Product Assembly Laboratory Manual

Objective

In this laboratory a working robotic chassis will be assembled utilizing skills, components and techniques developed in the previous lab sessions. This prototype will consist of a integration of mechanical, electrical, and software systems before performing a complete troubleshoot of the system.

Apparatus and Equipment Overview

For this lab students will require most of the components created in the previous labs, as well as some additional components.

Lab-Constructed Components

- 2 x turned Delrin standoffs
- 1 x milled polycarbonate sensor mount

Supplied Components

- 1 x Chassis kit
    - 1 x Acrylic chassis
    - 1 x Rear caster wheel
    - 2 x Rubber wheel
    - 4 x Acrylic fasteners
    - 6 x M3*30mm screw
    - 4 x M3*8mm screw
    - 8 x M3*6mm flanged screw
    - 8 x M3 nut
    - 4 x M3*12mm hex standoff
    - 1 x AA battery holder
    - 4 x AA batteries
    - 1 x ON/OFF switch
    - 2 x DC (Direct Current) geared motors
    - 4 x 6in wire
- 1 x Electronic kit
    - 1 x Arduino Uno
    - 1 x USB A-B cable
    - 1 x L293D Motor Shield
    - 1 x Ultrasonic sensor
    - 1 x AT-09 BLE module
    - 8 x Female to female jumper wire

- 1 x 2 pin Male header

Tools

- Wire strippers
- Soldering iron (+ solder)
- Small screwdriver

Pre-Lab Preparation

Before arriving, review the wiring diagram for the electrical components involved. It may also be beneficial to review the Arduino sketches that will be uploaded during assembly and get an idea of how these programs work.

Review Questions

How is the microcontroller mounted to the chassis?

_____

_____

How are the wheels connected to the chassis?

_____

_____

What programming language does the car use to move?

_____

_____

What is the role of the ultrasonic sensor?

_____

_____

Why is it important to solder the wires?

_____

_____

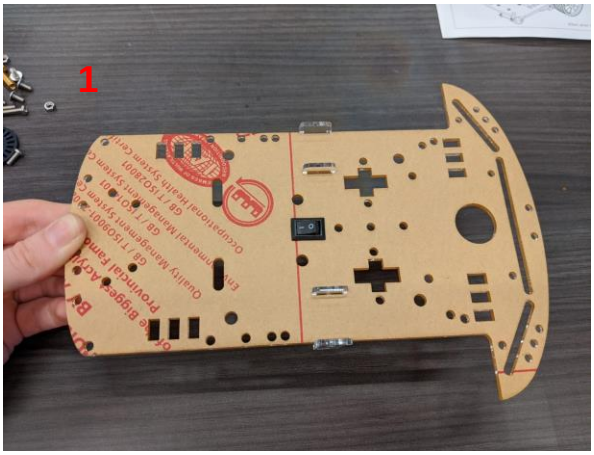
Procedure

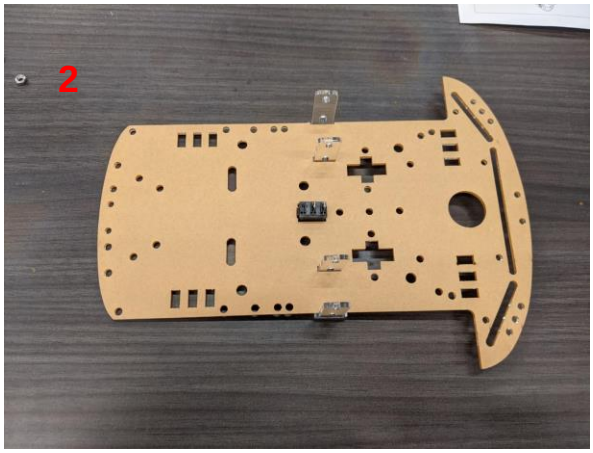
Sensitive electrical components like the control boards and sensors used in this lab can be damaged or destroyed from static discharge from mishandling during assembly and use. Before setting up this lab, students should ensure that their workstation is statically prepared (non-metal workstation, no carpet/other fibrous materials), and they have grounded themselves by touching a large metal object before handling the components. There are accessories to improve the assembly workstation for static resistance, such as an antistatic mat and static wrist/ankle straps, which should be used if available.

The following instructions are divided into 3 sections, hardware, electronics and software.

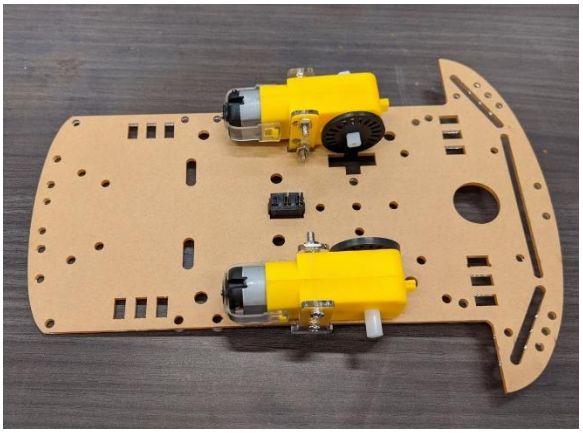
Part A – Chassis Assembly

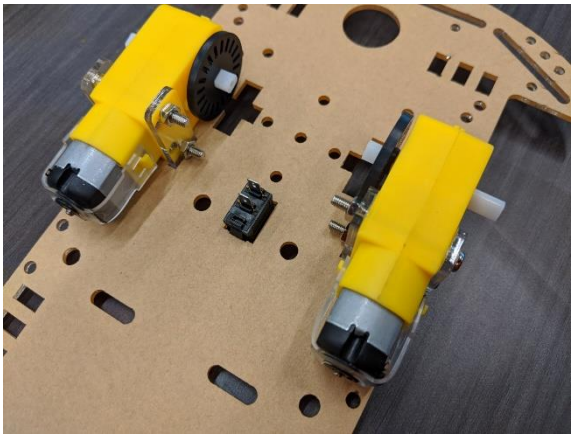
1. First **place** the switch and the 4 acrylic fasteners in the chassis like in the following pictures, do not worry if the side ones fall off right now. **Flip** the chassis upside down.



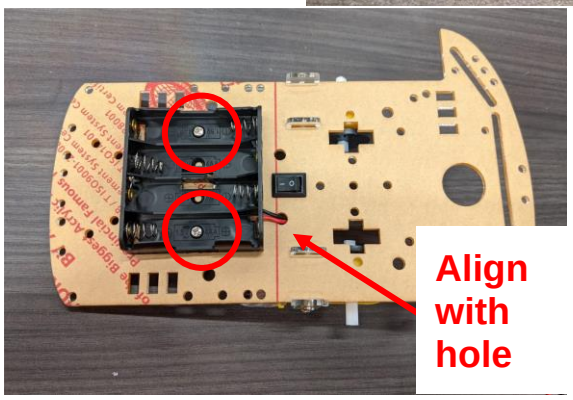


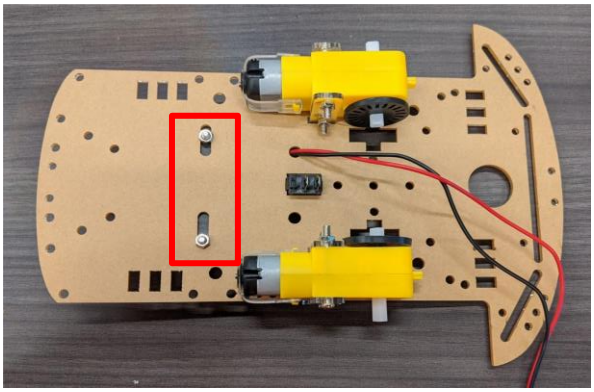
2. **Mount** both motors to the chassis using the longer M3 screws and nuts. With a screwdriver **tighten** the screws until the nut starts slipping in your fingers. You may want to tighten the top screws first. This will help stabilized the motor while tightening the bottom screws.



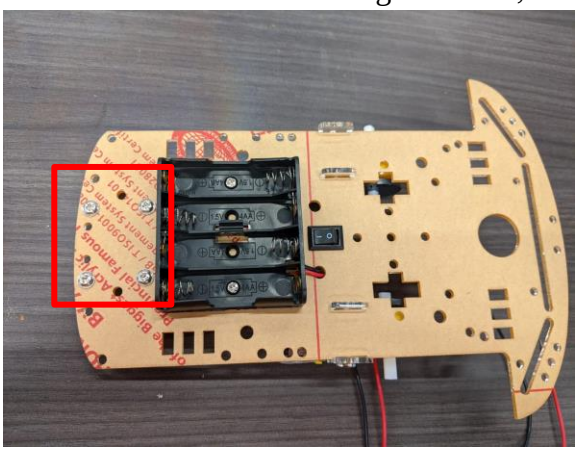


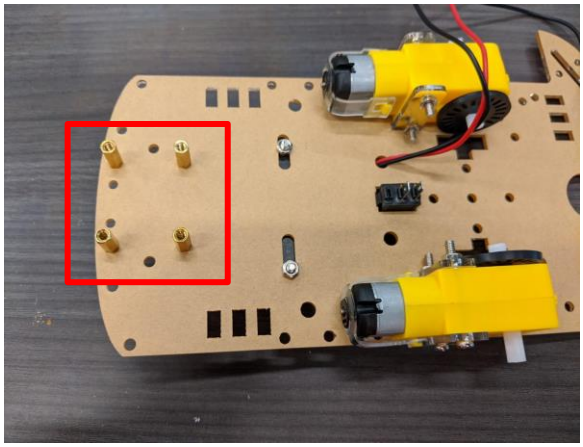
3. There are 2 types of small screws, a shorter one with a flange (M3*6mm) and a slightly longer one (M3*8mm). **Use** the longer one to assemble the battery holder with 2 nuts, use the screwdriver again to tighten. With the slots in the chassis, **align** the wires of the holder with a circular hole in the chassis.
    - If your kit did not have these screws, you can use tape or something else to secure the battery holder.



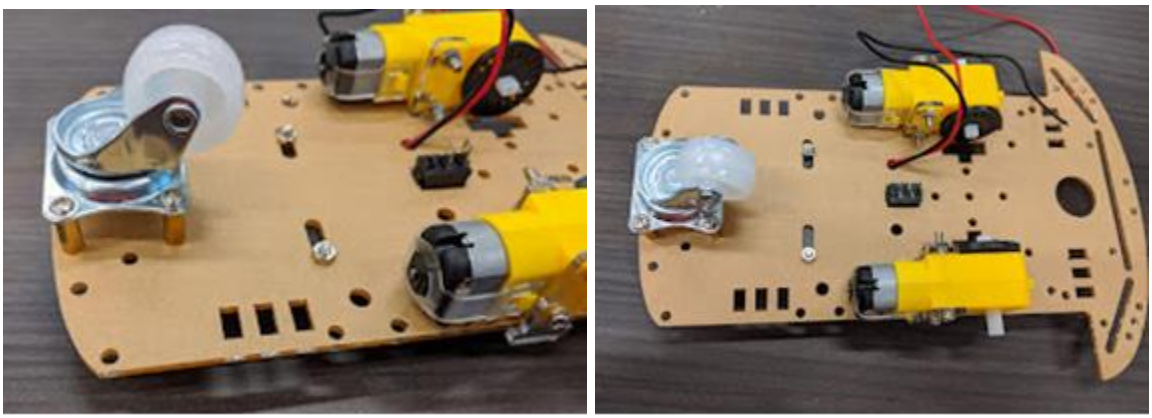


4. With 4 of the shorter flanged screws, **fix** the standoffs to the bottom of the chassis.

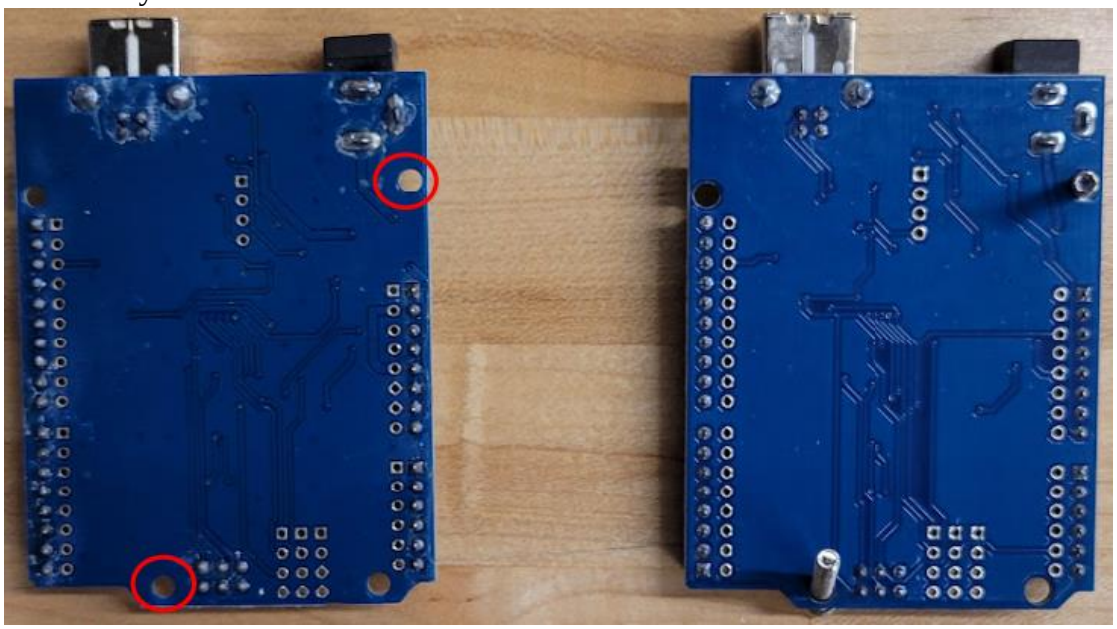




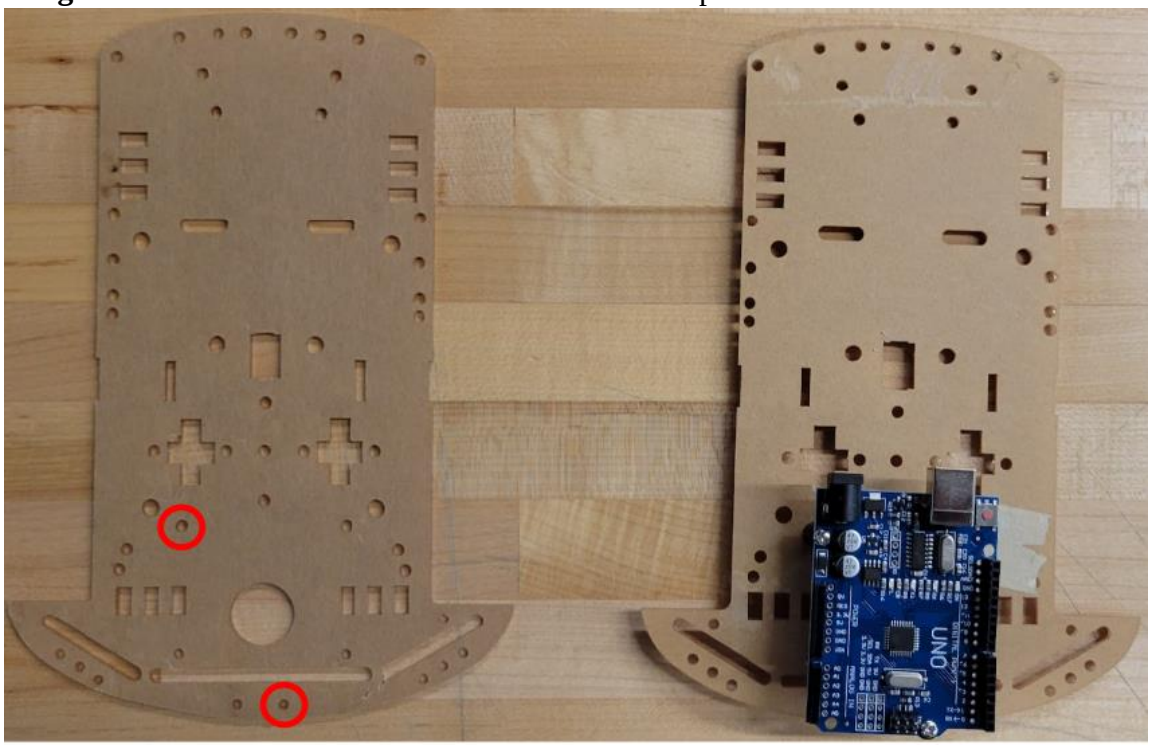
5. **Fasten** the caster to the standoffs with 4 other screws. **Tighten** with the screwdriver.



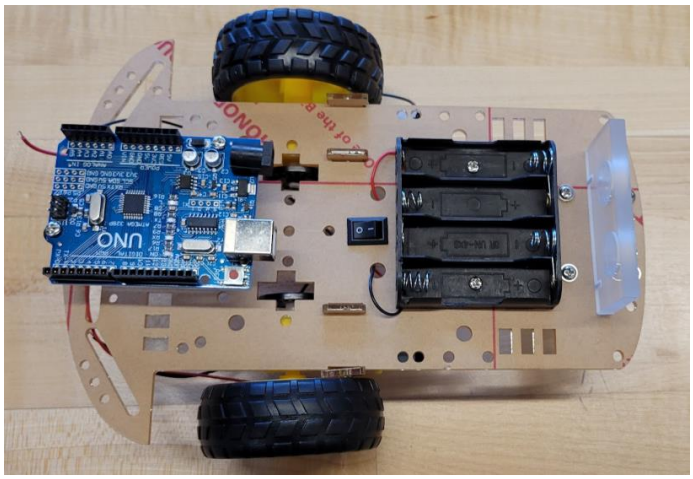
6. **Insert** 2 x M3*30mm bolts into the mounting holes on the Arduino microcontroller and slide the cylindrical standoffs onto the screws inserted into the Arduino.

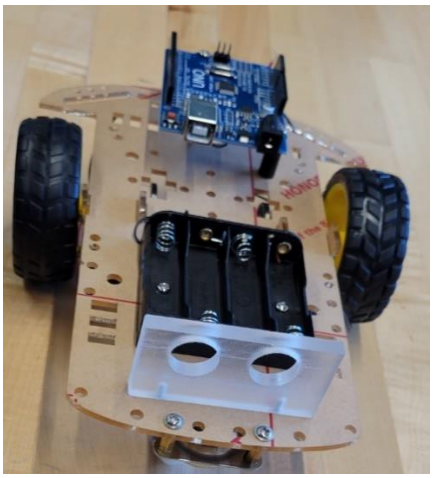


7. **Place** the Arduino onto the base using the screws in the locations shown in the next figure. **Tighten** the M3 nuts to fix the Arduino standoffs in place.



8. Using 2 M3x8mm bolt, **place** the polycarbonate sensor mount in the two middle holes of the chassis as shown in the picture below.

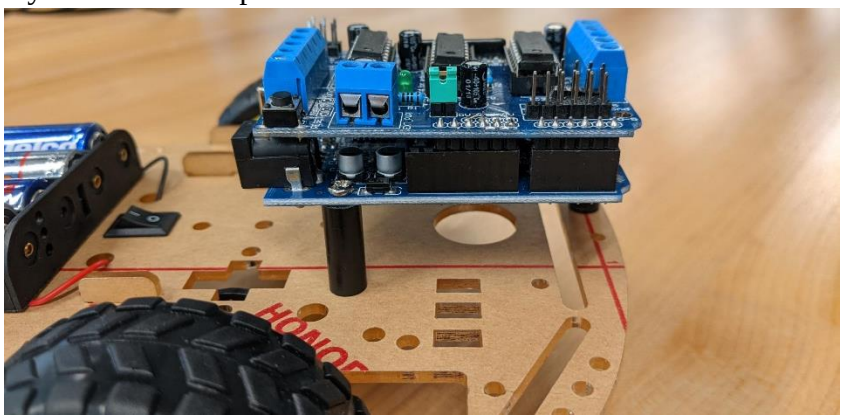




Part B – Electronic Control Components

1. **Solder** the 2-pin header into the analog pin A0-A1 of the motor shield if it's not already done.

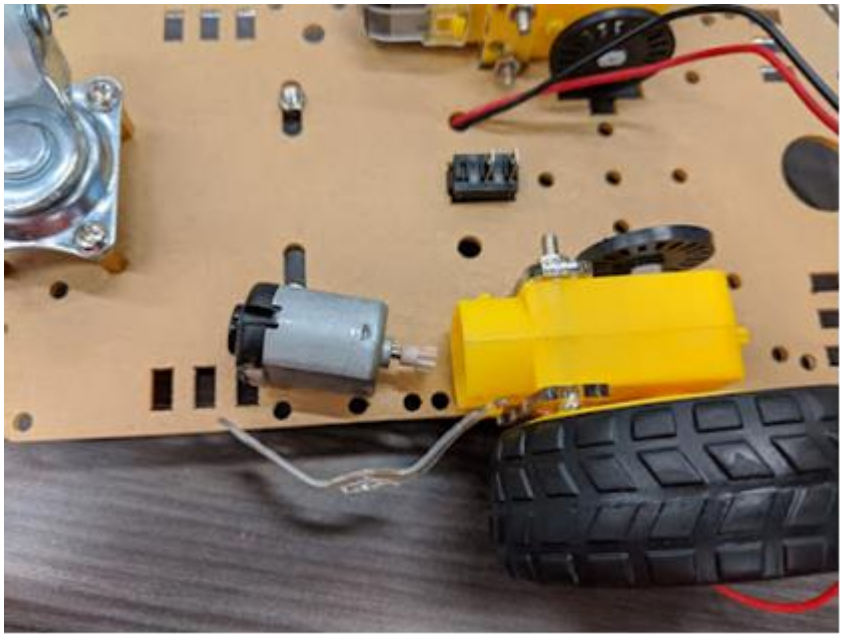
2. Next, **place** the motor shield on top of the Arduino, **aligning** the male pins from the shield farthest away from the USB port on the Arduino.



3. Using 4 female-female jumper wires, **connect** the AT-09 Bluetooth module to the motor shield by making the following connections:
   - AT-09 module GND to the SER1 (-) pin in the motor shield
   - AT-09 module VCC to the SER1 (+) pin in the motor shield
   - AT-09 module TXD to the SER1 (S) pin in the motor shield
   - AT-09 module RXD to the SER2 (S) pin in the motor shield
4. Using 4 female-female jumper wires, **connect** the ultrasonic sensor to the motor shield by making the following connections:

- Ultrasonic sensor VCC to the SER2 (+) pin in the motor shield
- Ultrasonic sensor Trig to the A0 pin in the motor shield
- Ultrasonic sensor Echo to the A1 pin in the motor shield
- Ultrasonic sensor GND to the SER2 (-) pin in the motor shield

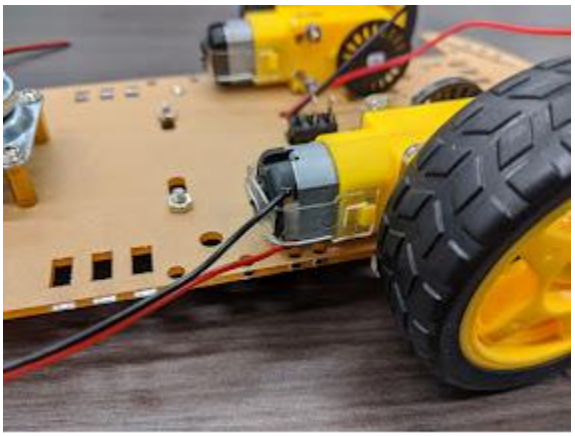
5. **Place** the sensor inside of the sensor mount.
6. **Unclip** the transparent plastic cover to release the motors.

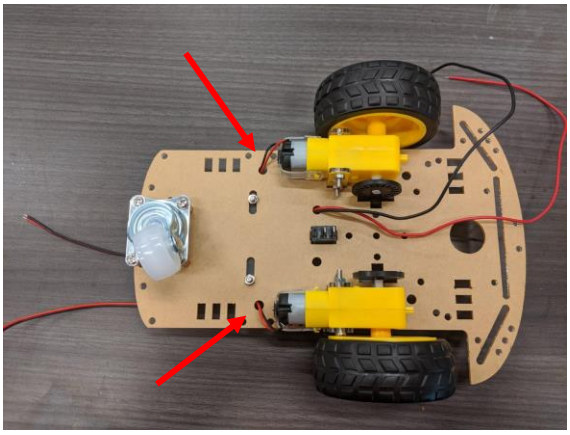


7. **Bend** the ends of 2 wires **to hook** them in the DC motor terminals and with a small amount of solder, **fix** them in place. The color of the wires does not matter here. **Be careful since the small terminals are fragile and can be easily ripped off.

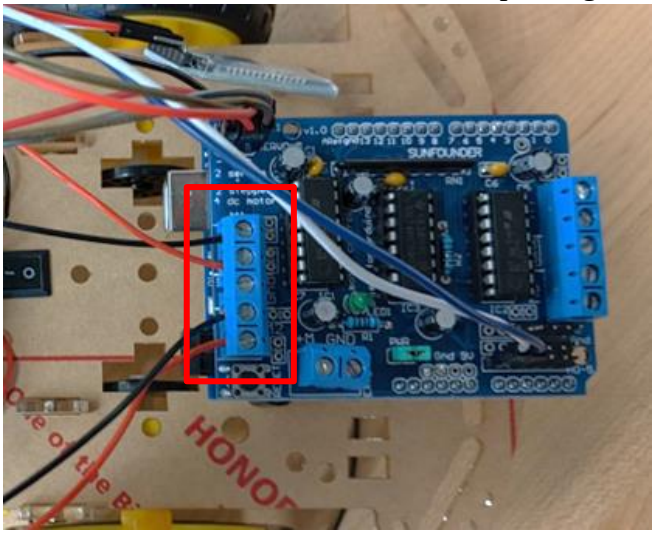
   - Tape can be a suitable non-permanent alternative.
   - You may receive a kit with wire already soldered, check that the connection is good and fix it if necessary.

8. **Place** both motors back in their casing and **clip** the transparent plastic back in place. **Feed** the wires through a circular hole in the chassis.

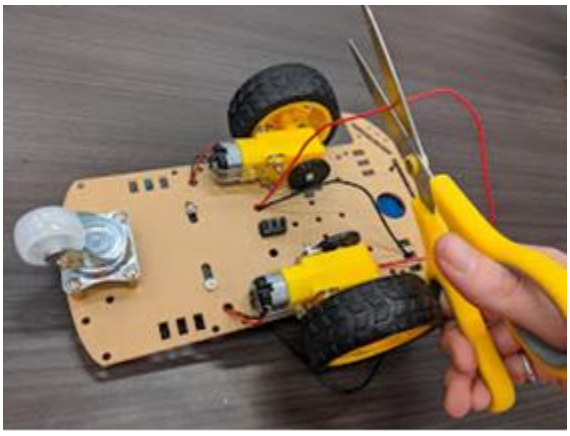
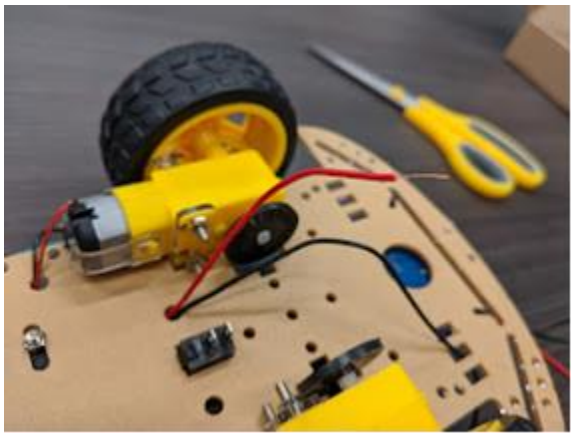




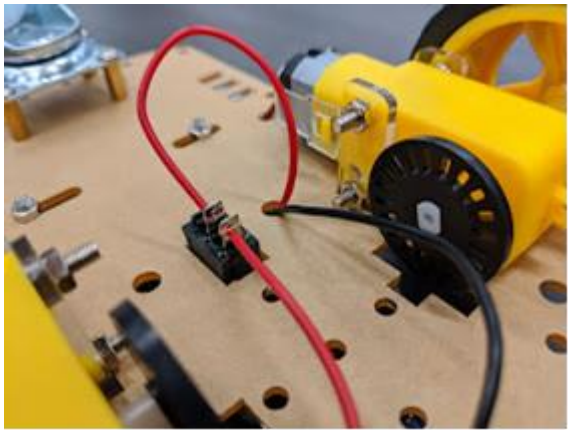
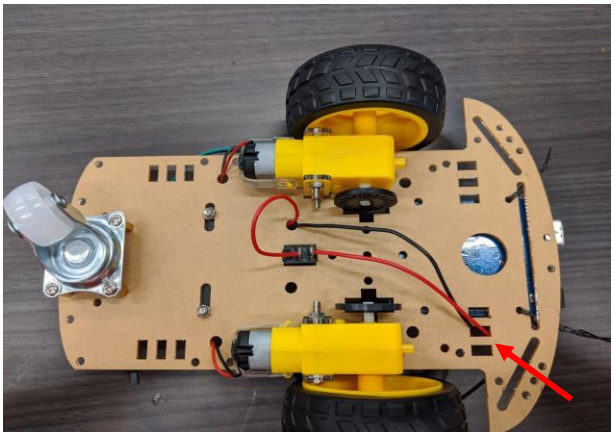
9. **Take** the wires from the DC motors and **put** them in the M1 and M2 terminal blocks in the shield and with a small flathead or Phillips screwdriver **screw in** the top to lock them in place. A pointy knife, tweezers or the flat end of nail clippers could also work for this.
    - Note that orientation of the wires will matter so you will have to change them later when you test your code to make sure the wheels are spinning in the right direction.



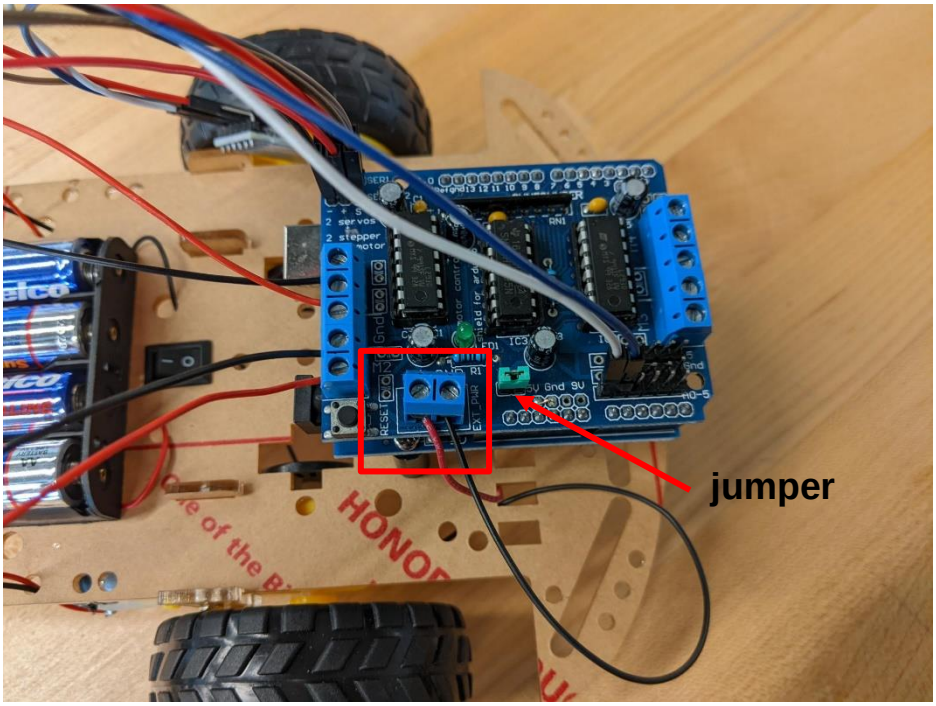
10. You will now **prepare** the on/off switch. You will interrupt the power coming from the battery pack so **cut** the **red** wire from the batteries with scissors about 2.5-3in from the chassis (this may already be done for you). With the scissors, **gently cut** the plastic isolation from the cable to reveal the copper stranded wire (Here is an example on how to strip the wire with scissors). Twist the copper strands together so that they are not a mess. **Do** the same with the other side of the cut wire.

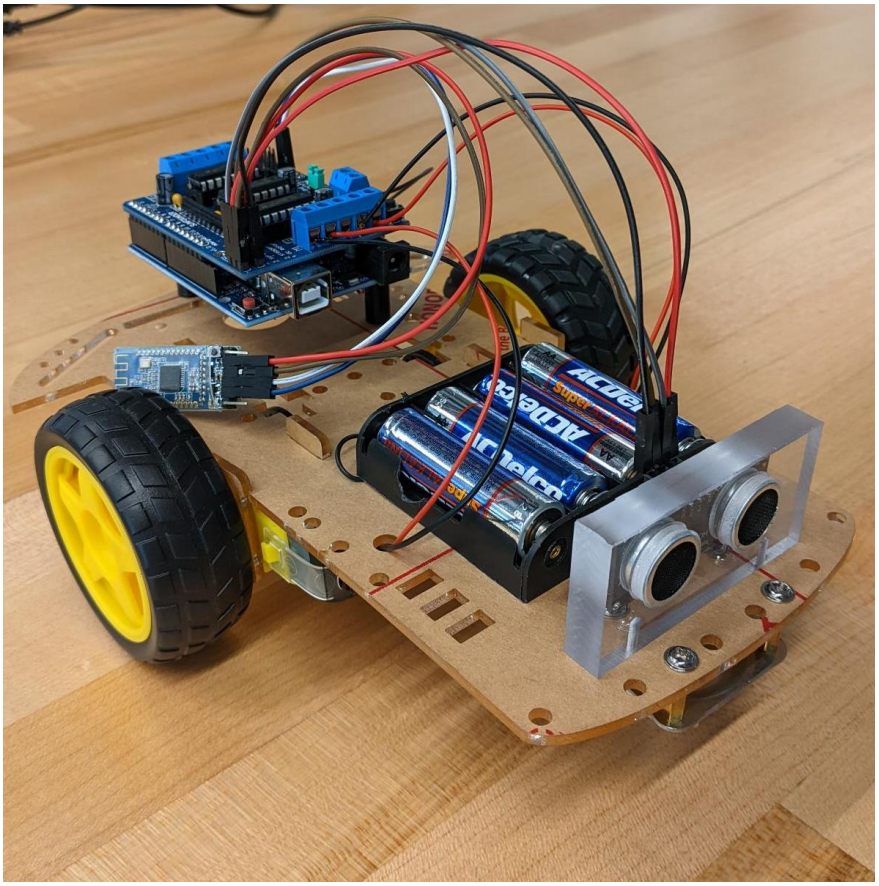
11. **Take** both stripped wires and **wrap** them around each terminal of the switch. **Solder** them in place. Then **take** both the **red** and **black** wires from the battery pack and **feed** them through a square hole in the chassis on the left by the top of the Arduino.

12. **Take** the battery wires and **screw** them into the power port on the motor shield. +M is positive (**red**) and GND is negative (**black**). **Ensure** the black (or yellow or green) jumper is plugged in on the motor shield.

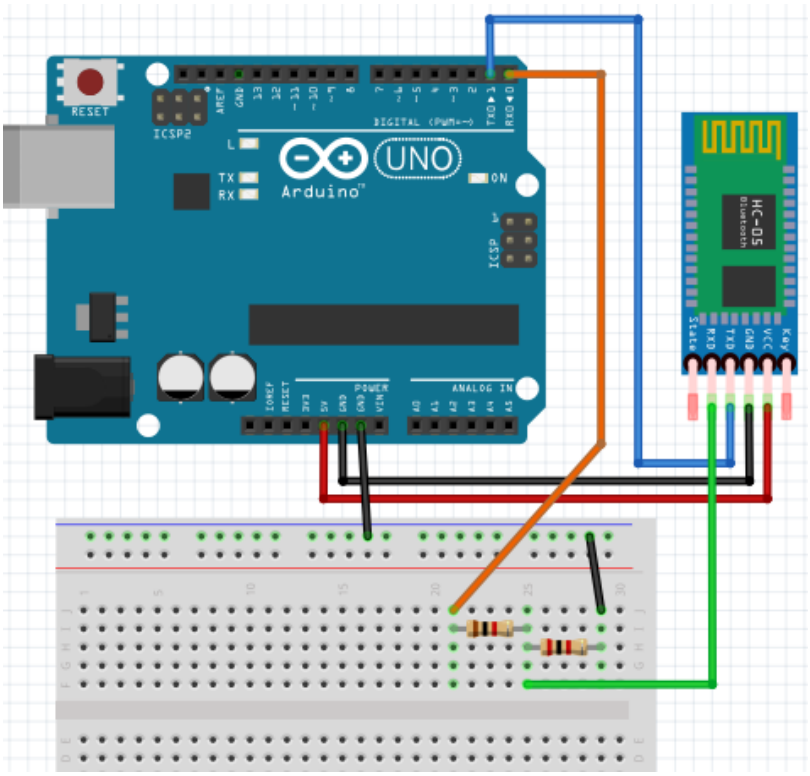


Here is the complete chariot:

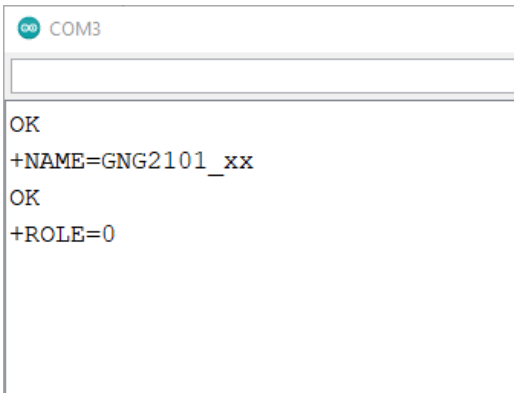


Part C – Software

1. **Remove** the motor shield from the Arduino and disconnect the Bluetooth sensor.
2. Program the control board **by connecting** the Arduino to the computer via USB and **launch** the Arduino IDE software. **Open** a blank sketch in Arduino by going to File→New. **Ensure** that the correct board and serial port are selected and upload it to the board.
    - Make sure nothing is connected to the Arduino when you do this.
3. You must configure the Bluetooth sensor before being able to use it with your chariot. **Connect** the Bluetooth sensor by following the diagram below (1 kΩ and 2 kΩ resistors).



4. **Open** the serial monitor and make sure the baud rate is ‘9600’ and the line ending is ‘Both NL and CR’. **Type** ‘AT’ and press enter. You should get the response ‘OK’ in the monitor.
    - If you get an error press the restart button on the Arduino and see if it fixes it or unplug the sensor and plug it back in.
    - Make sure the RXD and TXD pins are wired correctly.
5. **Configure** the name of the sensor by typing ‘AT+NAMEGNG2101_xx’ but replace xx with your group number and press enter.
6. **Type** ‘AT+ROLE’ and press enter to see the current role configuration. If the role is not 0 (slave), set it to 0 by typing ‘AT+ROLE0’. Your serial monitor should look like the following picture.



Note: Other AT commands exist and may be used to evaluate other input/output information from the system. You can type AT+HELP to see the full list of commands.

7. Now **upload** the Arduino sketch created in a previous lab for the chariot control to the board. Put the shield back on the Arduino and reconnect the Bluetooth sensor to the chariot.
8. **Test** the car by powering everything up (**insert** batteries and **press** the switch) and connect to the BLE module using the mobile app (Simple BLE Joystick). **Test** to make sure the motors spin correctly and in the right direction, and that the sensor functions as expected. If this is not the case, backtrack and try to discover where the error(s) in cart are. **Show** the TA your finished product.
    - Android app: https://apkcombo.com/simple-ble-joystick/ru.experementy.simpleblejoystick/

Questions

What could you do to improve the functionality of the car and controller?

_____

_____

Could this design be simplified/altered to be more easily manufactured?

_____

_____

Notes

If you notice your car does not go straight when instructed to, it is because the motors are not turning exactly at the same speed because of manufacturing defects. To fix this you can

change the speed of one of the motors in the Arduino code to equalize the physical speeds, the value can be between 0-255.

```
void setup() {
  leftmotor.setSpeed(200);
  rightmotor.setSpeed(200);

  // put your setup code here, to run once:
  mySerial.begin(9600);
  Serial.begin(9600);
```