

1. Servo Wiring

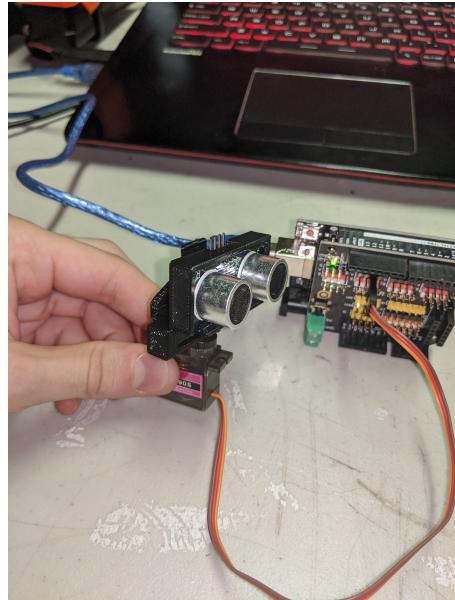
- a. Open the Arduino IDE.
- b. Write a short script that will set the servo's position to 90 degrees. This is so that we can place the sensor onto the servo such that it points forward on the robot when at 90 degrees. This will allow the sensor to be able to point fully right and left on the robot since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



```
sketch_jun29b:1
1 #include <Servo.h>
2 Servo servol;
3
4 void setup() {
5     // put your setup code here, to run once:
6     servol.attach(9);
7     servol.write(90);
8 }
9
10 void loop() {
11     // put your main code here, to run repeatedly:
12
13 }
14
```

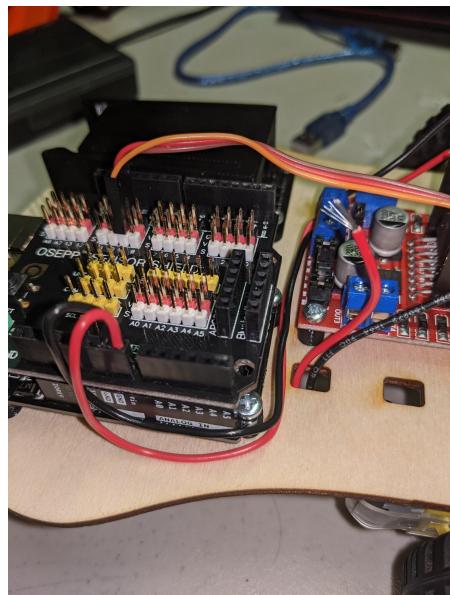
i.

- d. Upload this code to the arduino
 - i. Connect servo to pin 9 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



2. Motor controller Wiring

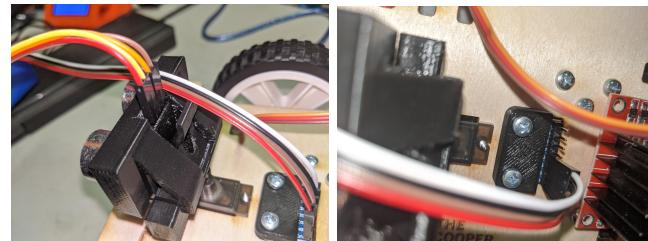
- a. Take the solid core wires connected to the motor controller and bend them appropriately into the Vin and GND pins on the sensor shield
 - i. GND is Black, Vin is Red
 - ii. GND should be the center terminal on the motor controller
 - iii. +V should be the left terminal on the motor controller (the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



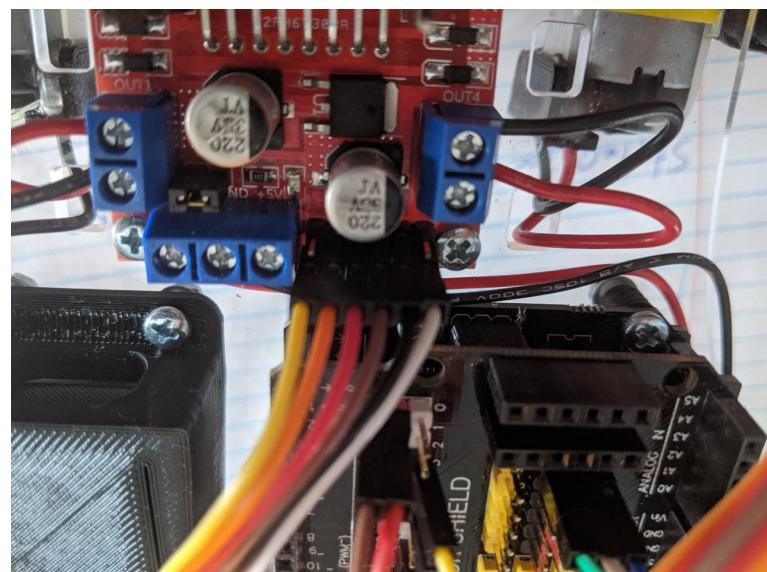
ii.



- 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 5,7,8,11,12,6 (these pins are on the sensor shield and have a white bottom)

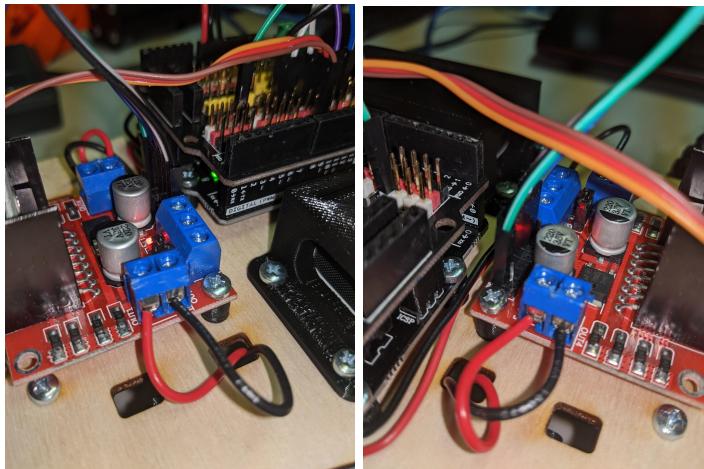
1. ENA: 5
2. IN1: 7
3. IN2: 8
4. IN3: 11
5. IN4: 12
6. ENB: 6



ii.

- iii. Remove small jumper that is connected on the outer 2 pins if necessary.
(Use screw driver 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
- Note: you may have to loosen the screw terminals to be able to push the wire in
 - 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



- Now we will test to see if the motors are rotating properly
- Open the Arduino IDE
- 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
- Type the following code:

The screenshot shows the Arduino IDE interface. The title bar reads "sketch_jul07a | Arduino 1.8.13". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu is a toolbar with icons for upload, verify, and other functions. The main area displays the following C++ code:

```
1 #include <motor_control.h>
2
3 void setup() {
4     // put your setup code here, to run once:
5     motor_setup();
6 }
7
8 void loop() {
9     // put your main code here, to run repeatedly:
10    raw_motor_control(200,200);
11 }
```

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!

3. Connecting Ultrasonic Distance sensor to Sensor Shield

- a. Plug in set of 4 jumper wires into distance sensor
- b. Connect Vcc -> 5V pin (Red on the shield)
- c. Connect Trig -> to Digital Pin #10 (white on the shield)
- d. Connect Echo -> Digital Pin #2 (white on the shield)
- e. Connect GND -> GND (Black on the shield)

The screenshot shows the Arduino IDE interface. The title bar reads "sketch_jun29b | Arduino 1.8.13". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu is a toolbar with icons for save, upload, and other functions. The main area displays the following C++ code:

```
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
10 void setup() {
11     // put your setup code here, to run once:
12     Serial.begin(9600);
13
14     distance_sensor.initialize();
15     distance_sensor.start();
16 }
17
18 void loop() {
19     // put your main code here, to run repeatedly:
20     if(distance_sensor.isFinished())
21     {
22         Serial.println(distance_sensor.getRange());
23         distance_sensor.start();
24     }
25 }
26
```

The status bar at the bottom left shows "i. 15" and "Arduino Uno on COM3".

f. Connecting MPU6050 to sensor shield

- i. From top to bottom the pin order it (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

The screenshot shows the Arduino IDE interface. The title bar reads "sketch_jul02a | Arduino 1.8.7". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for upload, download, and other functions. The main code editor window contains the following C++ code:

```
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 }
```

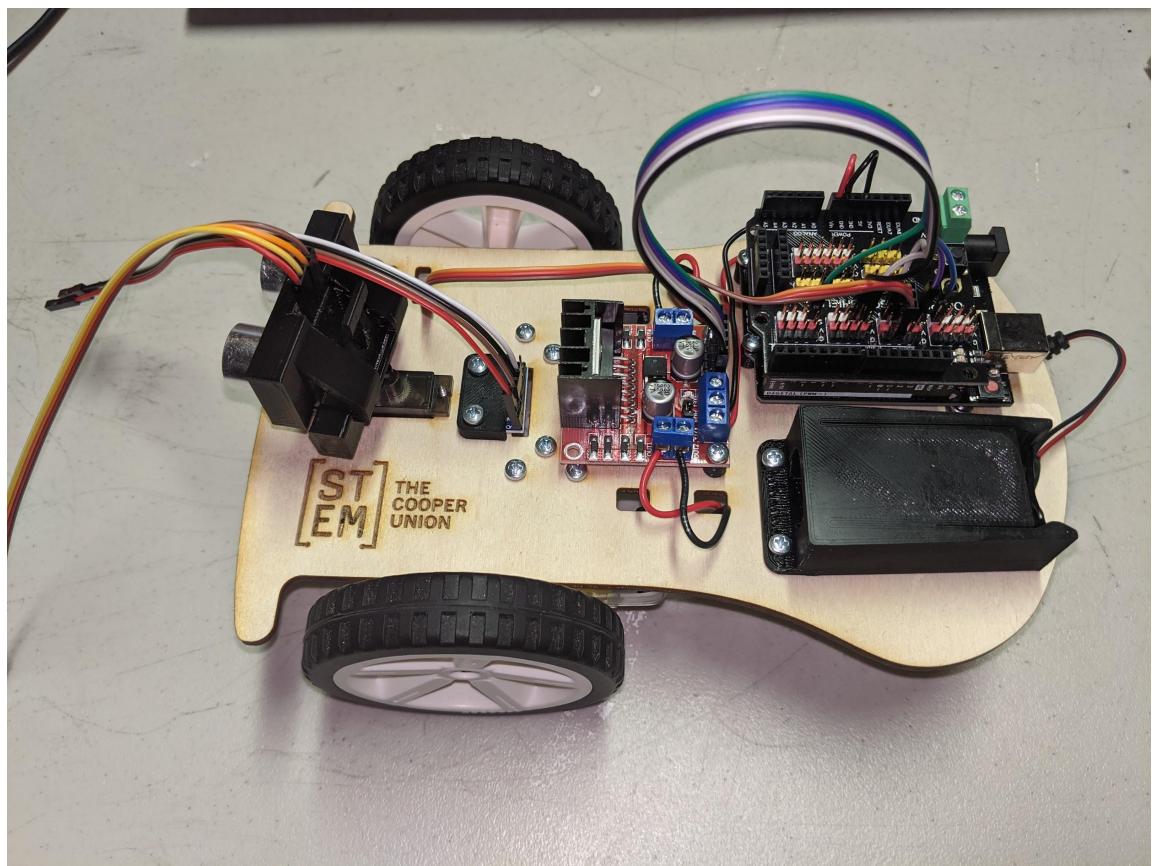
At the bottom of the code editor, there is a status message: "Done uploading." Below the status message, the terminal window displays memory usage information:

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

The status bar at the bottom shows the number "18".

g.

h. YOU'RE DONE!



i.

4. Encoder Wiring

- a. From left to right, the pins are GND, VCC, output
- b. The left encoder goes to digital pin 4
- c. The right encoder goes to digital pin 3



d.

5. Master Pin Guide

a.

Servo	9
-------	---

Motor Controller	5 7 8 11 12 6
Ultrasonic Distance	2 10
MPU6050	A5 A4
Encoder (left)	4
Encoder (right)	3