor Mount to Motor

- i. It can be tricky to get the washer and lock washer onto the fastener that is enclosed, be patient you'll get it.
- ii. Note that the direction of each motor mount is important, take a look at the picture to see the difference.



iii.

c. Mount Motor assemblies to Frame and pass wires through rectangular hole to top side of Frame



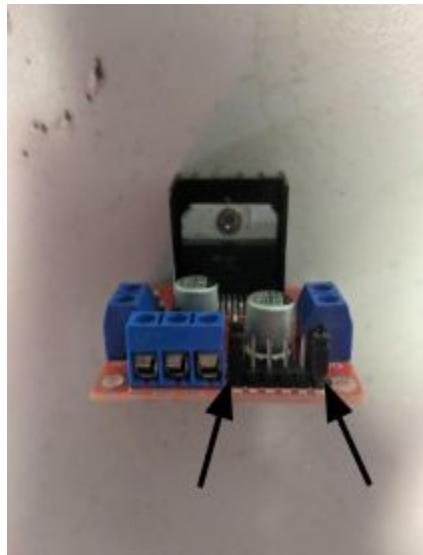
i.

4. Motor Controller Mounting and Power Wire routing

a. Retrieve:

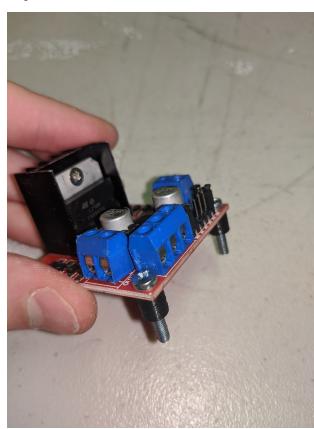
- i. Motor Controller
- ii. 2x M3x16mm
- iii. 2x M3 Nut
- iv. 2x M3 Lock Washer
- v. 2x M3 Washer
- vi. 2x M3 plastic spacer
- vii. Red and Black Solid Core Wire

b. Remove Jumpers from motor controller (These are connected to the ENA, and ENB pins)



i.

c. Pass M3 Fasteners through two holes shown and push plastic spacers onto fasteners. If plastic spacers do not want to cooperate, rotate them like a nut and they will move.



i.

d. Attach motor controller to frame



- i.
- e. Fasten the two solid core wires into proper screw terminals on motor controller
 - i. Loosen the screw terminals first
 - ii. Insert the wire into the terminal and tighten screw to hold wire in place
 - iii. Make sure you use the BLACK wire for the GND terminal
 - iv. Make sure you use the RED wire for the +12V terminal
 - v. GND is the center terminal
 - vi. +12V is the left terminal
 - vii. Note to bend the wires as shown for later

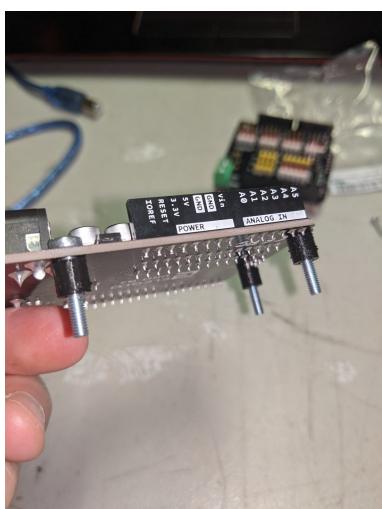
5. Arduino Attachment

a. Retrieve:

- i. Arduino UNO
- ii. 3x M3x16mm
- iii. 3x M3 Nut
- iv. 3x M3 Lock Washer
- v. 3x M3 Washer
- vi. 3x M3 Plastic spacer

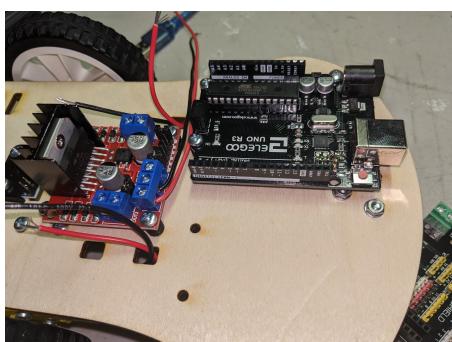
b. Like the motor controller pass the fasteners through the holes and attach plastic spacers as shown

- i. Note: one fastener hole will be empty



ii.

c. Fasten Arduino to Frame



i.

d. Arduino Uno and Sensor Shield prep

- i. Find the ICSP Pin Cover (it's a rectangular, hollow, plastic part)
- ii. Place ICSP Pin cover on arduino



1.

iii. Retrieve Sensor Shield and attach to UNO



iv.

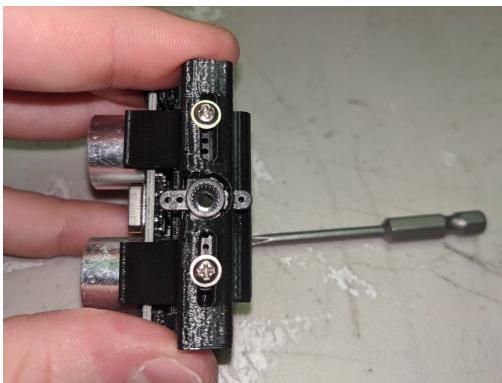
6. HC-SR04 sensor attachment w/Servo

- a. Retreive HC-SR04 and Ultrasonic Distance Sensor Bracket
- b. Retrieve mini servo motor bag that contains servo horns and fasteners
- c. Take out cross shaped servo horn and push into opening on bracket in proper orientation



i.

- d. Retrieve the two “self-tapping” screws from the servo bag. (They have the pointed end and are the longer of the two screw in the bag)
- e. Drive these screws into the outer-most holes on the servo horn using the phillips screw driver provided (if you have your own screw driver use that one as it may likely be better)



i.

- f. Open the Arduino IDE.

- i. Write a short script that will set the servo’s position to 90 degrees. This is so that we can place the sensor onto the servo such that it points forward on the robot when at 90 degrees. This will allow the sensor to be able to point fully right and left on the robot since the servo has a 180 degree rotation range.
- ii. Important lines of code

The screenshot shows the Arduino IDE interface with the following code in the editor:

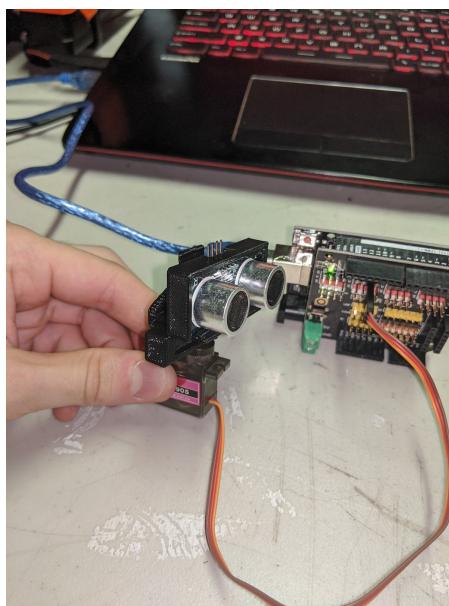
```
sketch_jun10a | Arduino 1.8.7
File Edit Sketch Tools Help
sketch_jun10a
1 #include <Servo.h>
2 Servo servol;
3
4
5 void setup() {
6     // put your setup code here, to run once:
7     servol.attach(9);
8     servol.write(90);
9 }
10
11 void loop() {
12     // put your main code here, to run repeatedly:
13 }
14 }
```

The status bar at the bottom indicates "Done uploading." and shows two error messages: "Invalid library found in D:\Documents\Robotics Crash Cour" and "Invalid library found in D:\Documents\Robotics Crash Cour".

1.

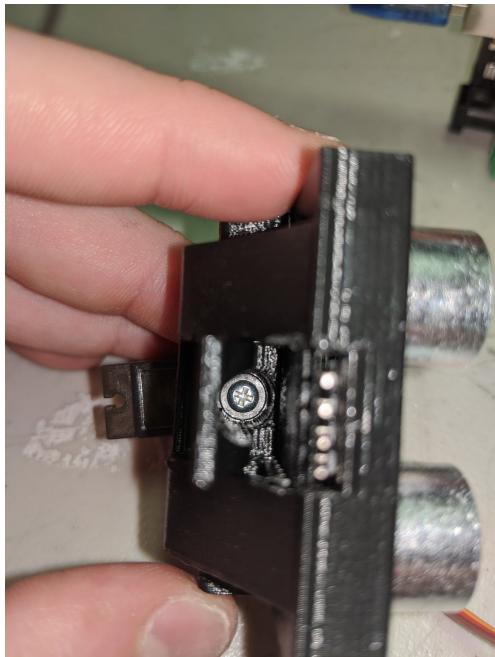
g. Upload this code to the arduino

- i. Connect servo to proper pin on sensor shield. Make sure orientation of connector is correct, G on the shield stands for GND (has a black bottom), which is the BROWN wire on the servo.
- ii. Make sure the servo has rotated to the proper position
- iii. Attach HC-SR04 as the picture shows (note the direction the wires are coming out of the servo in the picture) Don't worry if it is not perfectly straight, get it as close as possible



1.

h. Retrieve the last small screw in the servo bag, and fasten the servo horn to the servo.



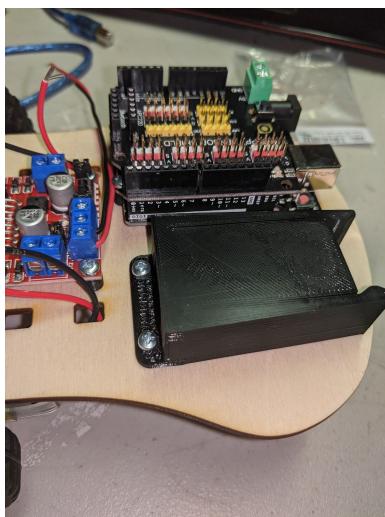
i.

7. 9V Battery Mount Mounting

a. Retrieve:

- i. 9V Battery Mount
- ii. 2x M3x16mm
- iii. You know the drill
- iv. No spacers here

b. Fasten mount to frame



i.

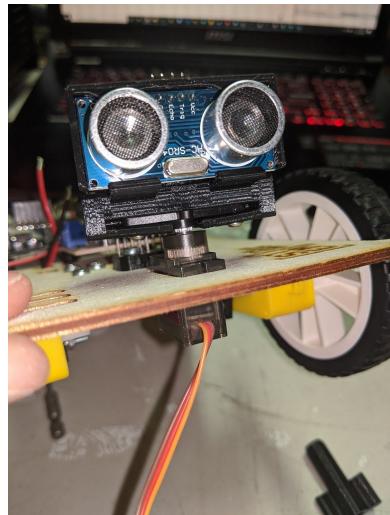
8. Mount Servo to Frame and route wire

a. Retrieve:

- i. Servo with sensor attached
- ii. 2x M2.2x9.5mm self tapping fastener
 1. This is the fastener with the pointed end in the fastener bag

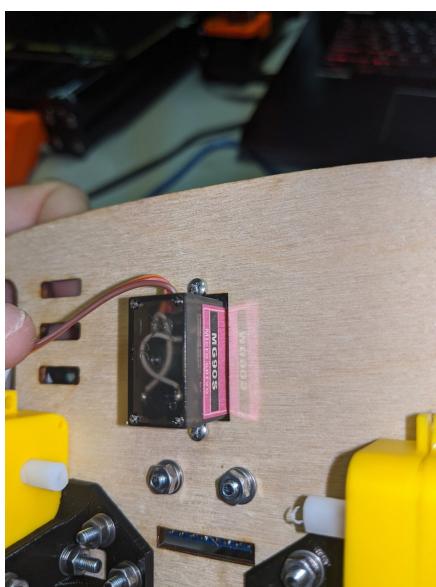
b. Take servo and pass wire through large rectangular hole

- i. Make sure wires are coming out of the servo towards the front of the robot



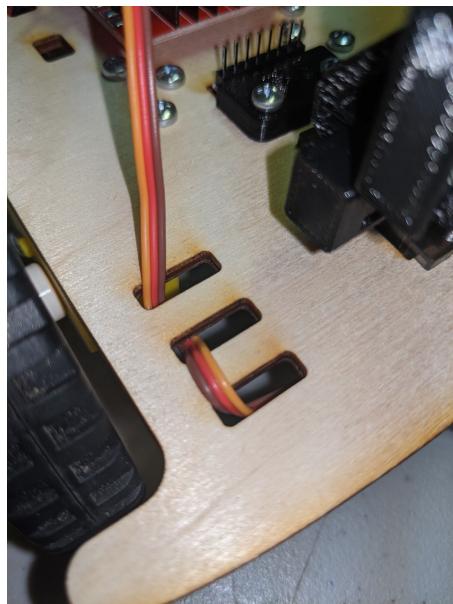
c. Fasten servo to frame

- i. Note: Head of screw will be on bottom of frame
- ii. Note: DO NOT OVER TIGHTEN, THE SCREW WILL EAT THROUGH THE PLASTIC OF THE SERVO AND NOT HOLD IT IN PLACE



iii.

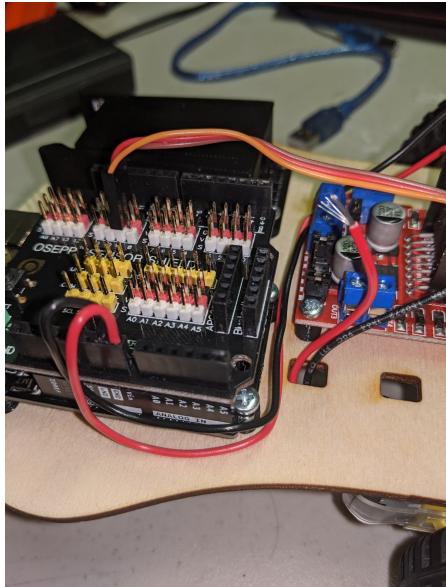
d. Route servo wire as follows



i.

9. Final Wiring and routing

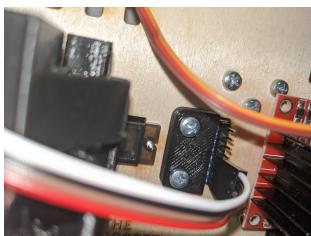
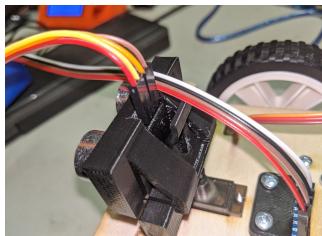
- a. Take the solid core wires connected to the motor controller and bend them appropriately into the Vin and GND pins on the sensor shield
 - i. GND is Black, Vin is Red
 - ii. GND should be the center terminal on the motor controller
 - iii. +V should be the left terminal on the motor controller (the one closer to the fastener)



iv.

- b. Connect jumper wires to sensors and motor controllers

- i. Set of 4 for MPU6050 and HC_SR04, Set of 6 for the Motor controller

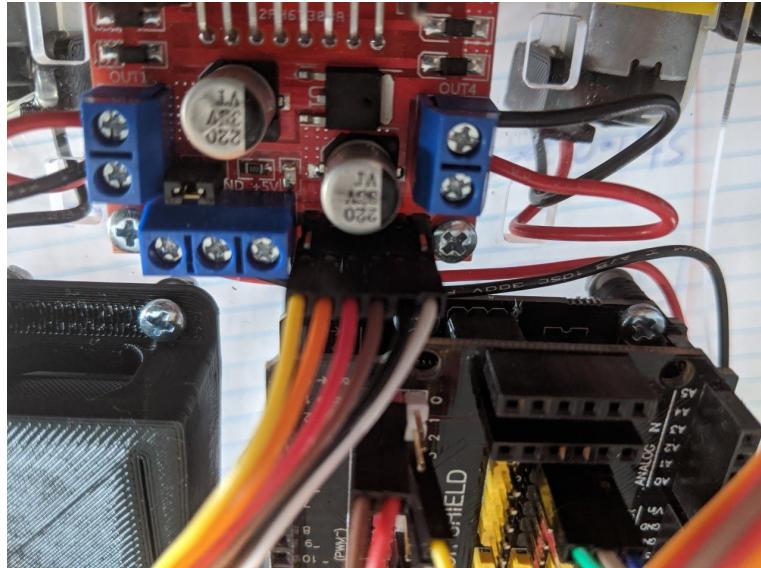


ii.

- c. Connect Motor Controller Jumper wires to sensor shield pins as follows

- i. If looking from the back of the robot to the motor controller there are six pins grouped together (two may have a jumper connected to it, remove this jumper), from left to right the pins go:

1. 6,7,8,11,12,3 (these pins are on the sensor shield and have a white bottom)
 2. ENA:6
 3. IN1: 7
 4. IN2: 8
 5. IN3: 11
 6. IN4: 12
 7. ENB: 3

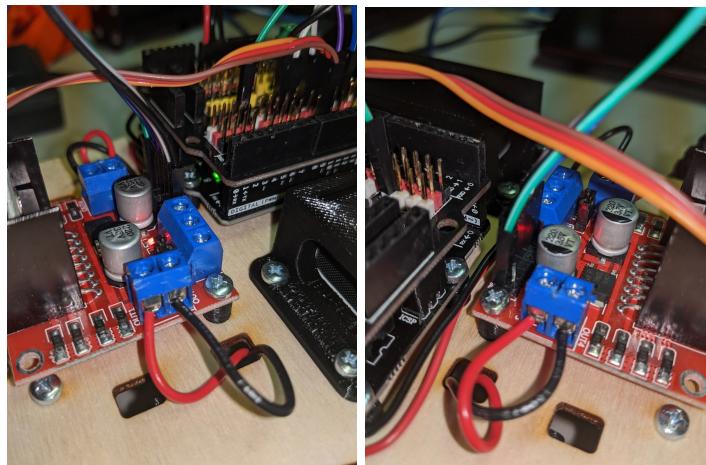


ii.

- iii. Remove small jumper that is connected on the outer 2 pins if necessary. (Use screw drive to pull them up and out)

d. Connect motor wires to the motor controller. Don't worry about the order right now, we will have to change them in a moment to make sure they're spinning in the proper direction

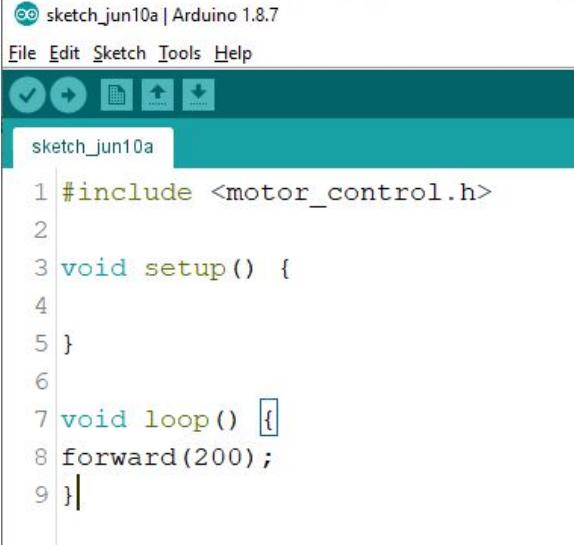
- i. Note: you may have to loosen the screw terminals to be able to push the wire in
- ii. Note: Make sure the wires are not fraying, if they are, spin them with your fingers so they are easy to insert into the screw terminal



iii.

e. Now we will test to see if the motors are rotating properly

- i. Open the Arduino IDE
- ii. Make sure the Sketchbook location is the arduino_code directory of your local repository on your computer or you'll get an error saying there is no file called motor_control.h
- iii. Type the following code:



```
#include <motor_control.h>
void setup() {
}
void loop() {
    forward(200);
}
```

- 1.
- iv. Upload this to the arduino
- v. Connect 9V battery to 9V battery connector
- vi. Plug 9V into arduino (be aware that the wheels will start spinning, so pick up the robot.)
- vii. Take a note on which wheels are spinning the wrong way (the wheels should be spinning such that the robot will move forward)
- viii. Flip the connection for any motor that is spinning in the opposite direction.
- f. Unplug the 9V battery from the robot!
- g. Connecting Ultrasonic Distance sensor to Sensor Shield
 - i. Plug in set of 4 jumper wires into distance sensor
 - ii. Connect Vcc -> 5V pin (Red on the shield)
 - iii. Connect GND -> GND (Black on the shield)
 - iv. Connect Trig -> to Digital Pin #10 (white on the shield)
 - v. Connect Echo -> Digital Pin #2 (white on the shield)

```
sketch_jul02a | Arduino 1.8.7
File Edit Sketch Tools Help
sketch_jul02a §
1 #include <HC_SR04.h>
2
3 #define ECHO_PIN 2
4 #define TRIG_PIN 10
5 #define ECHO_INT 0
6
7 HC_SR04 distance_sensor(TRIG_PIN, ECHO_PIN, ECHO_INT);
8
9 void setup() {
10 //Begin serial comms for debugging
11 Serial.begin(9600);
12
13 //Begin distance sensor
14 distance_sensor.begin();
15 distance_sensor.start();
16 }
17
18 void loop() {
19
20 if(distance_sensor.isFinished())
21 {
22   Serial.println(distance_sensor.getRange());
23   distance_sensor.start();
24 }
25 }
```

Done uploading.
Sketch uses 2706 bytes (8%) of program storage space. Maximum Global variables use 211 bytes (10%) of dynamic memory, leaving 1788 bytes free.

vi.

h. Connecting MPU6050 to sensor shield

- i. From top to bottom the pin order is (Vcc, GND, SCL, SDA)
- ii. Connect Vcc to 5V on shield
- iii. Connect GND to GND on shield
- iv. Connect SCL to A5 on shield
- v. Connect SDA to A4 on shield

sketch_jul02a | Arduino 1.8.7

File Edit Sketch Tools Help

sketch_jul02a §

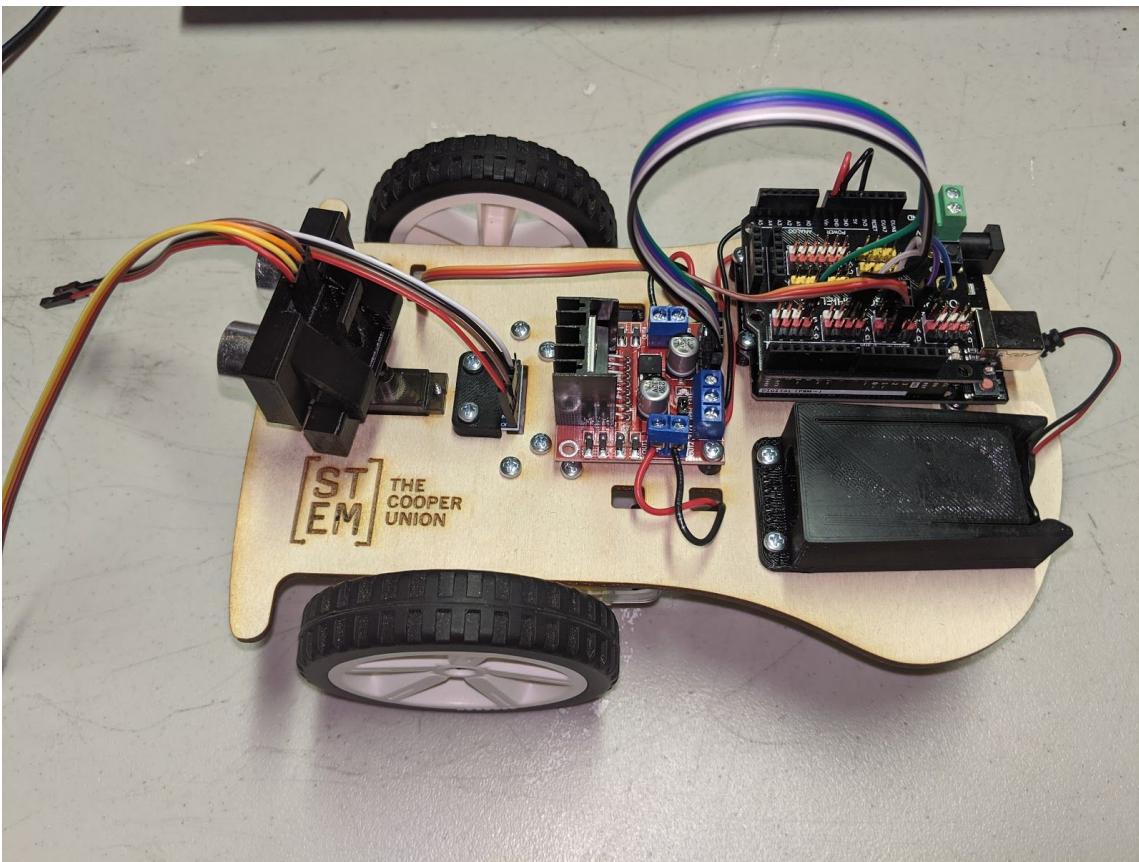
```
1 #include <MPU6050.h>
2
3 #define SDA 4
4 #define SCL 5
5
6 MPU6050 sensor(SDA, SCL);
7
8 void setup() {
9     //Begin serial comms for debugging
10    Serial.begin(9600);
11
12    //Begin distance sensor
13    sensor.initialize();
14    sensor.update();
15 }
16
17 void loop() {
18
19    sensor.update();
20    Serial.println(sensor.get_accel('x'));
21 }
```

Done uploading.

Sketch uses 5642 bytes (17%) of program storage space.
Global variables use 472 bytes (23%) of dynamic memory

vi.

j. YOU'RE DONE!



k.