TA0135

USER MANUAL

ARDUINO 4 WHEEL DRIVE WITH ULTRASONIC & LINE TRACER BLUETOOTH ROBOT KIT

AUS ELECTRONICS DIRECT.COM.AU

Contents

Chapter 1. Overview – TA0135

In this instruction, we will introduce you through the fun project of the Arduino 4 Wheel Drive with Ultrasonic & Line Tracer Bluetooth Robot Kit. This DIY Arduino based Bluetooth robot kit is based on Arduino Uno development board. It is based on a 4-wheel drive platform integrated with an ultrasonic sensor to avoid obstacles and line tracer sensors. Let’s get started!

Chapter 2. Getting started: the 4 Wheel Drive Ultrasonic & Line Tracer Bluetooth Robot Kit 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.

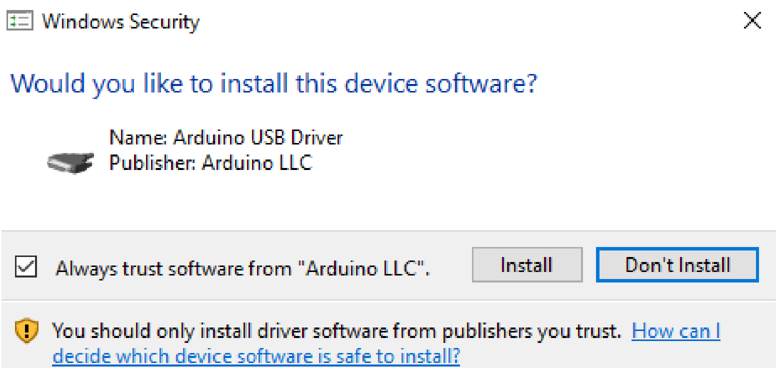
It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

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!



Windows Security

Would you like to install this device software?

Name: Arduino USB Driver
Publisher: Arduino LLC

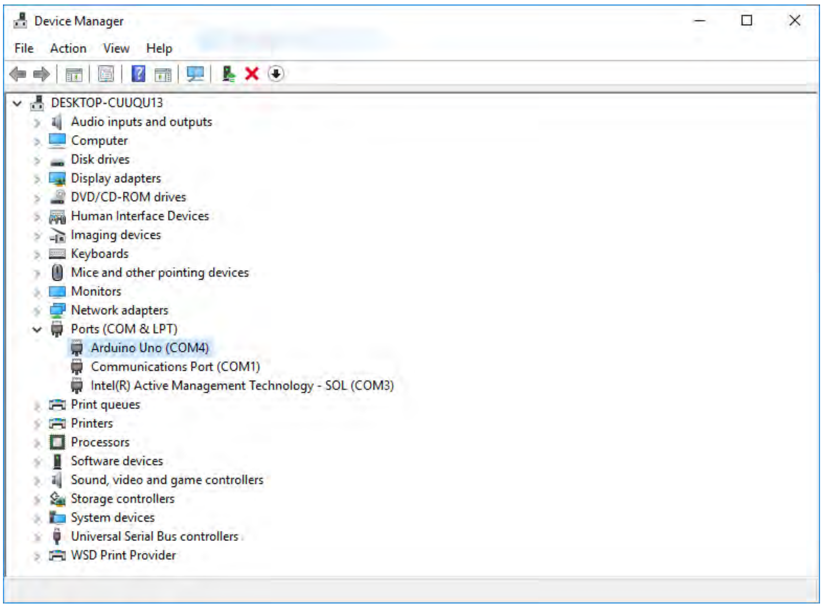
☑ Always trust software from "Arduino LLC". Install Don't Install

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

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.



Device Manager

File Action View Help

- DESKTOP-CUUQU13
  - Audio inputs and outputs
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - Human Interface Devices
  - Imaging devices
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Ports (COM & LPT)
    - Arduino Uno (COM4)
    - Communications Port (COM1)
    - Intel(R) Active Management Technology - SOL (COM3)
  - Print queues
  - Printers
  - Processors
  - Software devices
  - Sound, video and game controllers
  - Storage controllers
  - System devices
  - Universal Serial Bus controllers
  - WSD Print Provider

Find your Serial COM number and note it down.

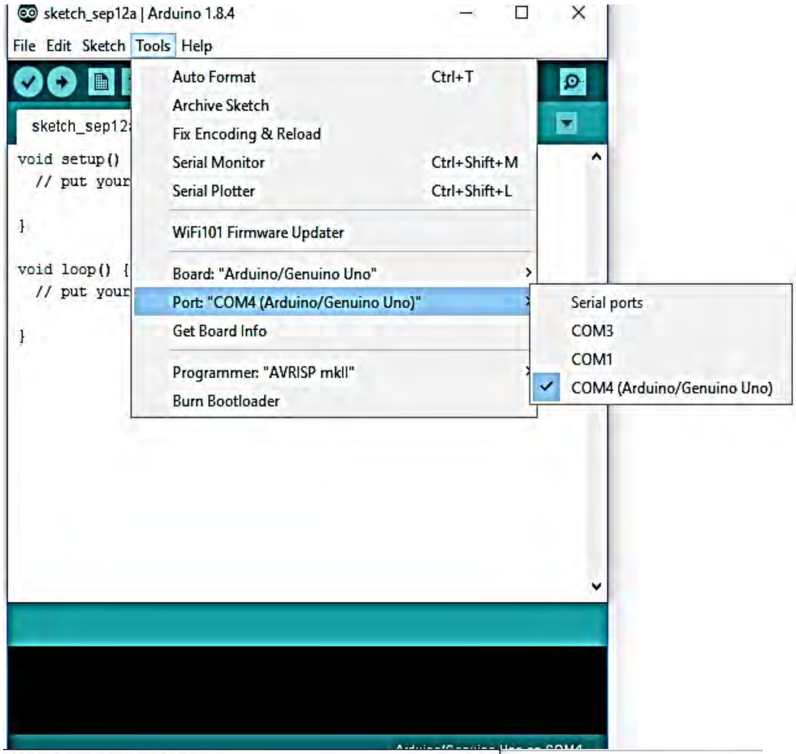
Figure 2 Check Your special COM and note it down the number

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 (COMx)

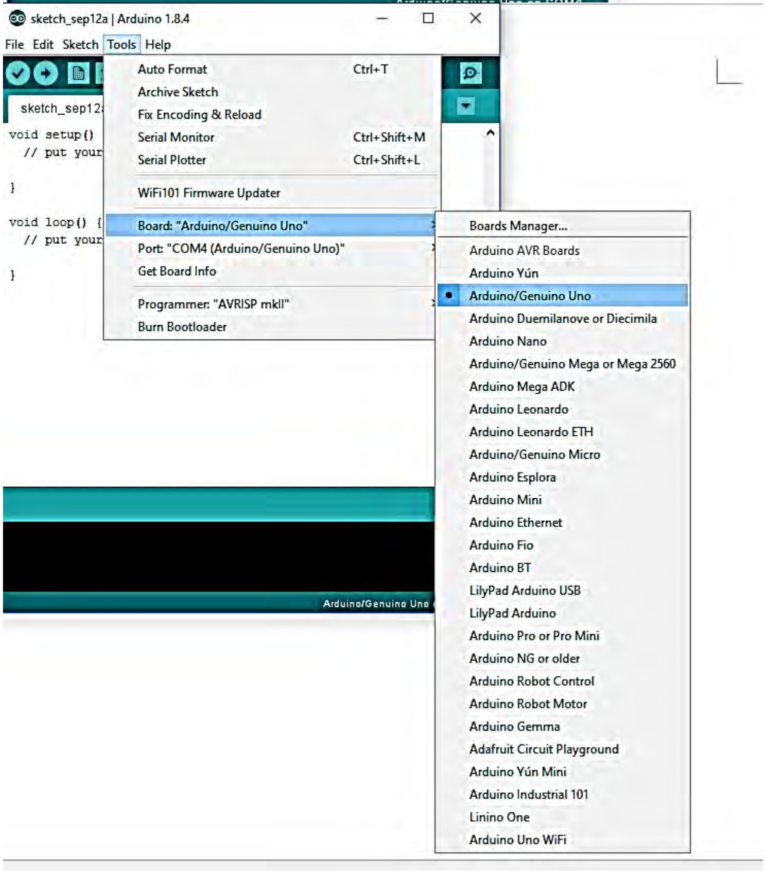
Note down the COM number as we require this later. As the COM port may vary from time to time, this step is vital. In this case for demonstration purpose, we are using the COM 4.

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:


sketch_sep12a | Arduino 1.8.4
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM4 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM3
COM1
✓ COM4 (Arduino/Genuino Uno)
sketch_sep12
void setup() {
// put your
}
void loop() {
// put your
}

Configuration of Ports

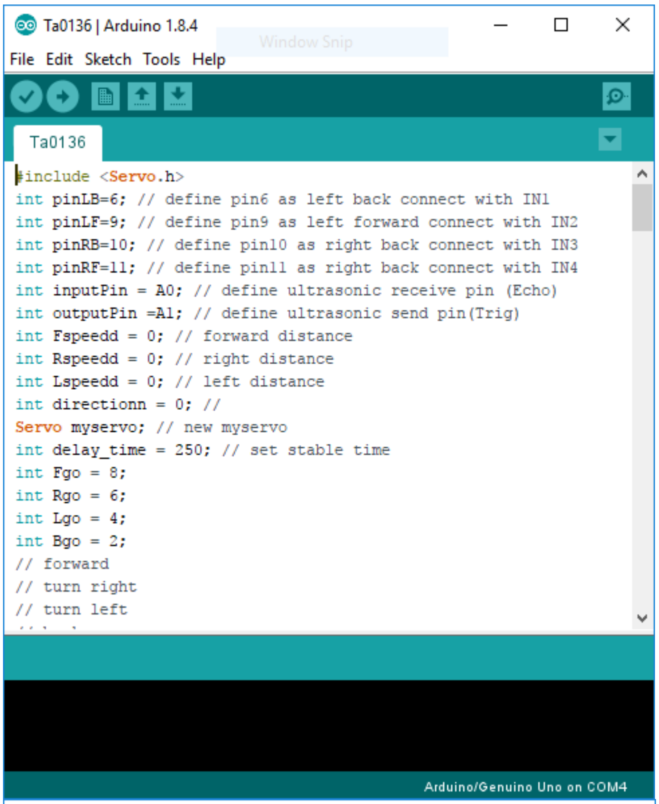


Configuration of the Board

Software installation

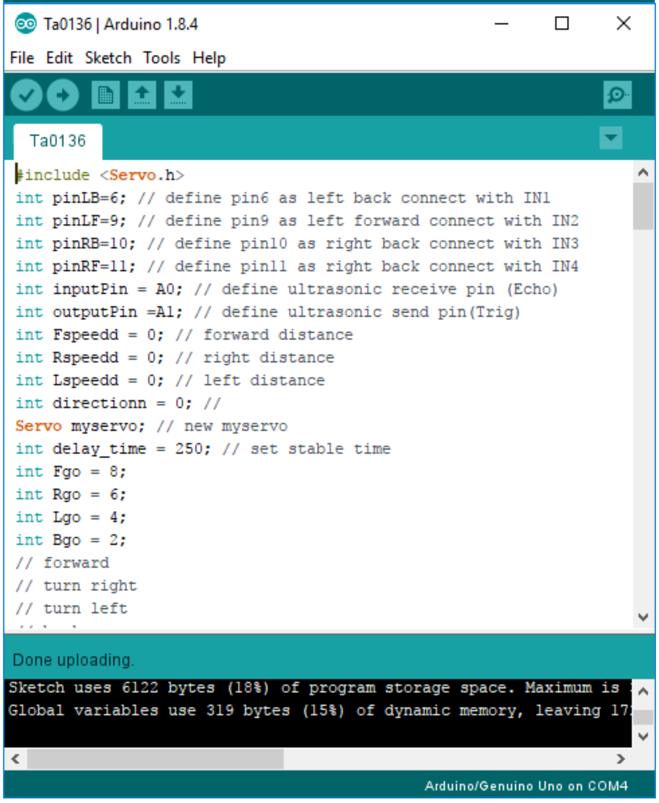
It’s time to play with you first simple example. Following the path by File | Examples | 01. Basics | Blink. A new code window would pop up, press the arrow symbol to upload. You will notice the orange LED is blinking almost every second.

3.3. Run your Arduino 2 Wheel Drive code



```
#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
```

Upload to UNO



```
#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
```

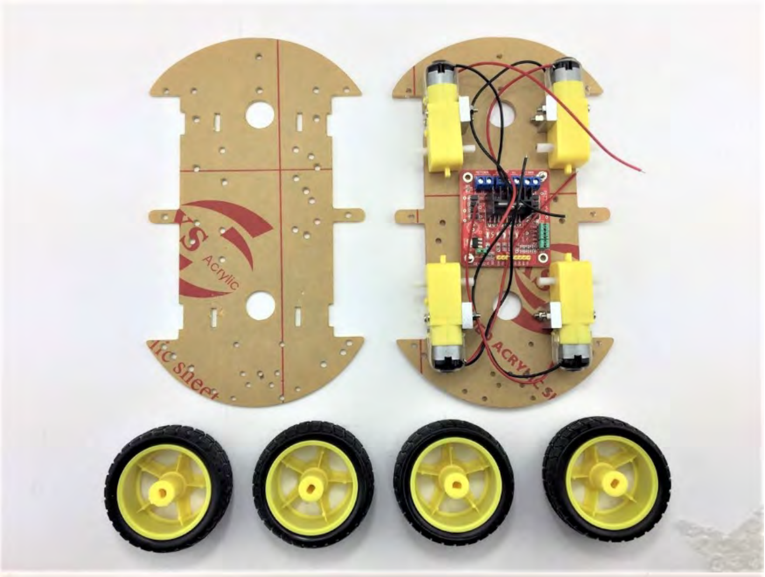
Done uploading.

```
Sketch uses 6122 bytes (18%) of program storage space. Maximum is
Global variables use 319 bytes (15%) of dynamic memory, leaving 17
```

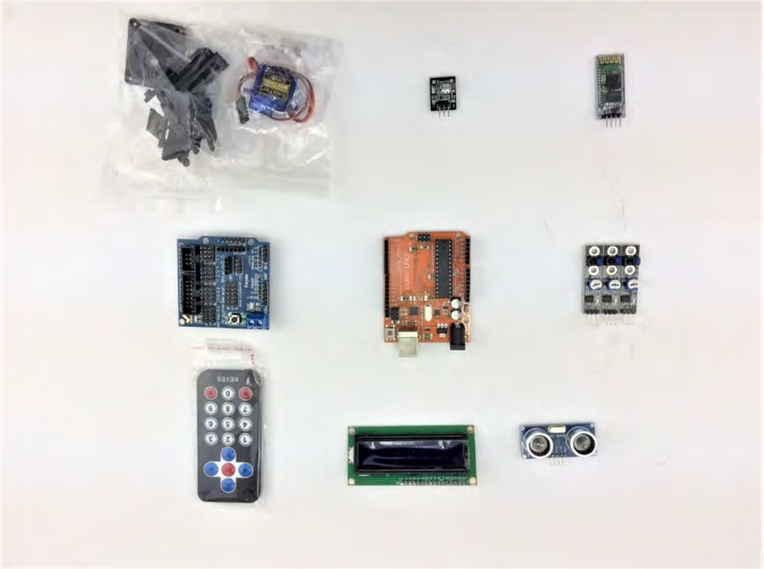
Done uploading!

Chapter 4. Hardware installation

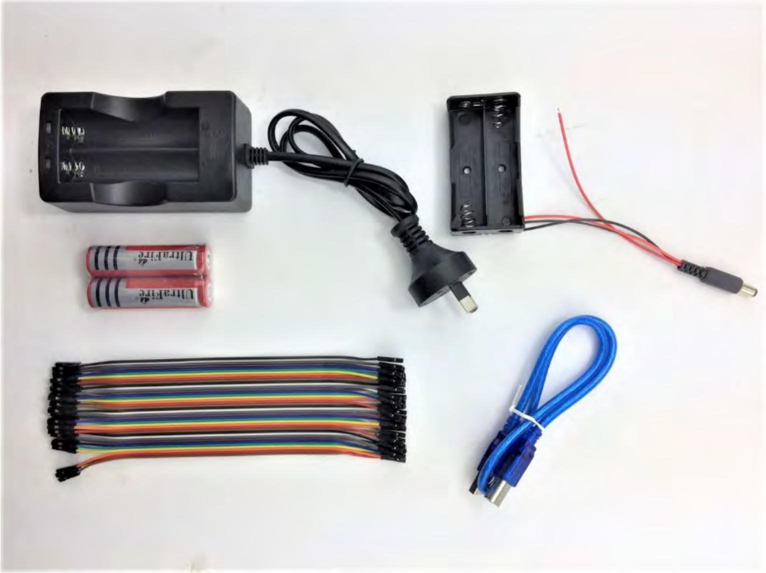
4.1. Unboxing and Component list



1. Acrylic Chassis with Protection Cover
2. Four DC Motors
3. L289N Motor Driver Board
4. Four Rubber wheels

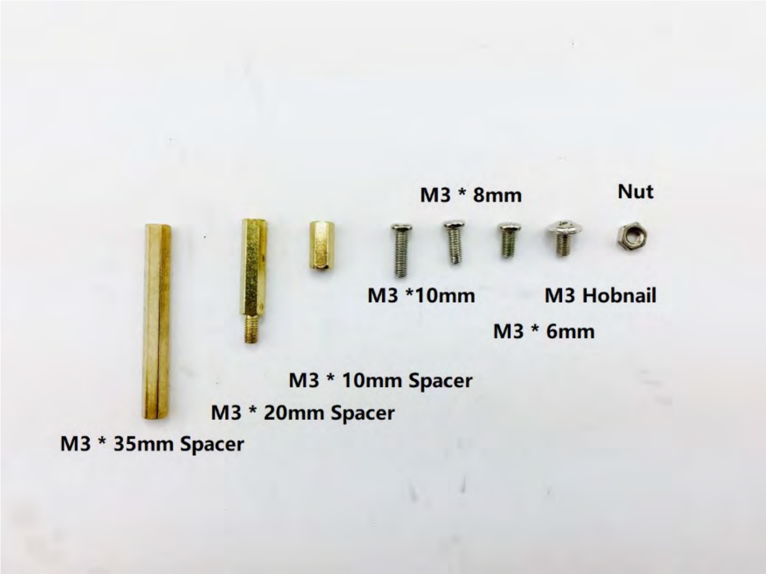


1. Ultrasonic Sensor
2. LCD1602
3. Remote Controller
4. Three Line track sensors
5. iDuino UNO Development Board
6. Arduino Sensor shield V5.0
7. HC-06 Bluetooth Module
8. Infrared receiver
9. Servo package (Horns and brackets included)



1. 2 x 18650 Batteries
2. Battery Charger
3. 18650 Battery Box
4. USB cable
5. Jumper wires (30)

Hardware installation



Fastener package:

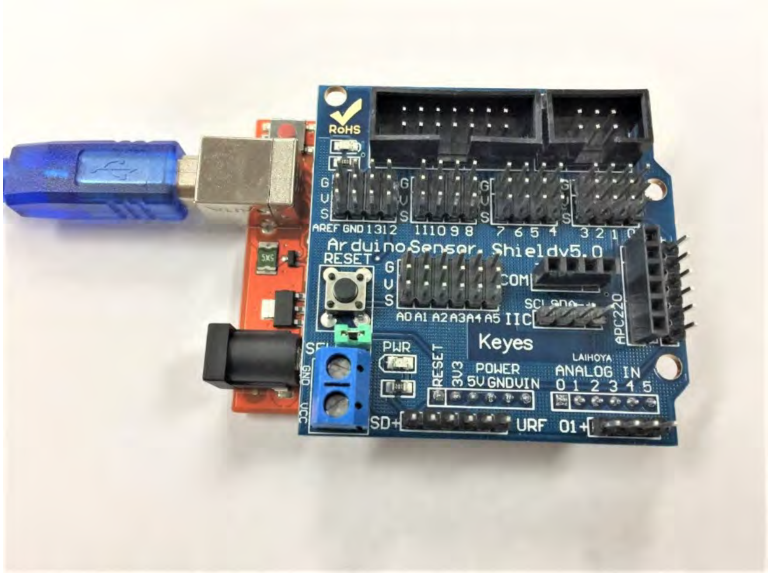
1. 6 x M3 35mm Spacers
2. 12 x M3 10mm Spacers
3. 6 x M3 10mm Screws
4. 8 x M3 8mm Screws
5. 14 x M3 6mm Screws
6. Hobnail (not used for this project)
7. 5 x M3 Nuts

4.2. Module Experiments

In this section, we are pre-testing and experimenting sensor modules separately to make sure that they function smoothly which are required to put together in the final Combination Module test. Be patient and have fun with each module.

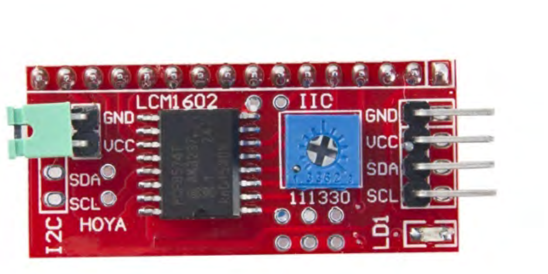
4.2.1 LCD1602 Module with I2C Interface

This module is ideal for low power display up to 16 characters by 2 rows for a total of 32. Together with Arduino Sensor Shield V5.0, this LCD1602 module is just at your hand. With this I2C interface LCD module, you only need 2 wires to display information. If you already have I2C devices in your project, this LCD module uses no more pins at all.



Firstly, place Arduino Sensor Shield on top of the Arduino UNO and make sure Pins are lined up correctly.

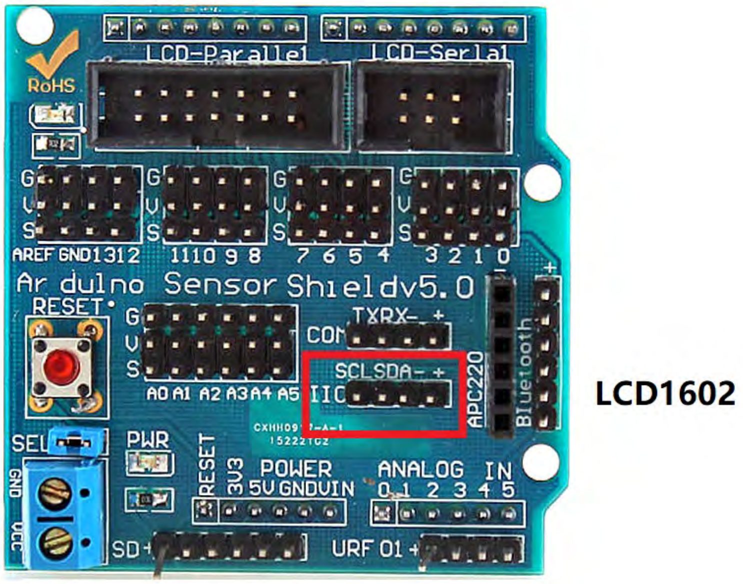
Hardware installation



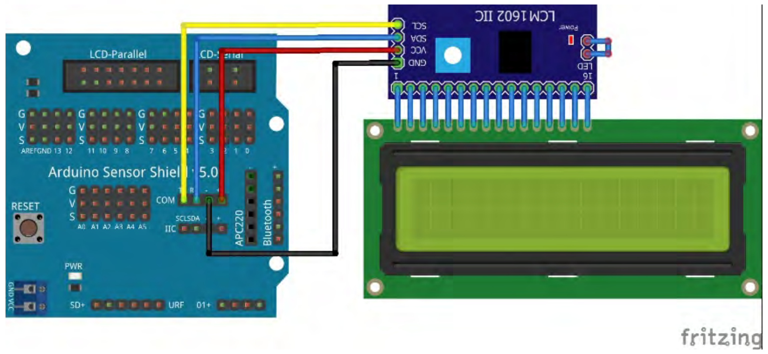
Identify the Pins from top to bottom:

- GND
- VCC
- SDA
- SCL

There is a backlight, and contrast adjustment potentiometer. You can adjust that to suit the ambient light.

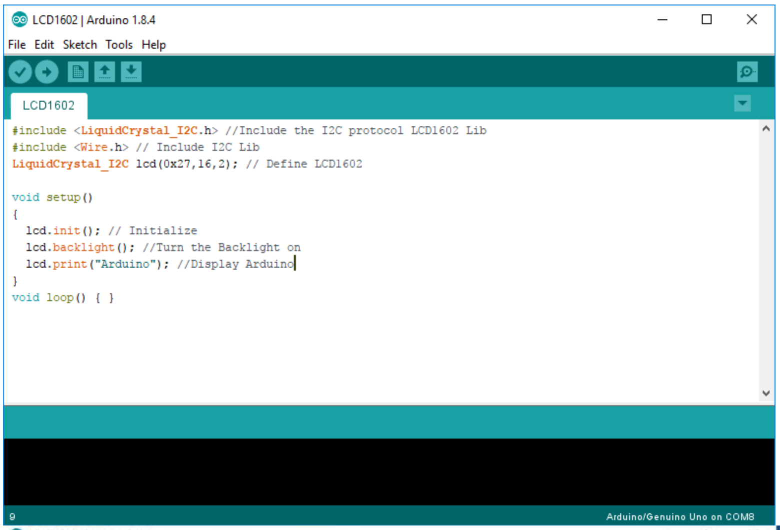


Use four jumper wires to connect LCD1602 module to the Arduino Sensor Shield



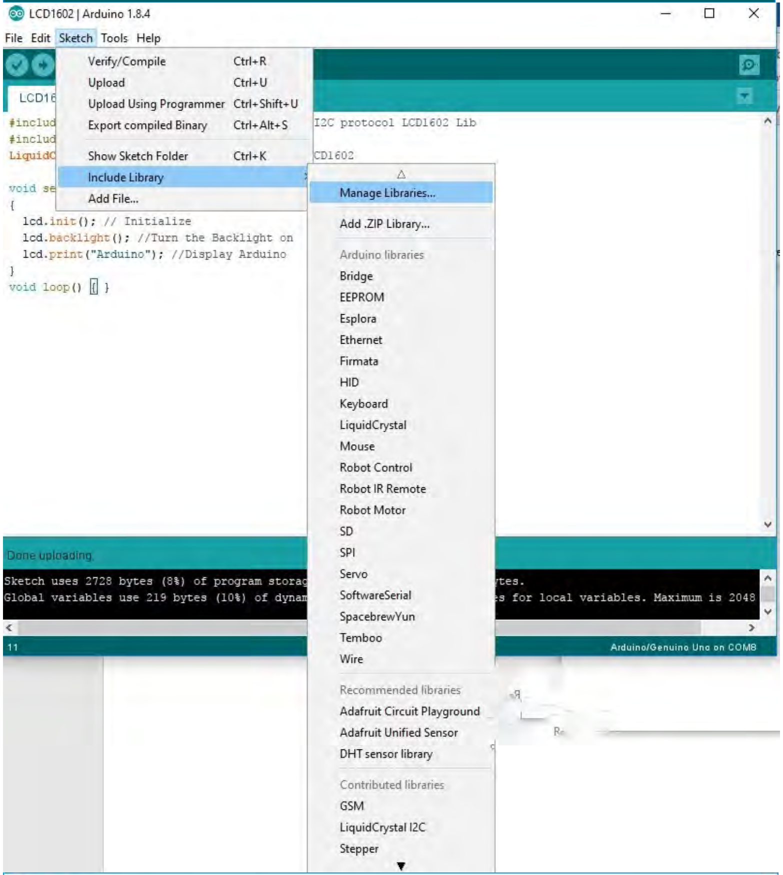
Schematic diagram

Hardware installation



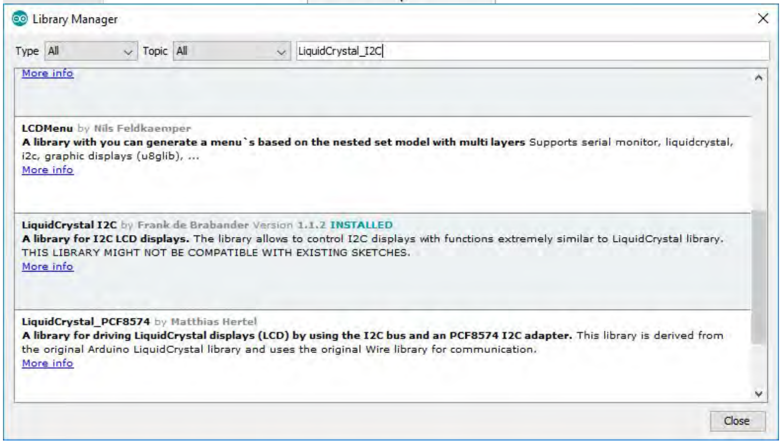
```
#include <LiquidCrystal_I2C.h> //Include the I2C protocol LCD1602 Lib
#include <Wire.h> // Include I2C Lib
LiquidCrystal_I2C lcd(0x27,16,2); // Define LCD1602

void setup()
{
  lcd.init(); // Initialize
  lcd.backlight(); //Turn the Backlight on
  lcd.print("Arduino"); //Display Arduino
}
void loop() { }
```

Open the LCD1602 sketch by double clicking the file (.ino)



```
Sketch uses 2728 bytes (8%) of program storag                          ytes.
Global variables use 219 bytes (10%) of dynam                          s for local variables. Maximum is 2048
```

If this is your first LCD1602 project, then you might need to import LCD1602 I2C library. Follow - Sketch | Include Library | Manage Libraries



Type All Topic All LiquidCrystal_I2C

More info

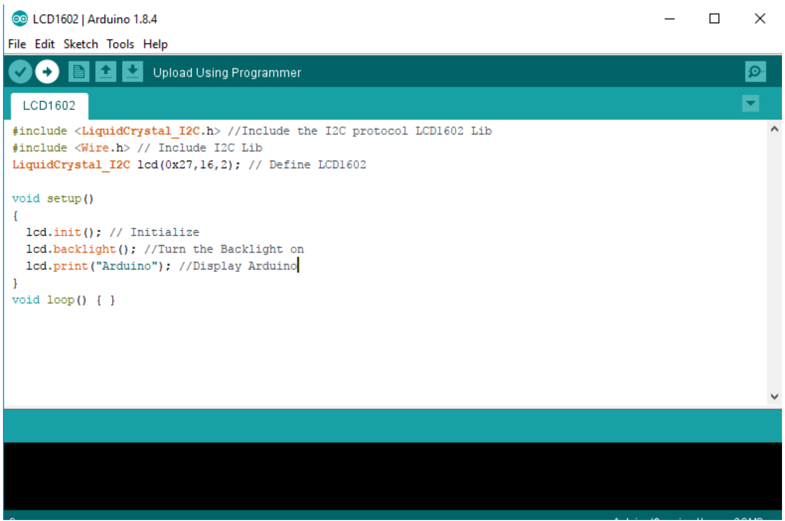
LCDMenu by Nils Feldkaemper
A library with you can generate a menu`s based on the nested set model with multi layers Supports serial monitor, liquidcrystal, i2c, graphic displays (u8glib), ...
More info

LiquidCrystal I2C by Frank de Brabander Version 1.1.2 INSTALLED
A library for I2C LCD displays. The library allows to control I2C displays with functions extremely similar to LiquidCrystal library. THIS LIBRARY MIGHT NOT BE COMPATIBLE WITH EXISTING SKETCHES.
More info

LiquidCrystal_PCF8574 by Matthias Hertel
A library for driving LiquidCrystal displays (LCD) by using the I2C bus and an PCF8574 I2C adapter. This library is derived from the original Arduino LiquidCrystal library and uses the original Wire library for communication.
More info

When this window pops up, type in the "LiquidCrystal_I2C" in the search bar on the top right corner and install it. Please wait for several minutes to complete the installation.

Hardware installation



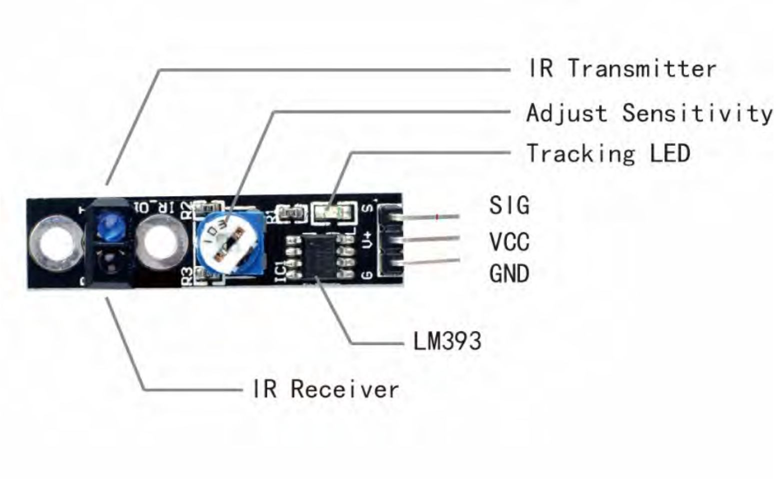
```
#include <LiquidCrystal_I2C.h> //Include the I2C protocol LCD1602 Lib
#include <Wire.h> // Include I2C Lib
LiquidCrystal_I2C lcd(0x27,16,2); // Define LCD1602

void setup()
{
  lcd.init(); // Initialize
  lcd.backlight(); //Turn the Backlight on
  lcd.print("Arduino"); //Display Arduino
}
void loop() { }
```

Hit the Upload button to upload to your UNO board

Now the first line should show" Arduino" as in the code

4.2.2 Track Line Module

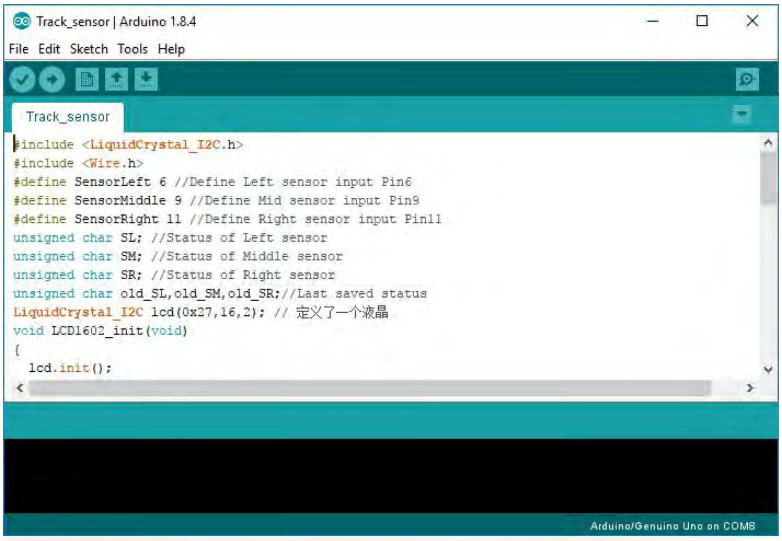
This reflective optical sensor has three channels as you can see three pairs of "Black/Blue" IR LED's on the sensors. This module measures the reflectivity of a surface with an infrared emitter/detector pair. If infrared light is detected, the module S pin will send digital high voltage signal to Arduino board. Now let's test it out!



This diagram shows a single-channel track line sensor pins

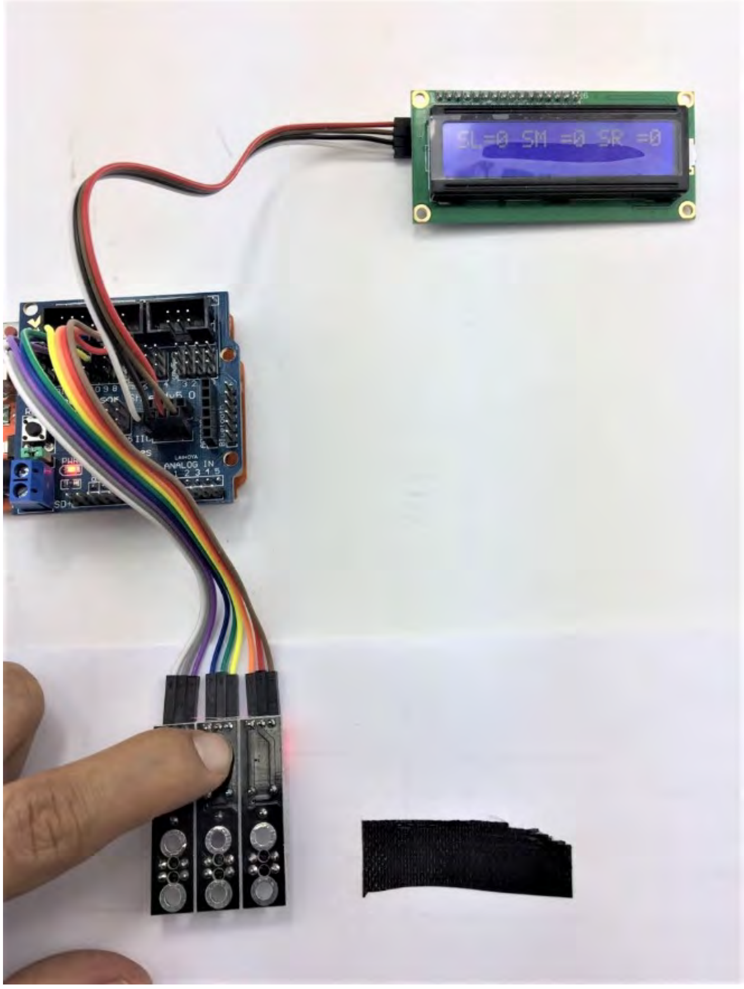
Track Line Sensor Board	Arduino Sensor shield
Left Sensor - Signal	6
Middle Sensor - Signal	9
Right Sensor - Signal	11
GND	GND
VCC	+5V

Hardware installation


Track_sensor | Arduino 1.8.4
File Edit Sketch Tools Help
Track_sensor
Arduino/Genuino Uno on COM8

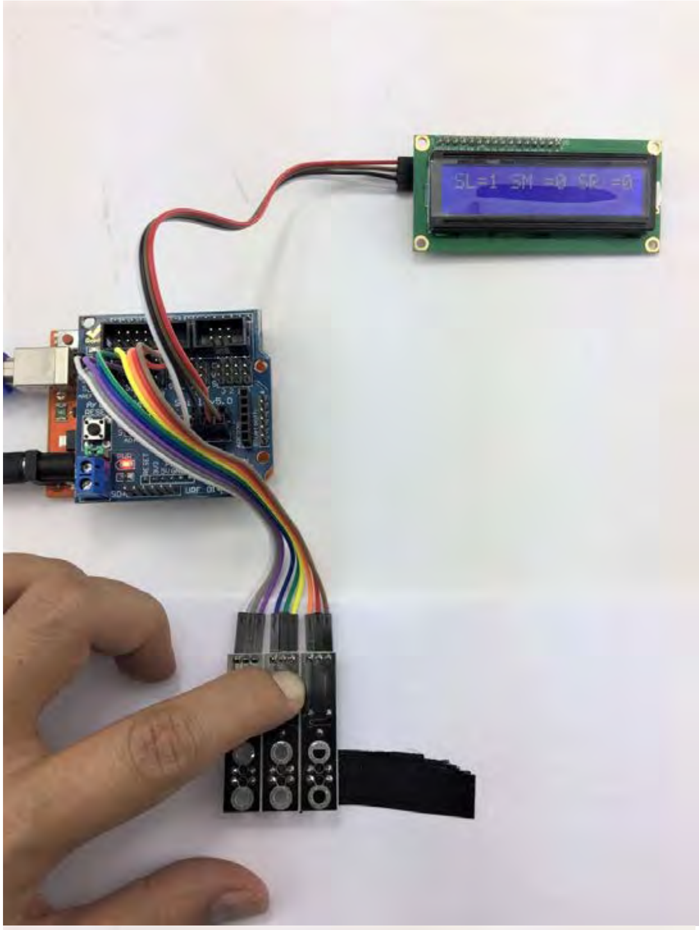
```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#define SensorLeft 6 //Define Left sensor input Pin6
#define SensorMiddle 9 //Define Mid sensor input Pin9
#define SensorRight 11 //Define Right sensor input Pin11
unsigned char SL; //Status of Left sensor
unsigned char SM; //Status of Middle sensor
unsigned char SR; //Status of Right sensor
unsigned char old_SL,old_SM,old_SR;//Last saved status
LiquidCrystal_I2C lcd(0x27,16,2); // 定义了一个液晶
void LCD1602_init(void)
{
  lcd.init();
```

As always, open sketch and upload it to UNO board

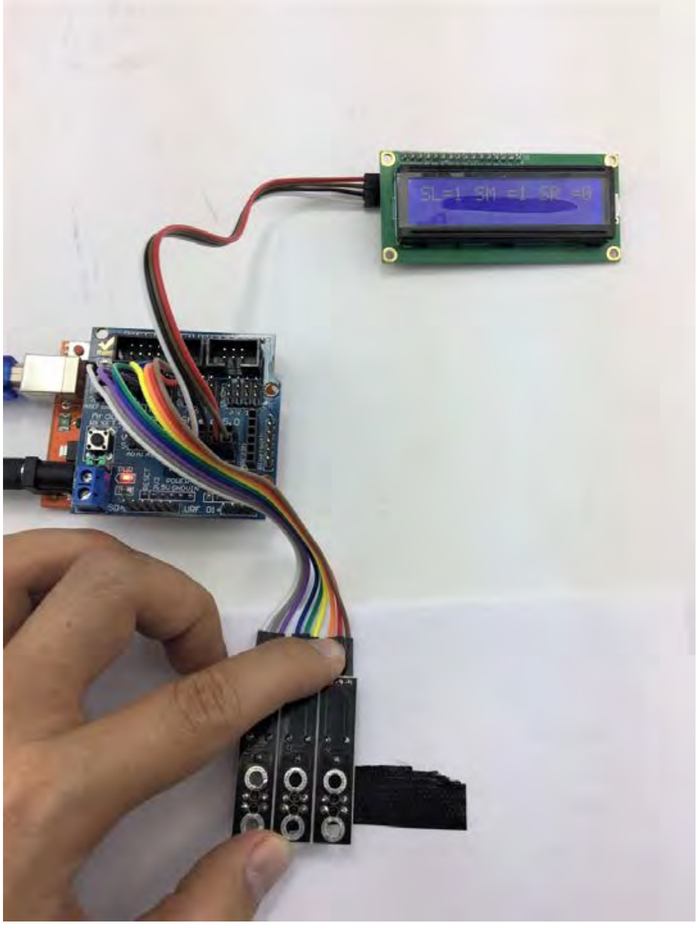


Based on the working principles of this reflective track line sensor, we are required to adjust their sensitivity by changing the corresponding potentiometers in white background with black strips.
Place it round 1cm above the white paper and now the red indicators should light up. Move it above the black strip and the indicator should turn off. If not, adjust the potentiometer clockwise to increase its sensitivity.

Hardware installation

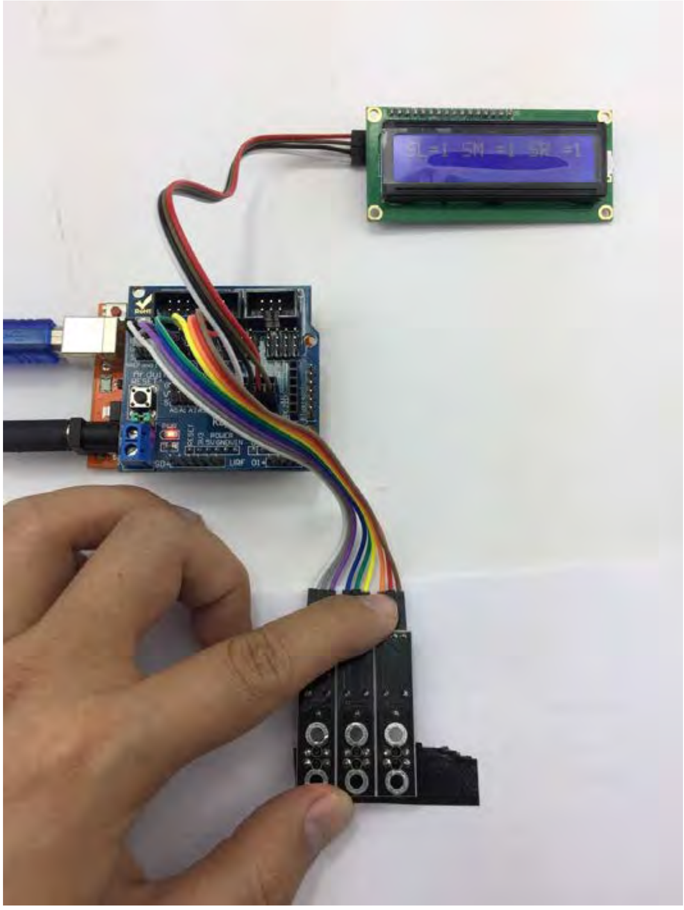


As shown in the LCD, the left IR detects the black line



As shown in the LCD, the Left and Middle IR detects the black line

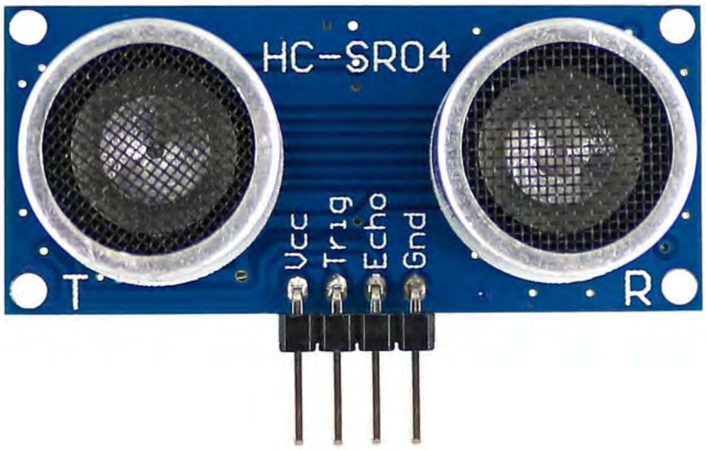
Hardware installation



As shown in the LCD, the right, middle and left IR detects the black line

4.2.3 Ultrasonic sensor Module

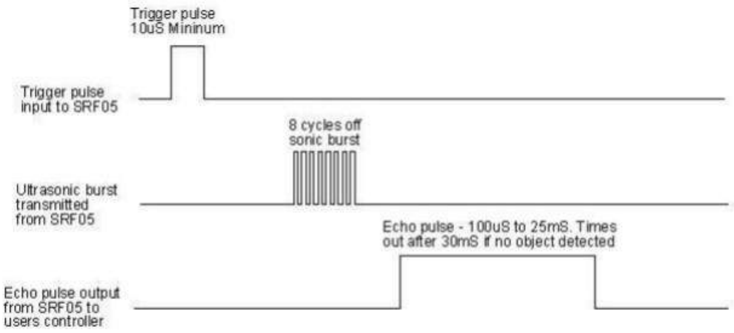
Ultrasonics sensor distance measuring module which a non-contact distance measurement module.



The sensor has 4 pins:

- VCC (5V)
- GND
- Trigger
- Echo

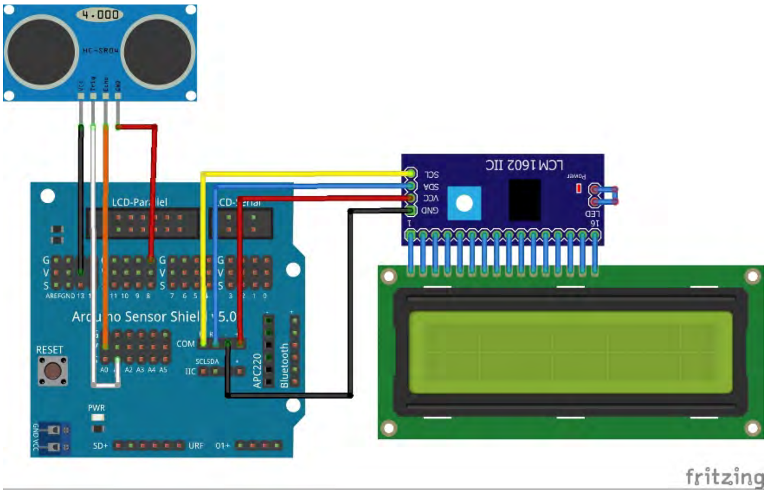
Hardware installation



The working principle is:

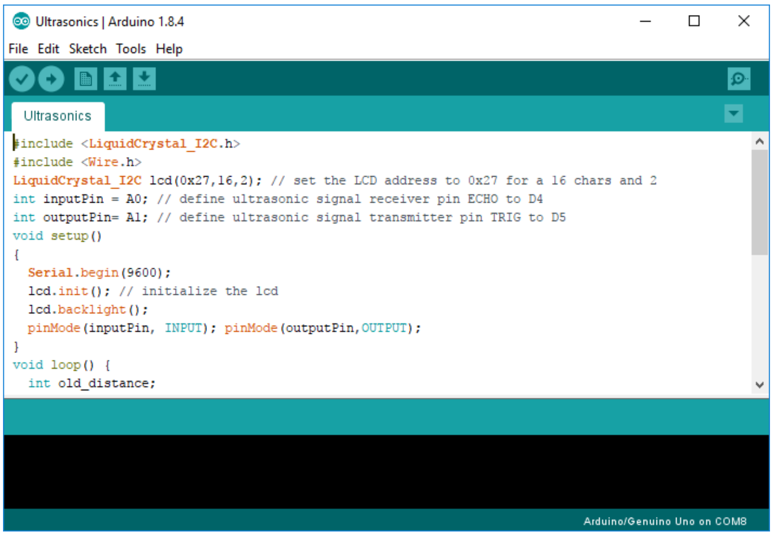
- Initialize TRIG pin with low voltage and then send at least 10µs High voltage from TRIG pin
- The sensor will send 8 cycles of square waves with 40KHz sonic burst
- Then sensor will automatically detect echo waves from the ECHO pin
- Then the distance is determined by the difference time

Distance = Duration of High voltage level * 170 m/s



Pin connection diagram

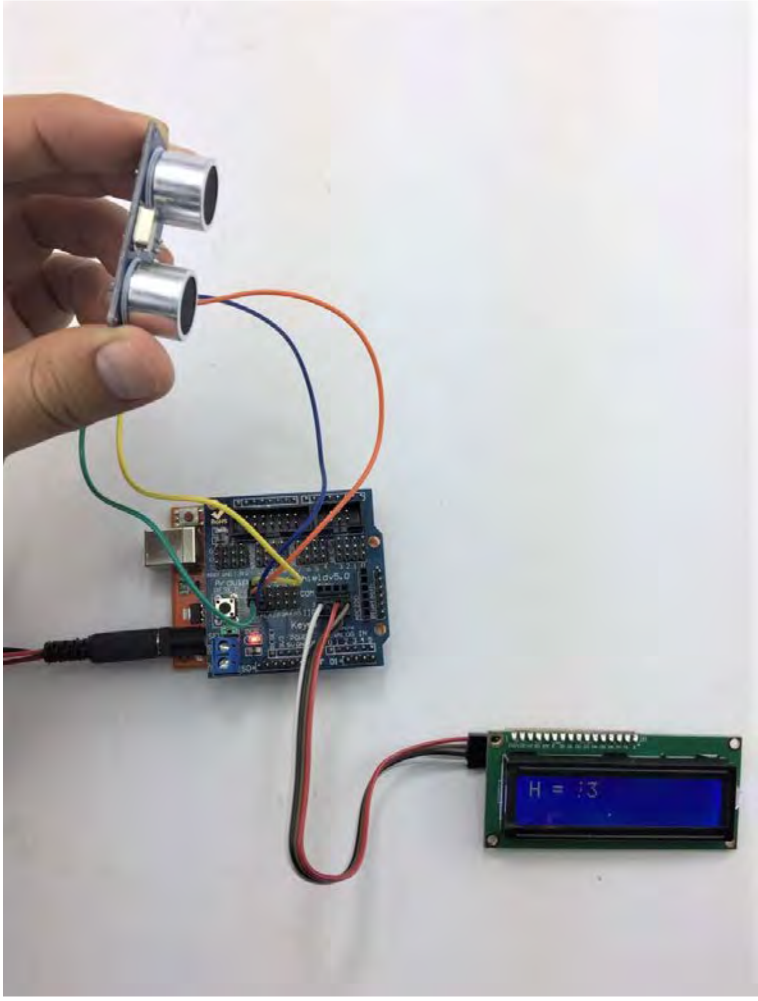
Ultrasonics sensor	Arduino Sensor shield
Trig	A0
Echo	A1


Ultrasonics | Arduino 1.8.4
File Edit Sketch Tools Help
Ultrasonics
Arduino/Genuino Uno on COM8

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2
int inputPin = A0; // define ultrasonic signal receiver pin ECHO to D4
int outputPin= A1; // define ultrasonic signal transmitter pin TRIG to D5
void setup()
{
  Serial.begin(9600);
  lcd.init(); // initialize the lcd
  lcd.backlight();
  pinMode(inputPin, INPUT); pinMode(outputPin,OUTPUT);
}
void loop() {
  int old_distance;
```

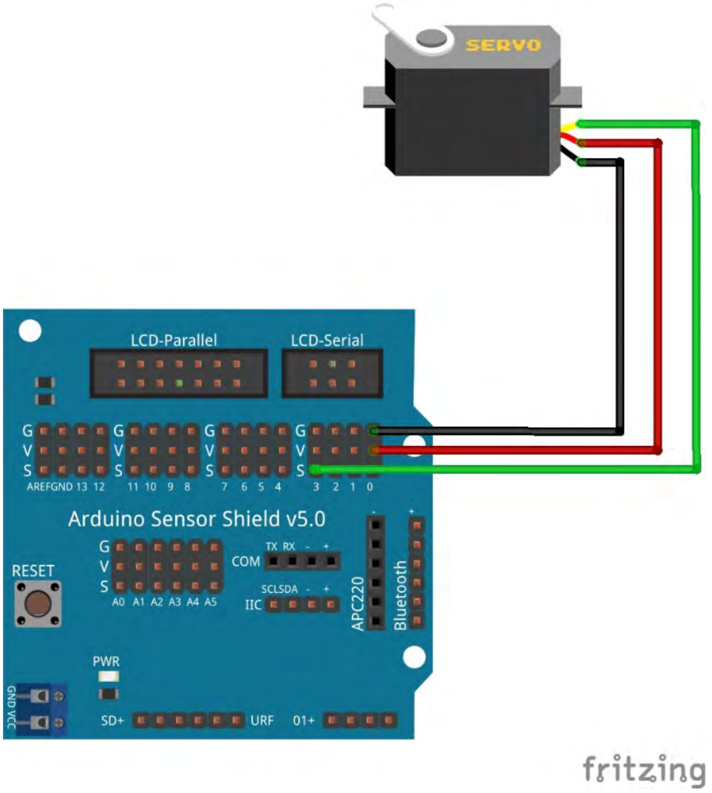
Upload the sketch and position the ultrasonic sensor to different angles to test

Hardware installation



The distance is shown in the LCD1602 in the unit of centimetre.

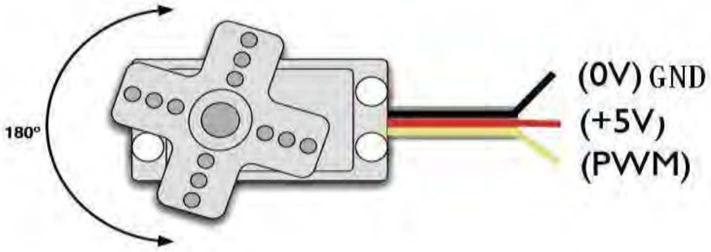
4.2.4 Servo Module



Rotation angle ranges from 0 ~ 180

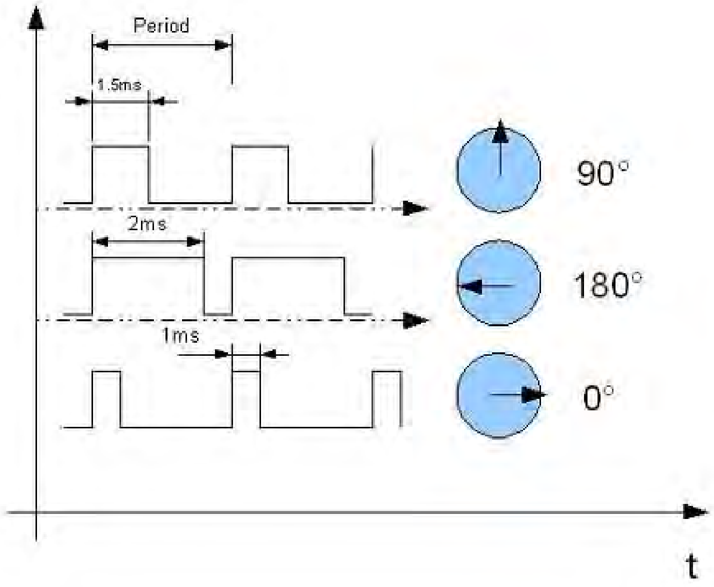
Servo sensor	Arduino Sensor shield
Signal	3

Pin connection



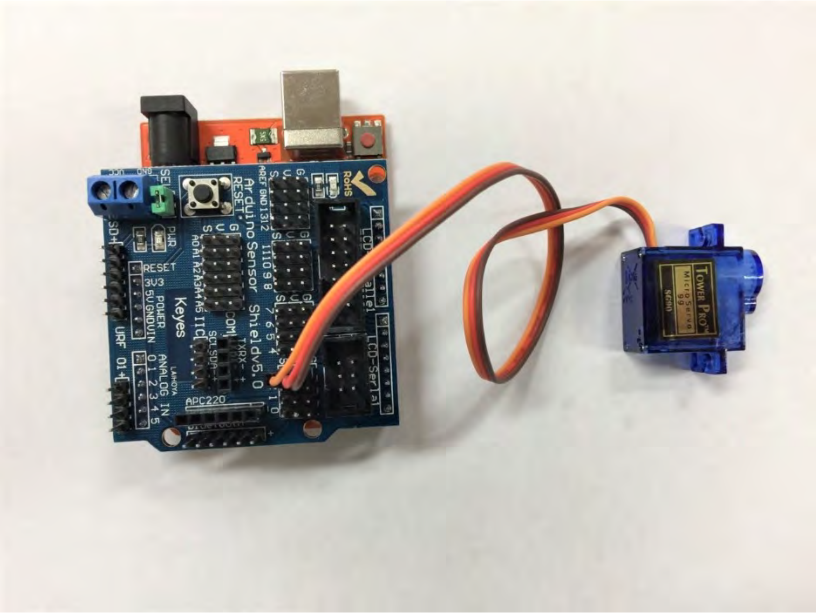
Servo has three pins:

- Signal
- GND
- VCC

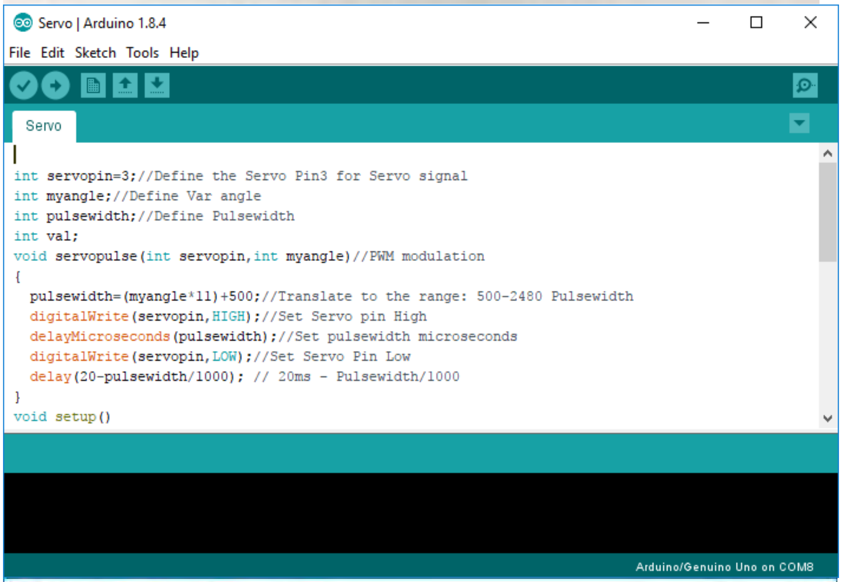


The rotation angle is regulated by PWM (pulse width modulation) signal duty cycle.
The frequency of PWM usually is in range from 30 to 60Hz – this is so called refresh rate. If this refresh rate is too small then then accuracy of servo reduces as it starts losing its position periodically if rate is too big, then servo can start chatter. It is important to select optimal rate, that servo motor could lock its position.

Hardware installation

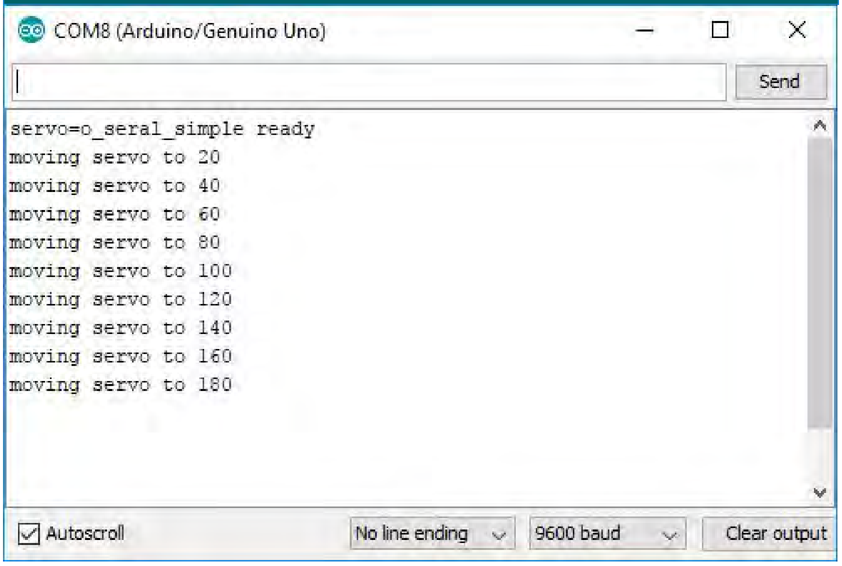


Connect the light orange signal wire to the Arduino Sensor shield digital pin 3 as shown on the left picture



```
int servopin=3;//Define the Servo Pin3 for Servo signal
int myangle;//Define Var angle
int pulsewidth;//Define Pulsewidth
int val;
void servopulse(int servopin,int myangle)//PWM modulation
{
  pulsewidth=(myangle*11)+500;//Translate to the range: 500-2480 Pulsewidth
  digitalWrite(servopin,HIGH);//Set Servo pin High
  delayMicroseconds(pulsewidth);//Set pulsewidth microseconds
  digitalWrite(servopin,LOW);//Set Servo Pin Low
  delay(20-pulsewidth/1000); // 20ms - Pulsewidth/1000
}
void setup()
```

Upload the sketch and open the Serial Monitor (remember to choose the corresponding Baud rate 9600)



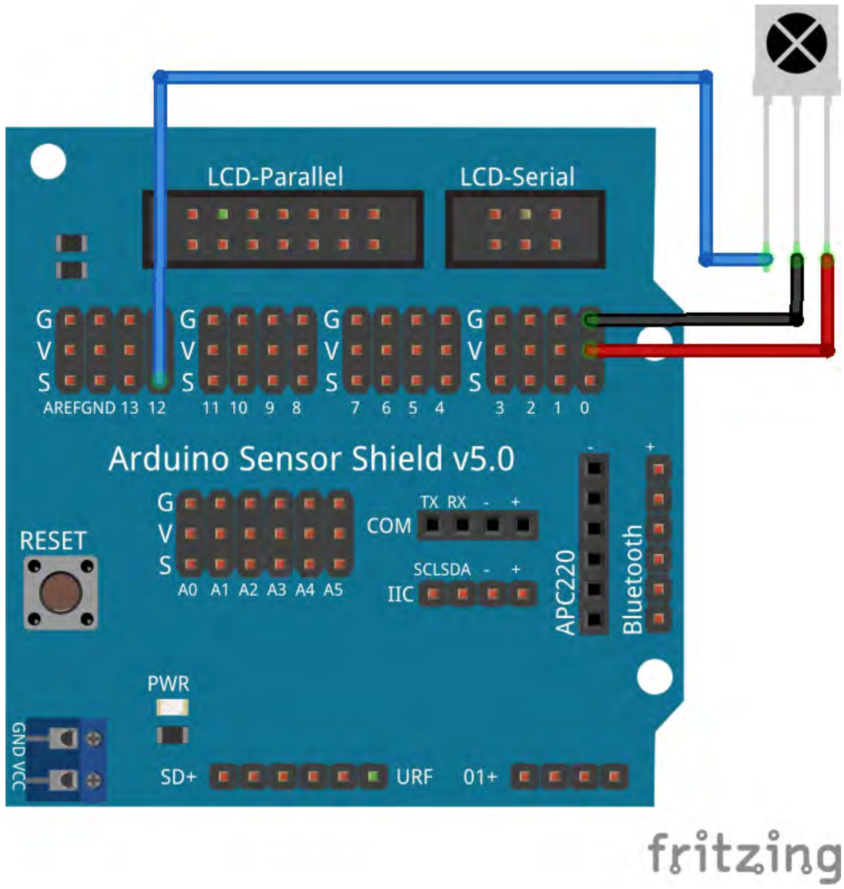
```
servo=o_seral_simple ready
moving servo to 20
moving servo to 40
moving servo to 60
moving servo to 80
moving servo to 100
moving servo to 120
moving servo to 140
moving servo to 160
moving servo to 180
```

☑ Autoscroll — No line ending — 9600 baud

By typing the angle in the dialog bar in the Serial Monitor and observe the servo change.

4.2.5 Infrared Sensor Module

This is an infra-red emitting diode which produces near -infrared Light (black light) which mainly used in a variety of optical switches, remote transmitter circuit, infrared emission control structures.



Pins connection diagram

-

This infrared sensor module has three pins:

- Signal
- GND
- VCC

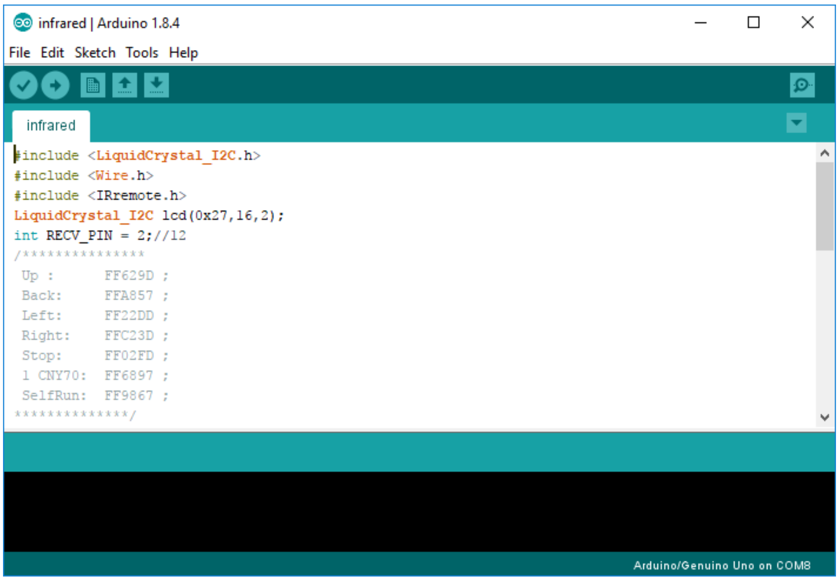
Hardware installation

Use three jumper wires to extend infrared sensor pins.

Remove the plastic battery cover on the bottom of the remote controller, which used to isolate remote controller and button battery and extend the battery life

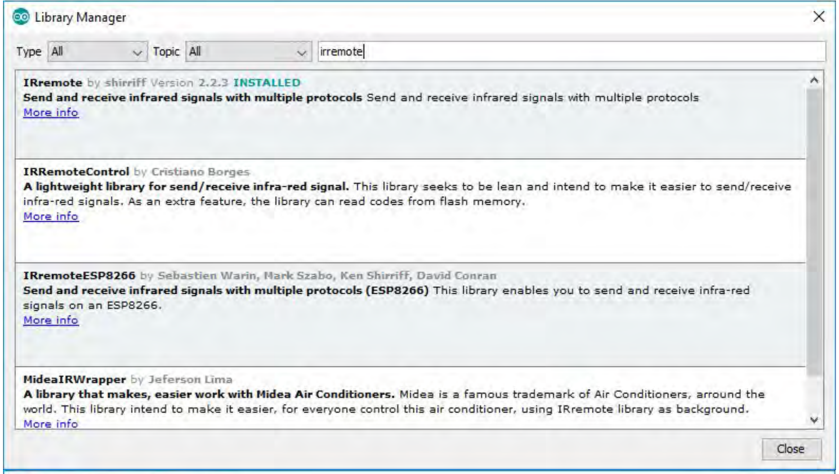
Pull out the battery cover

Hardware installation



```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <IRremote.h>
LiquidCrystal_I2C lcd(0x27,16,2);
int RECV_PIN = 2;//12
/***************
 Up :      FF629D ;
 Back:     FFA857 ;
 Left:     FF22DD ;
 Right:    FFC23D ;
 Stop:     FF02FD ;
 1 CNY70:  FF6897 ;
 SelfRun:  FF9867 ;
**************/
```

Open the sketch



Type: All · Topic: All · irremote

IRremote by shirriff Version 2.2.3 INSTALLED
Send and receive infrared signals with multiple protocols Send and receive infrared signals with multiple protocols
More info

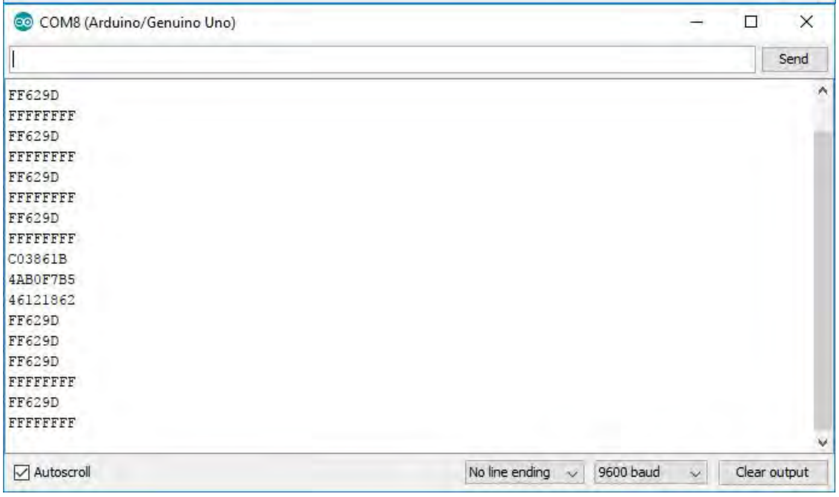
IRRemoteControl by Cristiano Borges
A lightweight library for send/receive infra-red signal. This library seeks to be lean and intend to make it easier to send/receive infra-red signals. As an extra feature, the library can read codes from flash memory.
More info

IRremoteESP8266 by Sebastien Warin, Mark Szabo, Ken Shirriff, David Conran
Send and receive infrared signals with multiple protocols (ESP8266) This library enables you to send and receive infra-red signals on an ESP8266.
More info

MideaIRWrapper by Jeferson Lima
A library that makes, easier work with Midea Air Conditioners. Midea is a famous trademark of Air Conditioners, arround the world. This library intend to make it easier, for everyone control this air conditioner, using IRremote library as background.
More info

If this is your first remote control project, then you might need to import remote control library. Follow - Sketch | Include Library | Manage Libraries.

Search “irremote” in the search bar and install the package.



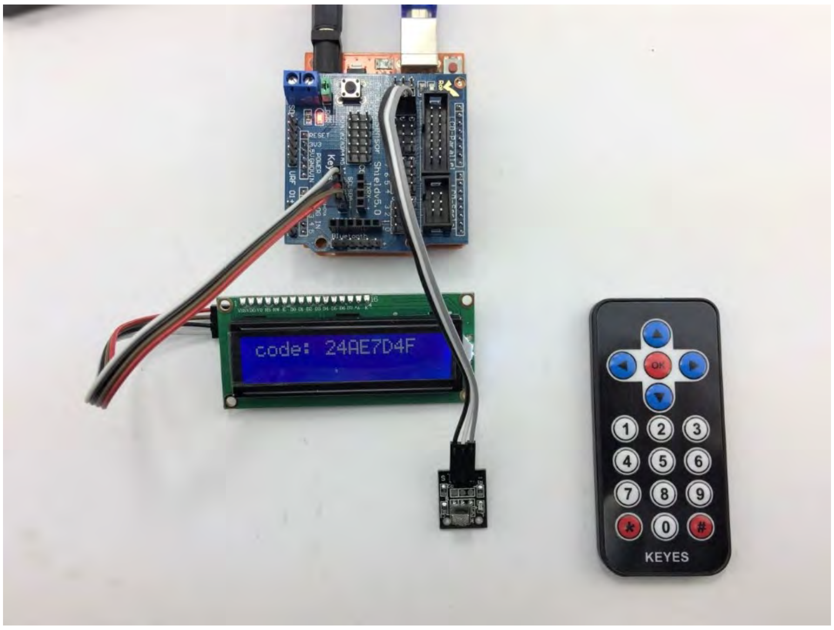
☑ Autoscroll · No line ending · 9600 baud

```
FF629D
FFFFFFFF
FF629D
FFFFFFFF
FF629D
FFFFFFFF
FF629D
FFFFFFFF
C03861B
4AB0F7B5
46121862
FF629D
FF629D
FF629D
FFFFFFFF
FF629D
FFFFFFFF
```

Aim the remote receiver and press the buttons and check the corresponding code for each button

Button on the remote controller:	IR receiver code:
a) Up	a) FF629D
b) Back	b) FFA857
c) Left	c) FF22DD
d) Right	d) FFC23D
e) Stop	e) FF02FD
f) 1	f) FF6897
g) 2	g) FF9867

Hardware installation



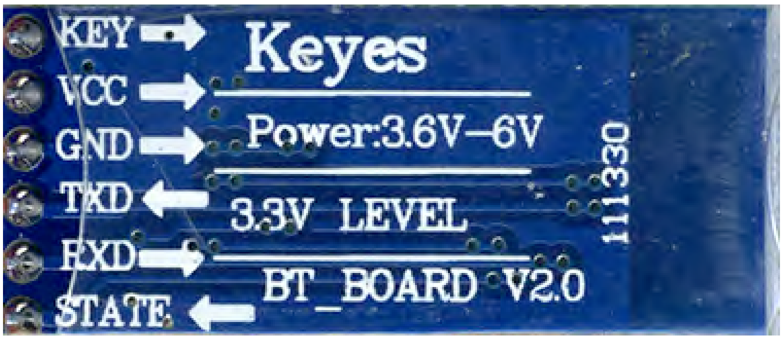
Codes are shown on the LCD1602

4.2.6 Bluetooth Module

This inexpensive HC06 Bluetooth module is used to control the robot by sending data between an Arduino or compatible board and a Bluetooth-equipped device such as an Android smartphone or tablet.

Here we will guide you through the communication between Arduino and Bluetooth device running serial terminal software, in this case, an Android pad. Please note that this control process is not compatible with IOS devices.

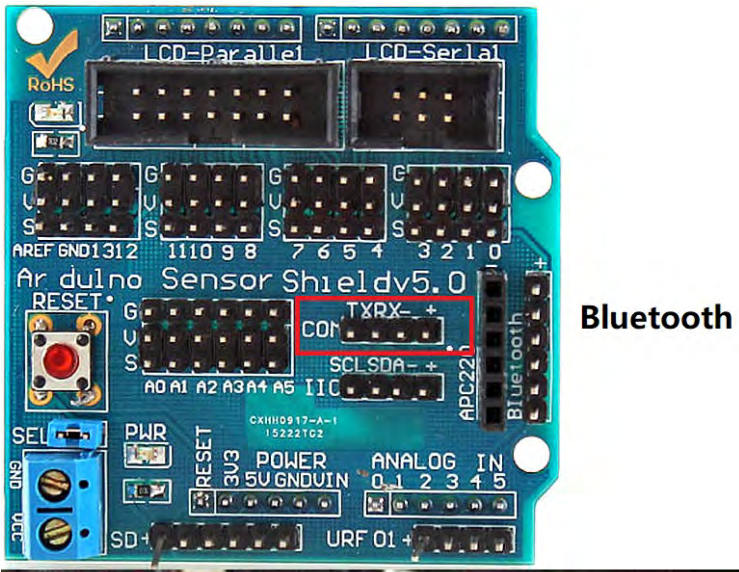
So, let’s get started! Our end goal for this Bluetooth module is to send data from your android devices and display information on LCD1602 previous you just built.



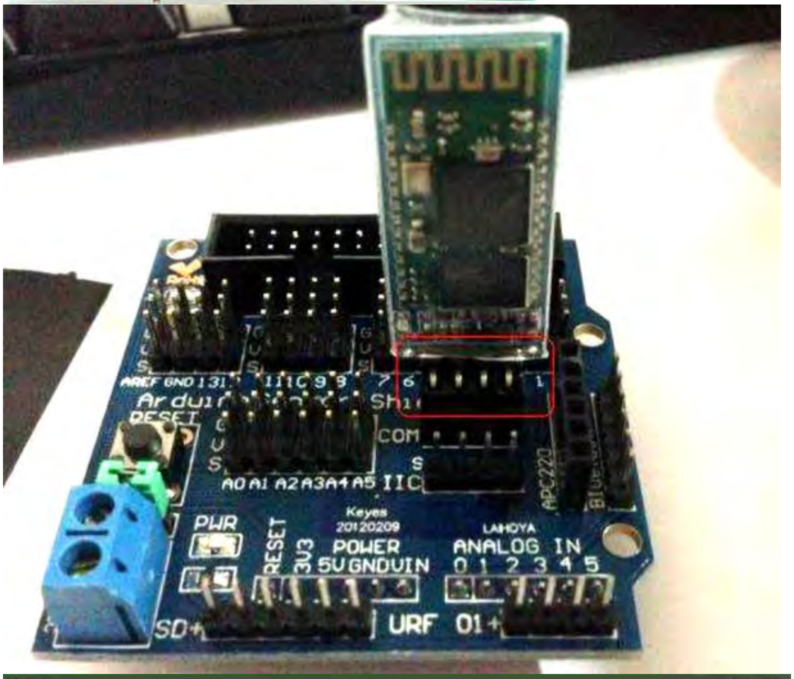
Pins configuration:

- GND: System / Arduino Ground
- VCC
- STATE: Tells if connected or not
- KEY: If brought HIGH before power is applied, forces AT Command Setup Mode. LED blinks slowly (2 seconds)
- TXD: Transmit Serial Data from HC-05 to Arduino Serial Receive. NOTE: 3.3V HIGH level: OK for Arduino
- RXD: Receive Serial Data from Arduino Serial Transmit

Hardware installation

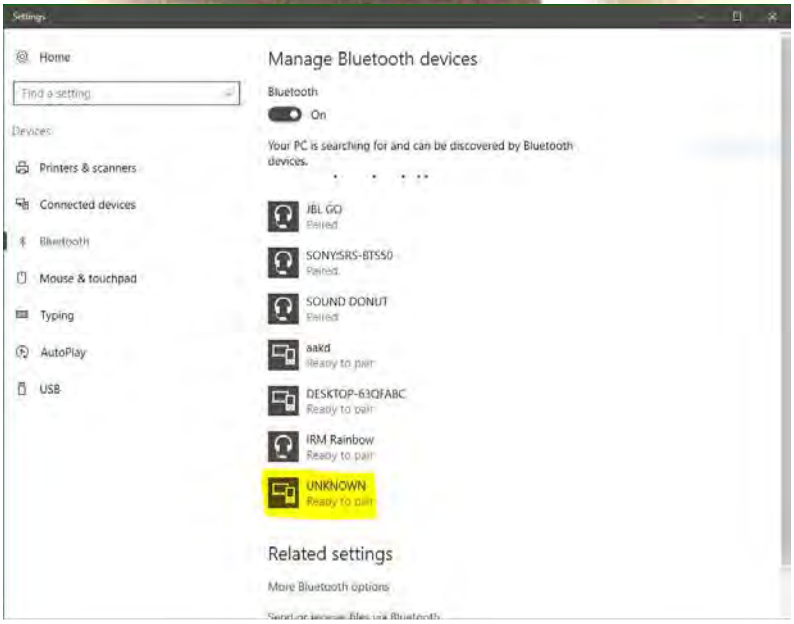


Plug the HC06 module in the corresponding COM slot



Once the Bluetooth module has been connected and power applied, the LED will blink rapidly. This means that it has not been "paired" with another Bluetooth device. The LED stays on continuously when paired.

Use USB cable to connect your Arduino UNO board and your PC, in this case, we are using Windows 10



Settings

Home

Find a setting

Devices

Printers & scanners

Connected devices

Bluetooth

Mouse & touchpad

Typing

AutoPlay

USB

Manage Bluetooth devices

Bluetooth

On

Your PC is searching for and can be discovered by Bluetooth devices.

JBL GO
Paired

SONY:SRS-BTS50
Paired

SOUND DONUT
Paired

aakd
Ready to pair

DESKTOP-63QFABC
Ready to pair

IRM Rainbow
Ready to pair

UNKNOWN
Ready to pair

Related settings

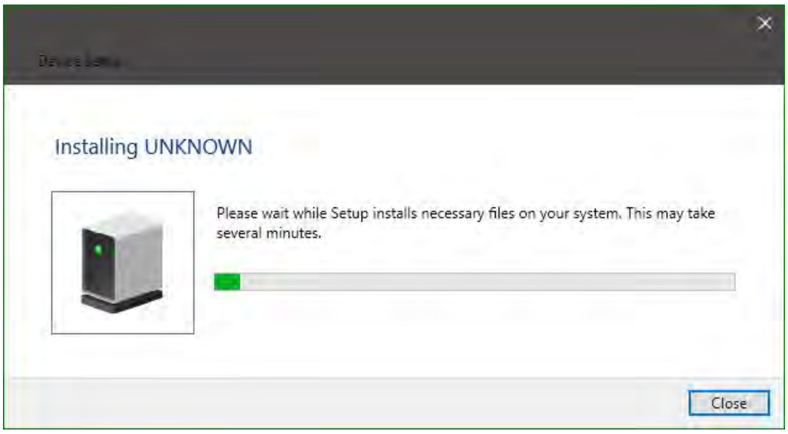
More Bluetooth options

Send or receive files via Bluetooth

Open the Settings | Devices | Bluetooth

Then you will notice there is new Unknown device

Hardware installation



Device Setup

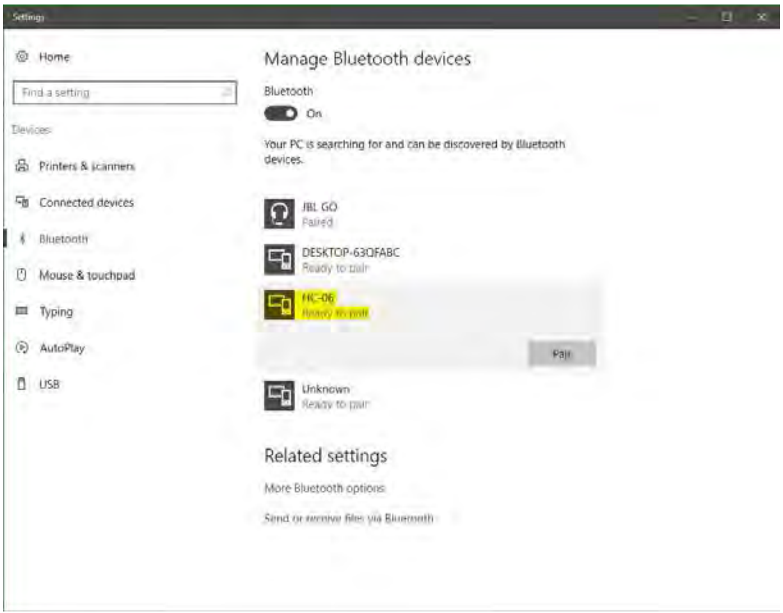
Installing UNKNOWN

Please wait while Setup installs necessary files on your system. This may take several minutes.

PC will automatically install the driver for HC-06 Bluetooth module

Wait for a second to download and install driver

If no response, then disconnect this module and re-insert in the COM slot



Settings

Manage Bluetooth devices

Bluetooth

On

Your PC is searching for and can be discovered by Bluetooth devices.

JBL GO
Paired

DESKTOP-63QFABC
Ready to pair

HC-06
Ready to pair

Unknown
Ready to pair

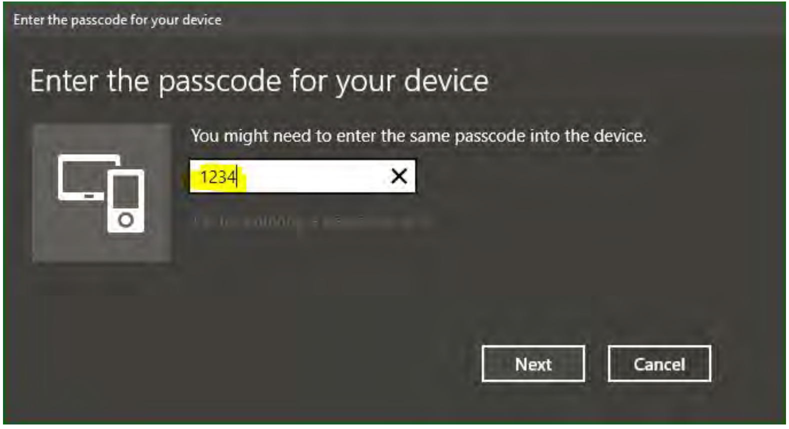
Related settings

More Bluetooth options

Send or receive files via Bluetooth

The HC-06 icon is shown and ready to pair.

Press pair to communicate with HC-06



Enter the passcode for your device

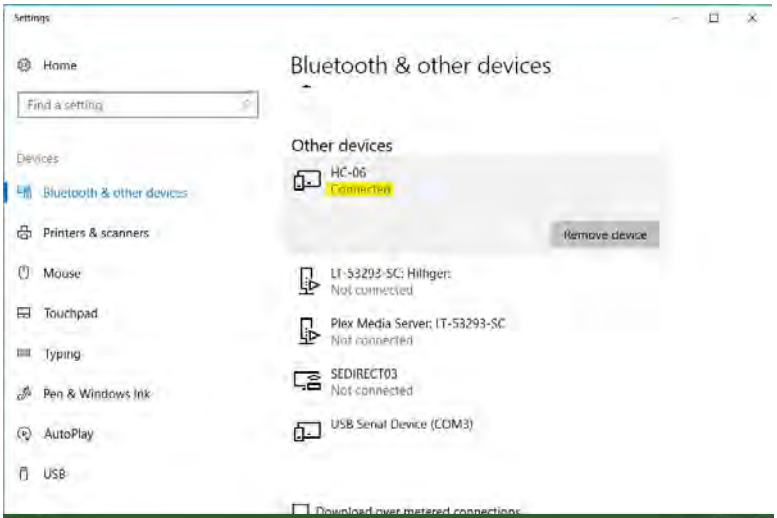
Enter the passcode for your device

You might need to enter the same passcode into the device.

1234

The default passcode is 1234. Then hit the Next Button

Hardware installation



Bluetooth & other devices

Other devices

HC-06
Connected

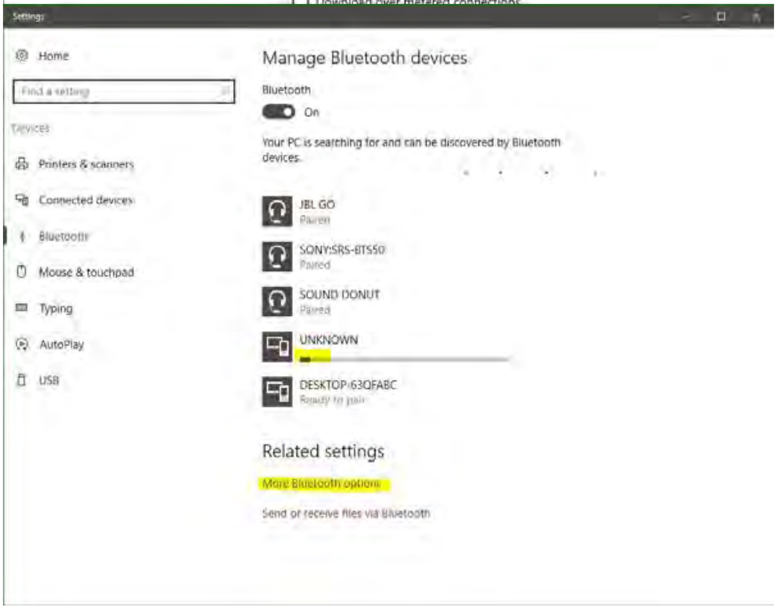
LT-53293-SC: Hilfiger:
Not connected

Plex Media Server: LT-53293-SC
Not connected

SEDIRECT03
Not connected

USB Serial Device (COM3)

□ Download over metered connections



Manage Bluetooth devices

Bluetooth
On

Your PC is searching for and can be discovered by Bluetooth devices.

JBL GO
Paired

SONY:SRS-BTS50
Paired

SOUND DONUT
Paired

UNKNOWN

DESKTOP-63QFABC
Ready to pair

Related settings

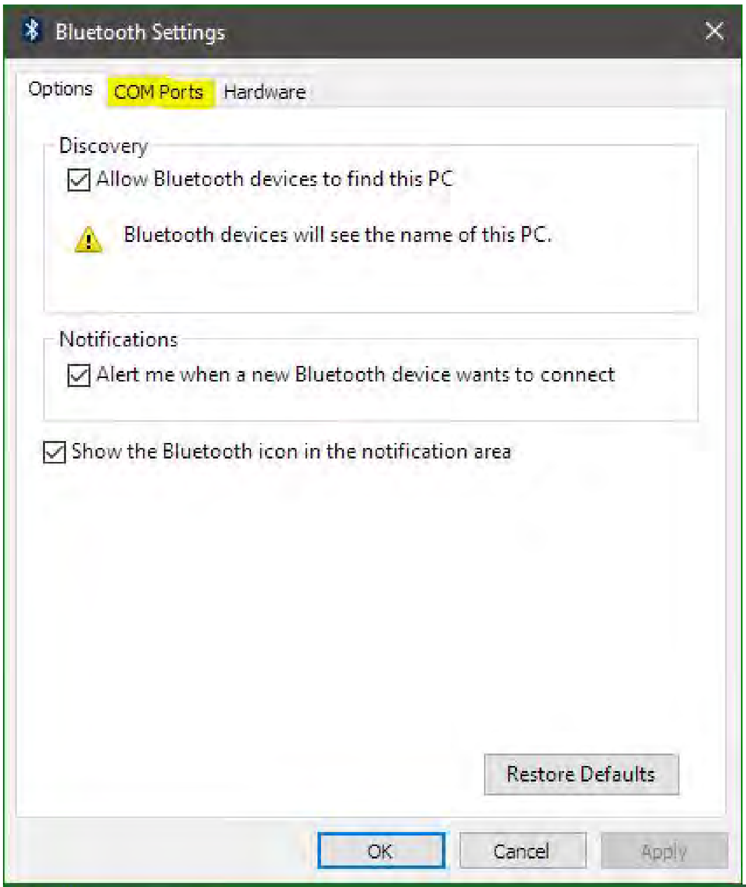
More Bluetooth options

Send or receive files via Bluetooth

Successfully connected. Now you will notice the red LED on the module is on continuously.

Click the More Bluetooth Options.

Hardware installation



Discovery

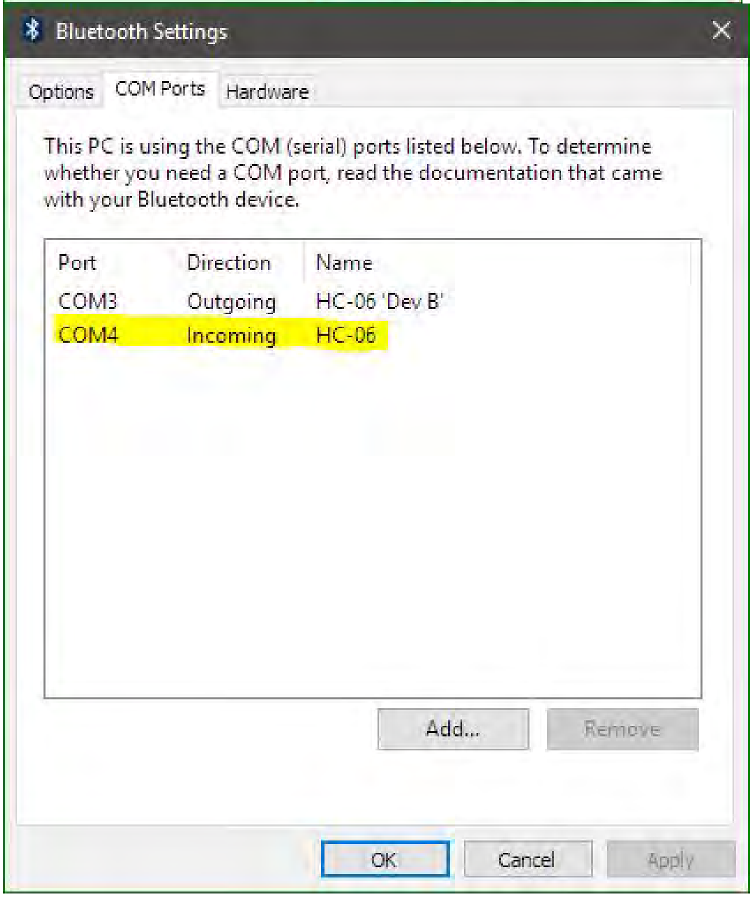
☑ Allow Bluetooth devices to find this PC

Bluetooth devices will see the name of this PC.

Notifications

☑ Alert me when a new Bluetooth device wants to connect

☑ Show the Bluetooth icon in the notification area



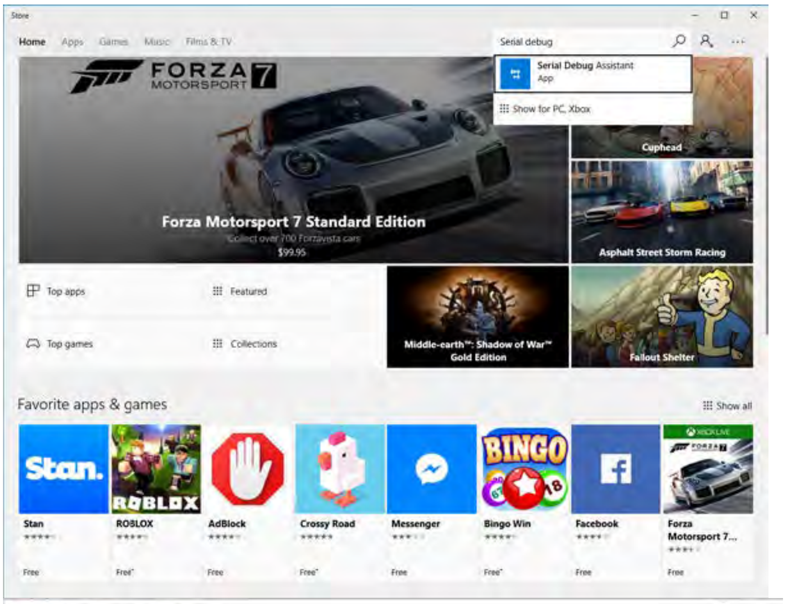
This PC is using the COM (serial) ports listed below. To determine whether you need a COM port, read the documentation that came with your Bluetooth device.

Port	Direction	Name
COM3	Outgoing	HC-06 'Dev B'
COM4	Incoming	HC-06

New dialog window pops up and choose the COM ports.

Note down the incoming Port number, in this case, it is shown as COM4, as required to configurate in the next step

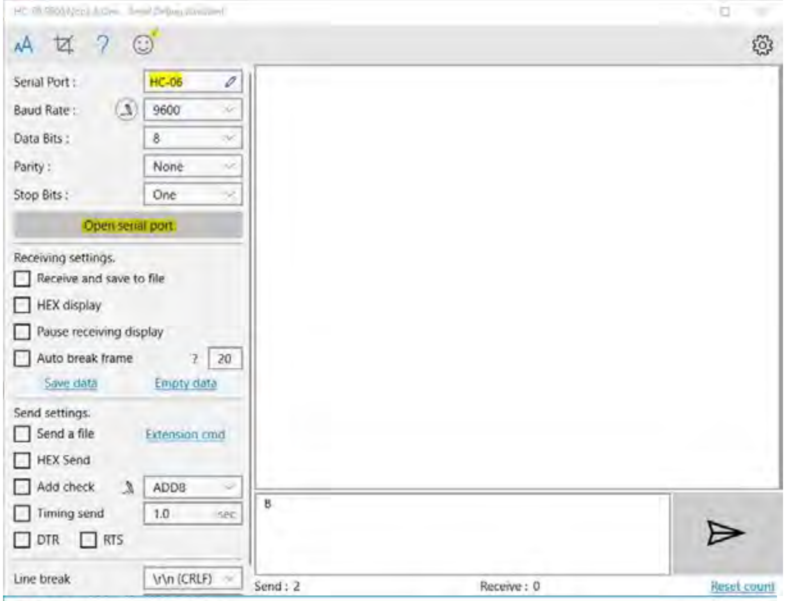
Hardware installation



Download the Serial Debug Assistant from Windows store.

Search the “Serial Debug Assistant” in the right corner search bar and install it

Open the software


Serial Port : HC-06
Baud Rate : 9600
Data Bits : 8
Parity : None
Stop Bits : One
Open serial port

Receiving settings.
□ Receive and save to file
□ HEX display
□ Pause receiving display
□ Auto break frame ? 20
Save data Empty data

Send settings.
□ Send a file Extension cmd
□ HEX Send
□ Add check ADD8
□ Timing send 1.0 sec
□ DTR □ RTS
Line break \r\n (CRLF)

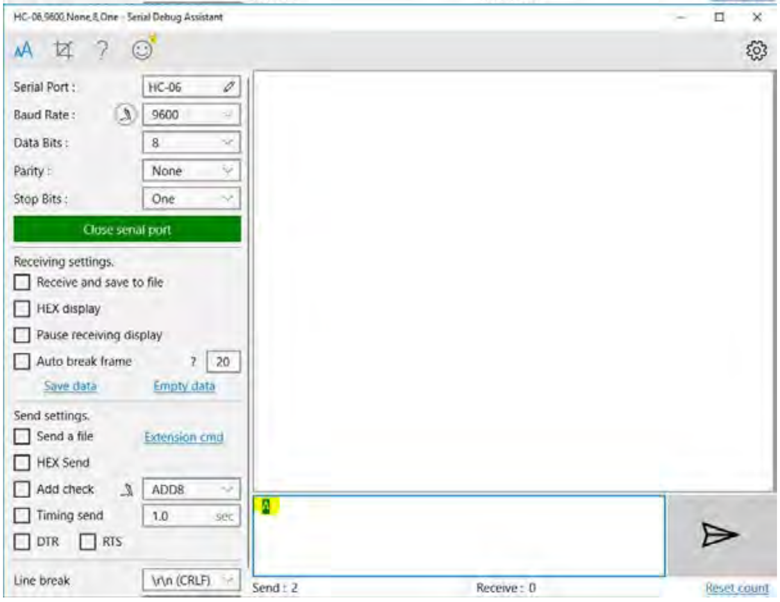
B

Send : 2 Receive : 0 Reset count

Choose the HC-06 device I the serial port (COM4 in this instance) and keep other settings default as shown on the left.

Baud Rate: 9600 bps
Data : 8 bits
Stop Bits: 1 bit
Parity : None
Handshake: None

Open the serial port


HC-06,9600,None,8,One - Serial Debug Assistant

Serial Port : HC-06
Baud Rate : 9600
Data Bits : 8
Parity : None
Stop Bits : One
Close serial port

Receiving settings.
□ Receive and save to file
□ HEX display
□ Pause receiving display
□ Auto break frame ? 20
Save data Empty data

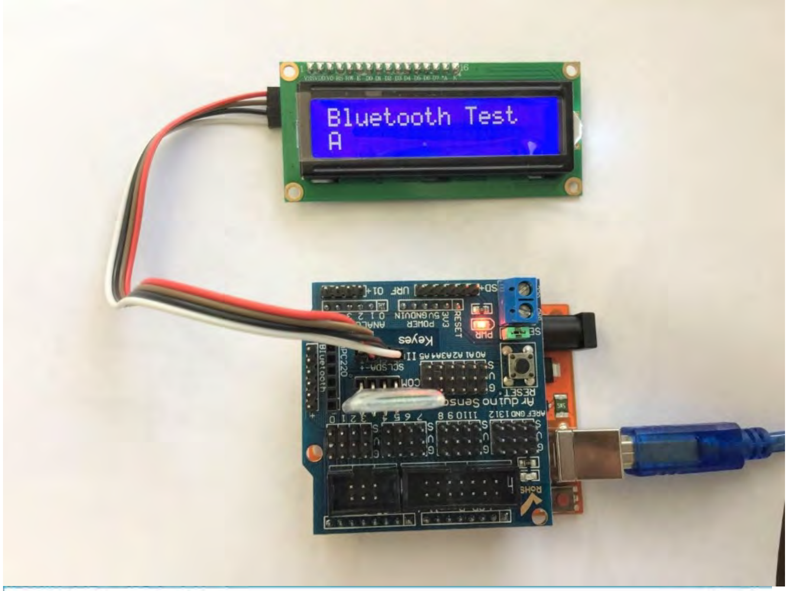
Send settings.
□ Send a file Extension cmd
□ HEX Send
□ Add check ADD8
□ Timing send 1.0 sec
□ DTR □ RTS
Line break \r\n (CRLF)

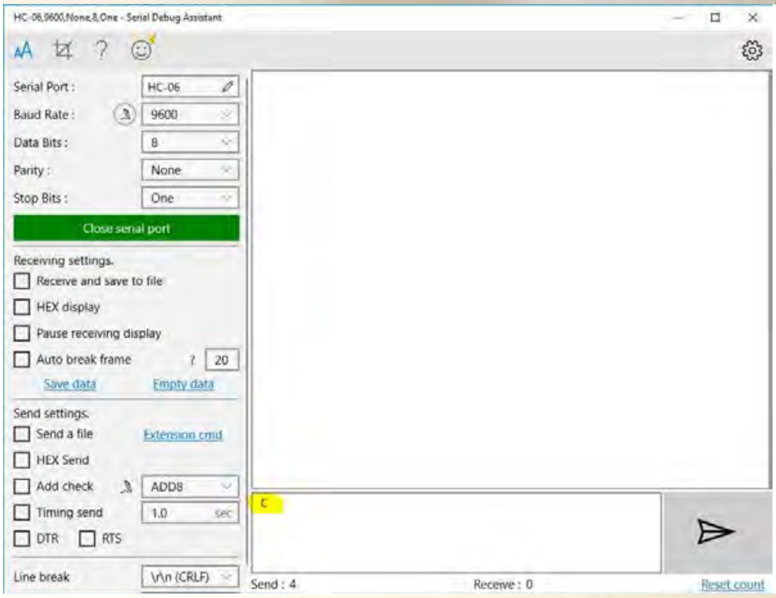
A

Send : 2 Receive : 0 Reset count

Type in character “A” and you will see it on the LCD1602

Hardware installation




HC-06,9600,None,8,One - Serial Debug Assistant

Serial Port : HC-06
Baud Rate : 9600
Data Bits : 8
Parity : None
Stop Bits : One

Close serial port

Receiving settings
□ Receive and save to file
□ HEX display
□ Pause receiving display
□ Auto break frame ? 20
Save data Empty data

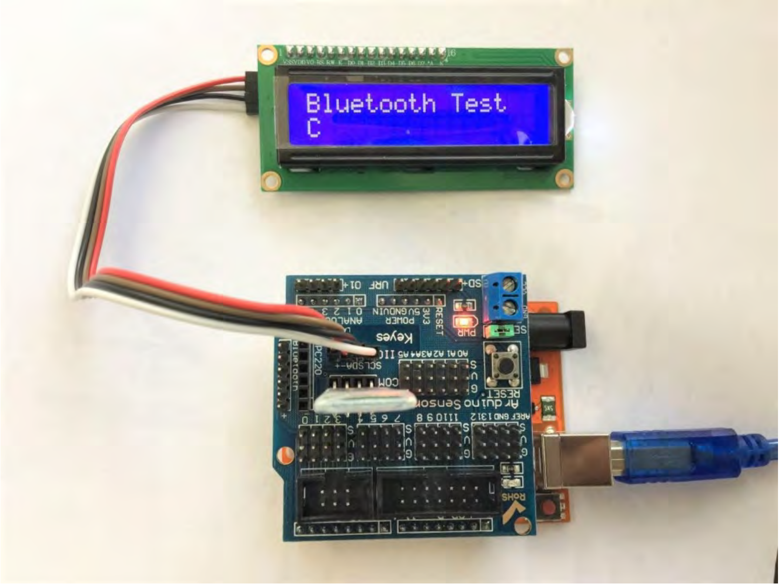
Send settings
□ Send a file Extension cmd
□ HEX Send
□ Add check ADD8
□ Timing send 1.0 sec
□ DTR □ RTS

Line break \r\n (CRLF)

C

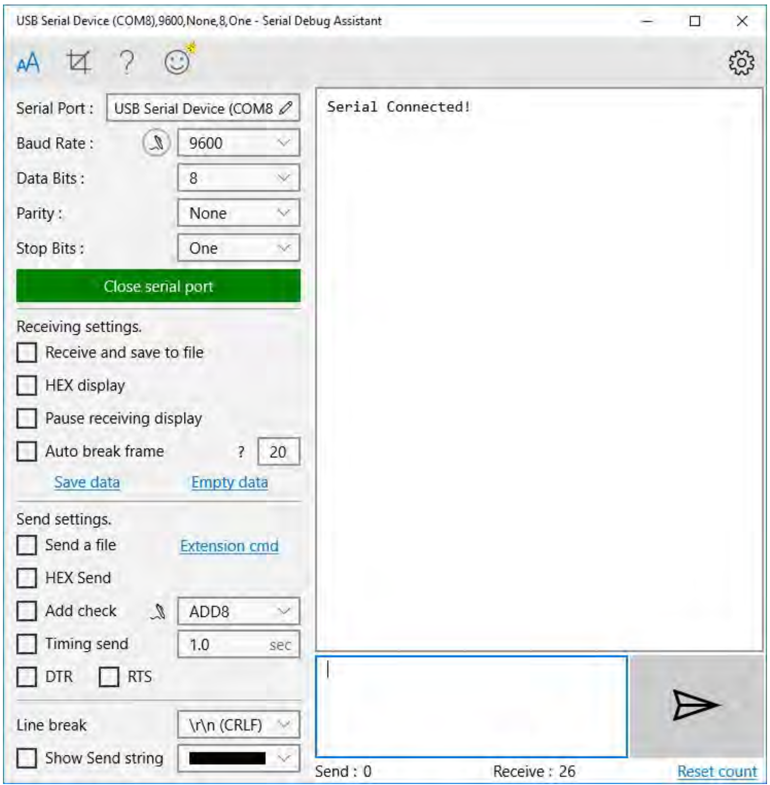
Send : 4 Receive : 0 Reset count

Try other character like a “C”



Now you have finished the Bluetooth test

Hardware installation



USB Serial Device (COM8),9600,None,8,One - Serial Debug Assistant

Serial Port : USB Serial Device (COM8
Baud Rate : 9600
Data Bits : 8
Parity : None
Stop Bits : One

Receiving settings.
□ Receive and save to file
□ HEX display
□ Pause receiving display
□ Auto break frame ? 20

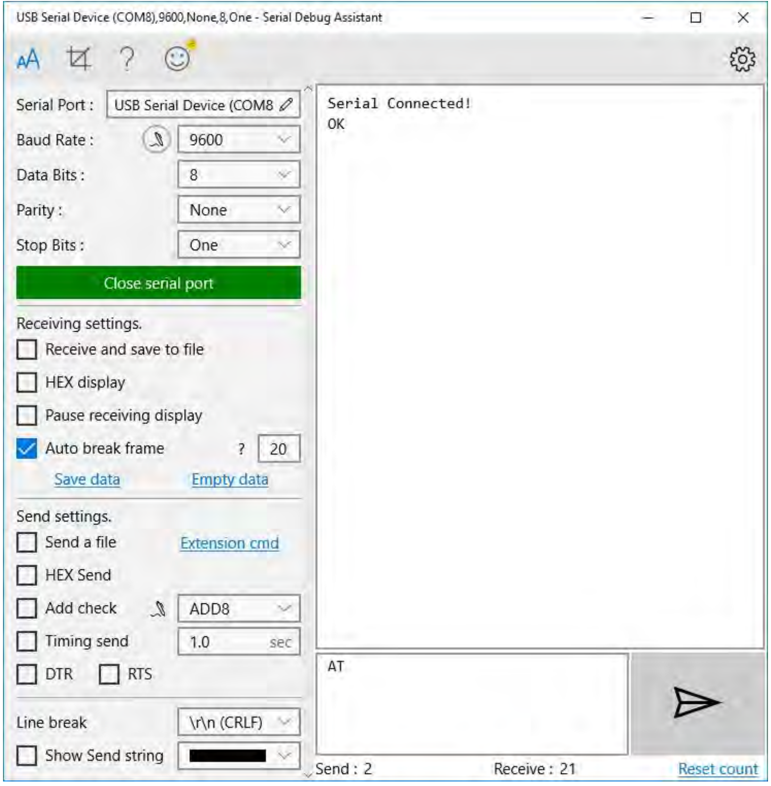
Send settings.
□ Send a file
□ HEX Send
□ Add check ADD8
□ Timing send 1.0 sec
□ DTR □ RTS

Line break \r\n (CRLF)
□ Show Send string

```
Serial Connected!
```

Send : 0 Receive : 26

The module has two modes of operation, Command Mode where we can send AT commands to it and Data Mode where it transmits and receives data to another Bluetooth module. You can also change the name or password of your unique Bluetooth module.



USB Serial Device (COM8),9600,None,8,One - Serial Debug Assistant

Serial Port : USB Serial Device (COM8
Baud Rate : 9600
Data Bits : 8
Parity : None
Stop Bits : One

Receiving settings.
□ Receive and save to file
□ HEX display
□ Pause receiving display
☑ Auto break frame ? 20

Send settings.
□ Send a file
□ HEX Send
□ Add check ADD8
□ Timing send 1.0 sec
□ DTR □ RTS

Line break \r\n (CRLF)
□ Show Send string

```
Serial Connected!
OK
```

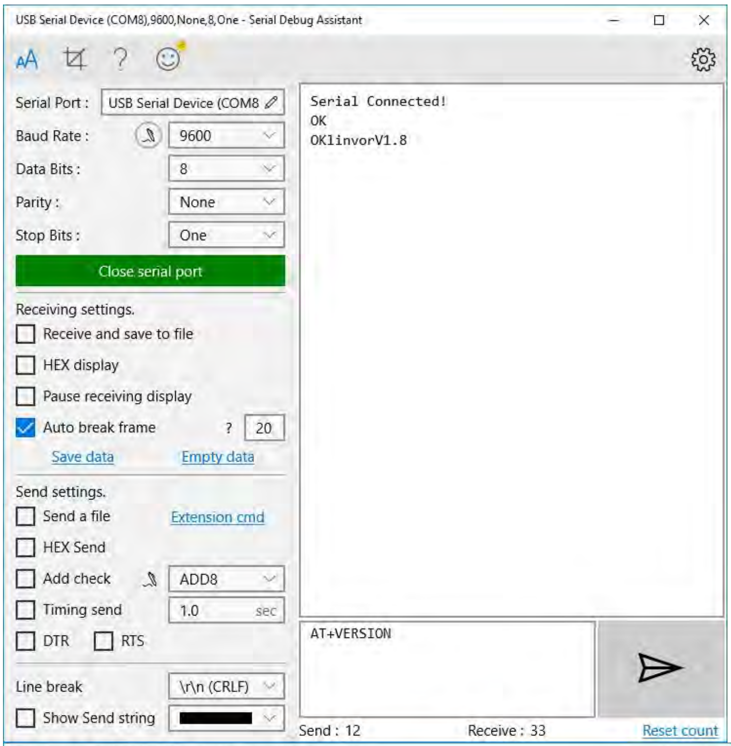
```
AT
```

Send : 2 Receive : 21

Command Mode Commands:
The format of commands is:
Always starts with "AT"
Then "+" followed by <ParameterName>
Then either:
? (returns current value of parameter)
= (New Value of parameter)

A few examples:
AT (AT Test command. Should respond with OK)
AT+VERSION? (show the firmware version)
AT+UART=9600,0,0 (Set baud rate to 9600, 1 stop bit, no parity)

Hardware installation


USB Serial Device (COM8),9600,None,8,One - Serial Debug Assistant

Serial Port : USB Serial Device (COM8
Baud Rate : 9600
Data Bits : 8
Parity : None
Stop Bits : One

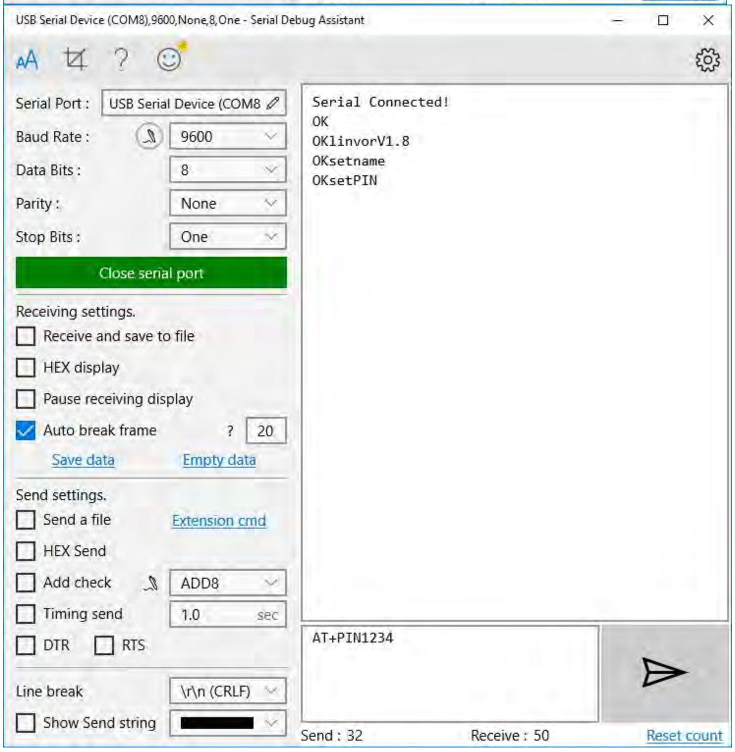
Receiving settings.
□ Receive and save to file
□ HEX display
□ Pause receiving display
☑ Auto break frame ? 20

Send settings.
□ Send a file
□ HEX Send
□ Add check ADD8
□ Timing send 1.0 sec
□ DTR □ RTS
Line break \r\n (CRLF)
□ Show Send string

```
Serial Connected!
OK
OKlinvorV1.8
```

```
AT+VERSION
```

Send : 12 Receive : 33


USB Serial Device (COM8),9600,None,8,One - Serial Debug Assistant

Serial Port : USB Serial Device (COM8
Baud Rate : 9600
Data Bits : 8
Parity : None
Stop Bits : One

Receiving settings.
□ Receive and save to file
□ HEX display
□ Pause receiving display
☑ Auto break frame ? 20

Send settings.
□ Send a file
□ HEX Send
□ Add check ADD8
□ Timing send 1.0 sec
□ DTR □ RTS
Line break \r\n (CRLF)
□ Show Send string

```
Serial Connected!
OK
OKlinvorV1.8
OKsetname
OKsetPIN
```

```
AT+PIN1234
```

Send : 32 Receive : 50

This step is optional:

Renaming your Bluetooth device:

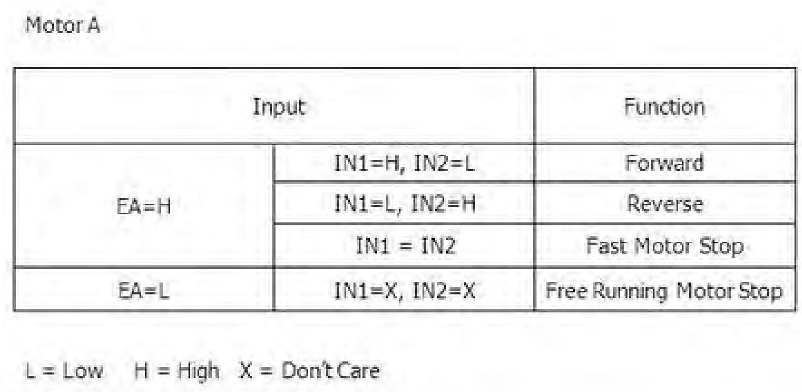
- Check firmware version.
- Change name
- Change PIN

4.2.7 Motor Module

This 4-wheel drive is driven by the H-bridge L289N Motor driver board. This allows you to control the speed and direction of two DC motors, or control one bipolar stepper motor with ease. The L298N H-bridge module can be used with motors that have a voltage of between 5 and 35V DC. There is also an onboard 5V regulator, so if your supply voltage is up to 12V you can also source 5V from the board.

This L289N Motor driver board is ideal for controlling this 4-wheel drive. The speed is modulated by either the enable pins (ENA/ENB) or IN1/IN2/IN3/IN4 with PWM modulation.

Hardware installation

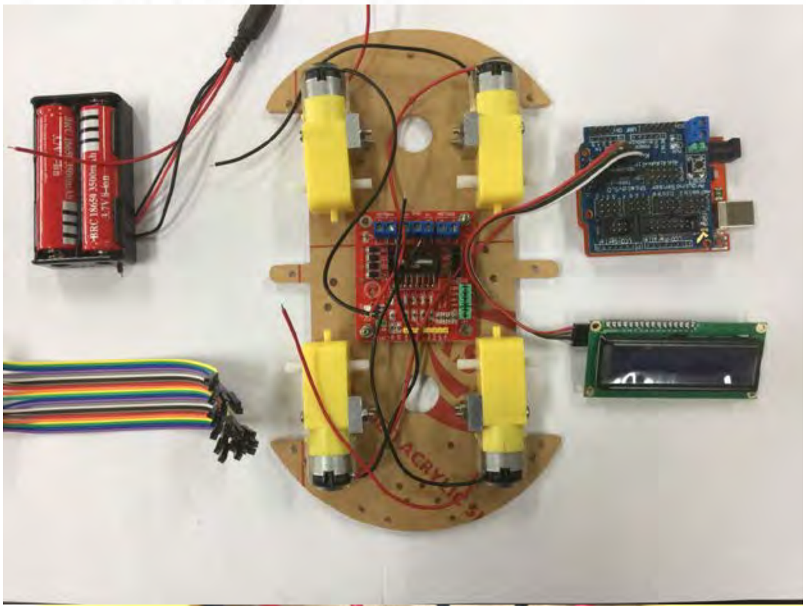


Motor A

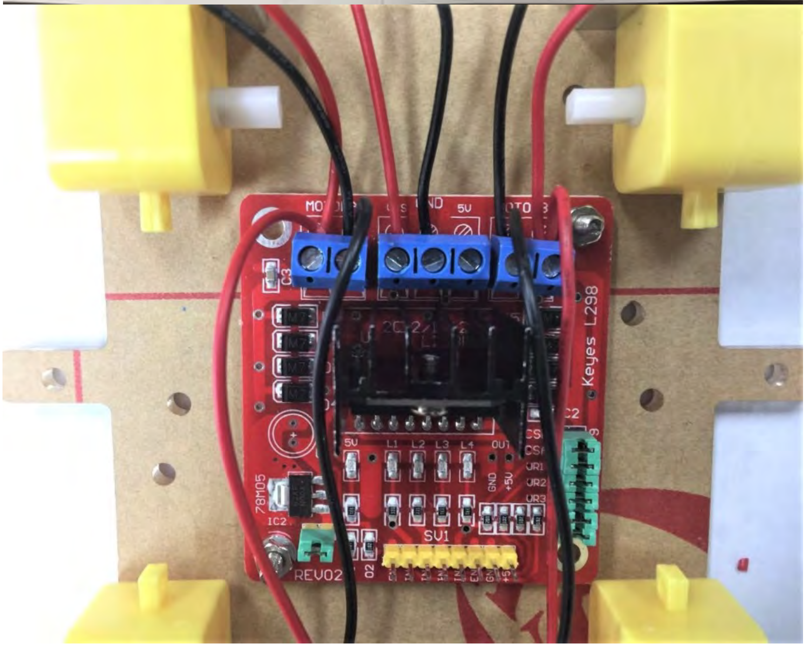
Input		Function
EA=H	IN1=H, IN2=L	Forward
	IN1=L, IN2=H	Reverse
	IN1 = IN2	Fast Motor Stop
EA=L	IN1=X, IN2=X	Free Running Motor Stop

L = Low H = High X = Don't Care

The motor direction is controlled by sending a HIGH or LOW signal to the drive for each motor (or channel). For example, for motor one, a HIGH to IN1 and a LOW to IN2 will cause it to turn in one direction, and a LOW and HIGH will cause it to turn in the other direction.



Prepare the jumper cable wires, LCD2602, Arduino Sensor Shield with UNO board, two 18650 batteries in the Battery Box and chassis (four motors and L289N Motor Driver Board included)



First step is to tighten the motor wire (red is positive and black is negative).

Then connect the Battery wires to L289N Motor Drive board. Red wire to VMS, black wire to GND

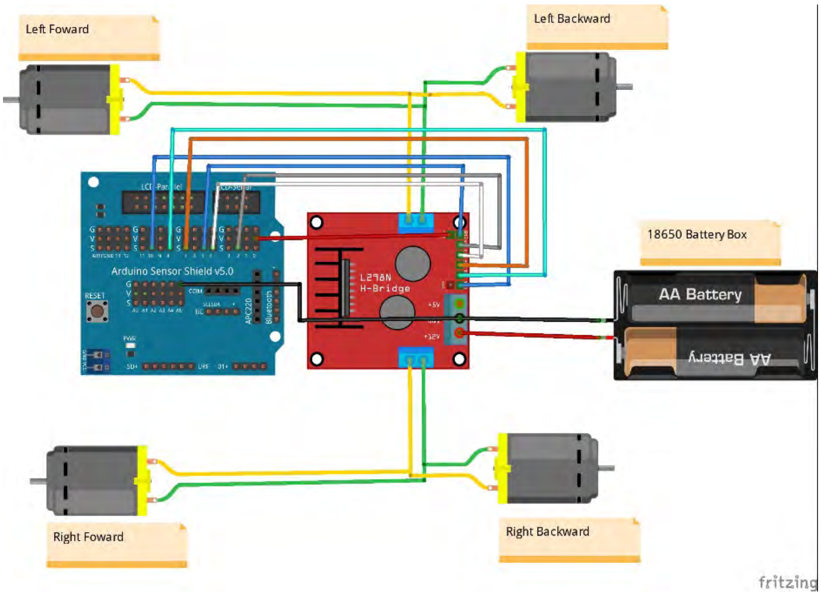
Please carefully secure the battery wires in the correct polarity to avoid burning our the L289N module.

Hardware installation

L289N Pin connection:

- ENA
- ENB
- IN1
- IN2
- IN3
- IN4
- GND
- VCC

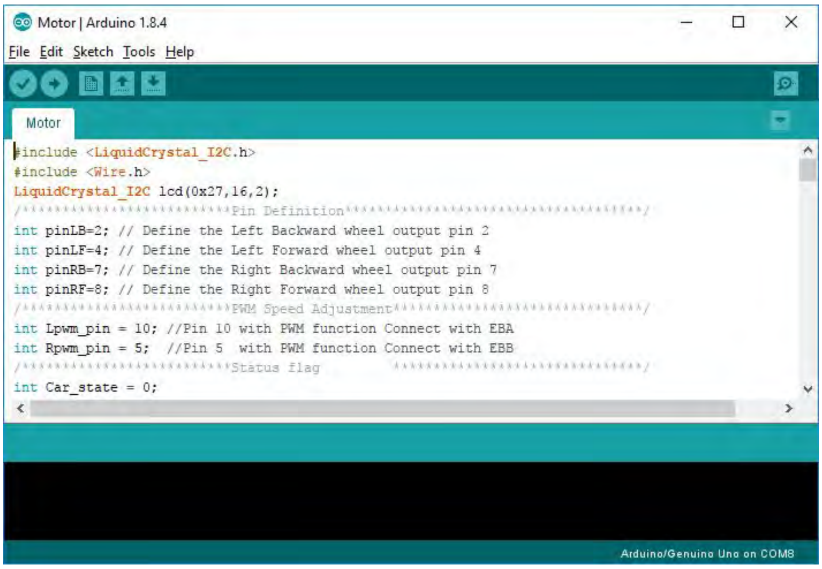
In this instance, left two wheels are controlled by IN1/IN2/ENA, and right two wheels are controlled by IN3/IN4/ENB.



Use jumper cables to configure the connection between four motors and your Arduino Sensor Shield:

- IN1 – pin 2
- IN2 – pin 4
- IN3 – pin 7
- IN4 – pin 8
- ENA – pin 10
- ENB – pin 5

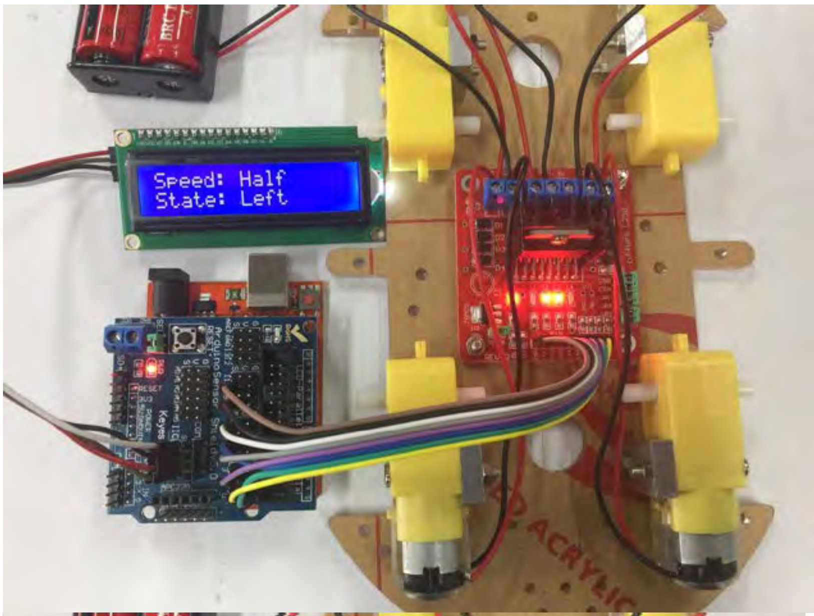
You will notice pin 10 and pin 5 has ~ indicators on the board, which means they are able to output PWM waves.



```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
LiquidCrystal_I2C lcd(0x27,16,2);
/*************************Pin Definition*********************************/
int pinLB=2; // Define the Left Backward wheel output pin 2
int pinLF=4; // Define the Left Forward wheel output pin 4
int pinRB=7; // Define the Right Backward wheel output pin 7
int pinRF=8; // Define the Right Forward wheel output pin 8
/*************************PWM Speed Adjustment***************************/
int Lpwm_pin = 10; //Pin 10 with PWM function Connect with EBA
int Rpwm_pin = 5;  //Pin 5  with PWM function Connect with EBB
/*************************Status flag      ******************************/
int Car_state = 0;
```

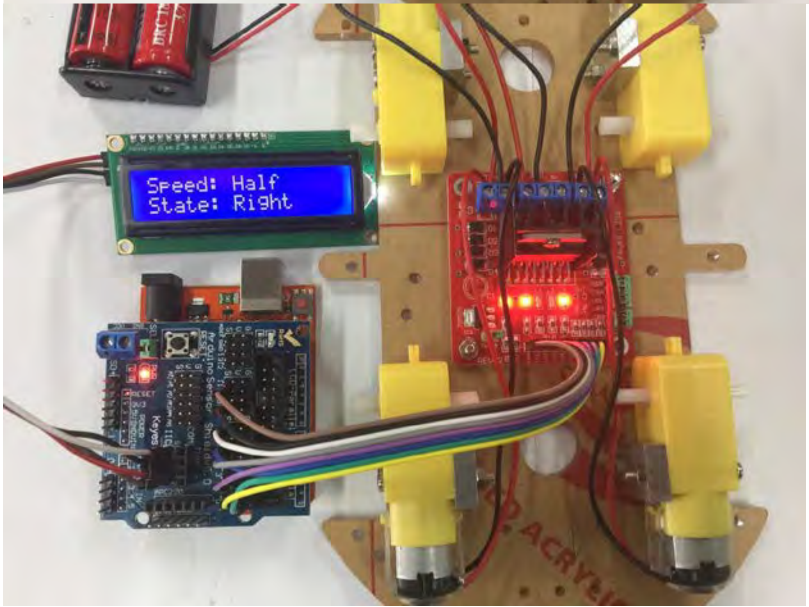
Once you finished the hardware connection, then upload the Motor sketch to your Arduino UNO board

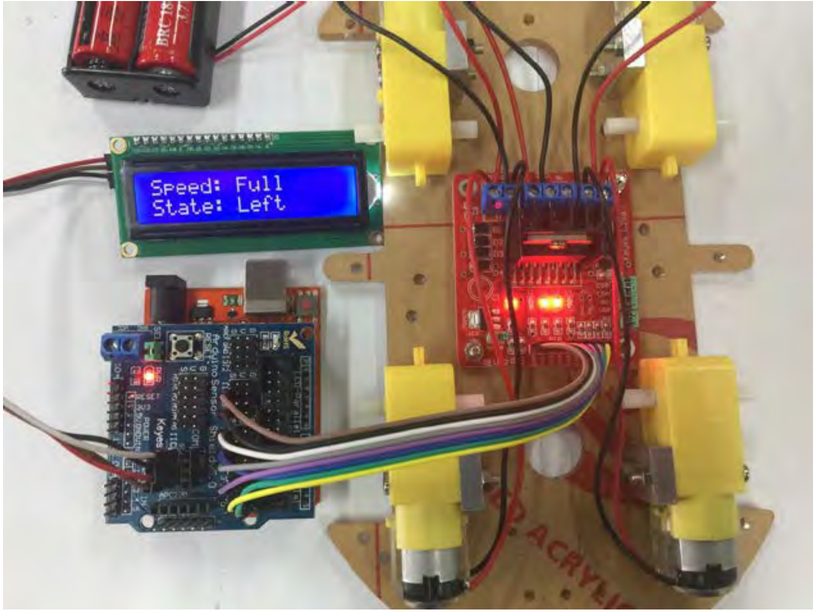
Hardware installation



Motor running status will be displayed on the LCD1602.

Motor speed is modulated by the PWM output pin 10 and pin 5



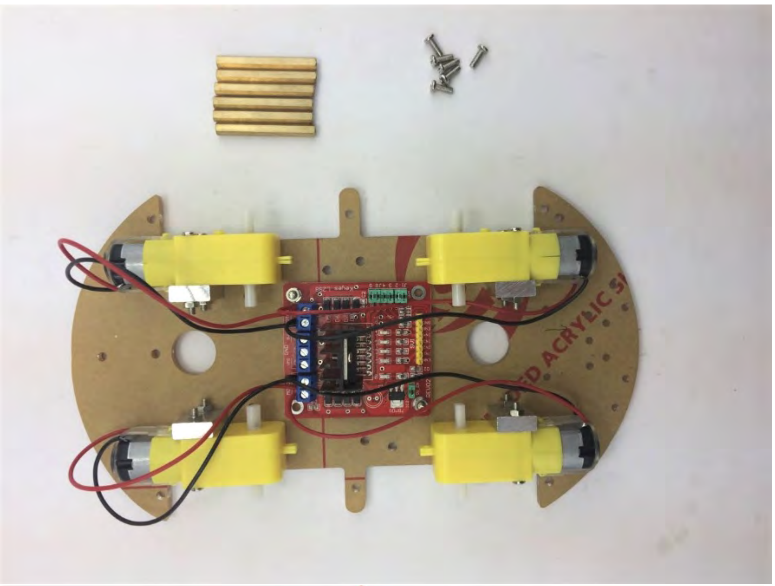


For now, all the sub-modules are working correctly.

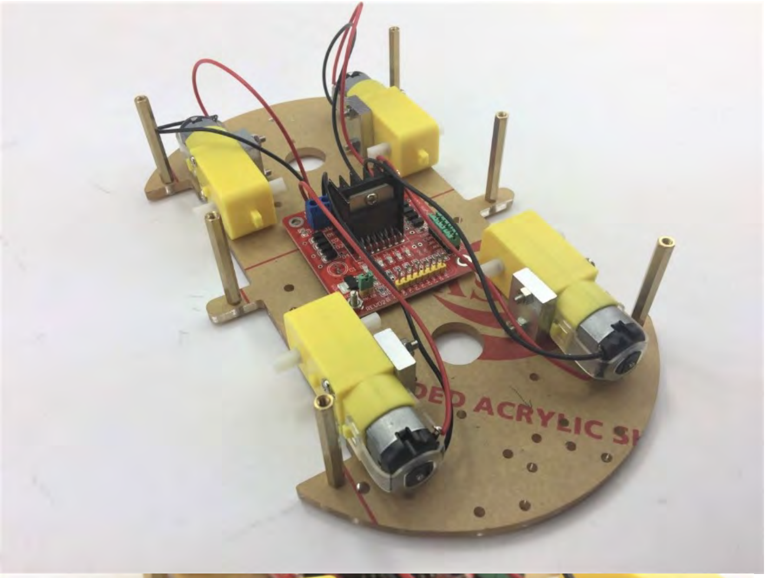
It is time to put everything together.

4.3. 4-wheel Drive installation

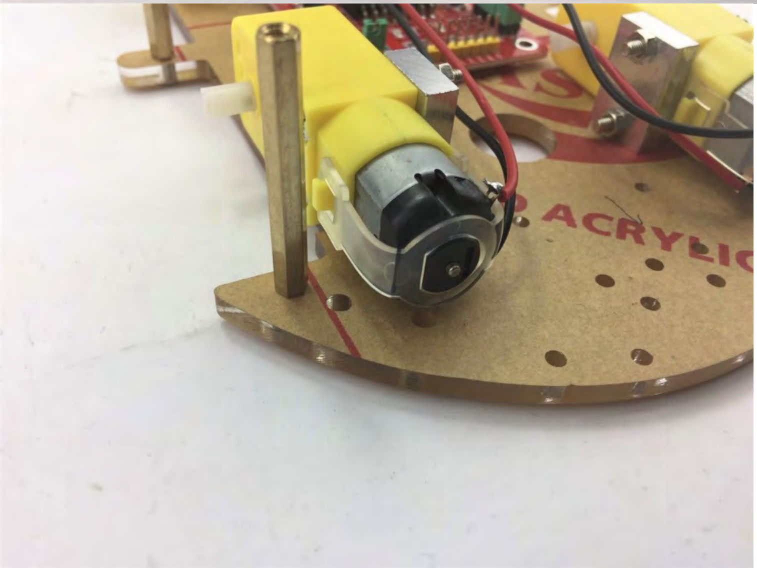
As each separate module works well, now we assembly them together.



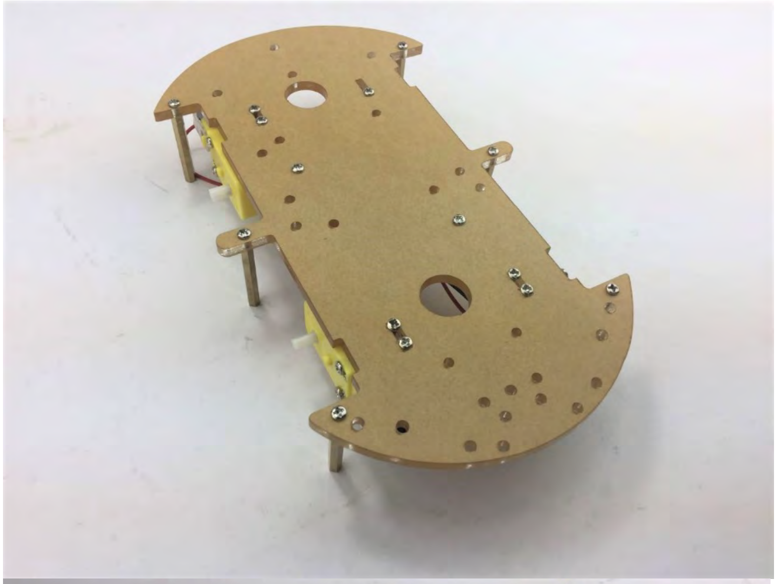
Prepare six M3 * 35mm Spacers and six M3 * 10mm screws



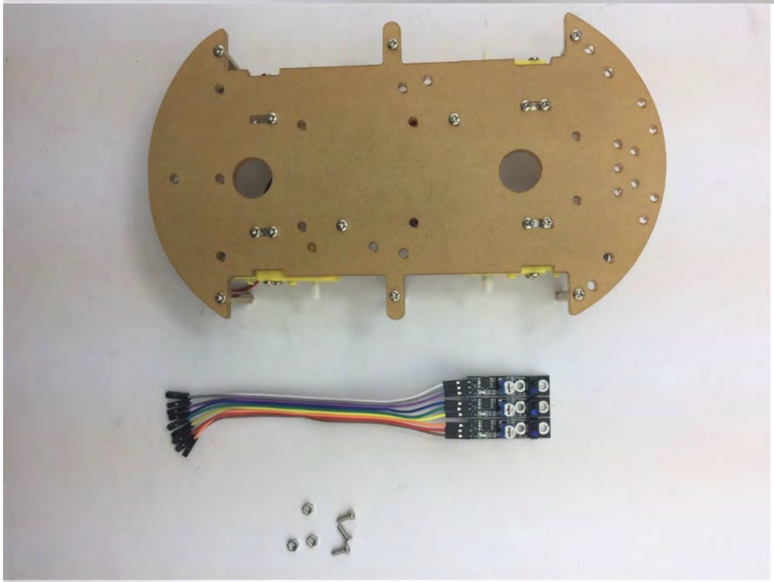
Secure the spacers on the chassis



Hardware installation

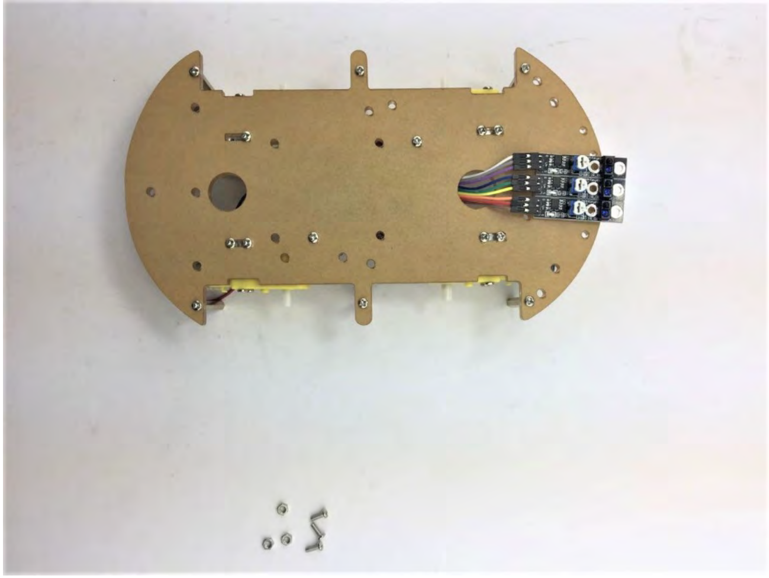


It should look like this



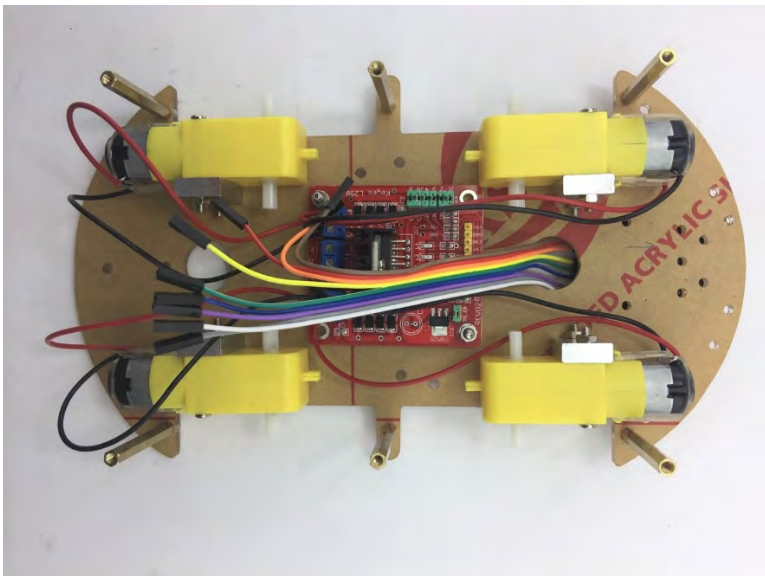
Use Jumper cables to extend connection pins on the Track line sensor.

Three M3 * 8mm screws and nuts to secure the Track line sensor

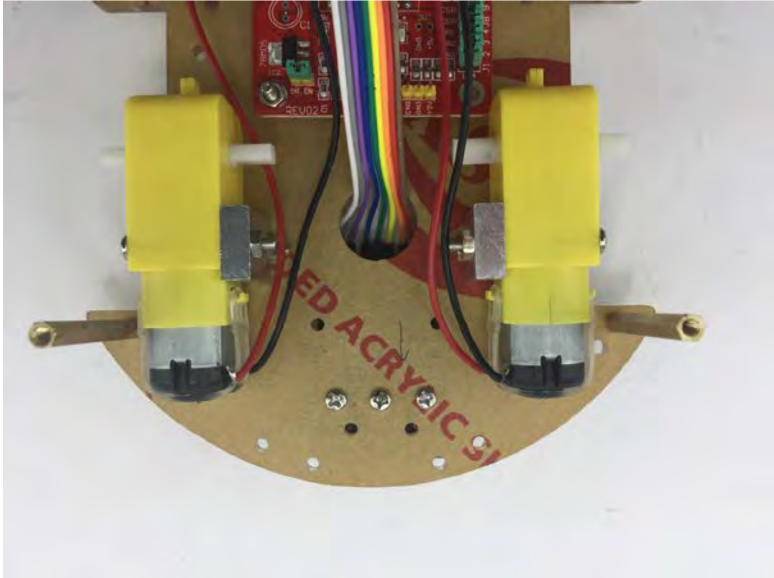


Please note do not change the potentiometer as we have already calibrated.

Hardware installation



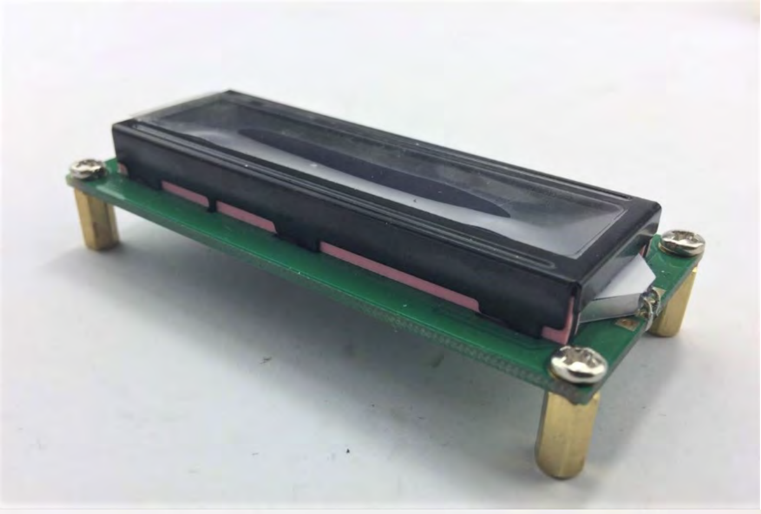
Please Note: The 8 Jumper cables will eventually attach to the Sensor Expansion Board from the L289N board a few steps from now. Please leave the opposite side free for the following steps.

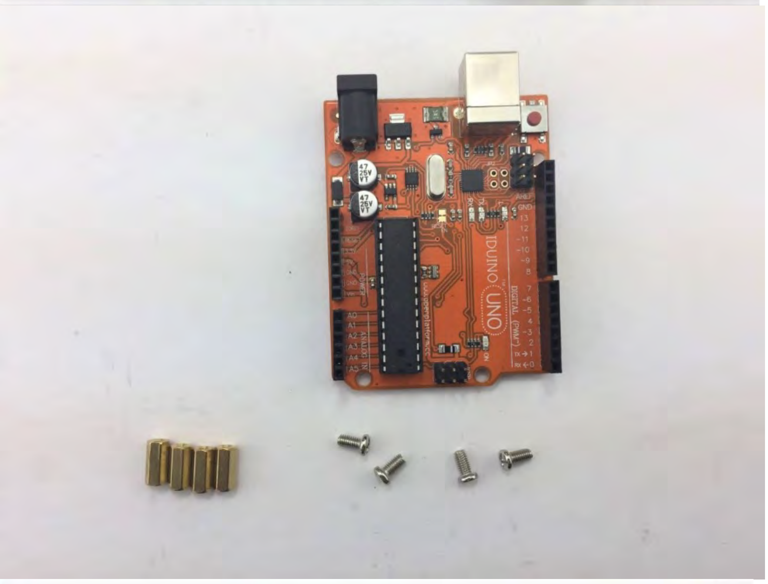


It should look like this

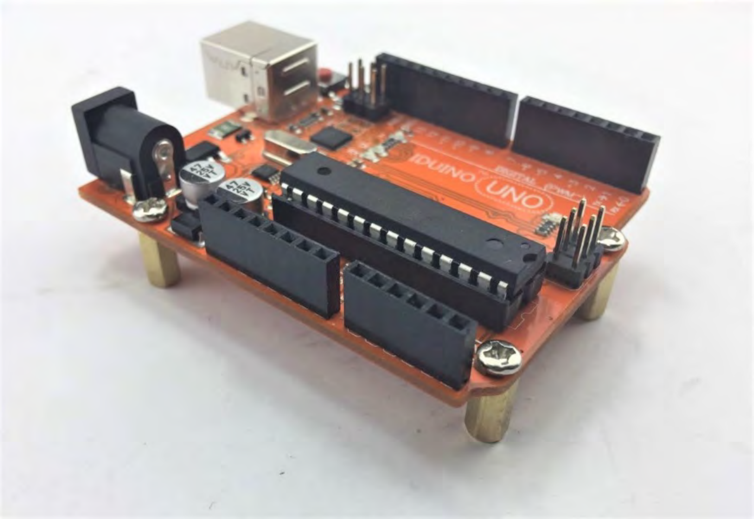
- LCD1602
- Four M3 * 10 mm Spacers
- Four M3 * 6 mm Screws

Hardware installation



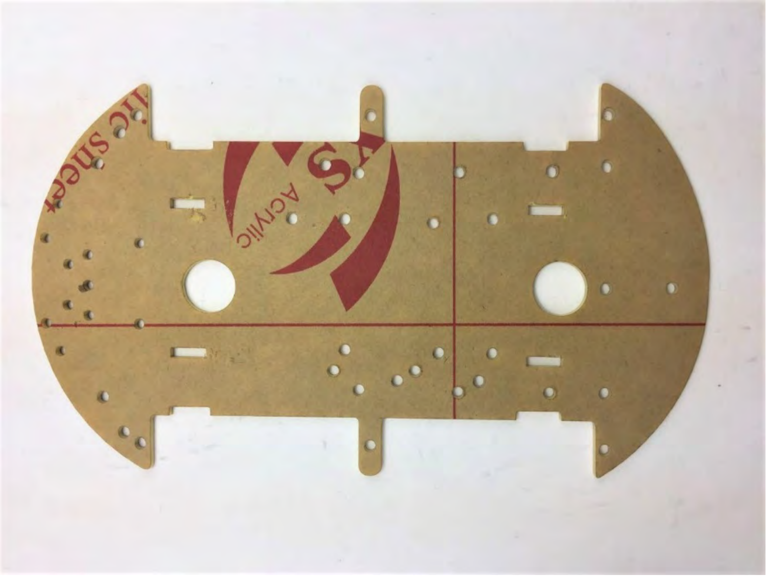


- Arduino UNO Board
- Four M3 * 10mm Spacers
- Four M3 * 6mm Screws

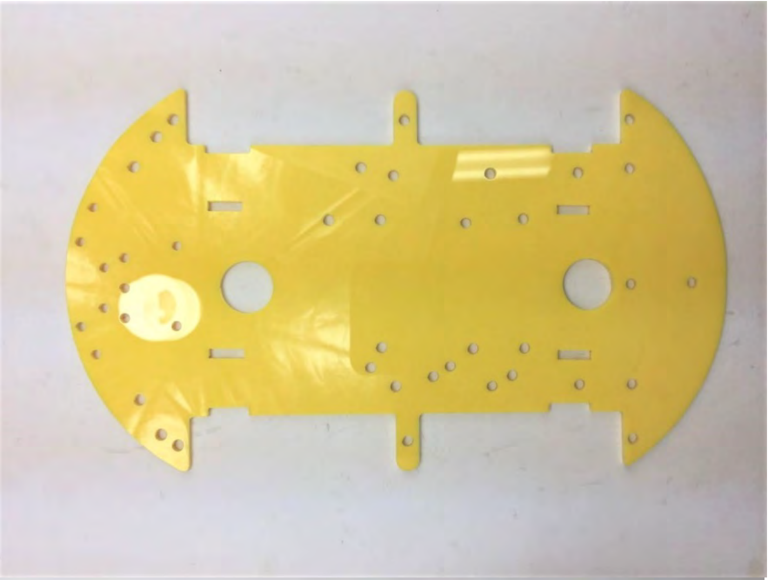


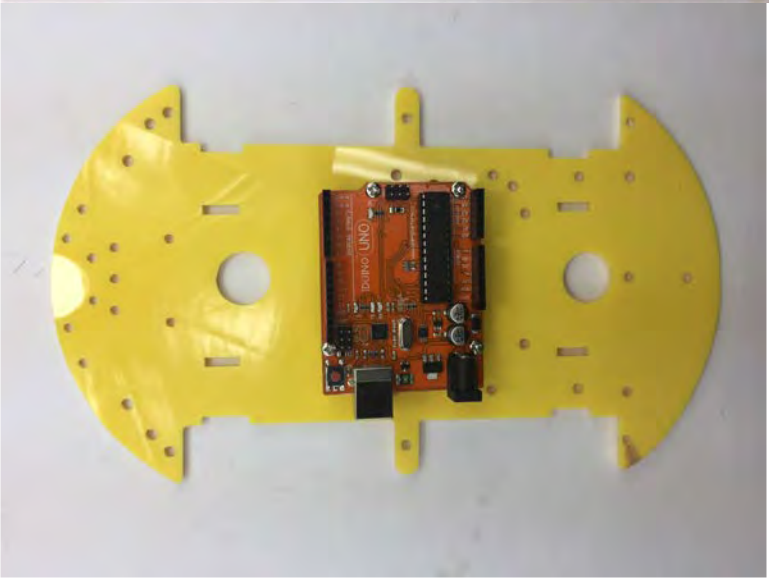
Secure the screws to the Spacers

Hardware installation



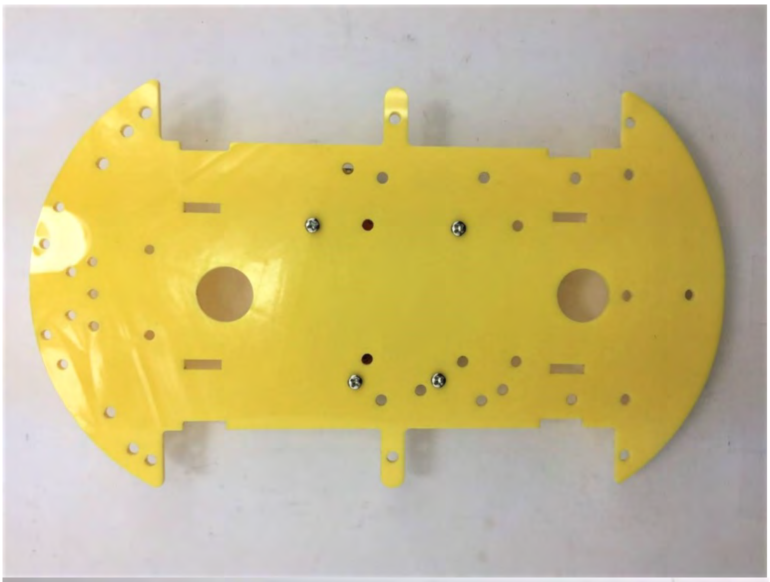
Remove the protection cover of top acrylic chassis



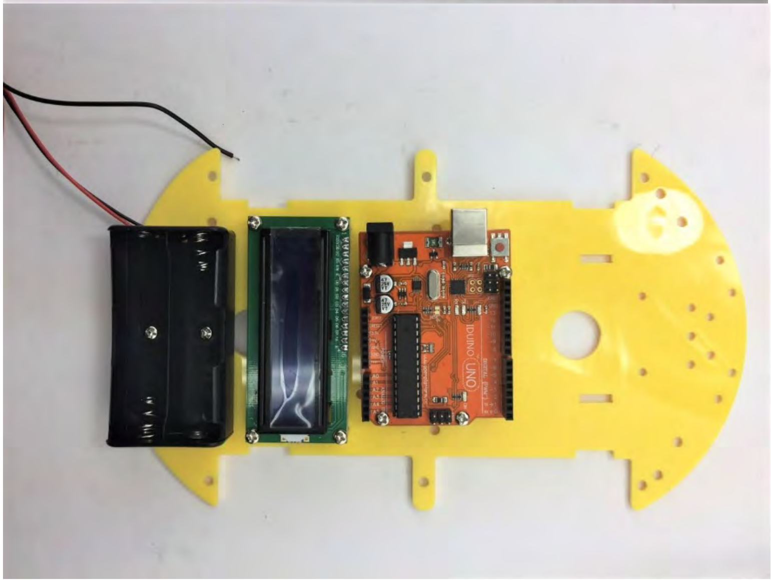


Place the Arduino UNO on the top of the chassis and secure it with four M3 * 8 mm screws

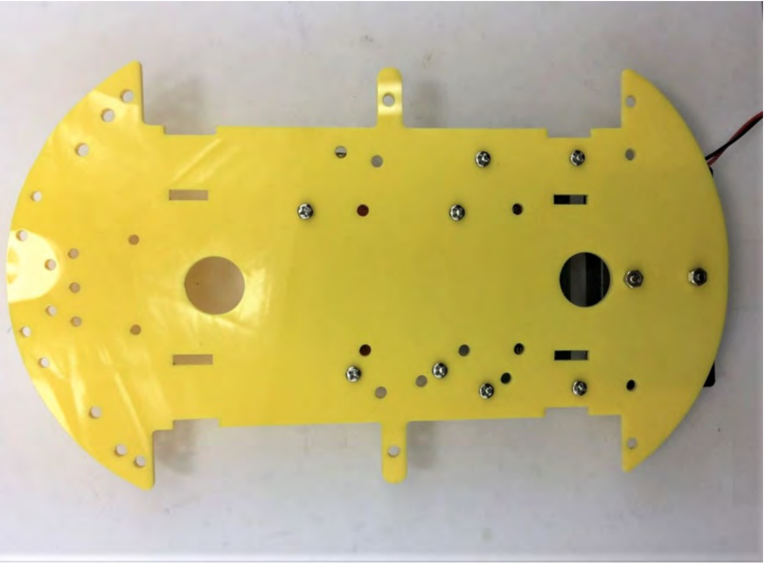
Hardware installation



Please fasten the screws as shown on the left.

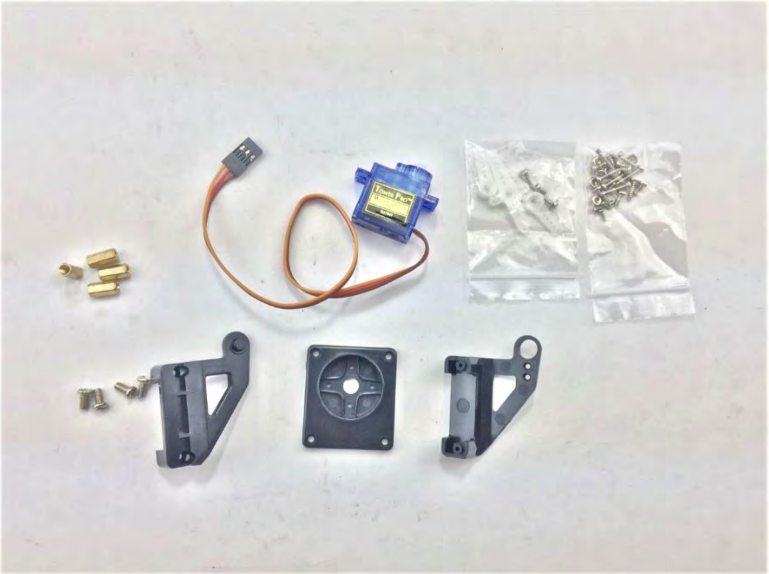


Place the LCD1602 and Battery Box on the top chassis and prepare six M3 * 8 screws to secure them

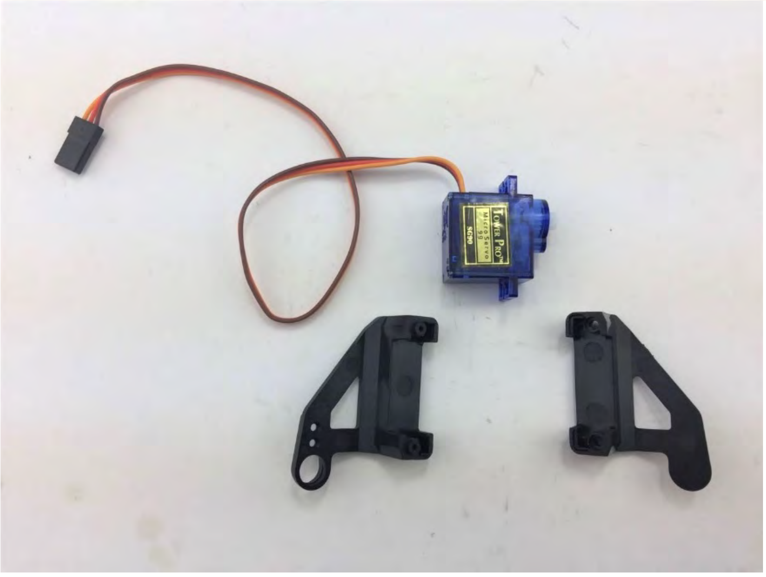


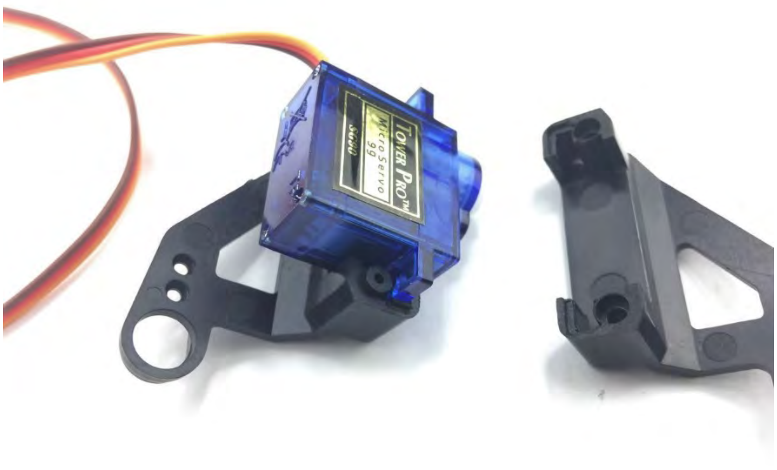
It should look like this

Hardware installation



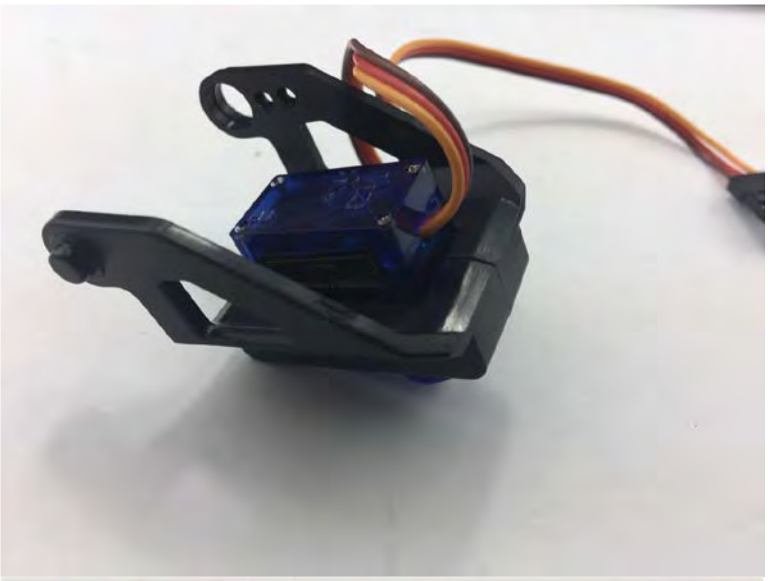
Prepare the Servo package parts



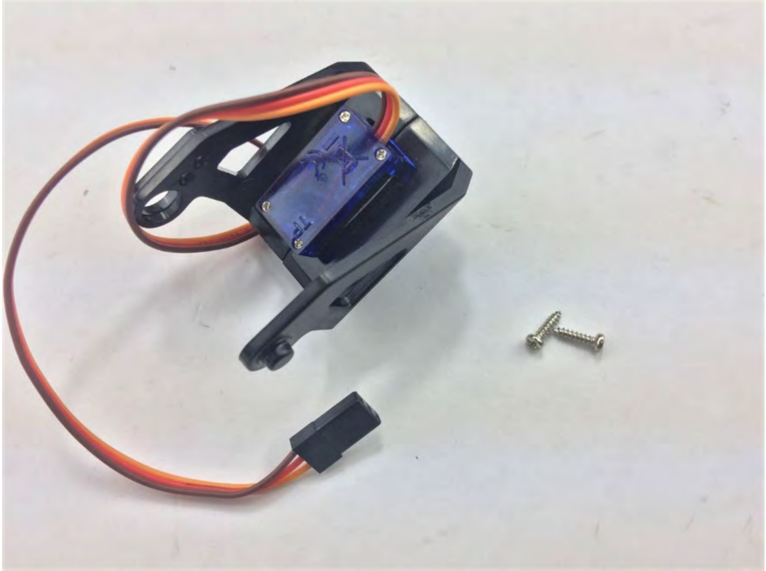


Firstly, fasten the left side and then right-side bracket

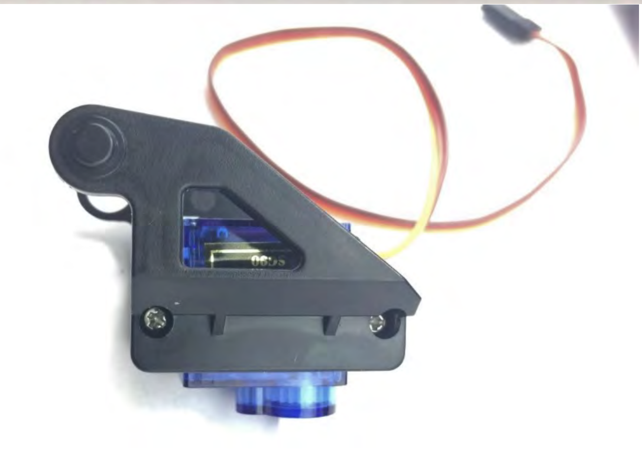
Hardware installation



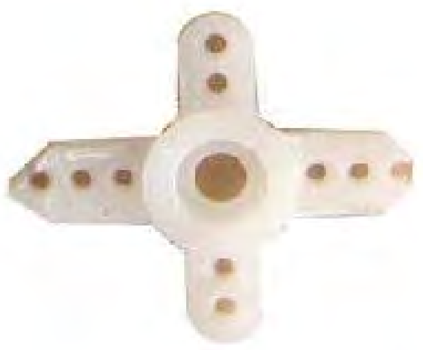
It should look like this

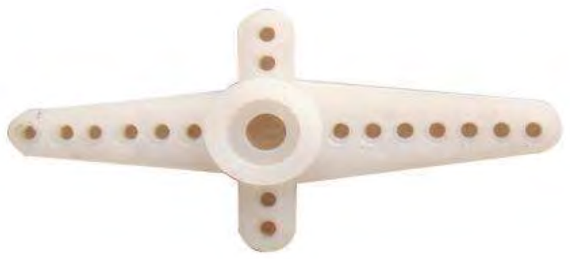


Secure the servo with two brackets and two self-tapping screws

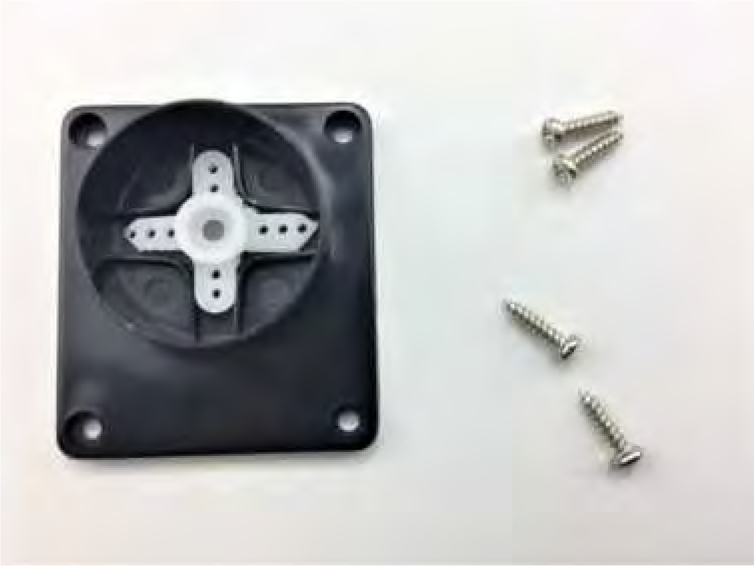


It should look like this



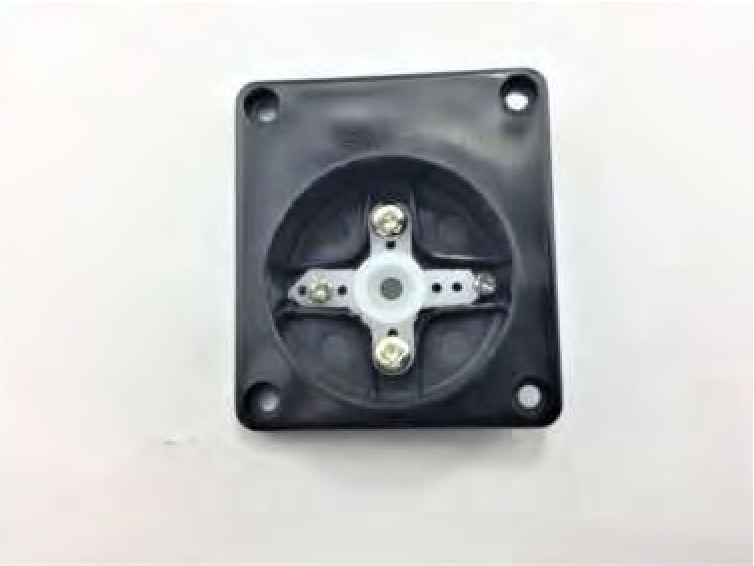


To fit the servo horn into the black holder we need to modify as pictured. Best tool to do this with is a side cutter
Caution: Be carefully with cutter and sharp edges of the horn

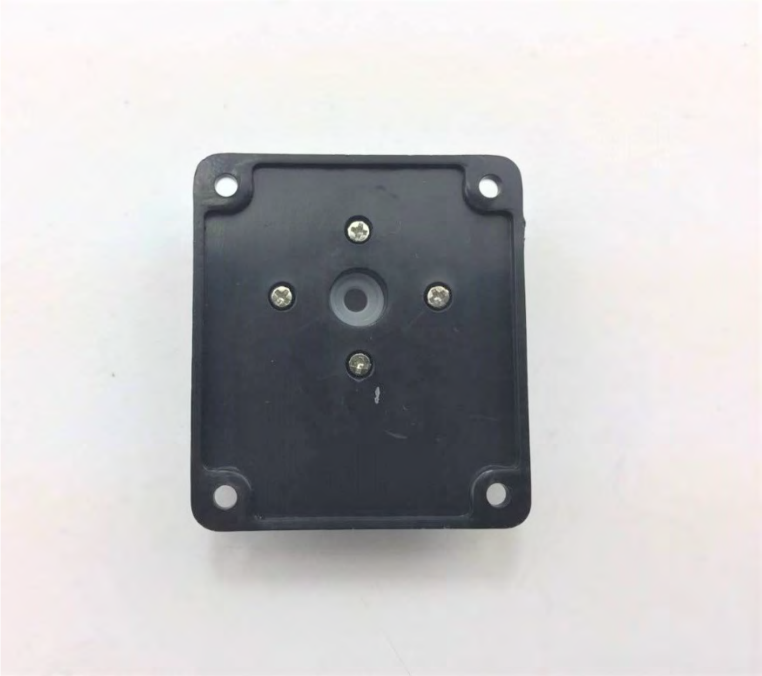


Place the modified servo horn inside the bracket and secure with 6 * M2.5 screws in the servo package

Hardware installation



It should look like this

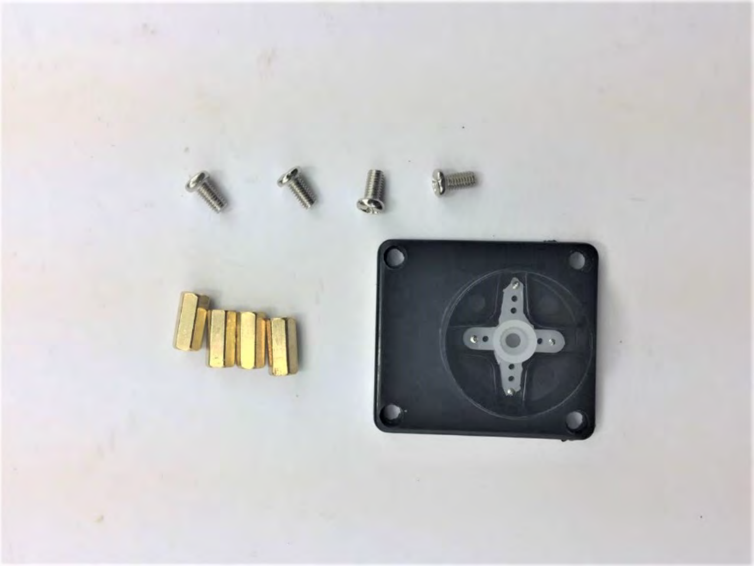


The back should look like this

To secure base mount on the Robot 4-wheel drive, four screw holes will need to be enlarged to fit for M3 screws. The original design is for small screw, not for M3.

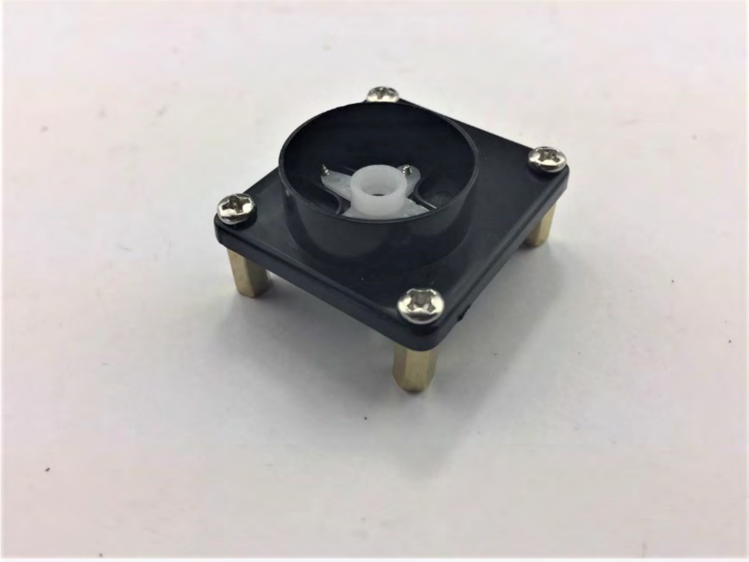
One way to suit for M3 screws is to use a drill or a small screwdriver to punch through the holes.

Hardware installation

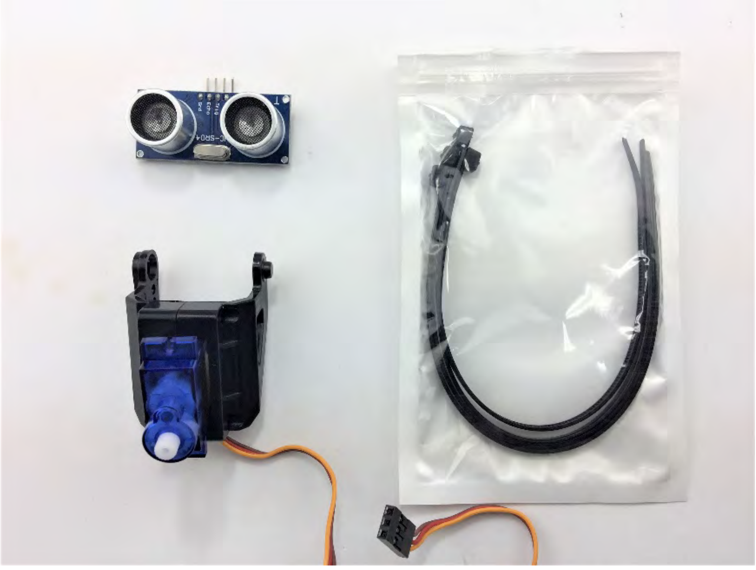


To fasten the rack, we need:

- Four M3* 10 mm Spacers
- Four M3* 6mm Screws

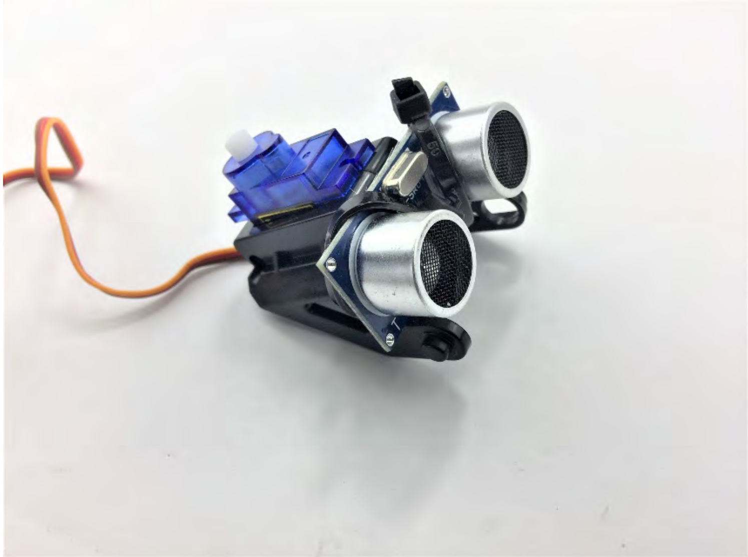


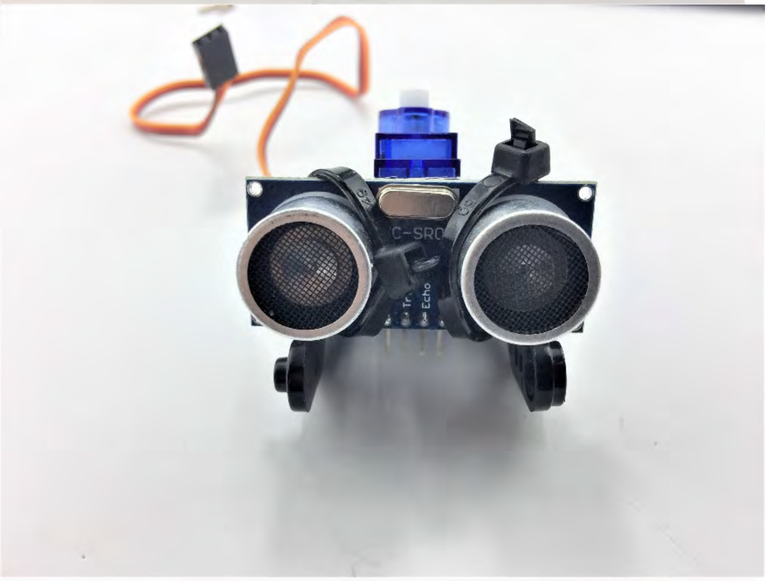
Tighten the screws as shown on the left

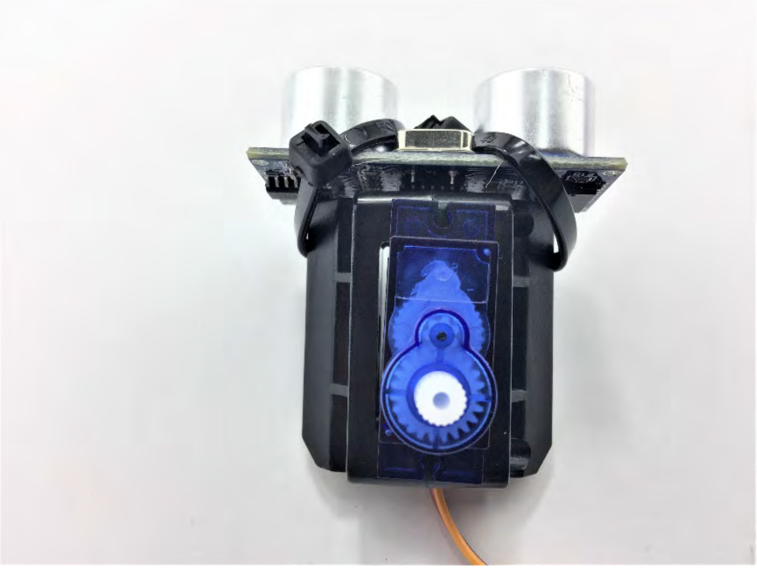


To tighten the ultrasonic sensor,
we can use several cable ties (Not included in the Kit).
Alternatively, you can use other
methods:
- Use rubber bands
- Thread or Plastic Strings

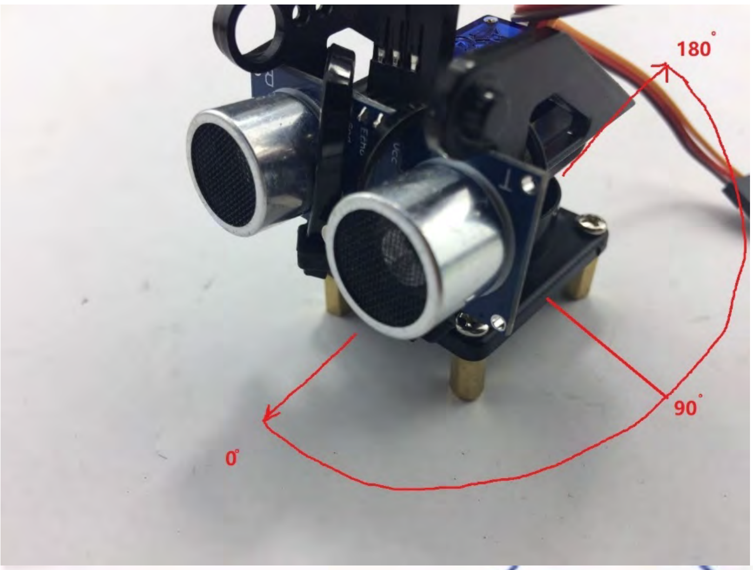
Hardware installation





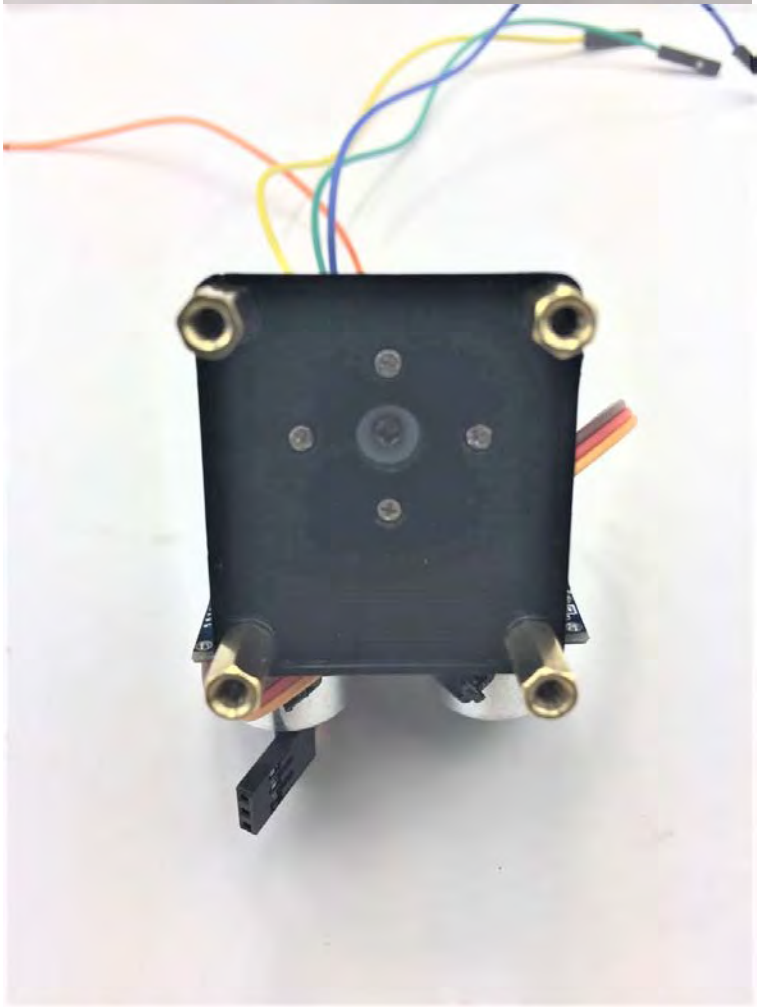


Hardware installation



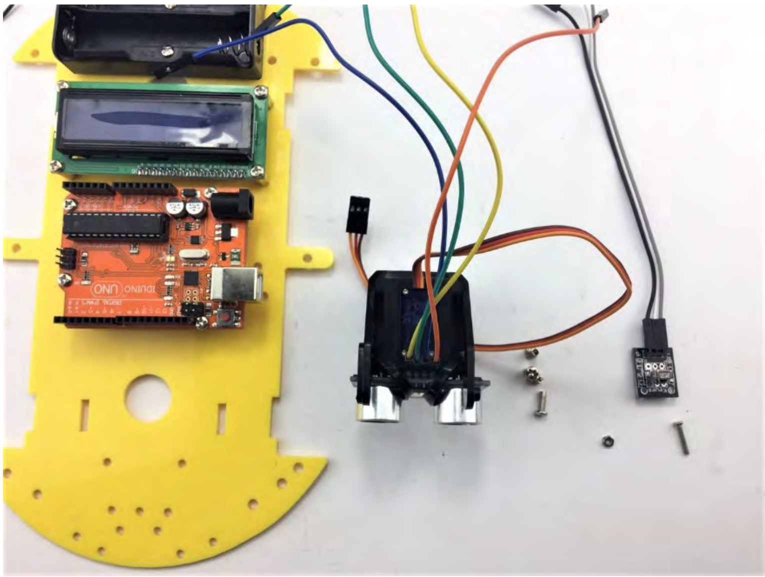
Make sure the Robot is able to detect the distance ranging from 0º to 180º, the Servo with Ultrasonic sensor can turn from the very left to very right smoothly, i.e., 0º to 180º without any jitter.

Note: angle 90º of servo is equivalent to facing front of the Robot 4-wheel drive.



Once finished adjusting the angle of servo, then prepare the Self-tapping screw
in the Servo package and tighten the sensor part with servo rack

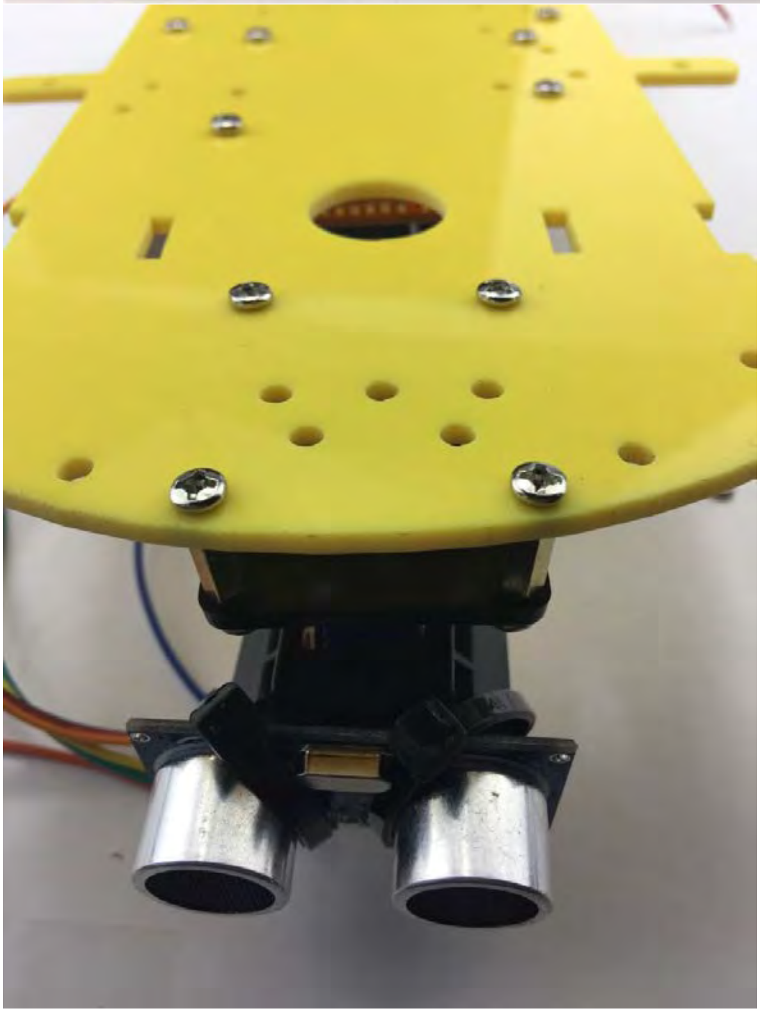
Hardware installation



The Ultrasonic sensor and Infrared sensor is ready to secure on the Chassis.

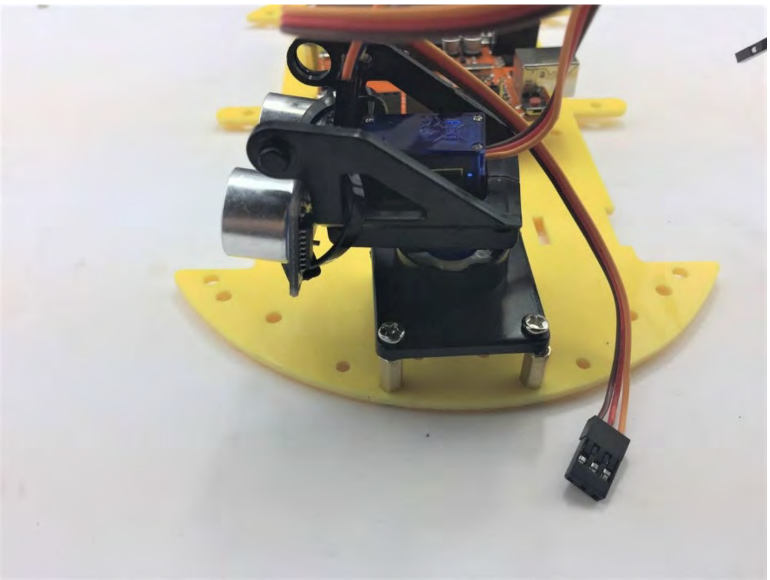
Use:

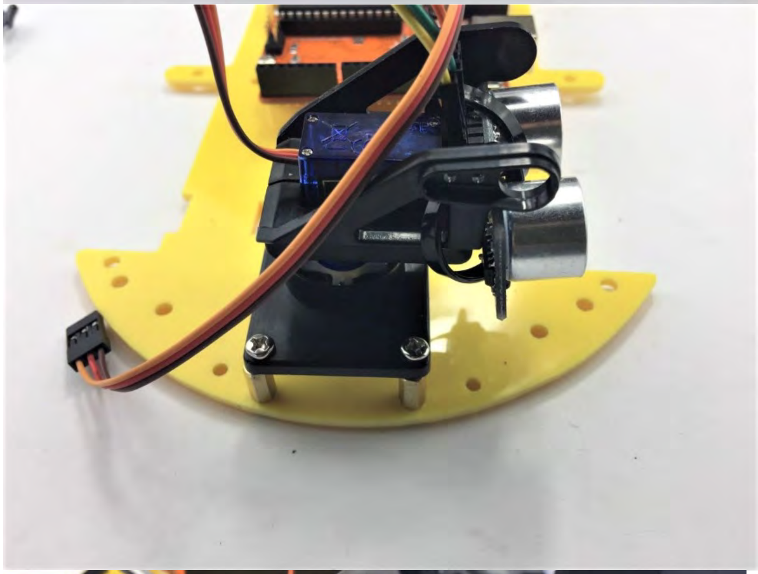
- Four M3 *8 mm Screws
- One M3 *10mm Screw
- One nut



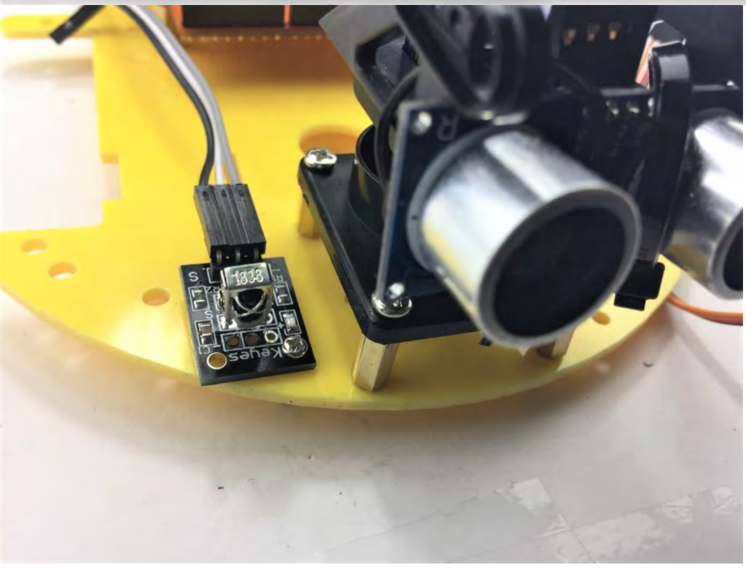
Ultrasonic sensor from the bottom of the Chassis perspective should look like this

Hardware installation



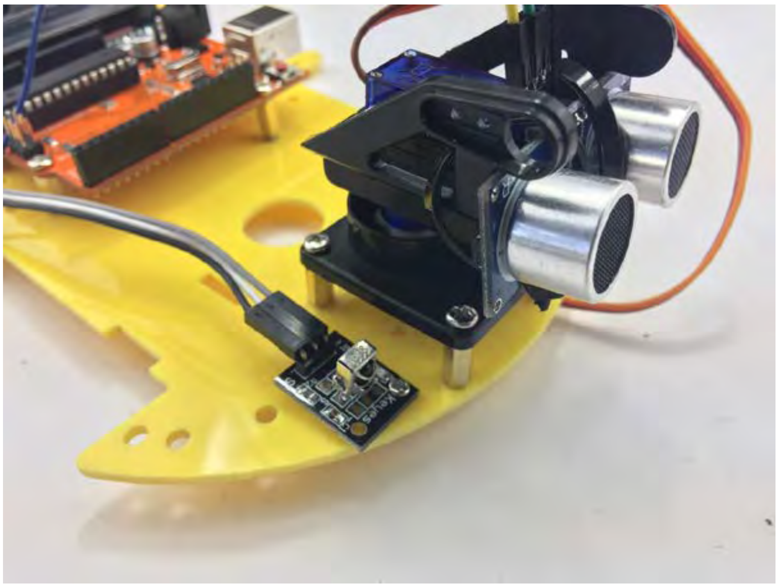


The Ultrasonic sensor can move from left to right



Fix the Infrared receiver on the side of the Ultrasonic sensor.

Hardware installation

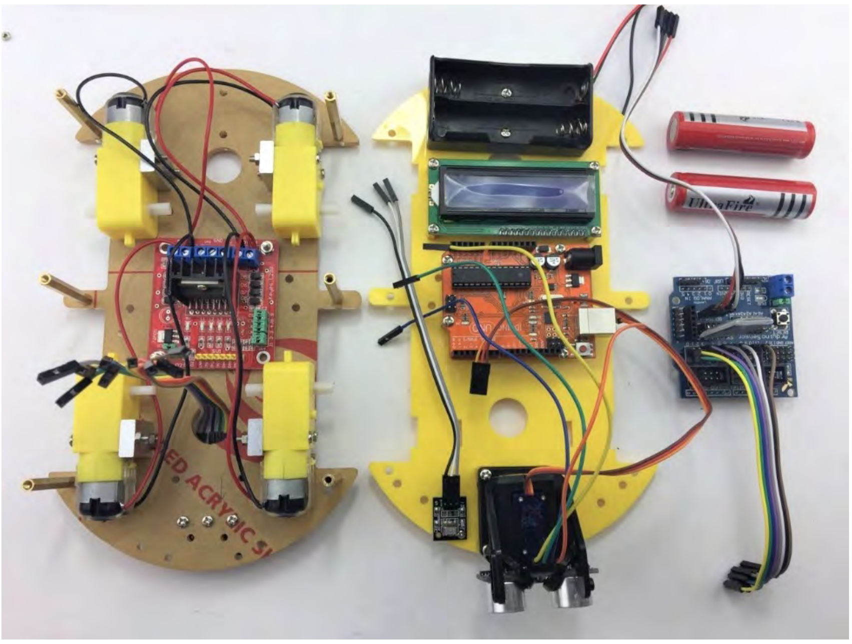


Please note do not place anything on the Infrared receiver and leave it in the open space in order to receive the signal easily.

4.4. Wire Connection

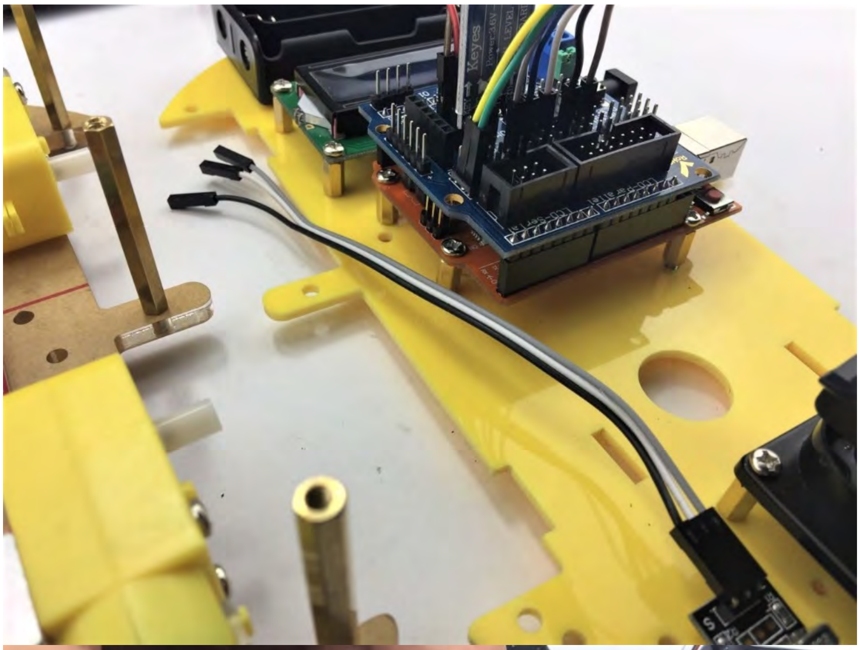
You are almost there. Final step is to wire the cables to power supply (i.e. Battery Box), UNO board, ultrasonic sensor and Servo. The following diagram shows the connection map. Don't panic if this is your first project, you can also follow the connection table 1. Take your time and be patient with the wiring.

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



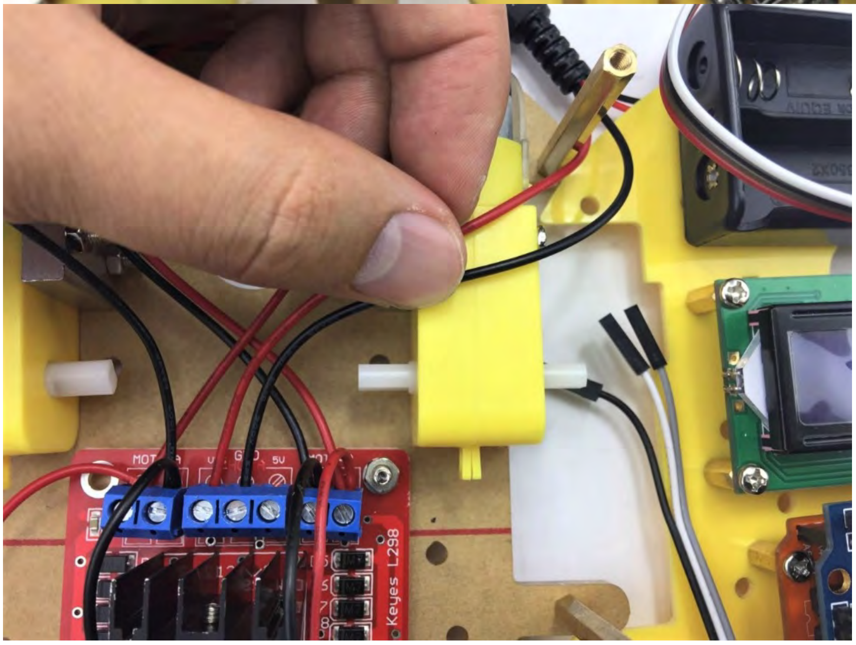
Collect bottom and top chastises and Arduino Sensor shield as well as two 18650 batteries.

Hardware installation



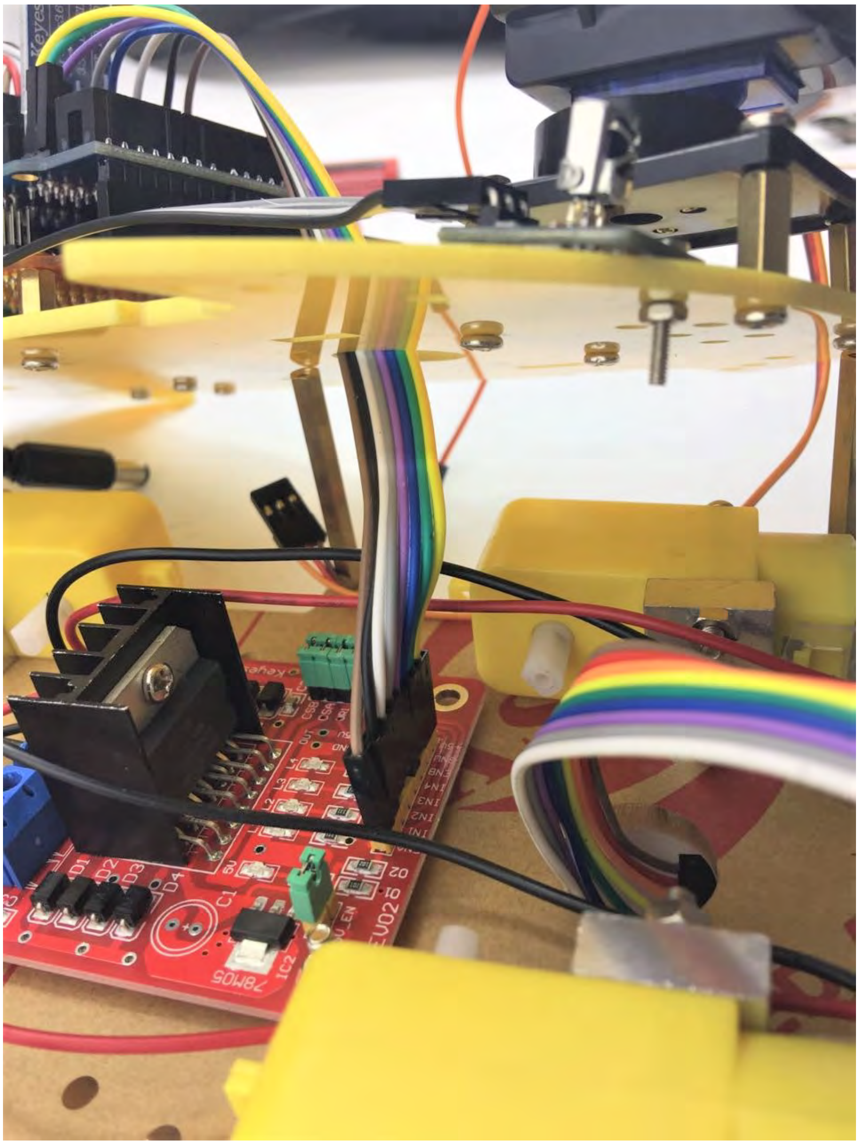
Place the Arduino Sensor shield on the top of the Arduino UNO Board.

Please note Arduino Sensor shield is perfectly aligned with Arduino UNO. Make sure each pin is firmly connected.



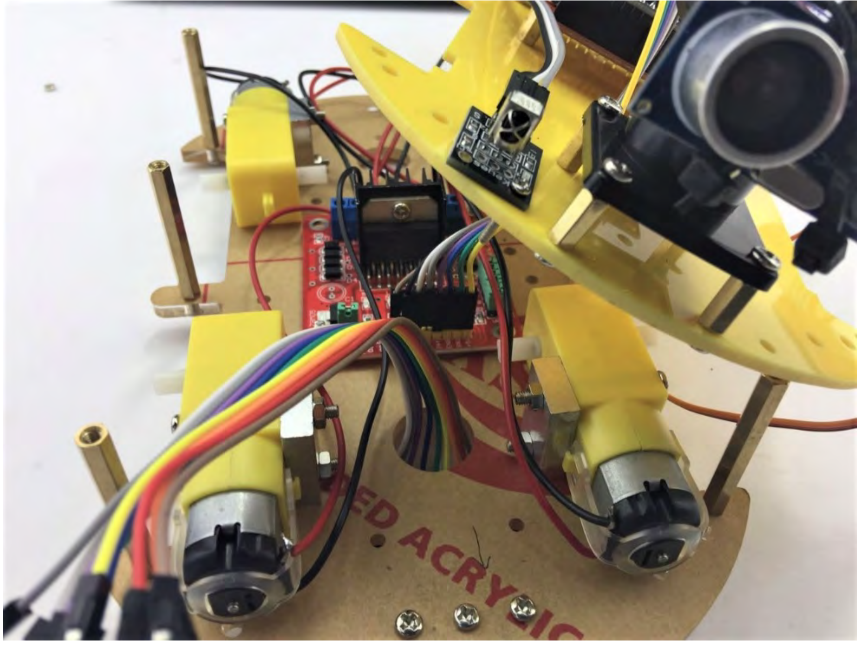
Firstly, connect battery box positive and negative cables to the L289N Motor Driver Board.

Hardware installation



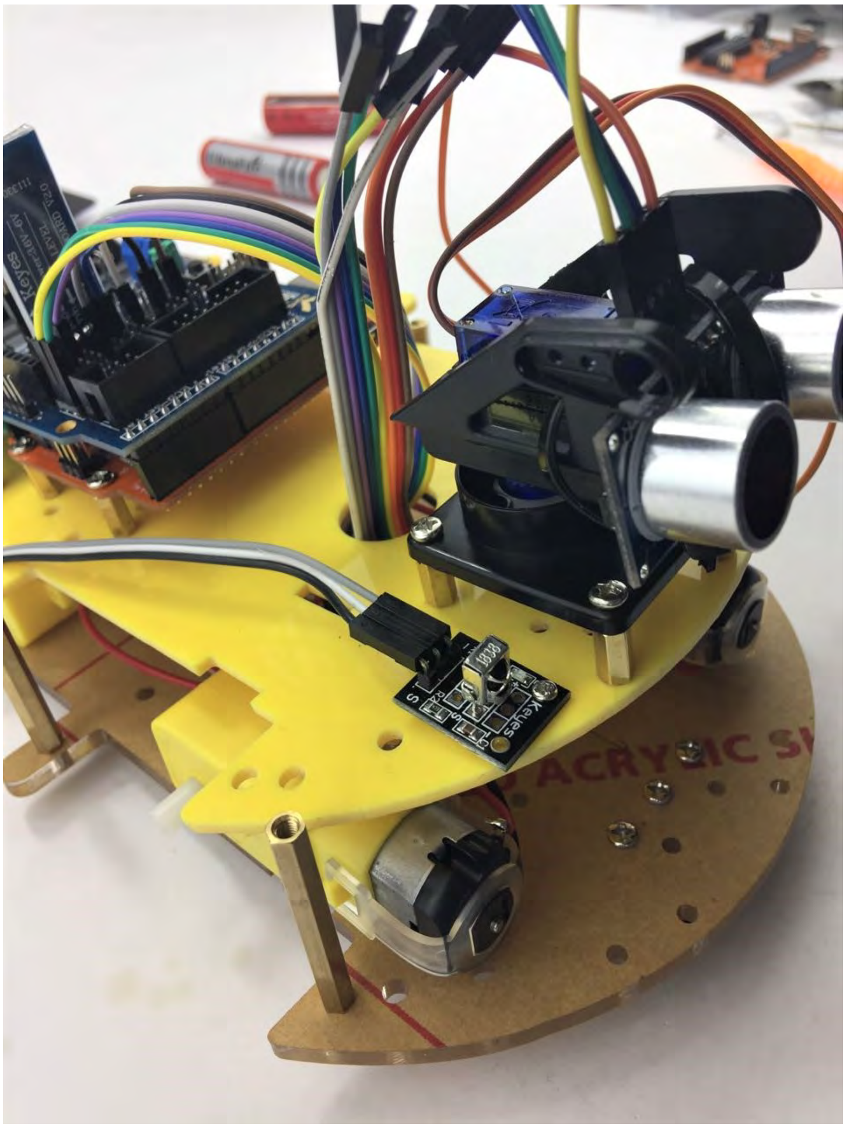
Use eight Jumper wires to connect Arduino Sensor shield and L289N Motor Driver Board.

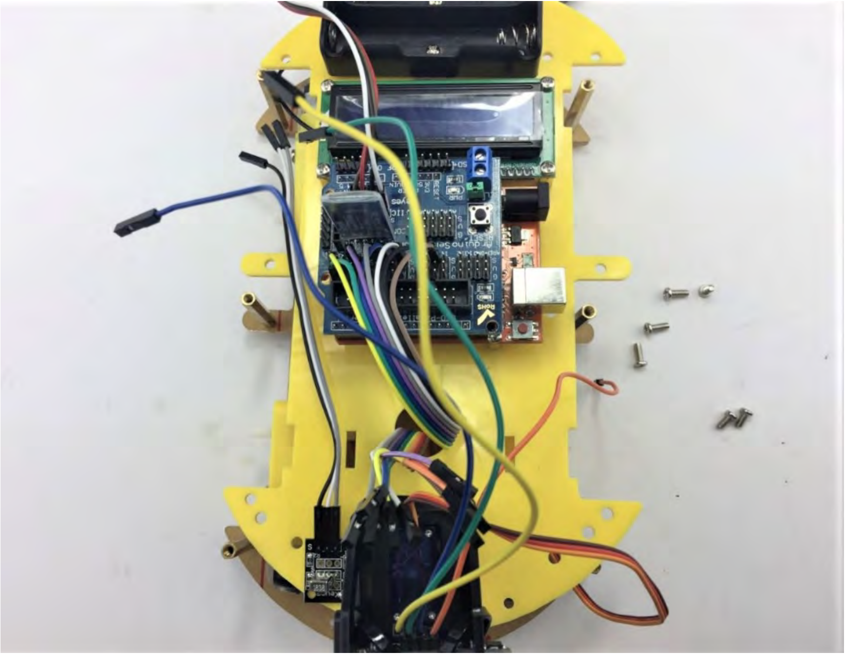
Please Note: The 8 Jumper cables will eventually attach to the Sensor Expansion Board from the L289N board a few steps from now. Please leave the opposite side free for the following steps.



Now pull the jumper wires of three Track line sensor through the hole of top chassis. As usual, leave the opposite side free for connection with Arduino Sensor Board for the following steps.

Hardware installation

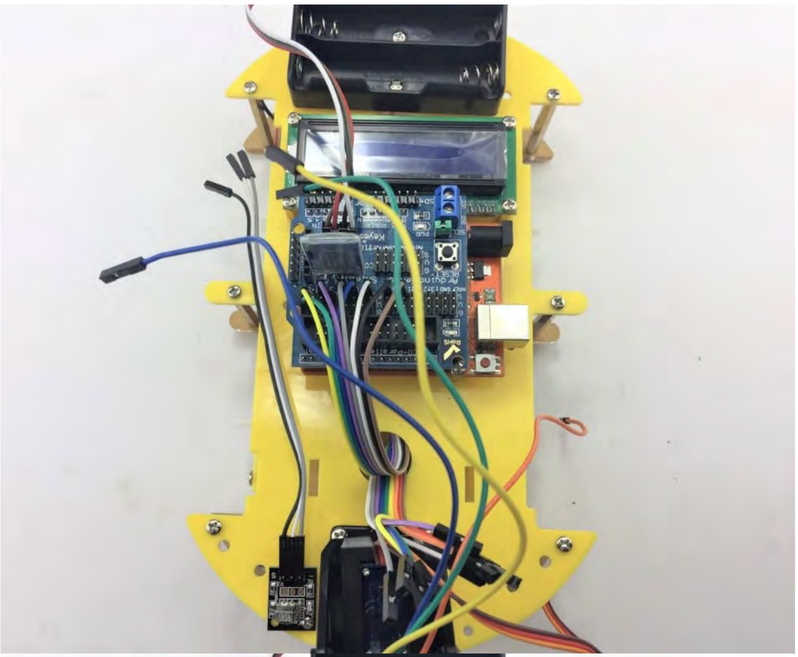


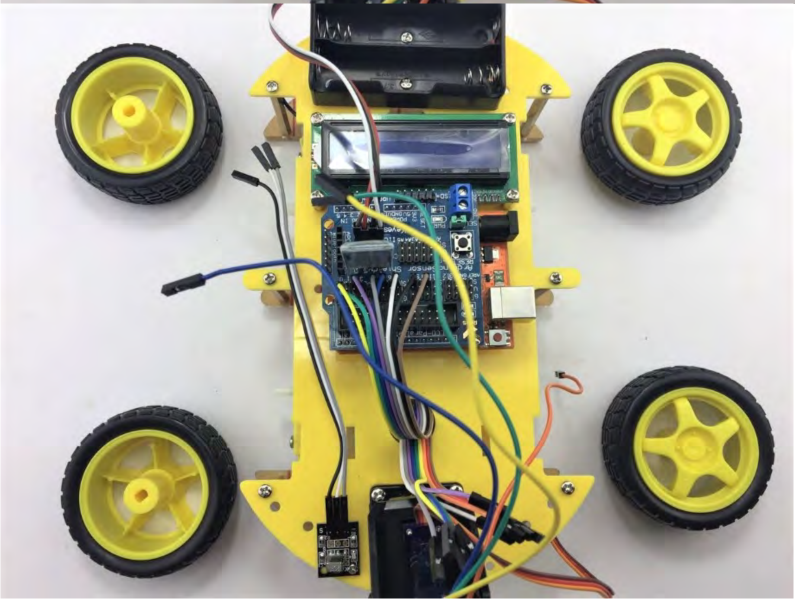


Secure the top chassis with:

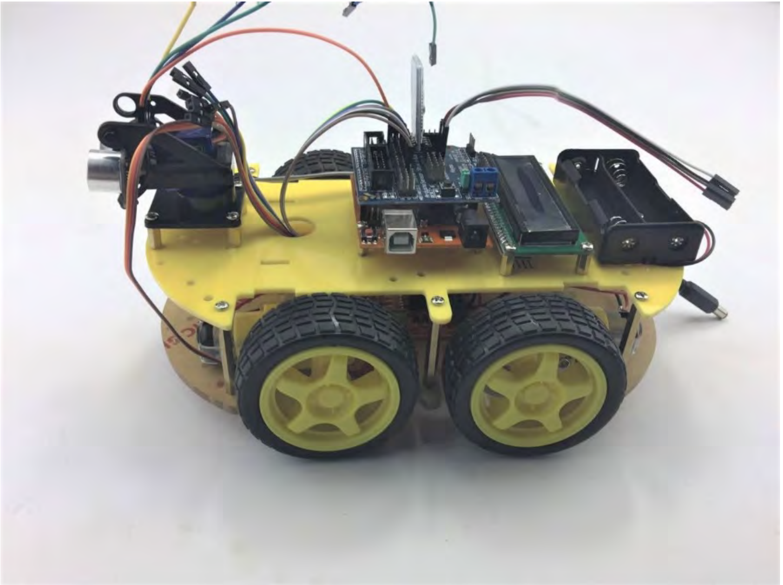
- Six M3 *8mm Screws

Hardware installation





Fix the four wheels on the Motors



Finish!

Hardware installation

Table 1 Connection table

UNO board Sensor Shield	L289N	Battery Box	Two Motors left	Two Motors Right	Servo	Ultrasonic sensor	Infrared sensor	LCD1602	Bluetooth
	GND	**GND**							
	VMS	**VMS**							
	+(Left)		+(red)						
	-(Left)		- (black)						
	+(Right)			+(red)					
	-(Right)			- (black)					
V	ENA								
5	IN1								
6	IN2								
10	IN3								
11	IN4								
V	ENB								
G	**GND**								
V	5V+								
9					S				
V					+				
G					**-**				
V						+			
12						Trig			
13						Echo			
S						-			
2							S		
V							+		
G							-		
SDA								SDA	
SCL								SCL	
V								+	
G								-	
TX									RX
RX									TX
G									-
V									+

Hardware installation

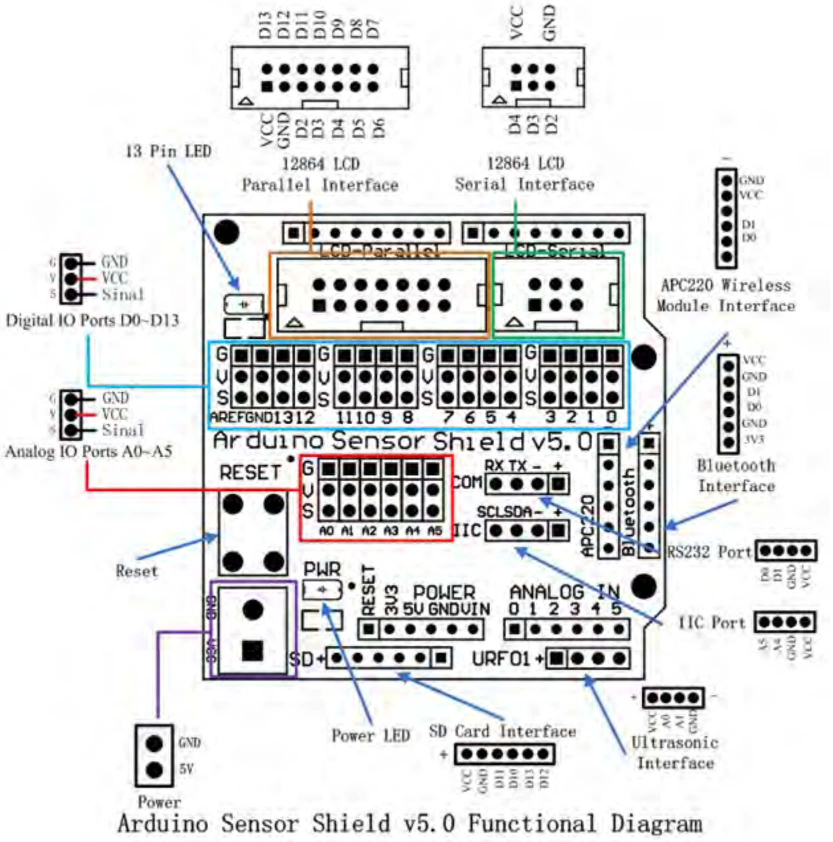


Figure 3 Arduino Sensor shield v5.0 Diagram

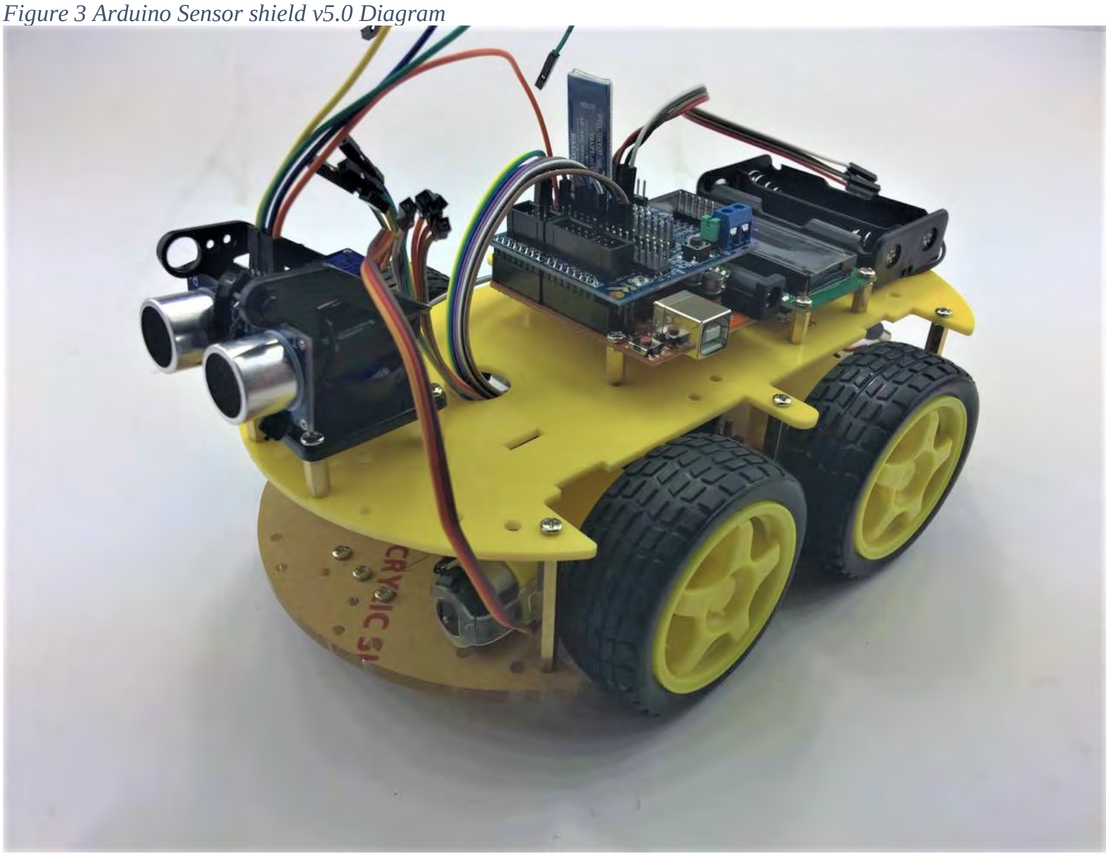
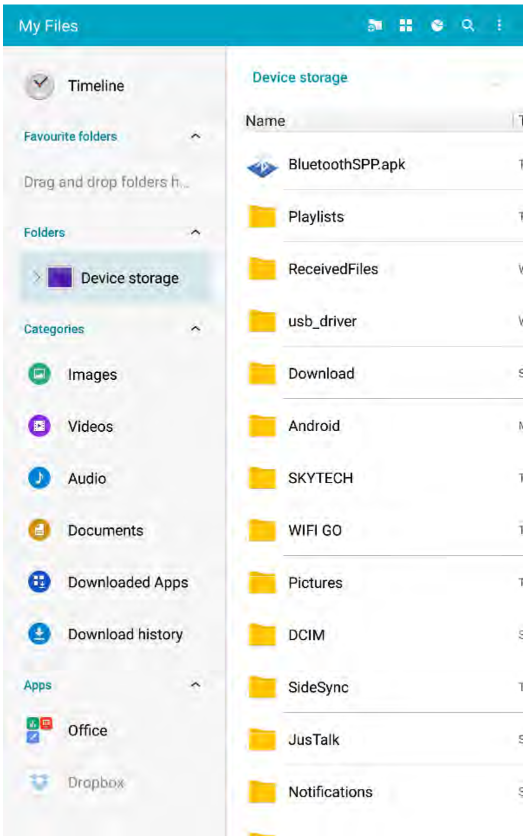


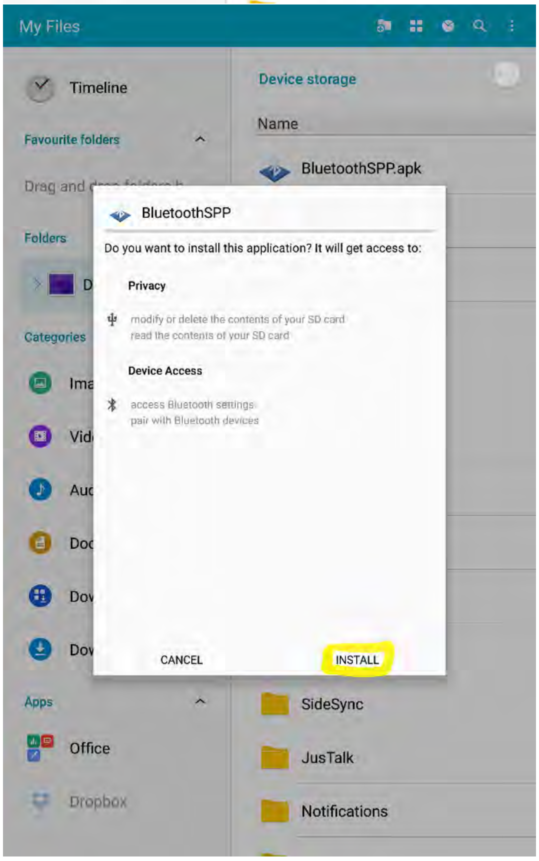
Figure 4 Overview

Chapter 5. Bluetooth Mobile Control

In this chapter, get your Android Device to control 4-wheel drive.



Copy the Bluetooth control application: “BluetoothSPP.apk” to your android device, in this instance, we are using Samsung Pad.



Do you want to install this application? It will get access to:

Privacy

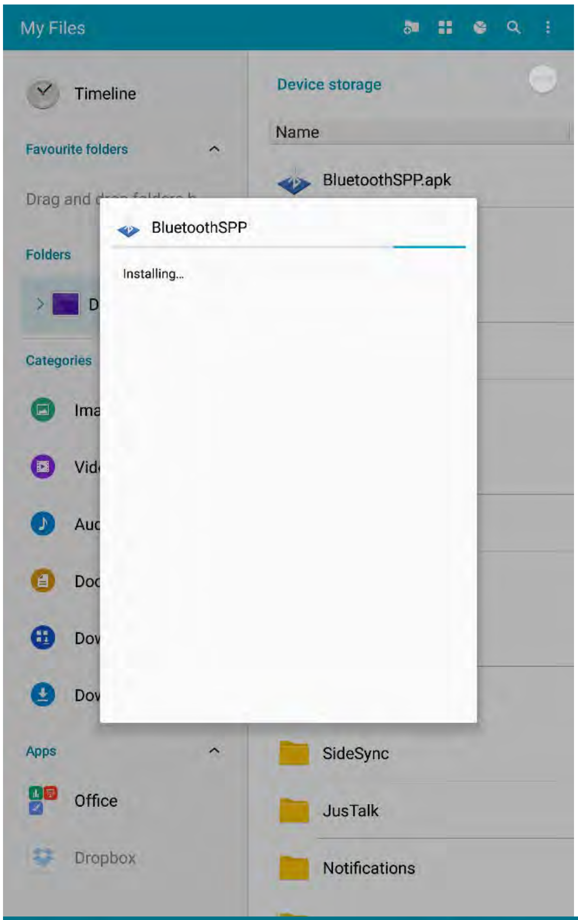
modify or delete the contents of your SD card
read the contents of your SD card

Device Access

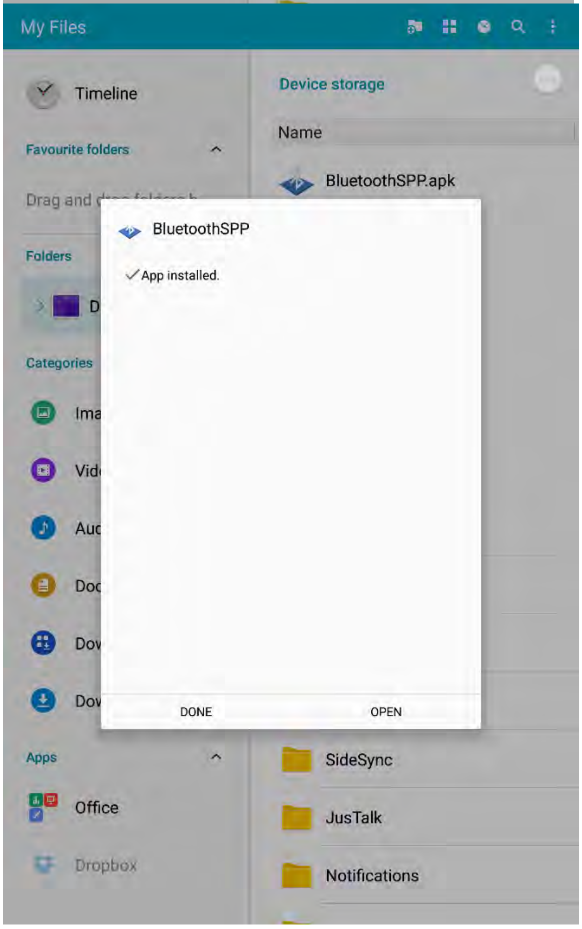
access Bluetooth settings
pair with Bluetooth devices

Install the app from the device file manager.

Bluetooth Mobile Control


BluetoothSPP

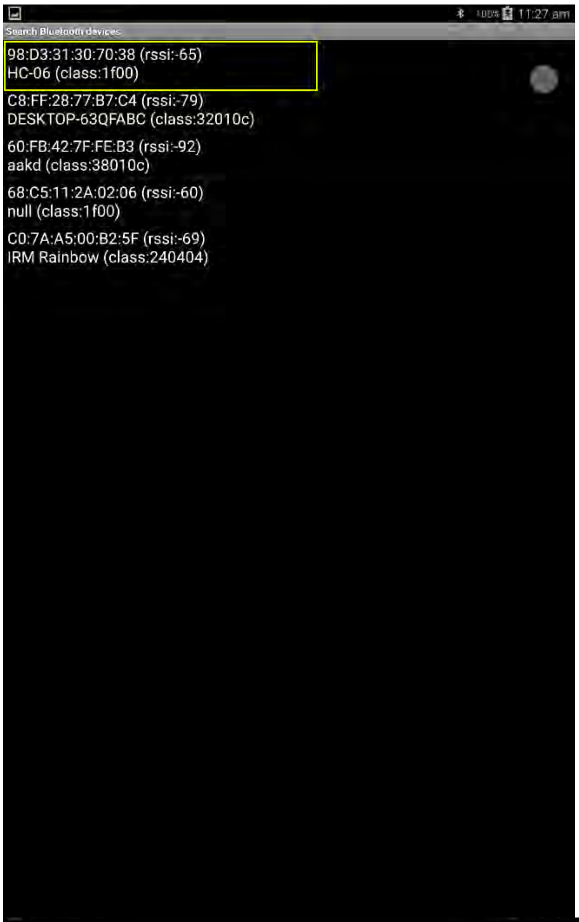
Installing...


BluetoothSPP

✓ App installed.

Application installed!

Bluetooth Mobile Control



Search Bluetooth devices

98:D3:31:30:70:38 (rssi:-65)
HC-06 (class:1f00)

C8:FF:28:77:B7:C4 (rssi:-79)
DESKTOP-63QFABC (class:32010c)

60:FB:42:7F:FE:B3 (rssi:-92)
aakd (class:38010c)

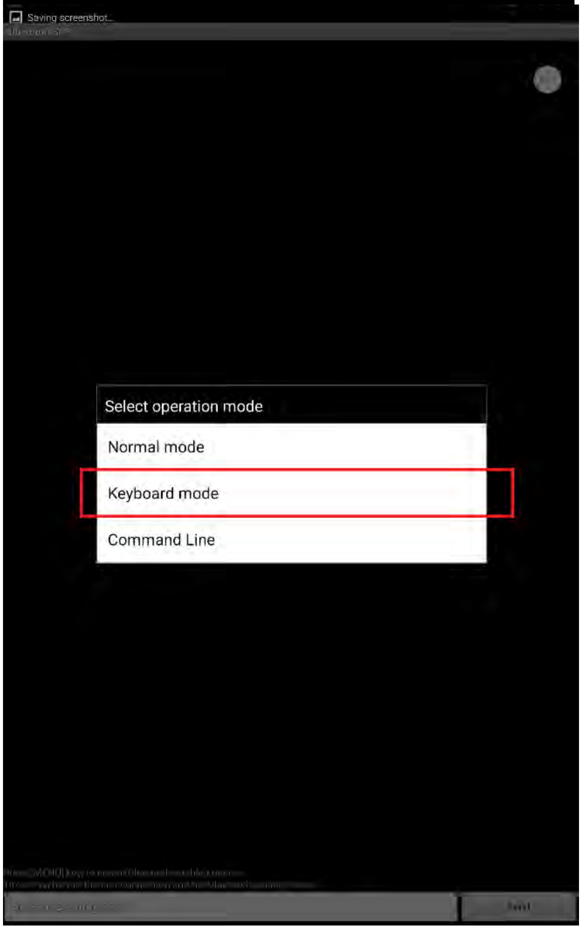
68:C5:11:2A:02:06 (rssi:-60)
null (class:1f00)

C0:7A:A5:00:B2:5F (rssi:-69)
IRM Rainbow (class:240404)

Open the app and it will initialize with searching for Bluetooth devices nearby.

The name of Bluetooth device of your Robot 4-wheel drive may not be the “HC06” shown on the left. This name is the default name for HC-06. It may vary as you have changed in the previous firmware AT command settings, see Chapter 4.2.6 for changed name and Pin.

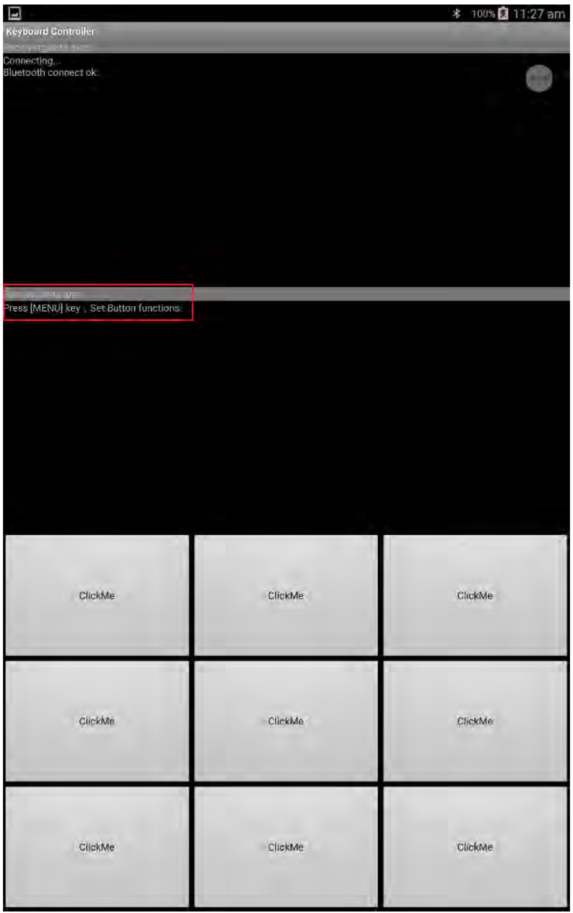
Please note enable the Bluetooth function on the Android device and turn on the Robot 4-wheel drive to ensure HC-06 is powered on.



Select operation mode
Normal mode
Keyboard mode
Command Line

Click the HC-06 device which is the one on the Robot 4-wheel drive and then choose the Keyboard Mode as we are controlling via the Keyboard Touch screen.

Bluetooth Mobile Control



Keyboard Controller

Receiving data area

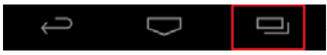
Connecting...
Bluetooth connect ok.

Sending data area

Press [MENU] key , Set Button functions.

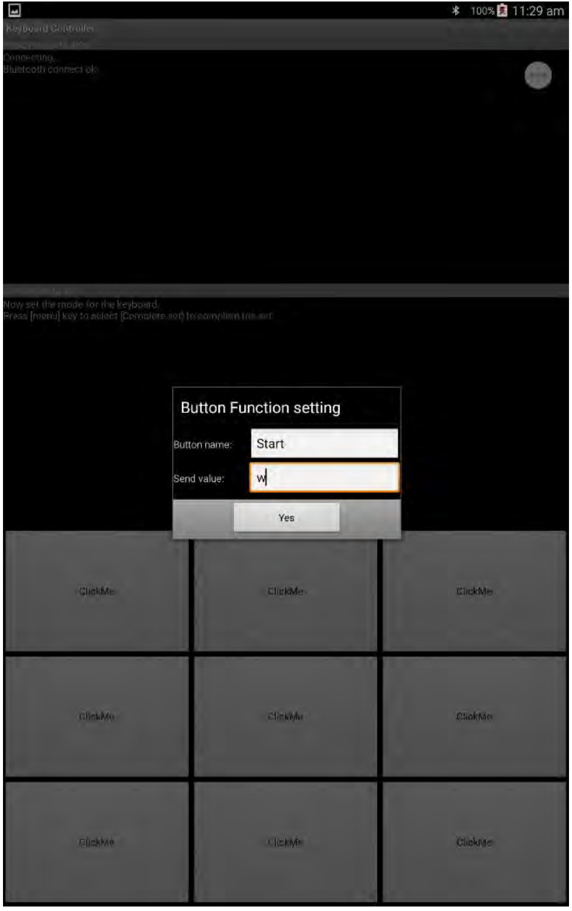
On the top left dialog, the app shows successfully connecting with your HC-06 on the Robot 3-wheel Drive. Now the red LED indicator on the Bluetooth Module should stop blinking and is on continuously, which means they paired successfully.

Define the Keyboard button by pressing the MENU button on your Android device (if there is no MENU button on your Android, please hold this button for seconds



)

Then a dialog window will pop up and choose the "Set Button function".



Keyboard Controller

Receiving data area

Connecting...
Bluetooth connect ok.

Sending data area

Now set the mode for the keyboard.
Press [menu] key to select [Complete set] to complete the set

Button Function setting

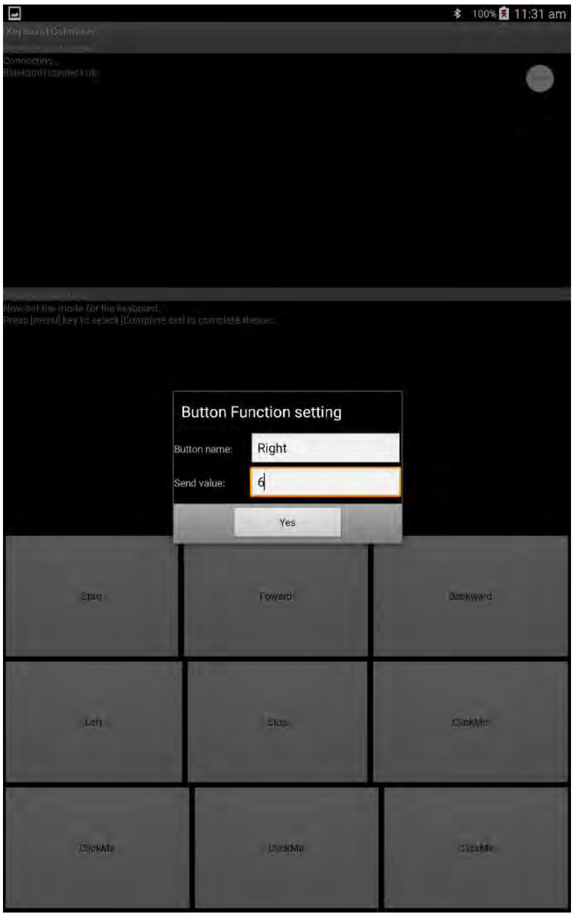
Button name: Start

Send value: w

Define the name and Serial Value.

Start: w

Bluetooth Mobile Control



Connecting...
Bluetooth connect ok

Now set the mode for the keyboard.
Press [menu] key to select [Complete set] to complete the set

Button Function setting

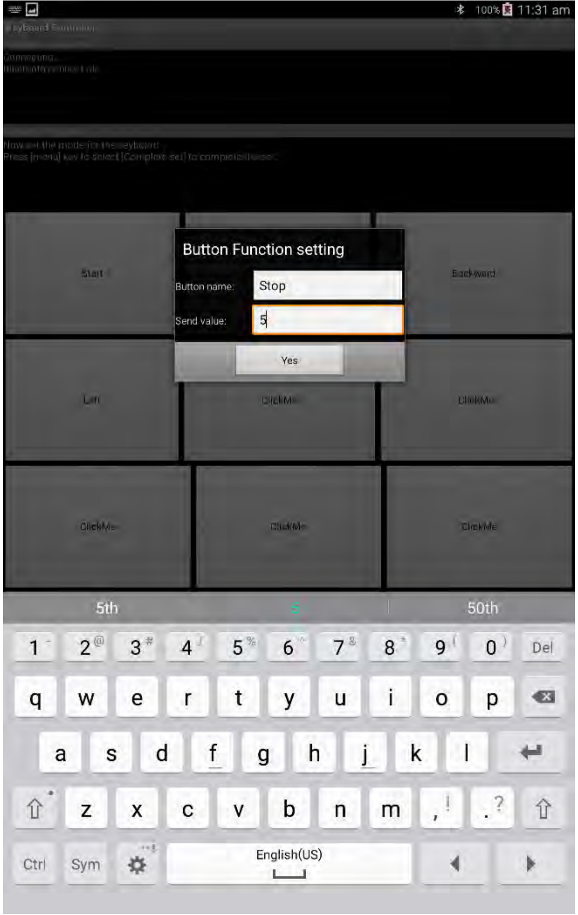
Button name: Right

Send value: 6

Yes

Define the name and Serial Value.

Right: 6



Button Function setting

Button name: Stop

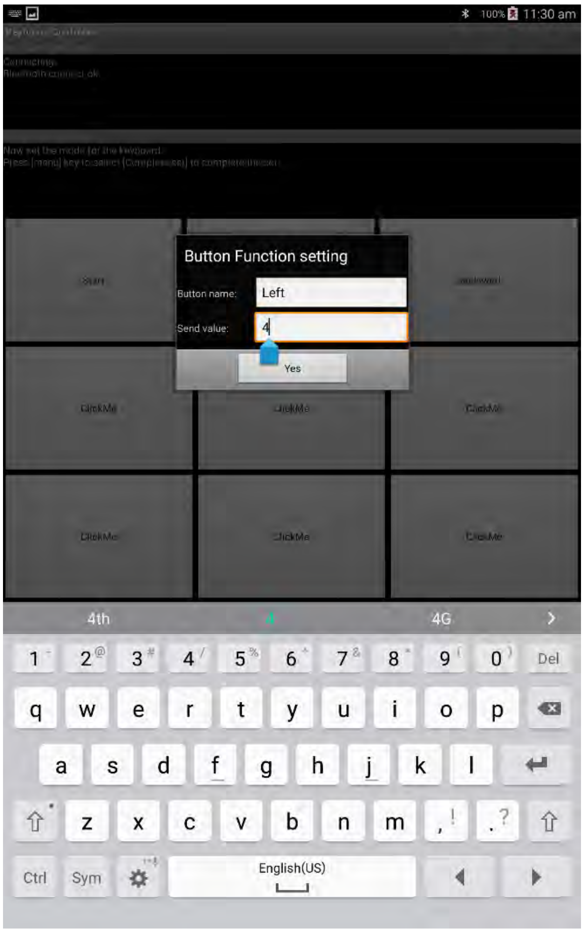
Send value: 5

Yes

Define the name and Serial Value.

Stop: 5

Bluetooth Mobile Control



Keyboard Controller

Connecting...
Bluetooth connect ok

Now set the mode for the keyboard.
Press [menu] key to select [Complete set] to complete the set.

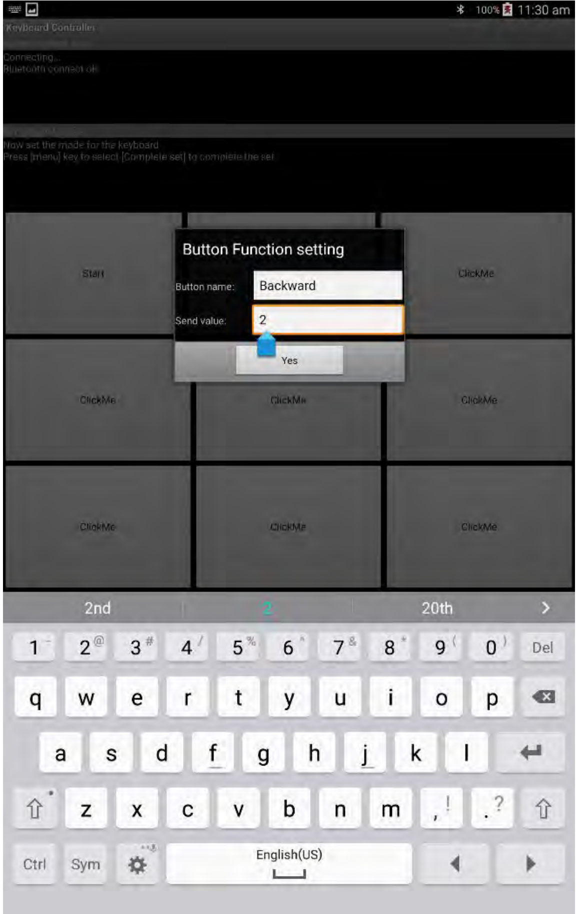
Button Function setting

Button name: Left

Send value: 4

Define the name and Serial Value.

Left: 4



Keyboard Controller

Connecting...
Bluetooth connect ok

Now set the mode for the keyboard
Press [menu] key to select [Complete set] to complete the set

Button Function setting

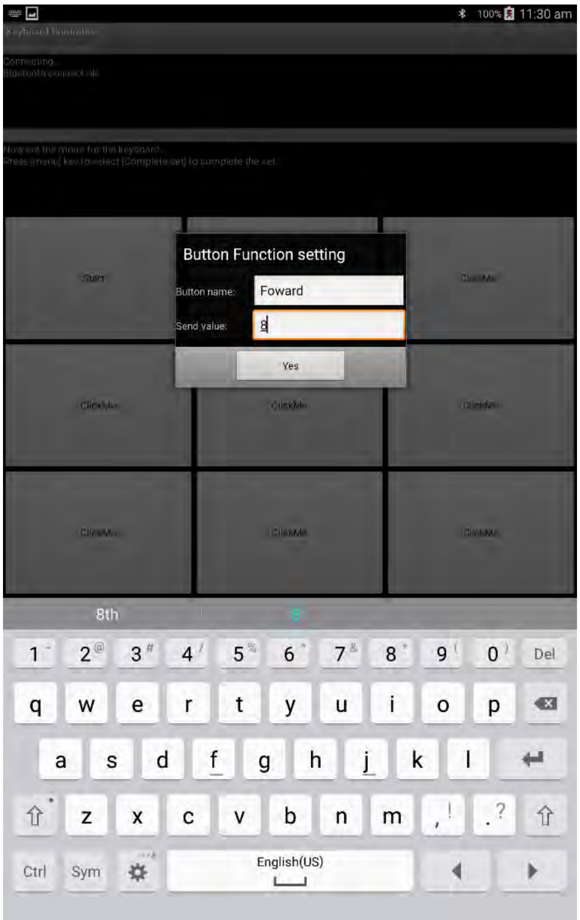
Button name: Backward

Send value: 2

Define the name and Serial Value.

Backward: 2

Bluetooth Mobile Control



Connecting...
Bluetooth connect ok.

Now set the mode for the keyboard.
Press [menu] key to select [Complete set] to complete the set.

Button Function setting

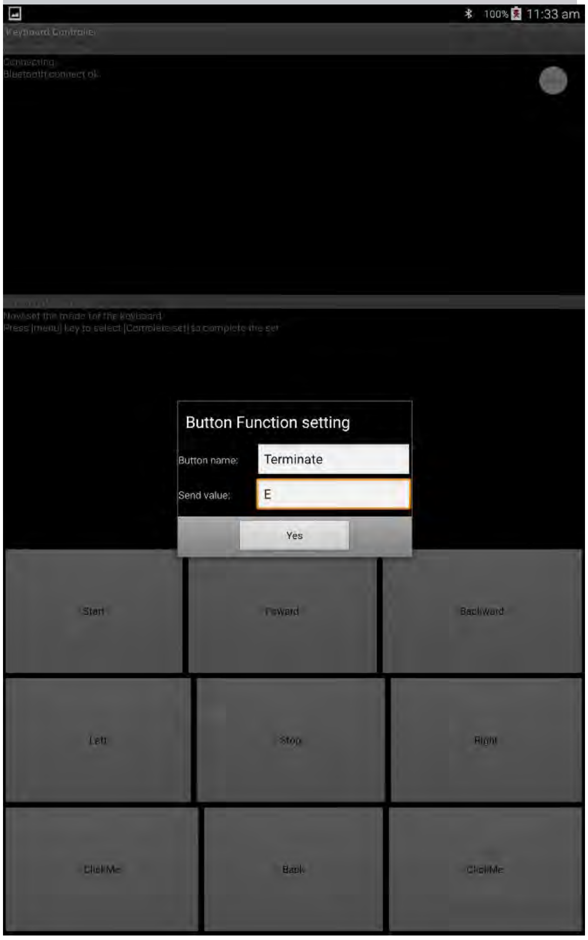
Button name: Foward

Send value: 8

Yes

Define the name and Serial Value.

Forward: 8



Connecting...
Bluetooth connect ok.

Now set the mode for the keyboard.
Press [menu] key to select [Complete set] to complete the set.

Button Function setting

Button name: Terminate

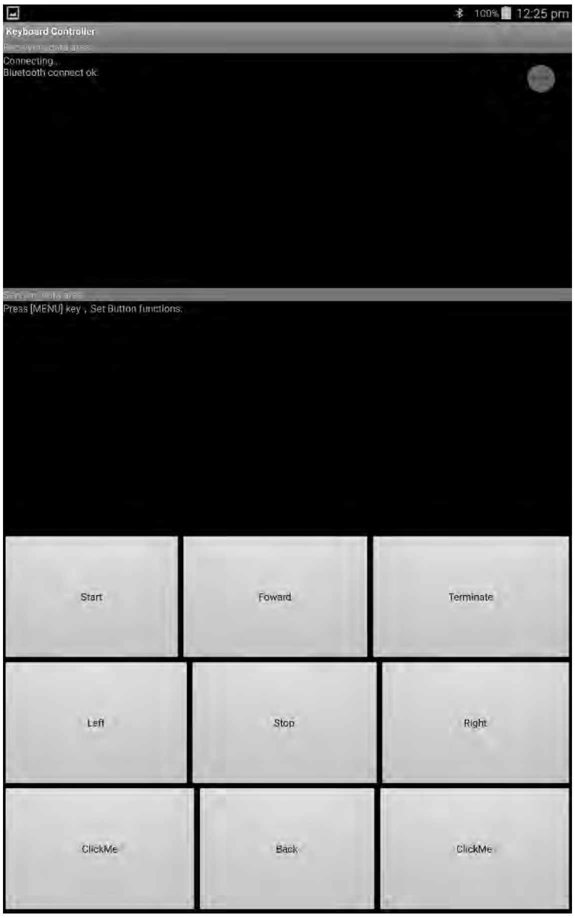
Send value: E

Yes

Define the name and Serial Value.

Terminate: E

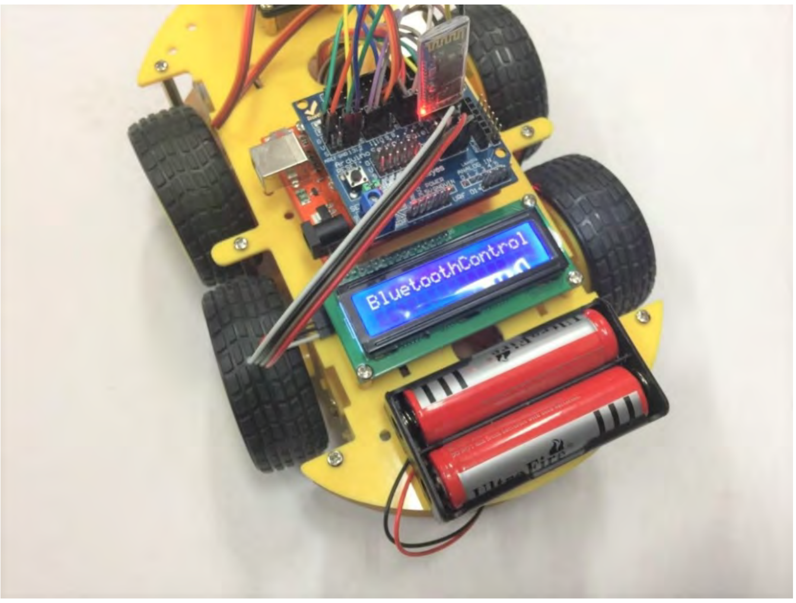
Bluetooth Mobile Control



Connecting...
Bluetooth connect ok.

Press [MENU] key , Set Button functions.

Keyboard definition is done!

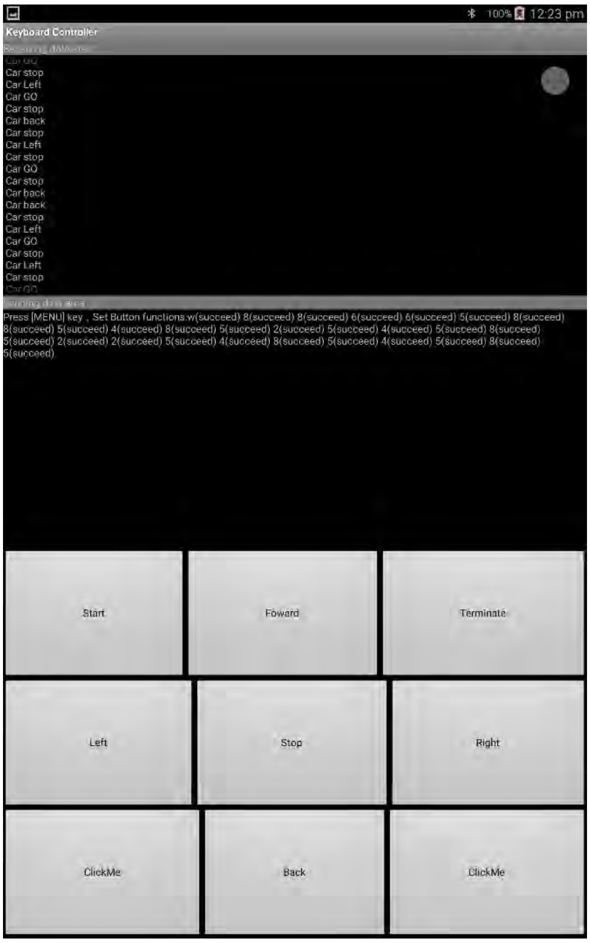


Now press the number 3 on the Remote controller to trigger the Bluetooth Mode on the Robot 4-wheel drive. When you press any button on the remote controller, the red LED on the Infrared receiver will be blinking twice.

Then you will notice LCD1602 display" BluetoothControl " which means the Robot is in the standby mode waiting for command from you Android Bluetooth device.

Then press the 3 again to tell Robot ready to go.

Bluetooth Mobile Control



```
Car GO
Car stop
Car Left
Car GO
Car stop
Car back
Car stop
Car Left
Car stop
Car GO
Car stop
Car back
Car back
Car stop
Car Left
Car GO
Car stop
Car Left
Car stop
Car GO
```

```
Press [MENU] key , Set Button functions.w(succeed) 8(succeed) 8(succeed) 6(succeed) 6(succeed) 5(succeed) 8(succeed)
8(succeed) 5(succeed) 4(succeed) 8(succeed) 5(succeed) 2(succeed) 5(succeed) 4(succeed) 5(succeed) 8(succeed)
5(succeed) 2(succeed) 2(succeed) 5(succeed) 4(succeed) 8(succeed) 5(succeed) 4(succeed) 5(succeed) 8(succeed)
5(succeed)
```

Now have fun with your Robot 4-Wheel drive!

Chapter 6. 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.

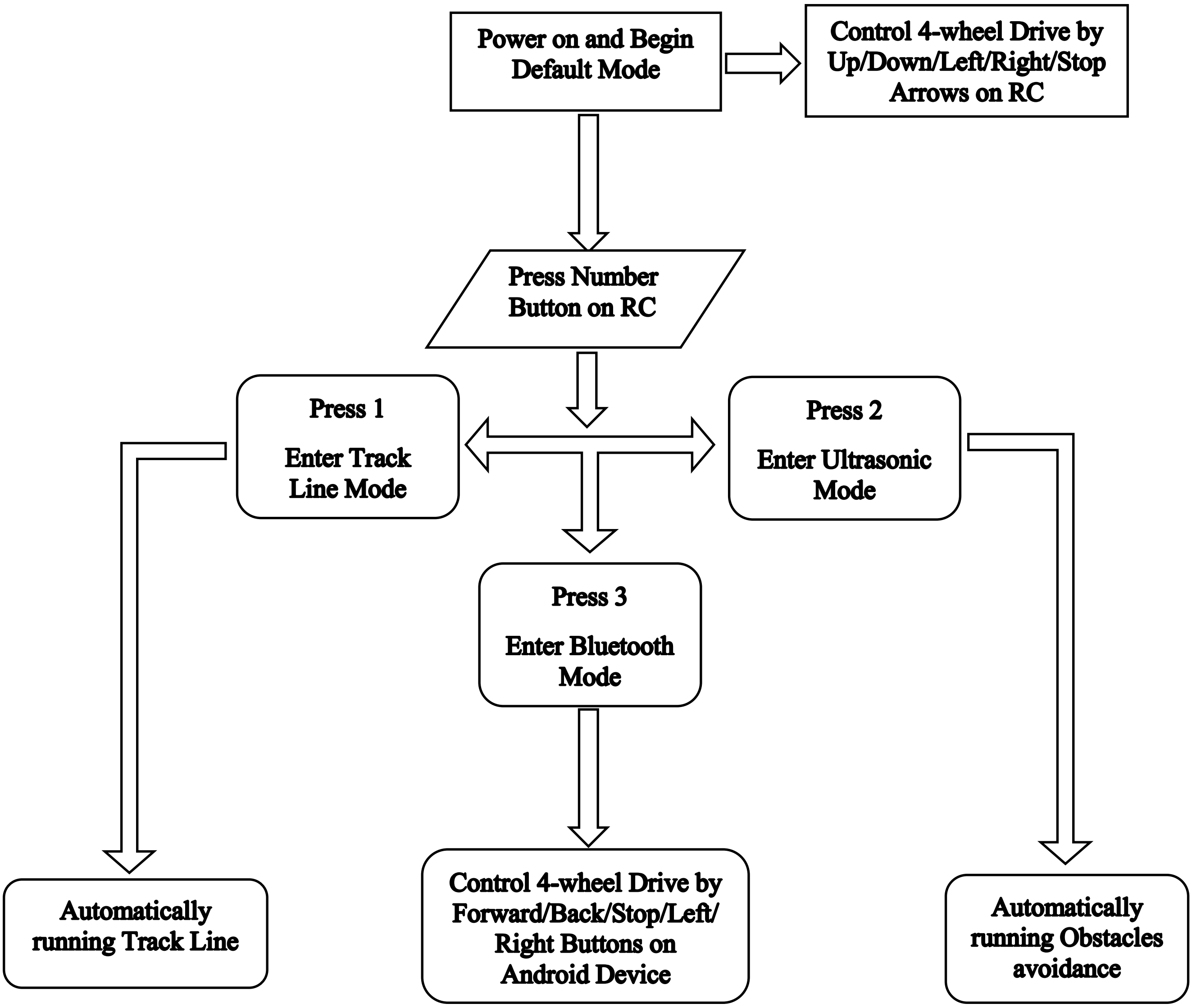

Power on and Begin Default Mode
Control 4-wheel Drive by Up/Down/Left/Right/Stop Arrows on RC
Press Number Button on RC
Press 1
Enter Track Line Mode
Press 2
Enter Ultrasonic Mode
Press 3
Enter Bluetooth Mode
Automatically running Track Line
Control 4-wheel Drive by Forward/Back/Stop/Left/ Right Buttons on Android Device
Automatically running Obstacles avoidance

Figure 5 Overall flow chart

Please unplug the Sensor board from the Arduino UNO board and disconnect 18650 power box supply to modify your code. Otherwise, it may cause irreparable harm to your Robot and PC as it may drive a large current through your USB port.

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

Check our website for more at Here. (https://www.auselectronicsdirect.com.au/arduino/arduino-starter-kit/)