

TA0136
USER MANUAL
ARDUINO 2 WHEEL
DRIVE ULTRASONIC
ROBOT KIT



Contents

Chapter 1. Overview – TA0136	1
Chapter 2. Getting started: the 2 Wheel Drive Ultrasonic Robot Kit using Arduino UNO	1
2.1. What is Arduino?	1
2.2. What is IDUINO UNO?.....	1
Chapter 3. Software installation	2
3.1. Arduino Software/IDE	2
3.2. Play with your first “Hello World” LED example.....	3
3.3. Run your Arduino 2 Wheel Drive code	4
Chapter 4. Hardware installation	5
4.1. Unboxing and Component list	5
4.2. Chassis Frame Installation	6
4.3. Arduino Installation	6 -12
4.4. Sensor installation	12 - 17
4.5. Switch	17 - 22
4.6. Wire Connection	22 - 24
Chapter 5. Have fun	24 - 26
Appendix	27 - 30

Chapter 1. Overview – TA0136

In this instruction, we will introduce you through the fun project of the Arduino 2 Wheel Drive Ultrasonic Robot Kit. Get your Arduino board kit. Let's get started!

Chapter 2. Getting started: the 2 Wheel Drive Ultrasonic Robot Kit using Arduino UNO

2.1. What is Arduino?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards can read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

2.2. What is IDUINO UNO?



The Iduino Uno is on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analogue inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Chapter 3. Software installation

In this section, we will introduce you the development platform where you translate creative mind into codes and let it fly.

3.1. Arduino Software/IDE

Download from [here](#). Open Windows-based app by double clicking it and follow the instruction to complete(Remember to install everything driver for Arduino). Easy!

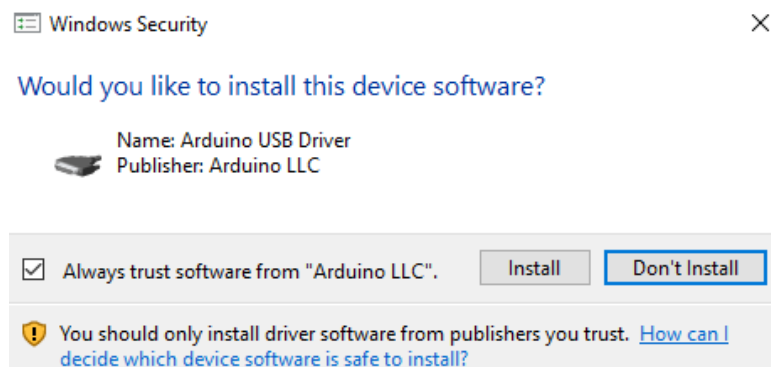


Figure 1 Installation of drivers

Connecting your UNO board with your computer

Connecting UNO and your PC by a blue USB cable, and if connected correctly you will see the green power LED light up and another orange LED is blinking.

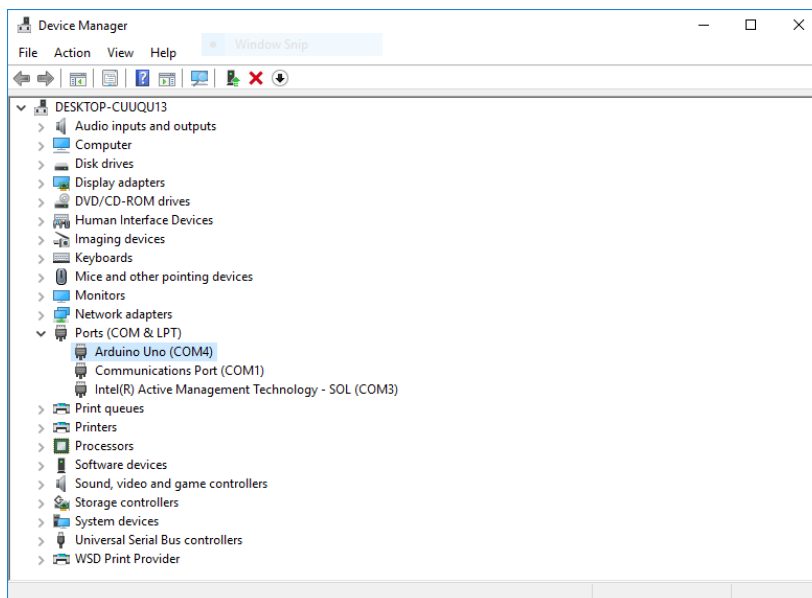


Figure 2 Check Your special COM and note it down the number

Find your Serial COM number and note it down.

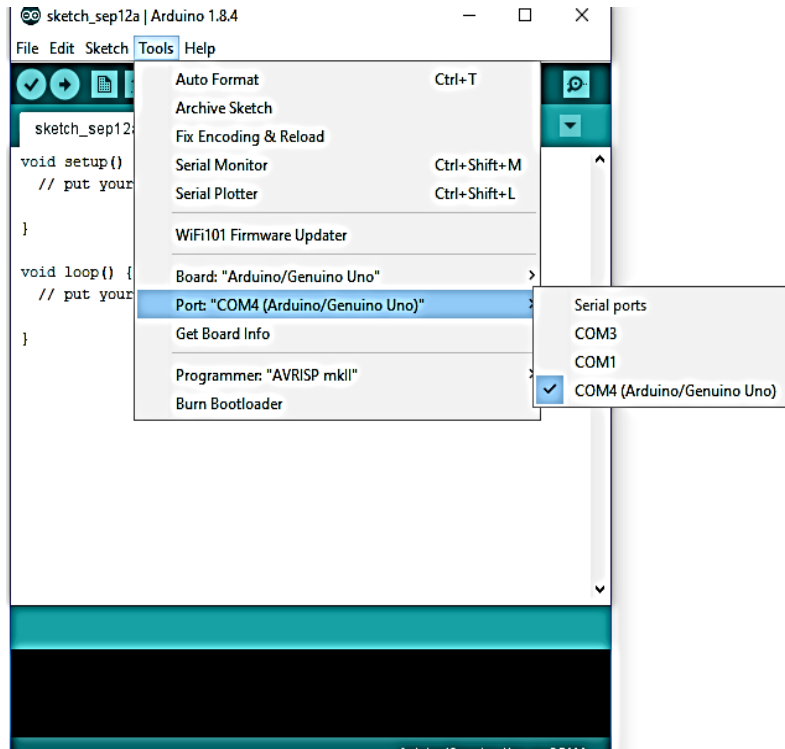
We need to figure out which channel COM is currently communicating between PC and UNO. Following the path: Control panel | Hardware and Sound | Devices and Printers | Device Manager | Ports (COM & LPT) | Arduino UNO (COM_x)

Chapter 3 Software installation

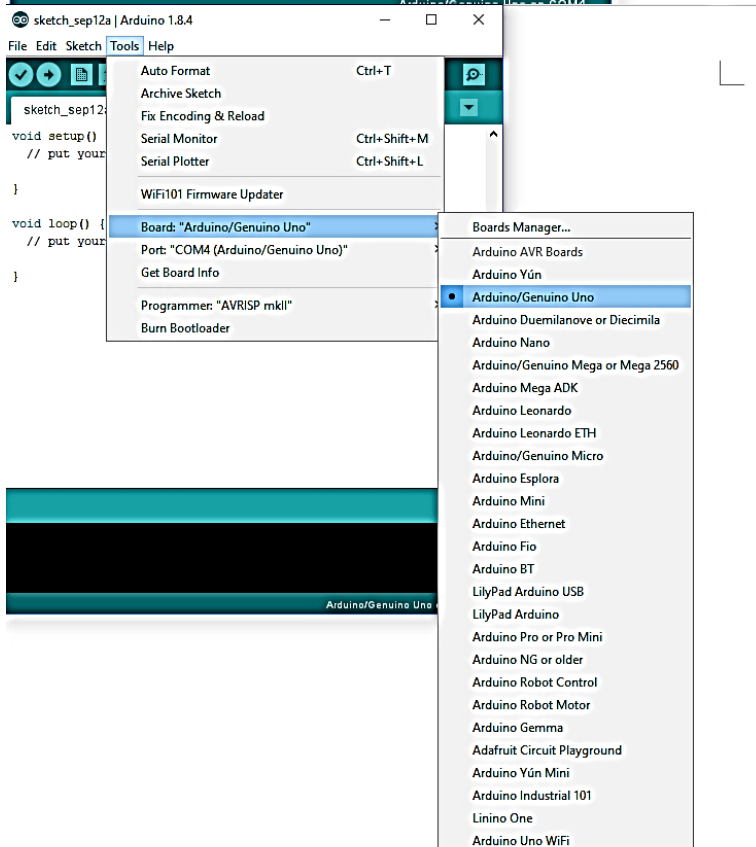
Note down the COM number as we require this later. As the COM port may vary from time to time, this step is vital. In this case for demonstration purpose, we are using the COM 4.

3.2. Play with your first “Hello World” LED example

Firstly, let's tell IDE where to find our Arduino port and which board you are currently using: The following instruction (Figure 3 and 4) shows the details:



Configuration of Ports

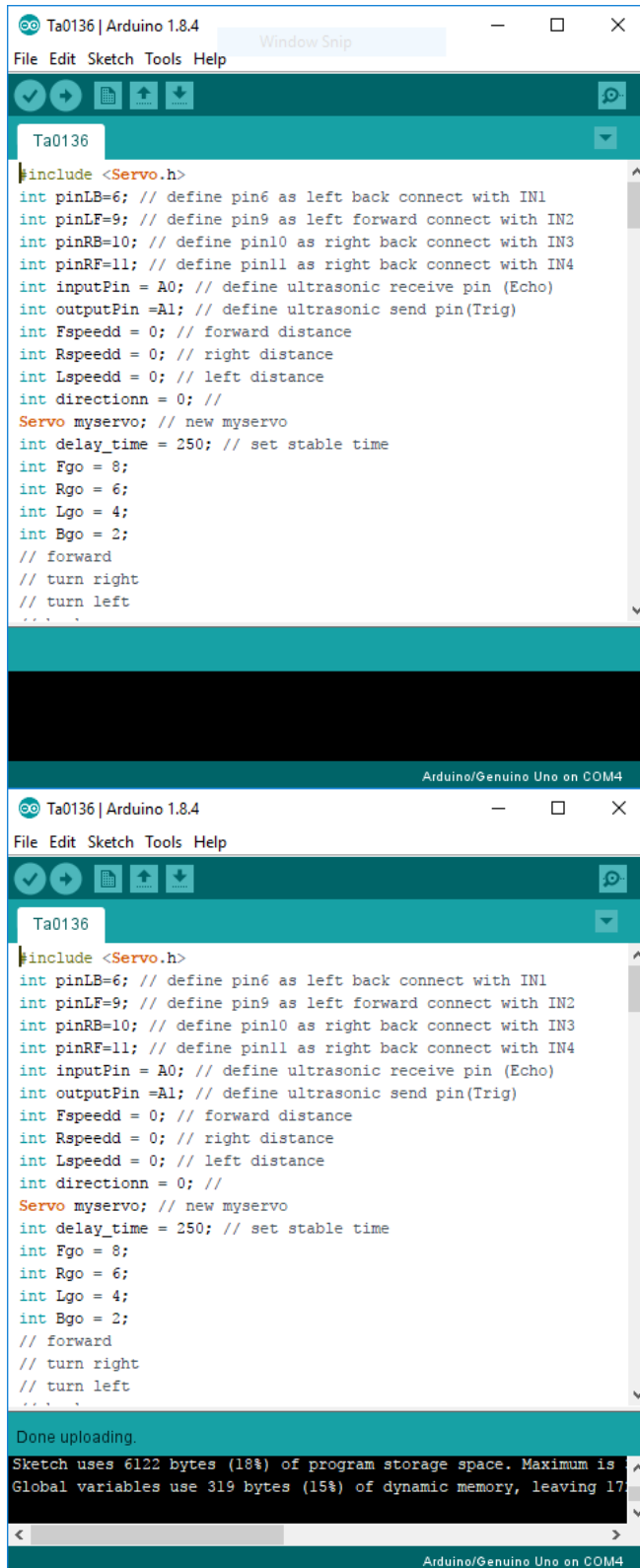


Configuration of the Board

Chapter 3 Software installation

It's time to play with your first simple example. Following the path by File | Examples | 01. Basics | Blink. A new code window would pop up, press the arrow symbol to upload. You will notice the orange LED is blinking almost every second.

3.3. Run your Arduino 2 Wheel Drive code



The image shows two screenshots of the Arduino IDE interface. The top screenshot shows the code editor with the following code:

```
#include <Servo.h>
int pinLB=6; // define pin6 as left back connect with IN1
int pinLF=9; // define pin9 as left forward connect with IN2
int pinRB=10; // define pin10 as right back connect with IN3
int pinRF=11; // define pin11 as right back connect with IN4
int inputPin = A0; // define ultrasonic receive pin (Echo)
int outputPin = A1; // define ultrasonic send pin (Trig)
int Fspeedd = 0; // forward distance
int Rspeedd = 0; // right distance
int Lspeedd = 0; // left distance
int directionn = 0; //
Servo myservo; // new myservo
int delay_time = 250; // set stable time
int Fgo = 8;
int Rgo = 6;
int Lgo = 4;
int Bgo = 2;
// forward
// turn right
// turn left
...
```

The bottom screenshot shows the same code editor with a status bar at the bottom that reads "Done uploading." Below the code editor, a message box displays the following information:

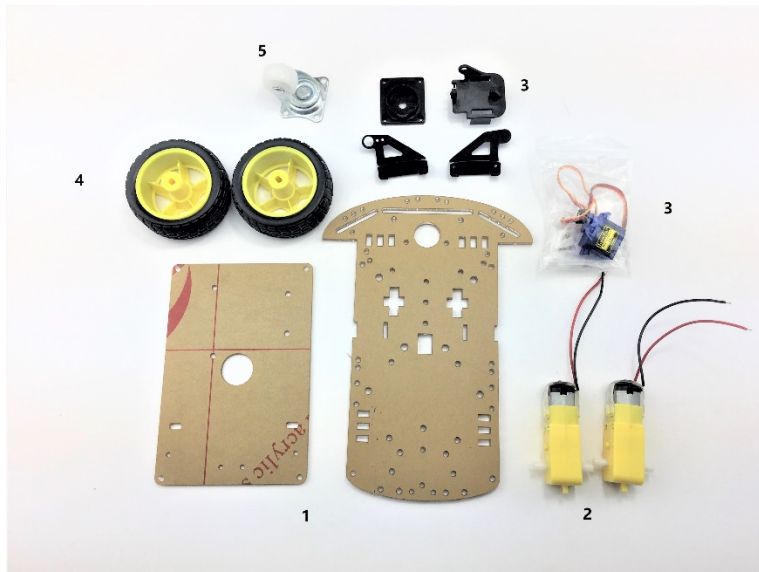
```
Sketch uses 6122 bytes (18%) of program storage space. Maximum is 32768 bytes.
Global variables use 319 bytes (15%) of dynamic memory, leaving 17579 bytes free.
```

Upload to UNO

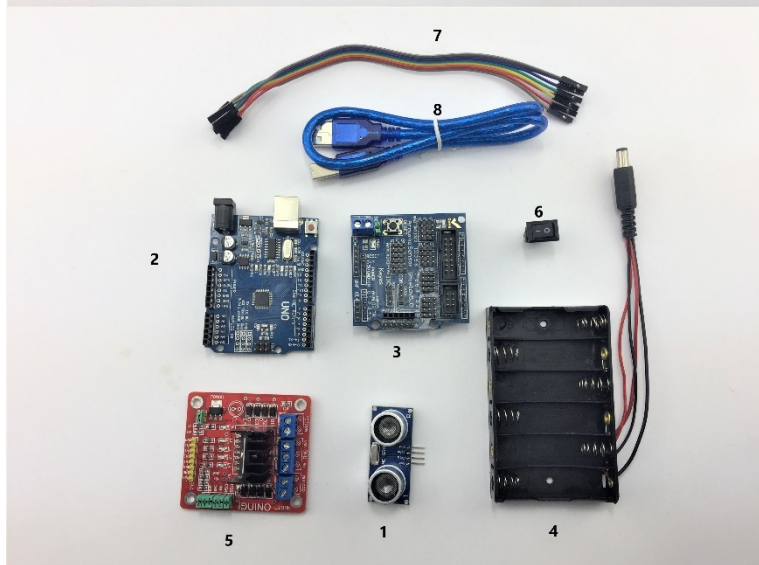
Done uploading!

Chapter 4. Hardware installation

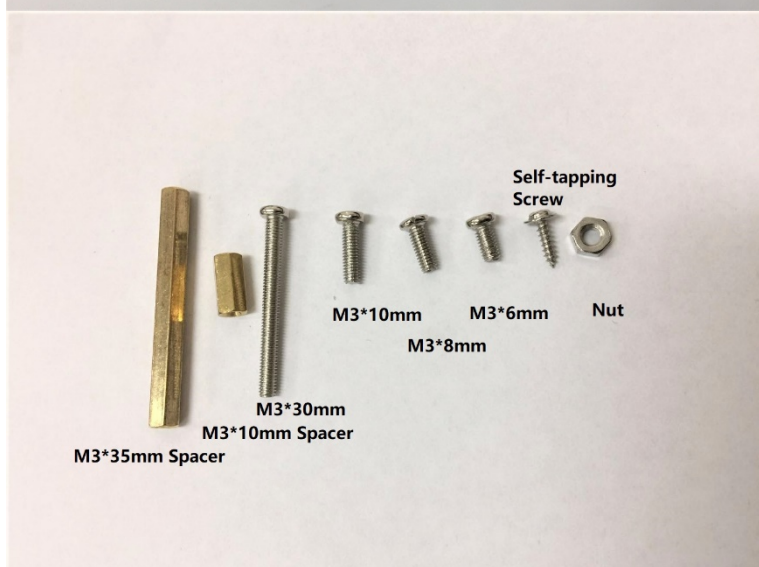
4.1. Unboxing and Component list



1. Acrylic Chassis
2. DC Motors
3. SG90 Servo with horns and Brackets
4. Rubber Wheels
5. Metal Pivot wheel



1. Ultrasonic Sensor
2. Arduino UNO Development Board
3. Arduino Sensor shield V5.0
4. 6 x AA Battery Box (Batteries not included)
5. L298N Motor Driver Board
6. Rocker Switch (Optional configuration please see Chapter 4.4)
7. Jumper Cable with 12 wires



Fastener package:

1. 4 x M3 30mm Screws
2. 4 x M3 10mm Screws
3. 4 x M3 8mm Screws
4. 8 x M3 6mm Screws
5. 4 x M3 35mm Spacers
6. 4 x M3 10mm Spacers
7. Self-tapping Screws
8. 8 x Nuts

4.2. Chassis Frame Installation



Remove the protective cover and prepare the items:

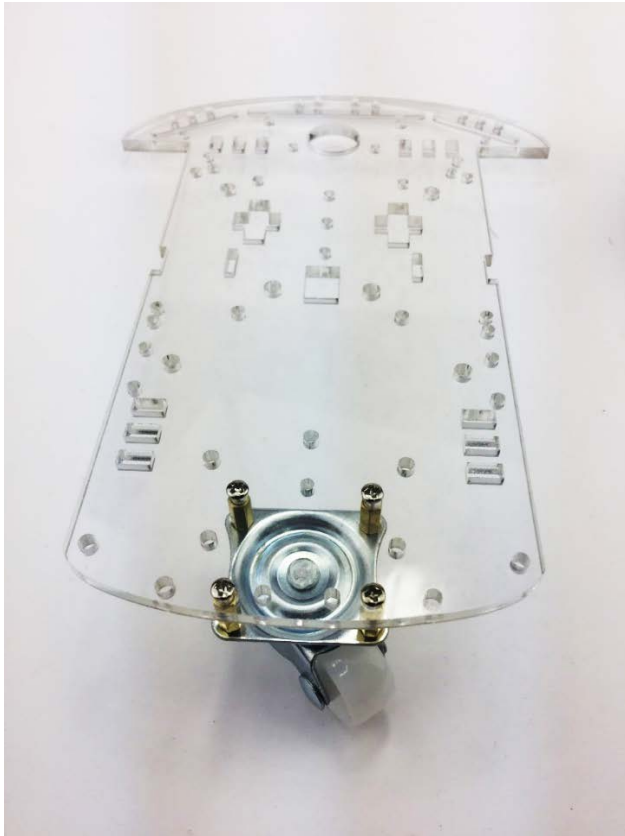
1. 4 x M3 6mm Screws
2. 4 x M3 10mm Spacers
3. 4 x M3 6mm screws
4. Metal Pivot Wheel



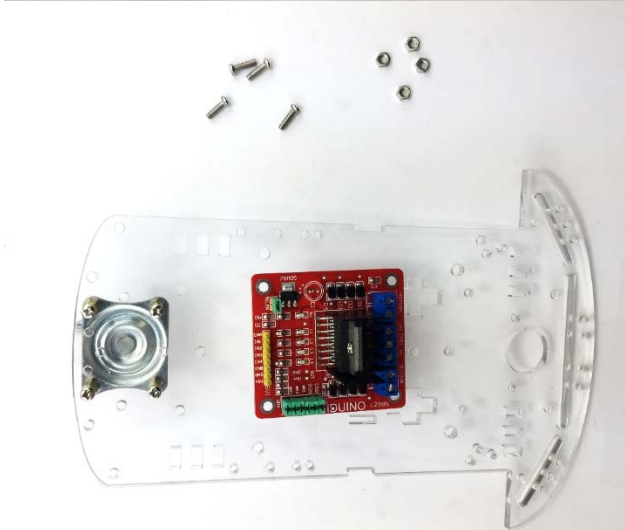
Assemble the M3*10 spacers and M3*6 screws onto the Metal Pivot Wheel



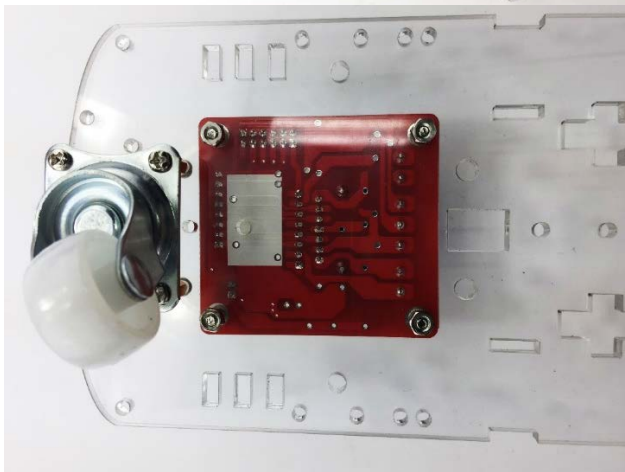
Chapter 4
Hardware installation



Screw in the metal pivot wheel to the chassis

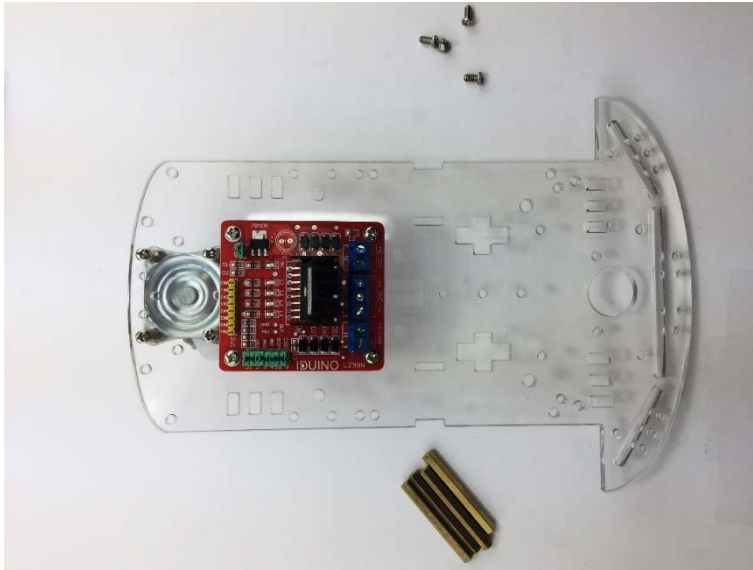


Gather 4 x M3 10mm screws and nuts to secure L289N Motor Drive board

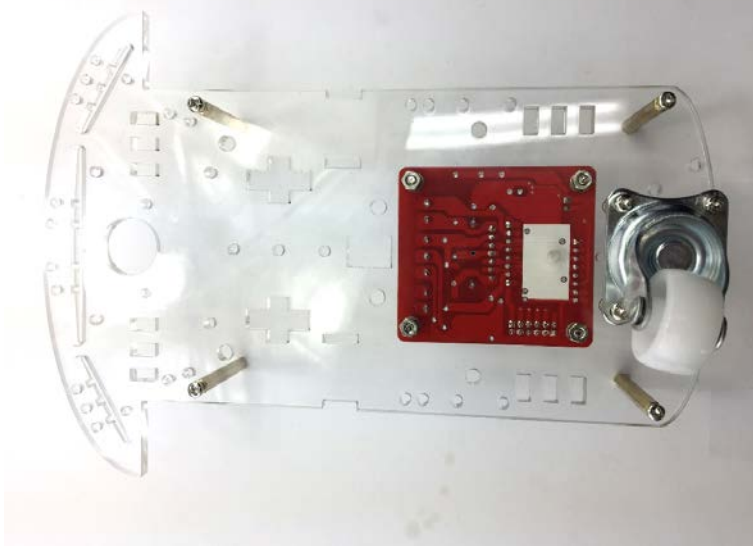


Screw the L289N Motor Drive board from the bottom chassis

Chapter 4
Hardware installation



Gather 4 x M3 8mm screws and 4 x M3 35mm spacers

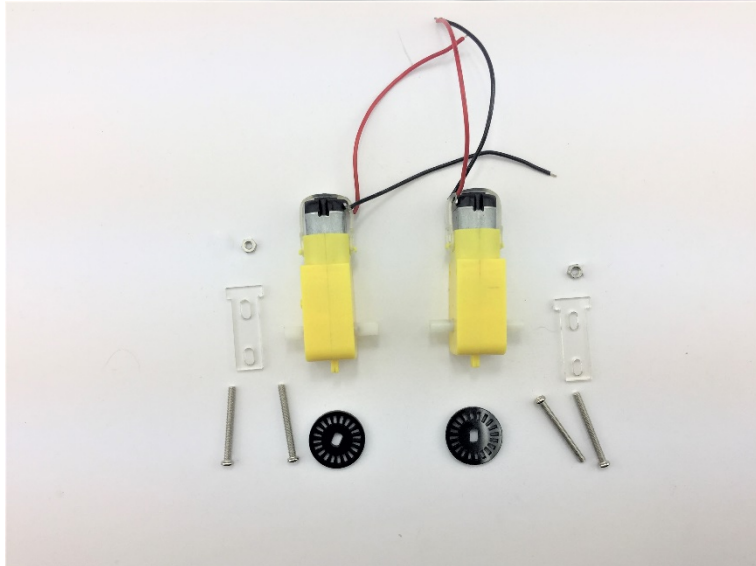
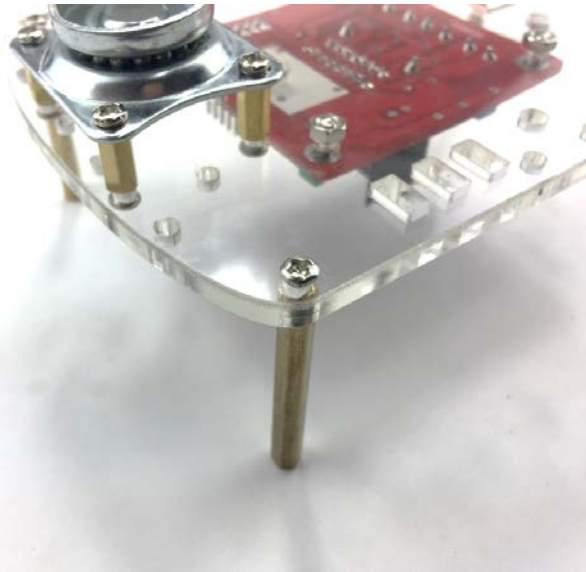


Secure Spacers onto the chassis as per the picture on the left.



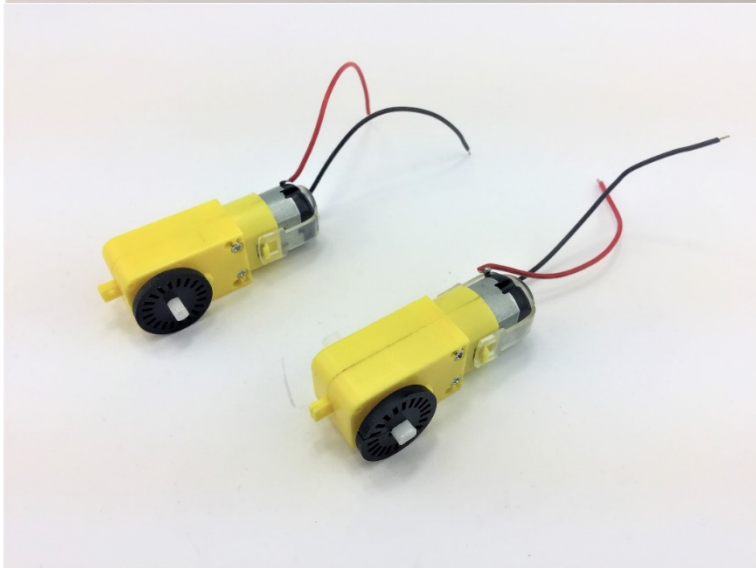
Spacers/Stand-offs should look like this

Chapter 4
Hardware installation



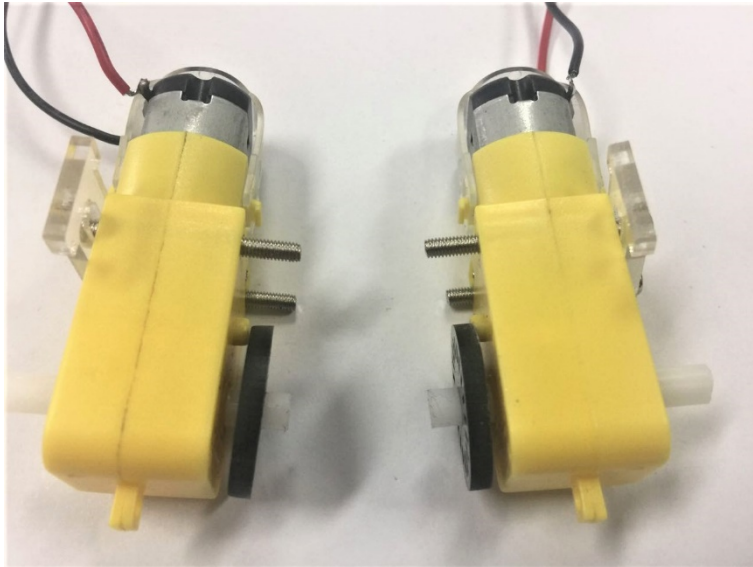
Gather the following components:

1. 2 x Acrylic Motor Brackets
2. 4 x M3 35mm Screws
3. 2 x Nuts

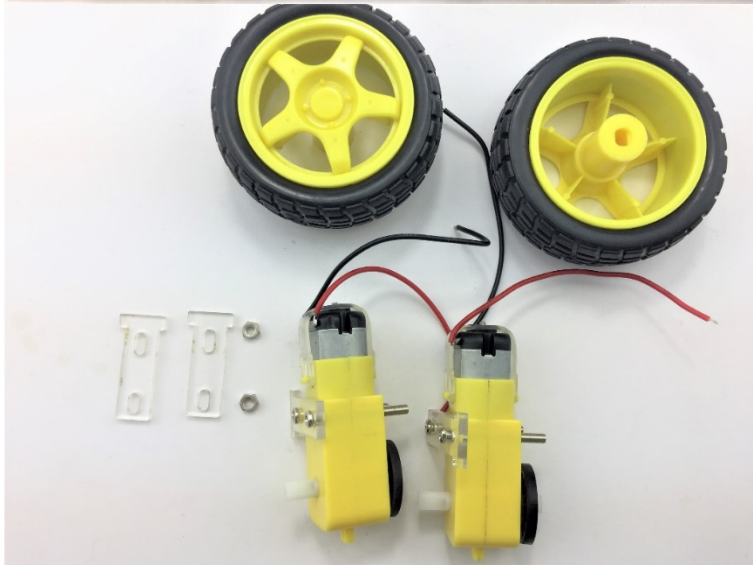


*Attaching wheel encoders are optional. These are not required for this particular project

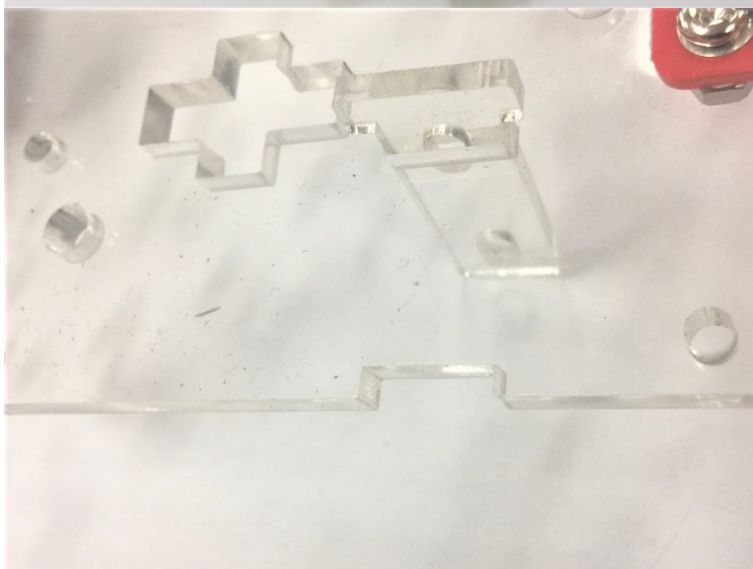
Chapter 4
Hardware installation



Place acrylic DC motor Brackets on both side of the motors as shown on the left

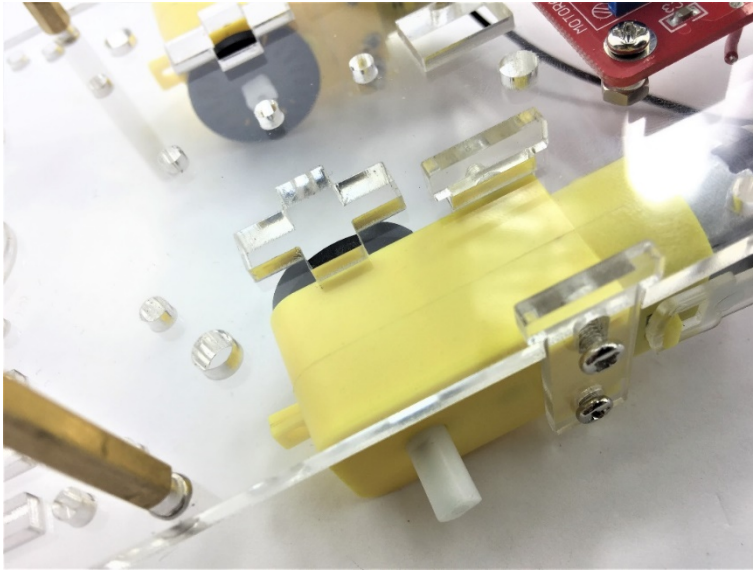


Gather another two acrylic fasteners and two wheels

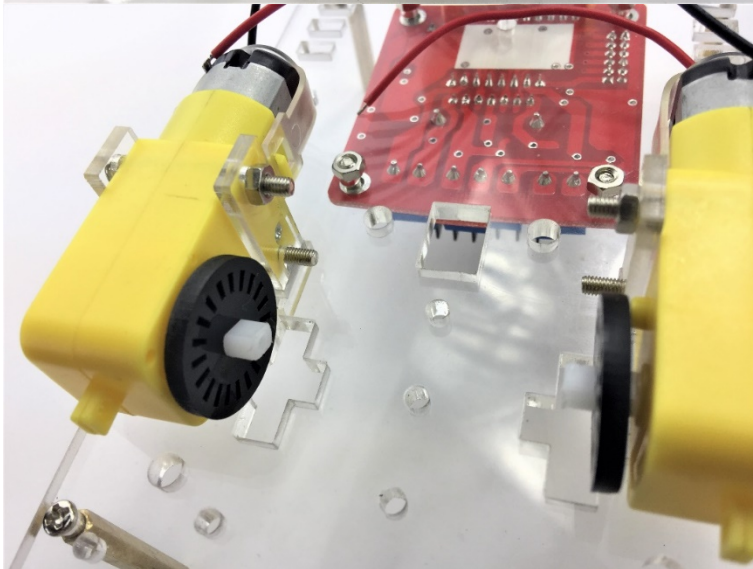


Insert the acrylic fasteners first in pre-cut slot

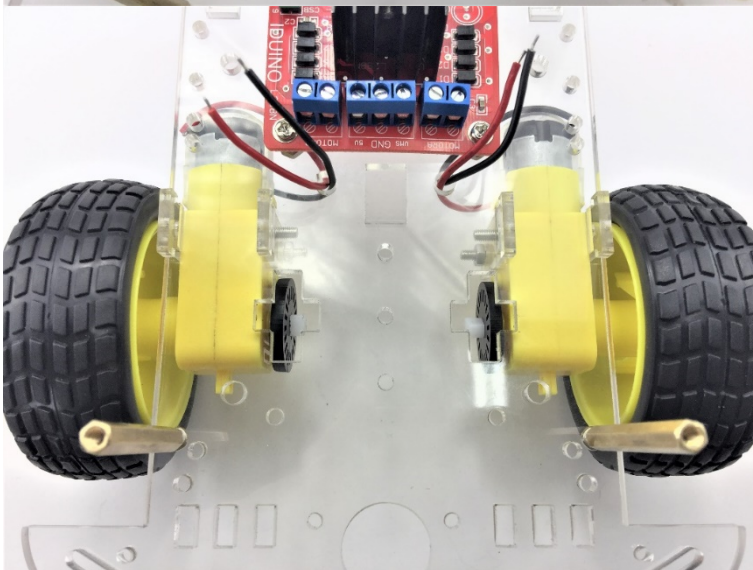
Chapter 4
Hardware installation



Then tighten and secure the DC motor with one nut on the other side



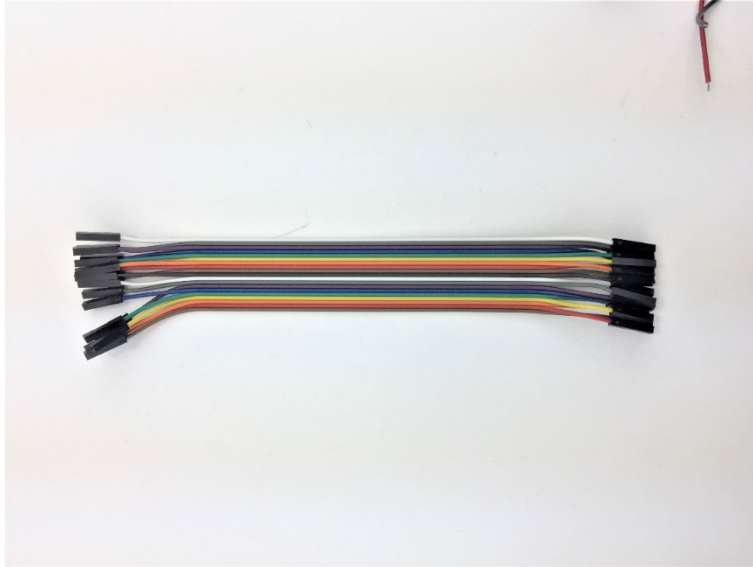
Fix the other motor as well



Pull the wire through as we need to connect them to the L289N board

4.3. Arduino Installation

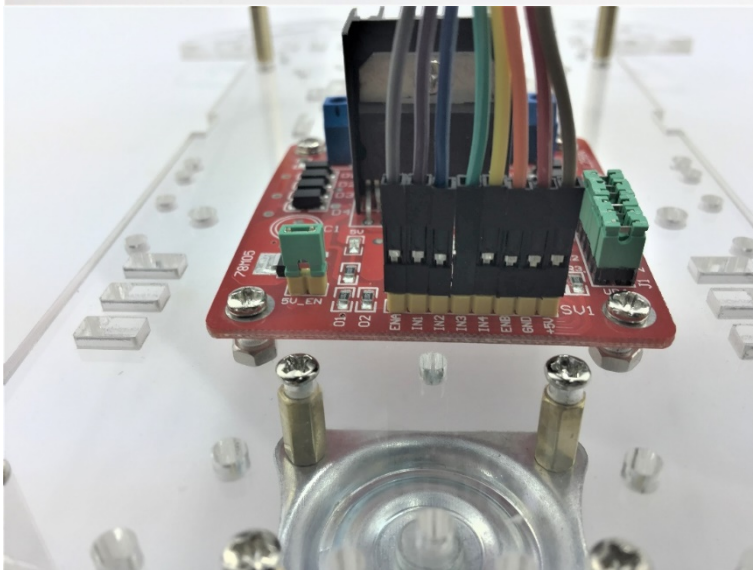
Let's fix the Arduino UNO and Sensor shield in the following steps.



Prepare the jumper cables



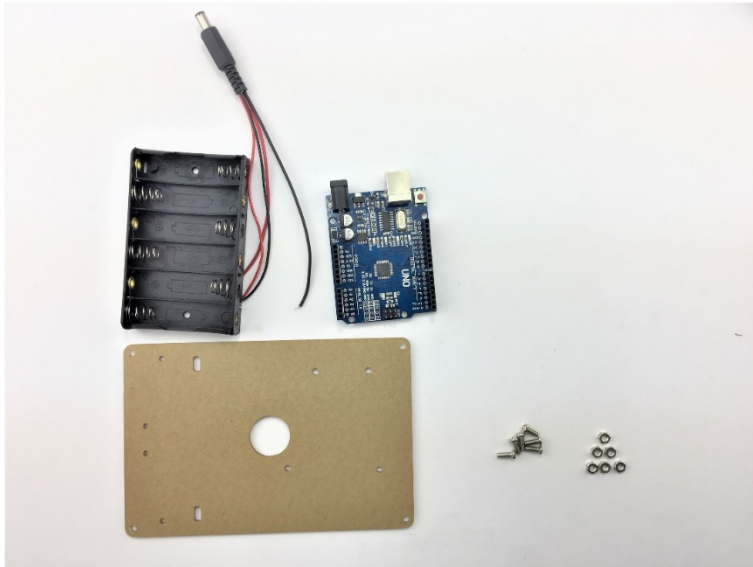
Separate the jumper cable set into four and eight configurations



Connect the 8 jumpers with L289N board as shown.

Please Note: The 8 Jumper cables will eventually attach to the Sensor Expansion Board from the L289N board a few steps from now. Please leave the opposite side free for the following steps.

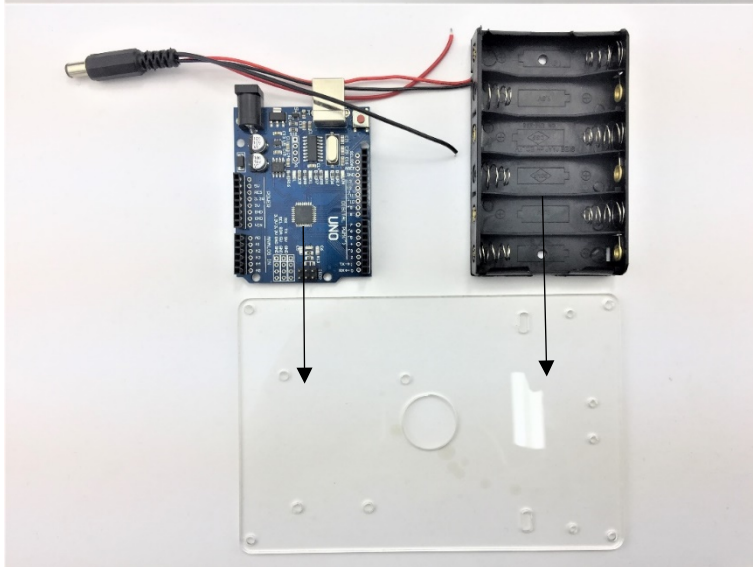
Chapter 4
Hardware installation



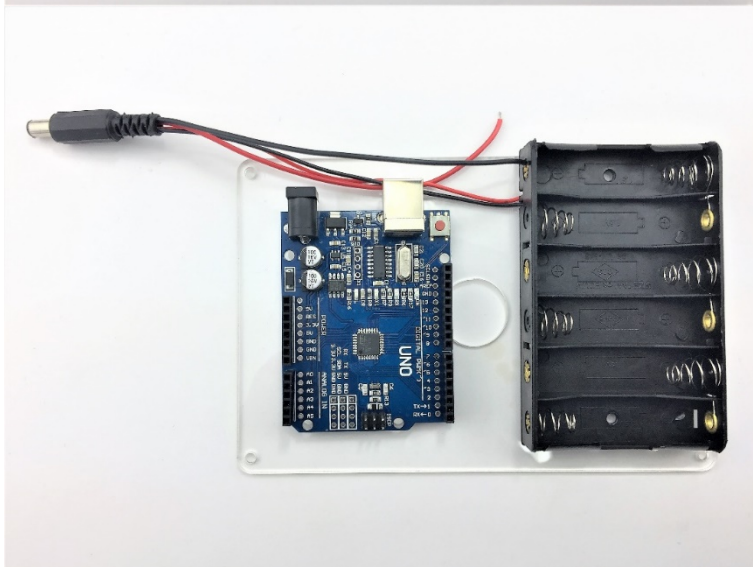
Prepare the following:

Battery pack:

- Arduino UNO development board
- Top acrylic chassis
- 6 x nuts
- 6 x M3 10mm screws

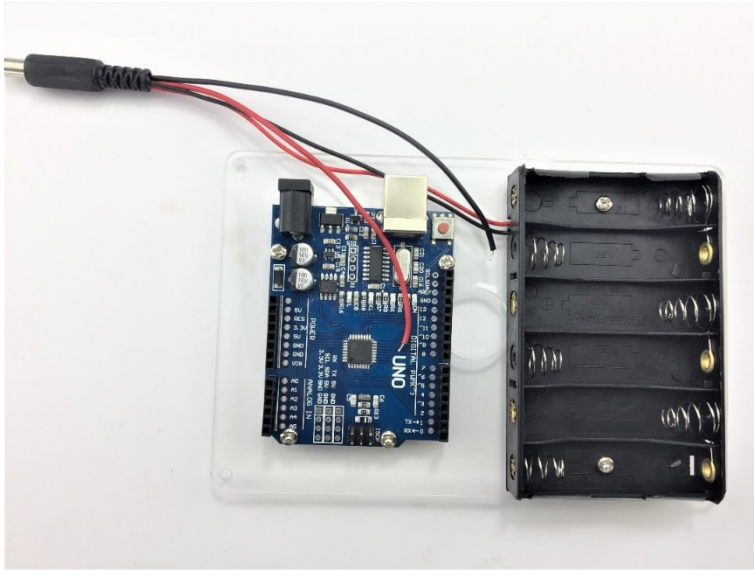


Peel the protective cover

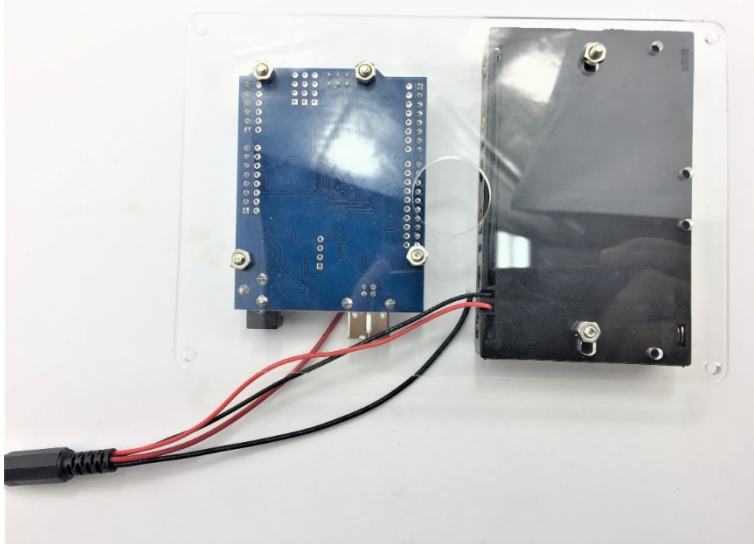


Place the battery pack and UNO board on the top acrylic chassis

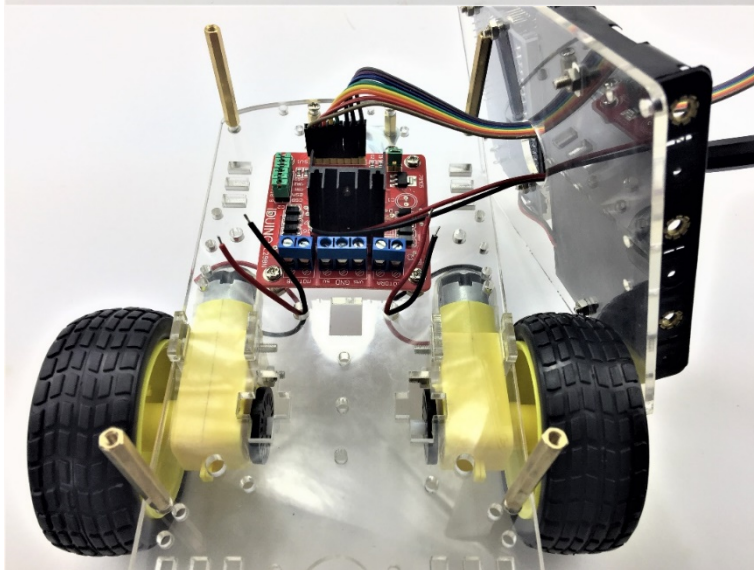
Chapter 4
Hardware installation



Secure the screws from the bottom

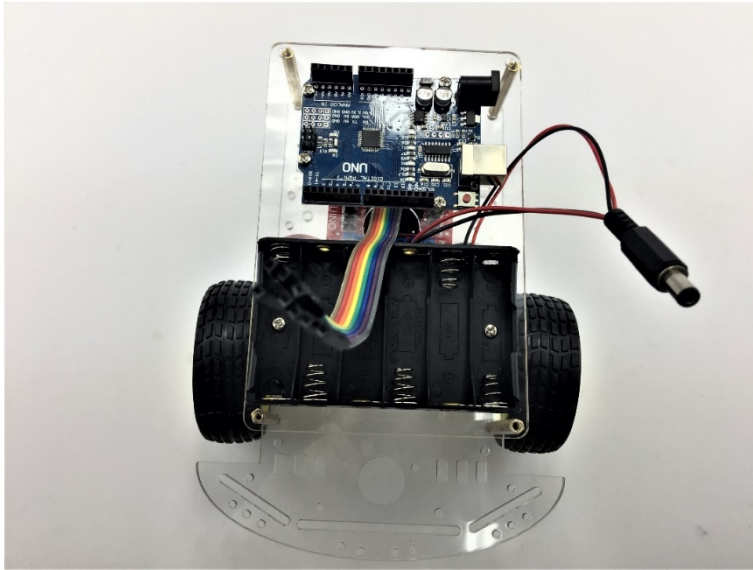


It should look like this

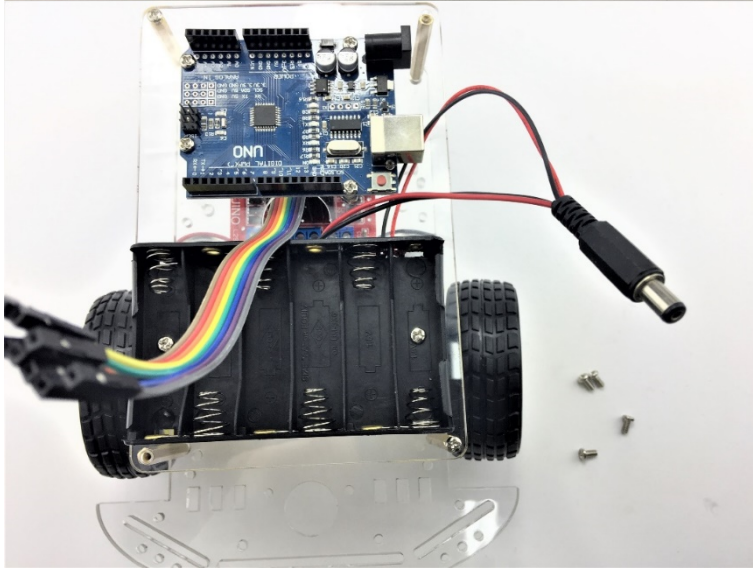


Feed the jumper wire from the L289N board and pull through to the top acrylic chassis hole to connect with the UNO board later

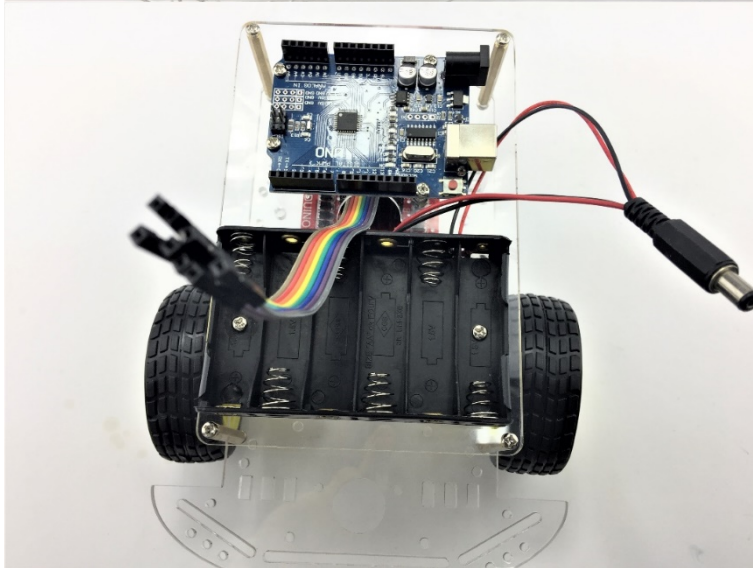
Chapter 4
Hardware installation



Mount the top acrylic chassis to the bottom chassis

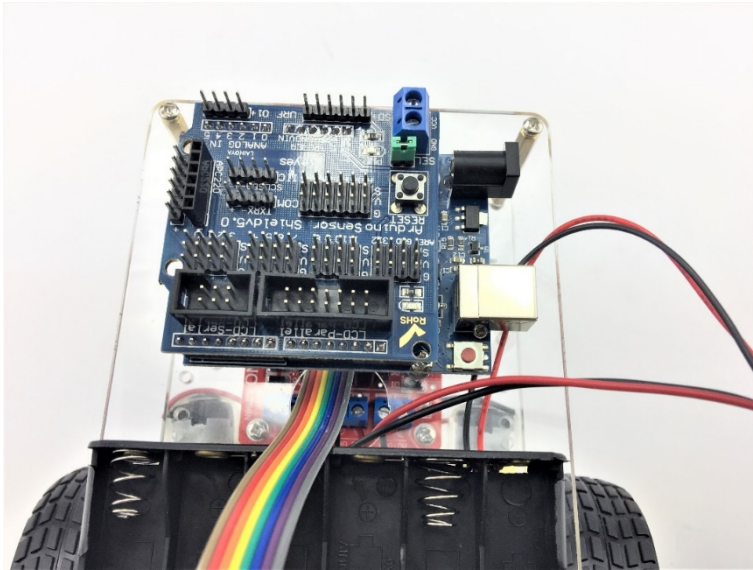


Use four M3*6 screws to secure top acrylic chassis.



It should look like this

Chapter 4
Hardware installation



Place the sensor shield on the top of the UNO board and ensure the pins line up with the Uno Board.

4.4. Sensor installation

Ultrasonic sensor acts as the eye of this 2-wheel drive car. In the following steps, we are going to guide you through the installation of ultrasonic sensor and servo.



Items required in this step

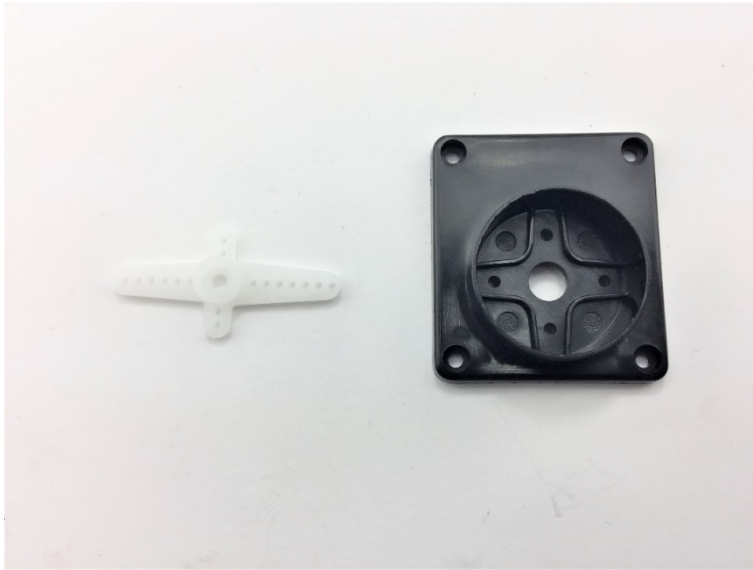


Assemble the FPV holder and servo



Use two self-tapping screws (in the FPV package) to tighten

Chapter 4
Hardware installation

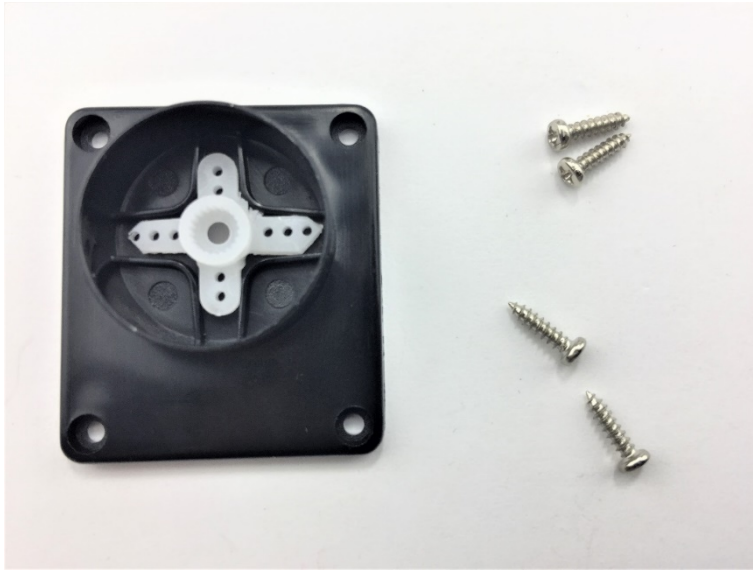


Prepare the items



To fit the servo horn into the black holder we need to modify as pictured. Best tool to do this with is a side cutter
Caution: Be carefully with cutter and sharp edges of the horn

Chapter 4
Hardware installation



Place the modified servo horn inside the bracket and secure with 6 * M2.5 screws in the servo package



It should look like this



To tighten the ultrasonic sensor, we can use several cable ties (Not included in the Kit). Alternatively, you can use other methods:

- Use rubber bands
- Thread or Plastic Strings

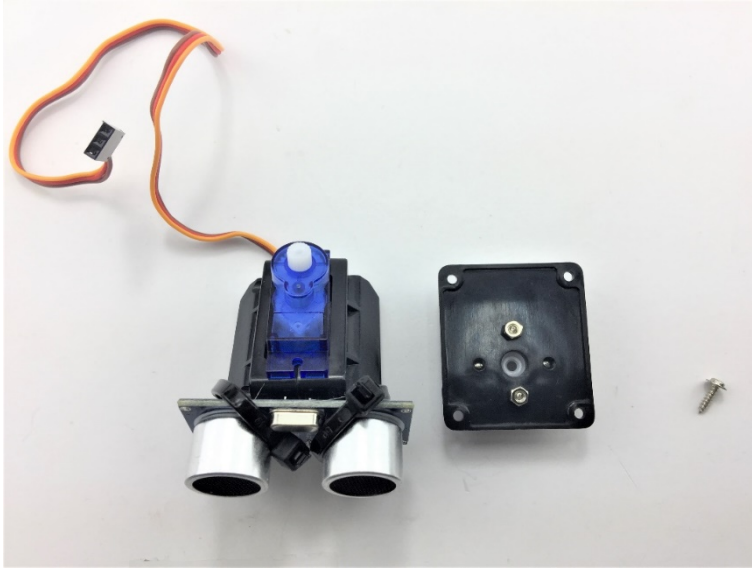
Chapter 4
Hardware installation



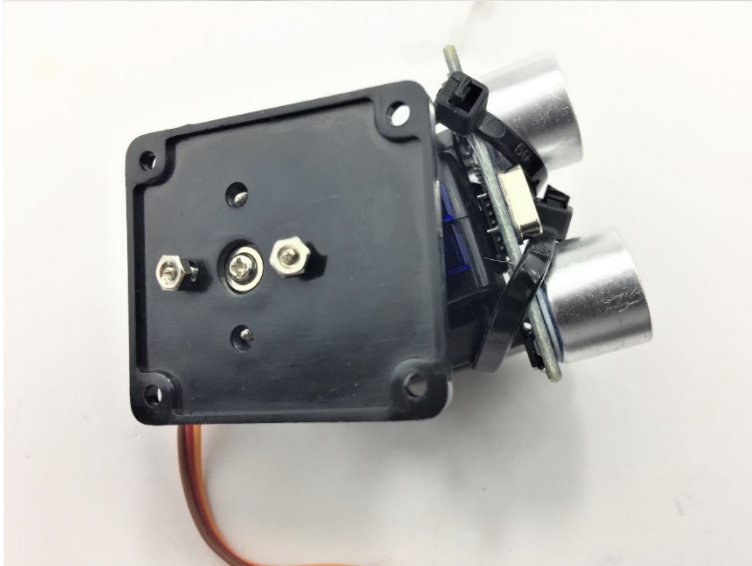
It should look like this



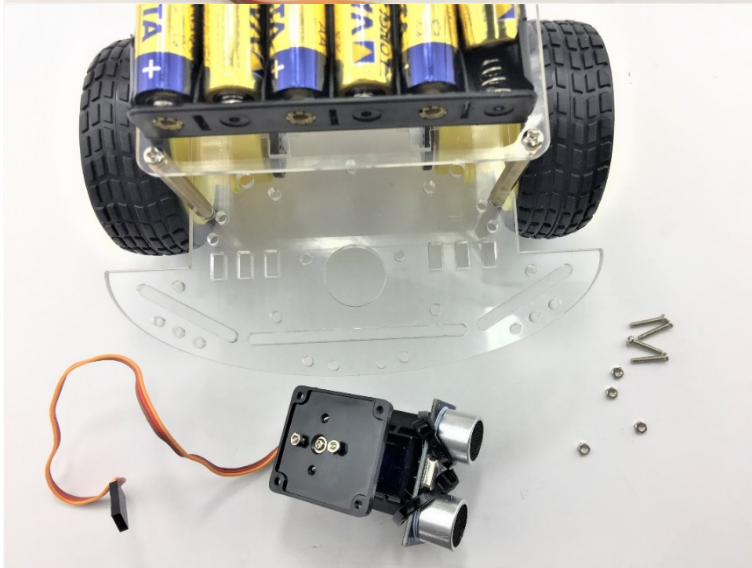
Chapter 4
Hardware installation



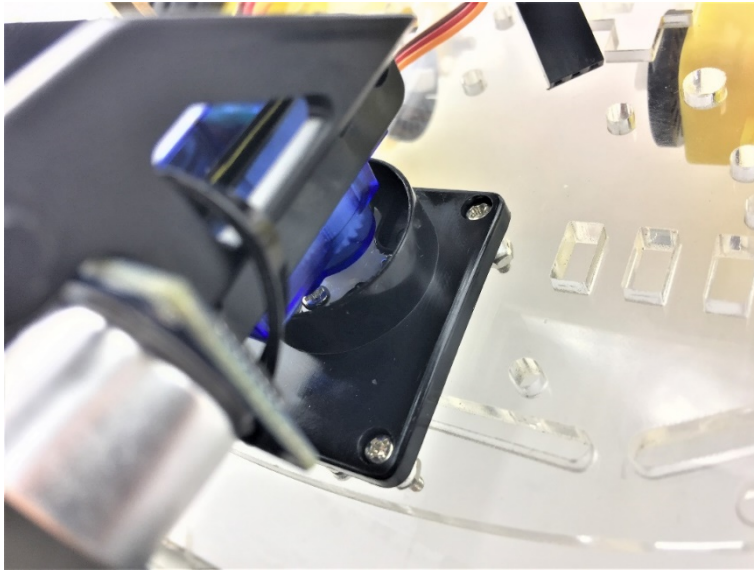
Prepare the Self-tapping screw in the Servo package and tighten the sensor part with servo rack



It should look like this



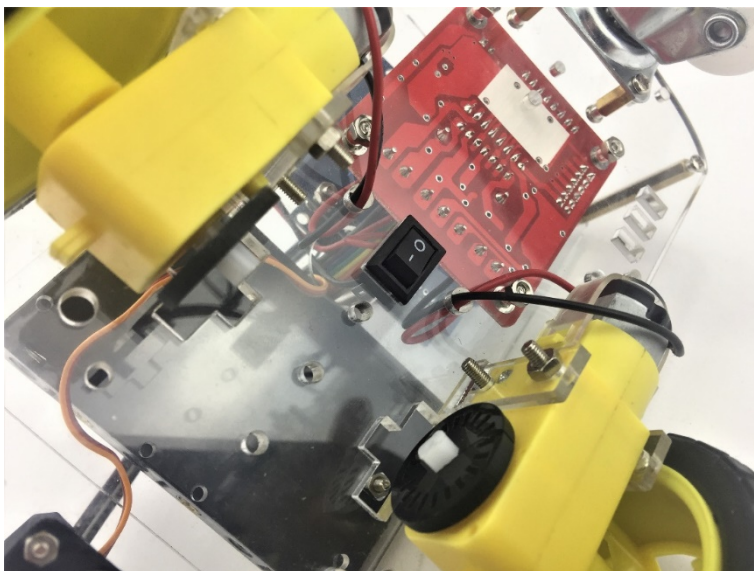
Gather four M2.5 screws and corresponding nuts in the FPV package to secure to the bottom chassis



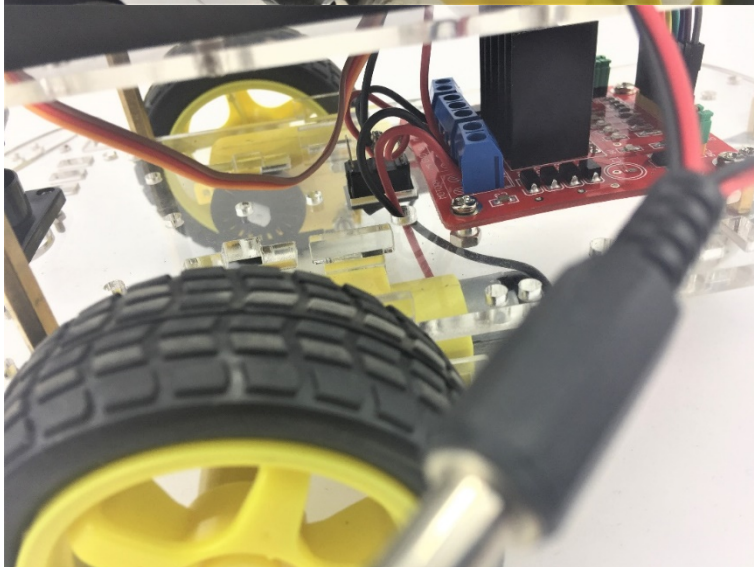
Finished!

4.5. Switch

The switch can save you a lot energy turning the car on and off. However, it's just an option. It can be done by putting the switch in line with the battery box power wire.



Insert the switch in pre-cut slot

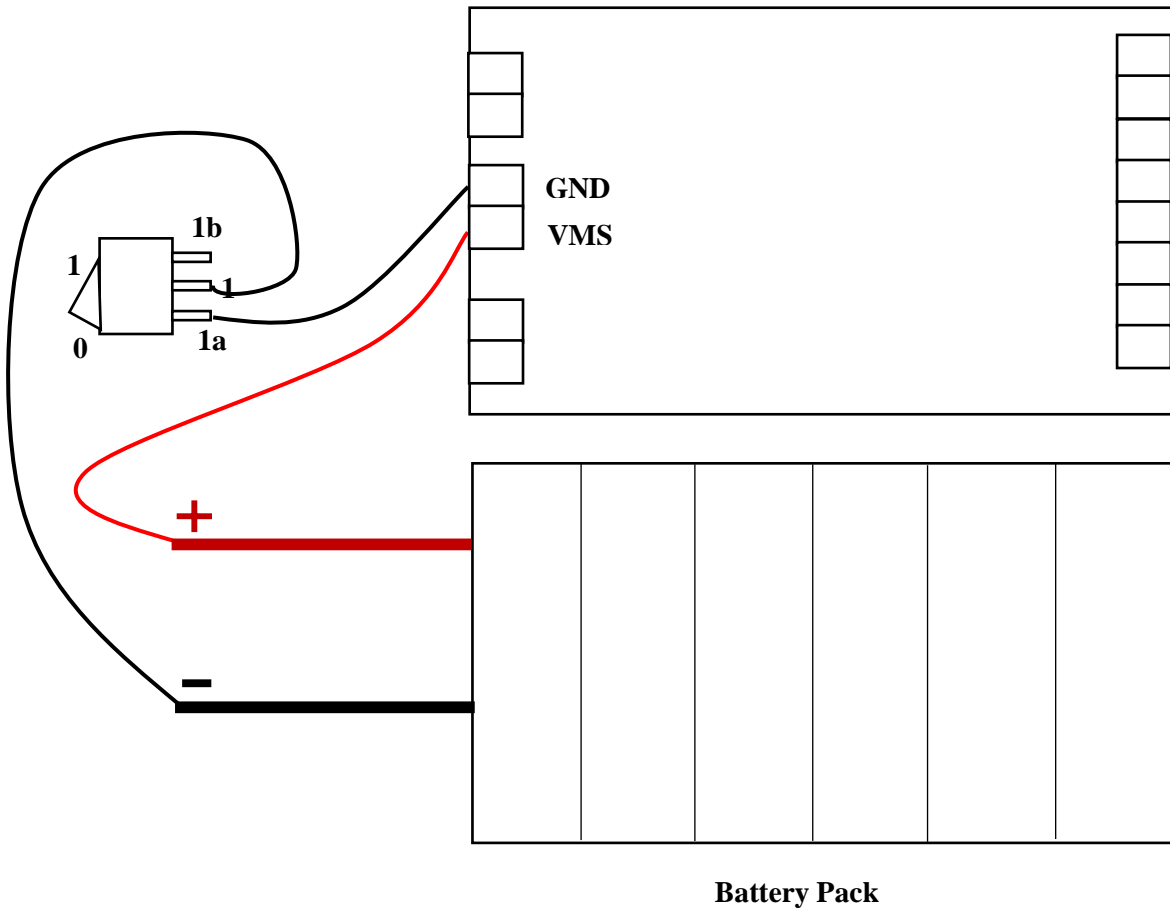


Solder the switch in line with the power cable

Chapter 4
Hardware installation

The following diagram shows the switch connection with battery pack and L289N.

Caution: Do not modify your programming code when USB cable plugged in Arduino UNO development board and your computer with wires connecting to L289N Motor Drive board as a large amount of current will drain from the USB cable which will burn out the computer USB port.



4.6. Wire Connection

You are almost there. Final step is to wire the cables to power supply (i.e. Battery Box), UNO board, ultrasonic sensor and Servo. The following diagram shows the connection map. Don't panic if this is your first project, you can also follow the connection table 2. Take your time and be patient with wires.

Caution: Any incorrect wire connection will lead to problems including device malfunction, device failure, damage to the device or damage to other property.

Table 1 Connection table

UNO board Sensor Shield	L298N	Battery Box	Motor left	Motor Right	Servo	Ultrasonic sensor
	GND	GND				
	VMS	VMS				
	+(Left)		+(red)			
	-(Left)		-(black)			
	+(Right)			+(red)		
	-(Right)			-(black)		
V	ENA					
6	IN1					
9	IN2					
10	IN3					
11	IN4					
V	ENB					
G	GND					
V	5V+					
5					S	
V					+	
G					-	
V						+
A1						Trig
A0						Echo
G						-

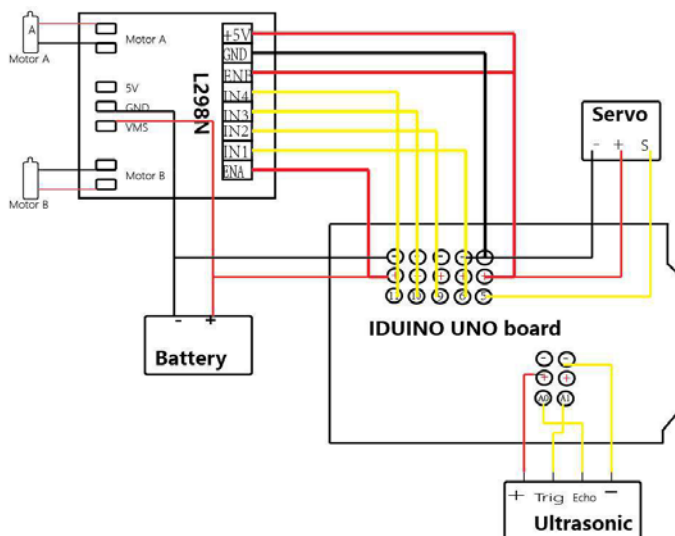


Figure 3 Connection map

Chapter 4
Hardware installation

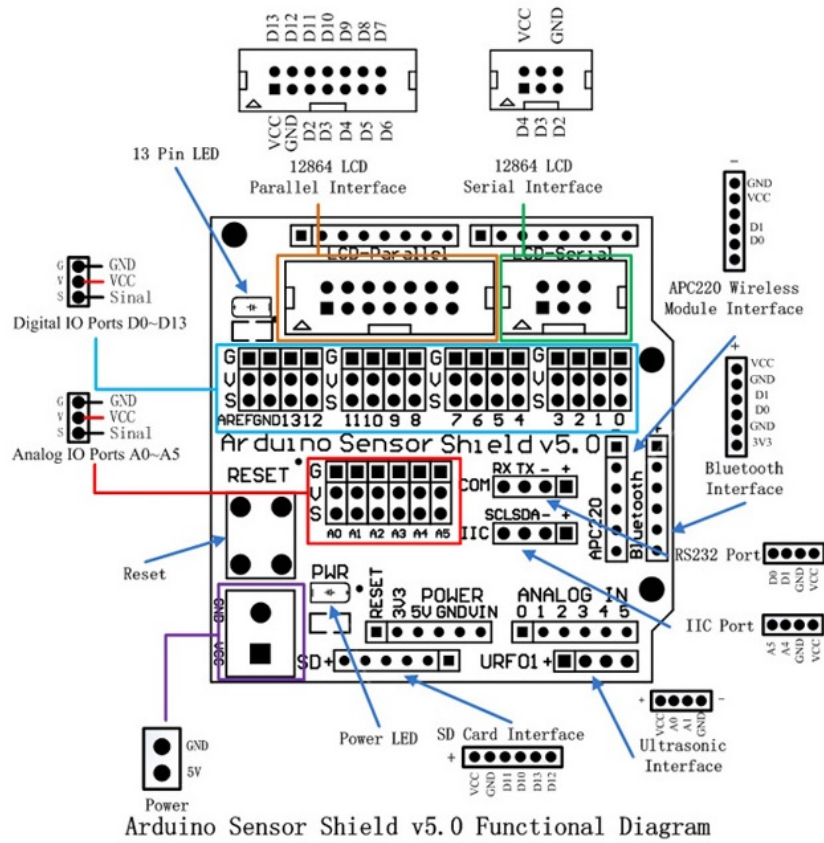


Figure 4 Arduino sensor shield v5.0 Diagram

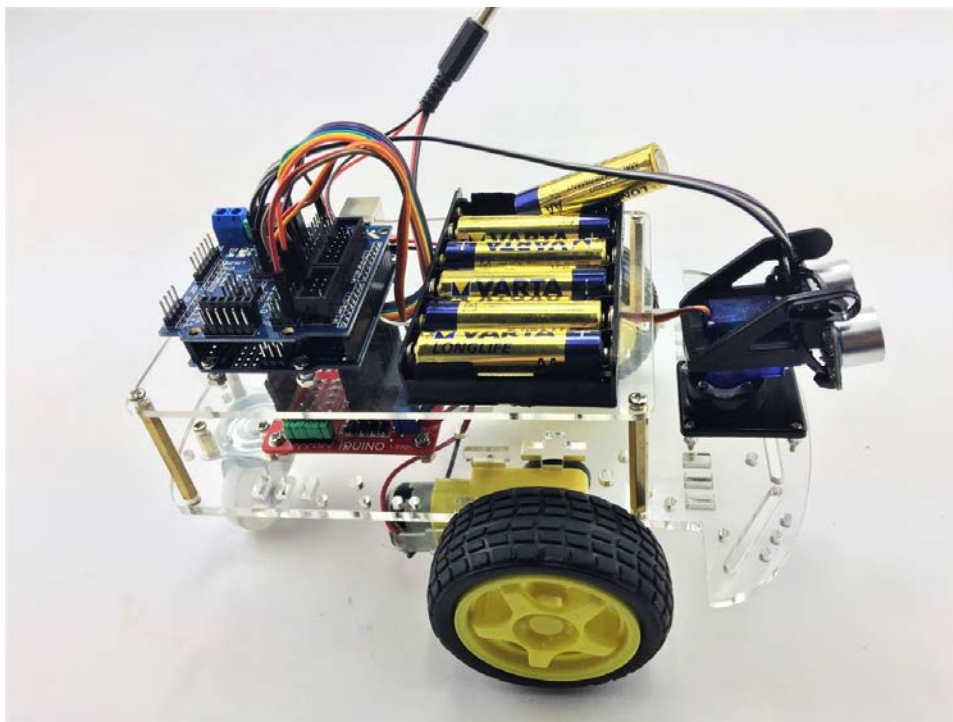


Figure 5 Overview

Chapter 5. Have fun

Now it's time to have fun! Turn the power on, and see how your DIY Arduino Robot car goes! After final assembly and activation, the Robot car may require adjustments and debugging. The Robot will perform on how it is programmed. Figuring out what the code is doing is part of the learning process. Reopen your Arduino IDE and we assure you will learn a lot once you gain a deep understanding of the code.

This kit is just a starting point and can be expanded to incorporate other sensors and modules. You are limited by your imagination.

We are also offering other Arduino Robot Kit versions where you can learn WIFI, Bluetooth, infrared remote control and so many more.

- [Arduino 4 Wheel Drive with Ultrasonic & Line Tracer Bluetooth Robot Kit](https://www.auselectronicdirect.com.au/arduino-4-wheel-drive-with-ultrasonic-line-tracer~3691)
(<https://www.auselectronicdirect.com.au/arduino-4-wheel-drive-with-ultrasonic-line-tracer~3691>)
- [Arduino 4 Wheel Drive with Ultrasonic & Line Tracer Robot Kit](https://www.auselectronicdirect.com.au/arduino-4-wheel-drive-with-ultrasonic-line-tracer~3691)
(<https://www.auselectronicdirect.com.au/arduino-4-wheel-drive-with-ultrasonic-line-tracer~3691>)
- [Arduino 2 Wheel Drive Wireless Bluetooth Robot Kit](https://www.auselectronicdirect.com.au/arduino-2-wheel-drive-wireless-bluetooth-robot-kit)
(<https://www.auselectronicdirect.com.au/arduino-2-wheel-drive-wireless-bluetooth-robot-kit>)
- [Arduino Robot Arm 4dof Mechanical Claw Kit](https://www.auselectronicdirect.com.au/arduino-robot-arm-4dof-mechanical-claw-kit)
(<https://www.auselectronicdirect.com.au/arduino-robot-arm-4dof-mechanical-claw-kit>)

Check our website at [Here. \(https://www.auselectronicdirect.com.au/arduino/arduino-starter-kit/\)](https://www.auselectronicdirect.com.au/arduino/arduino-starter-kit/)

Appendix

Code:

```
*****Code begin*****
```

```
#include <Servo.h>
int pinLB=6; // define pin6 as left back connect with IN1
int pinLF=9; // define pin9 as left forward connect with IN2
int pinRB=10; // define pin10 as right back connect with IN3
int pinRF=11; // define pin11 as right back connect with IN4
int inputPin = A0; // define ultrasonic receive pin (Echo)
int outputPin =A1; // define ultrasonic send pin(Trig)
int Fspeedd = 0; // forward distance
int Rspeedd = 0; // right distance
int Lspeedd = 0; // left distance
int directionn = 0; //
Servo myservo; // new myservo
int delay_time = 250; // set stable time
int Fgo = 8;
int Rgo = 6;
int Lgo = 4;
int Bgo = 2;
// forward
// turn right
// turn left
// back
void setup()
{
  Serial.begin(9600);
  pinMode(pinLB,OUTPUT);
  pinMode(pinLF,OUTPUT);
  pinMode(pinRB,OUTPUT);
  pinMode(pinRF,OUTPUT);
  pinMode(inputPin, INPUT);
  pinMode(outputPin, OUTPUT);
  myservo.attach(5); // define the servo pin(PWM)
}
void advance(int a) // forward
{
  digitalWrite(pinRB,LOW);
  digitalWrite(pinRF,HIGH);
  digitalWrite(pinLB,LOW);
  digitalWrite(pinLF,HIGH);
  delay(a * 40);
}
void turnR(int d) //turn right
{
  digitalWrite(pinRB,LOW);
  digitalWrite(pinRF,HIGH);
  digitalWrite(pinLB,HIGH);
  digitalWrite(pinLF,LOW);
  delay(d * 50);
}
void turnL(int e) //turn left
{
```

Appendix

```
digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,LOW);
digitalWrite(pinLB,LOW);
digitalWrite(pinLF,HIGH);
delay(e * 50);
}
void stopp(int f) //stop
{
digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,HIGH);
delay(f * 100);
}
void back(int g) //back
{
digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,LOW);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,LOW);
delay(g * 300);
}
void detection() //test the distance of different direction
{
int delay_time = 250; //
ask_pin_F(); // read forward distance
if(Fspeedd < 10) // if distance less than 10
{
stopp(1);
back(2);
}
if(Fspeedd < 25) // if distance less than 10
{
stopp(1);
ask_pin_L();
delay(delay_time);
ask_pin_R();
delay(delay_time);
if(Lspeedd > Rspeedd) //if left distance more than right distance
{
directionn = Rgo;
}
if(Lspeedd <= Rspeedd)//if left distance not more than right
//distance
{
directionn = Lgo;
}
}
//if left if (Lspeedd < 10 && Rspeedd < 10) distance and right
//distance both less than 10
{
directionn = Bgo;
}
}
else
{
directionn = Fgo; // forward go
}
```

Appendix

```
}
}
void ask_pin_F() // test forward distance
{
myservo.write(90);
digitalWrite(outputPin, LOW);
delayMicroseconds(2);
digitalWrite(outputPin, HIGH);
delayMicroseconds(10);
digitalWrite(outputPin, LOW);
float Fdistance = pulseIn(inputPin, HIGH);
Fdistance= Fdistance/5.8/10;
Serial.print("F distance:");
Serial.println(Fdistance);
Fspeedd = Fdistance;
}
void ask_pin_L() // test left distance
{
myservo.write(5);
delay(delay_time);
digitalWrite(outputPin, LOW);
delayMicroseconds(2);
digitalWrite(outputPin, HIGH);
delayMicroseconds(10);
digitalWrite(outputPin, LOW);
float Ldistance = pulseIn(inputPin, HIGH);
Ldistance= Ldistance/5.8/10;
Serial.print("L distance:");
Serial.println(Ldistance);
Lspeedd = Ldistance;
}
void ask_pin_R() // test right distance
{
myservo.write(177);
delay(delay_time);
digitalWrite(outputPin, LOW);
delayMicroseconds(2);
digitalWrite(outputPin, HIGH);
delayMicroseconds(10);
digitalWrite(outputPin, LOW);
float Rdistance = pulseIn(inputPin, HIGH);
Rdistance= Rdistance/5.8/10;
Serial.print("R distance:");
Serial.println(Rdistance);
Rspeedd = Rdistance;
}
void loop()
{
myservo.write(90);
detection();
if(directionn == 2)
{
back(8);
turnL(2);
Serial.print(" Reverse ");
}
}
```

Appendix

```
if(directionn == 6)
{
back(1);
turnR(6);
Serial.print(" Right ");
}
if(directionn == 4)
{
back(1);
turnL(6);
Serial.print(" Left ");

}
if(directionn == 8)
{
advance(1);
Serial.print(" Advance ");
Serial.print(" ");
}
}
}
*****Code End*****
```