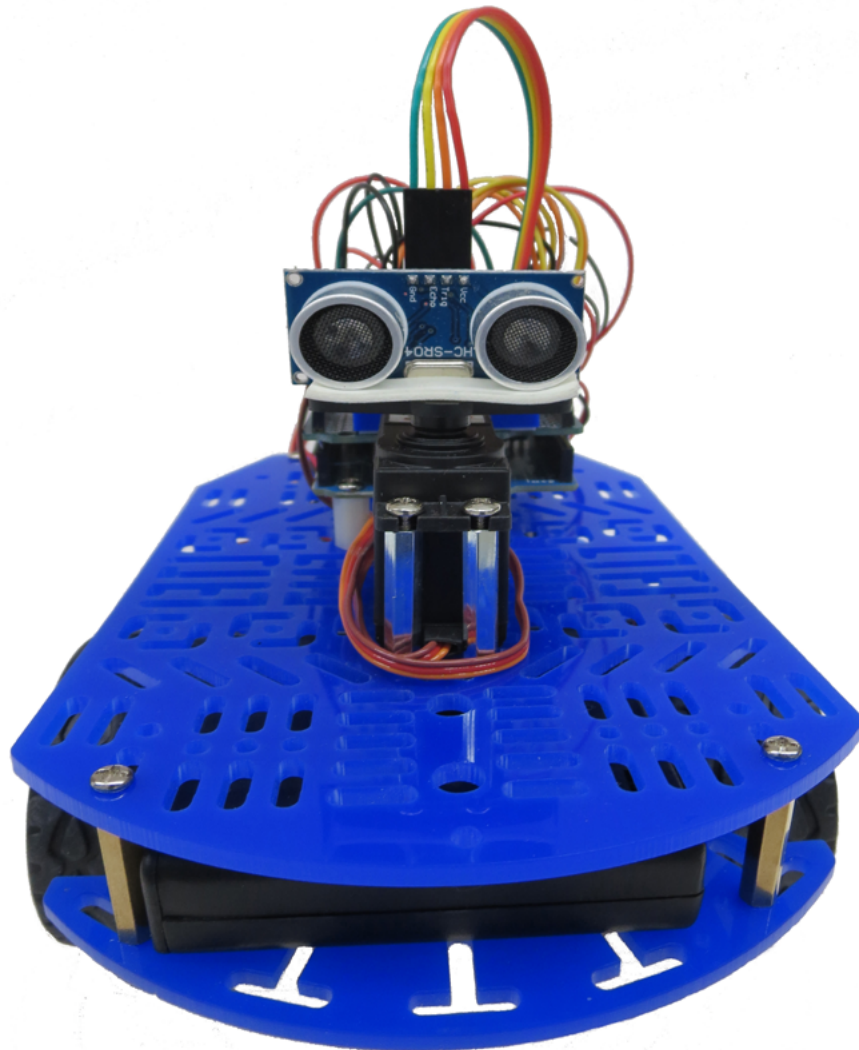


ABRA

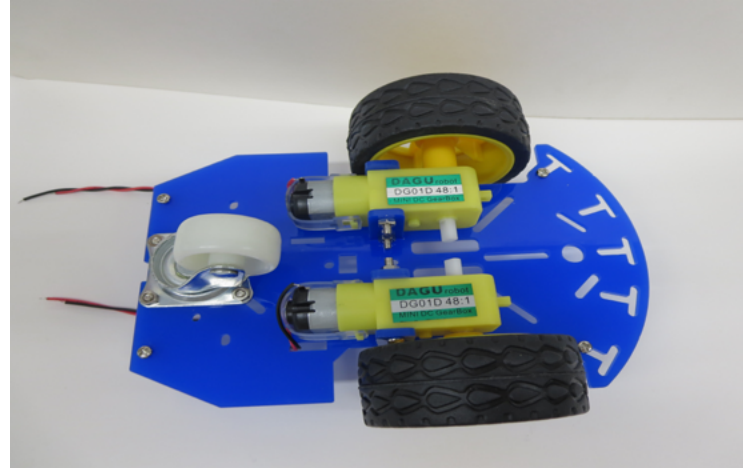
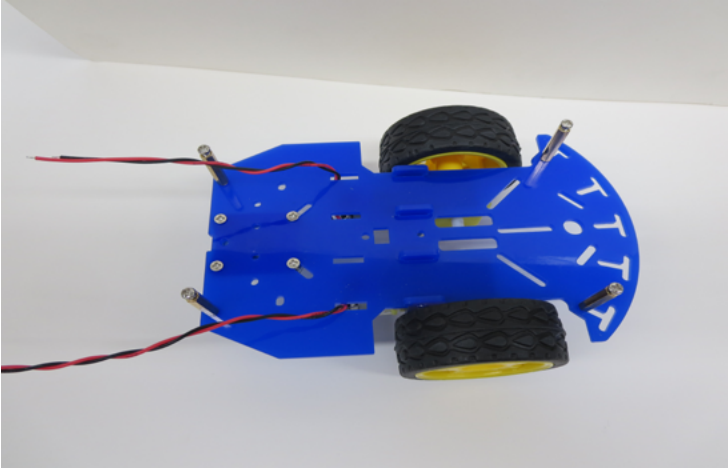


ARDUINO 2WD ROBOT KIT



A. CHASSIS

1. Assemble the 2 Wheel Drive Chassis following the instructions provided in the package.



Note: in Step 2 the Speed Board Holders are not used and can be discarded.

Note: in Step 6 the 4AA battery holder with 2.1mm connector is not used and can be discarded. A 4AA battery holder with switch and wire leads is included that will be used for motor power.

2. Stop after Step 7 and install the Arduino Uno on the top plate using the three provided nylon spacers as shown in Fig. 3, 4.

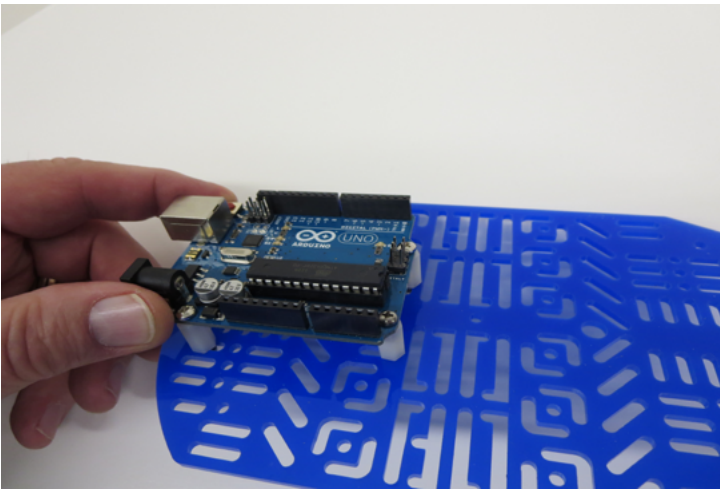


Fig.3

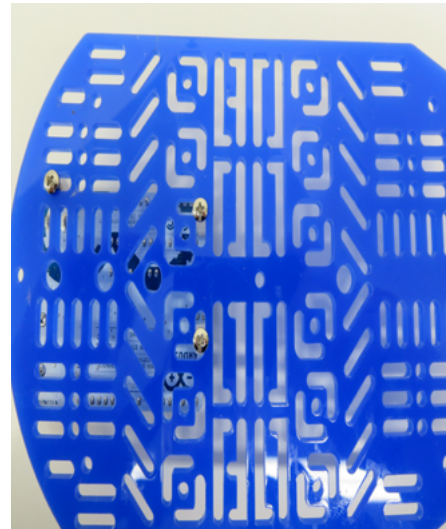


Fig.4

Install the Servo motor with the four metal spacers provided as shown in Fig. 5, 6.

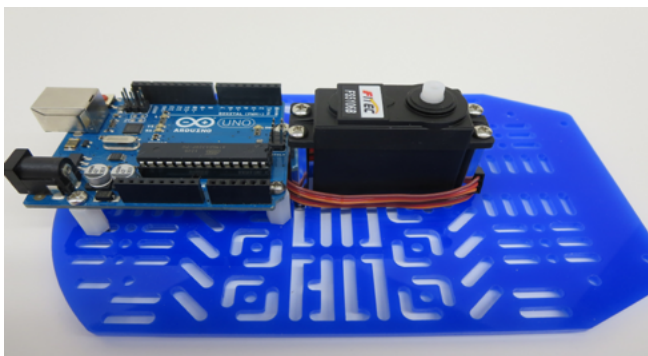


Fig.5

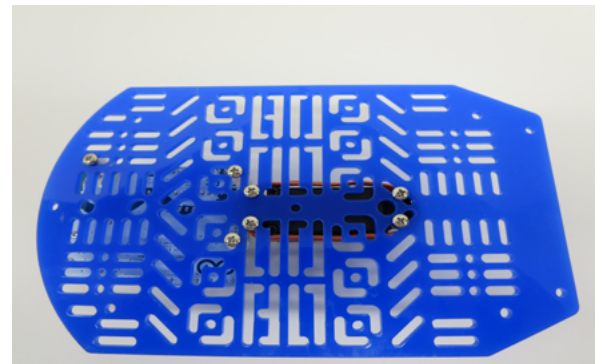


Fig.6

3. In Step 8. attach the top plate with the mounted Arduino Uno and Servo motor to the chassis using the M3*6 screws in the chassis kit.

B. MOTOR CONTROLLER

1. Slide together 2 of the 2 position terminal blocks as shown in Fig. 7, 8 to make a 4 position terminal block for Motor Connections.

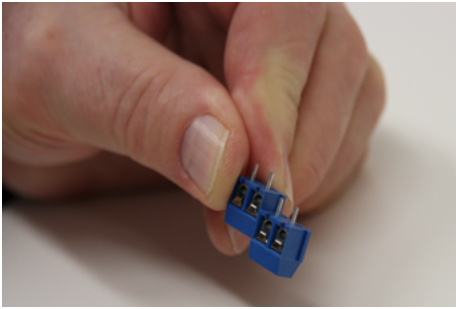


Fig.7

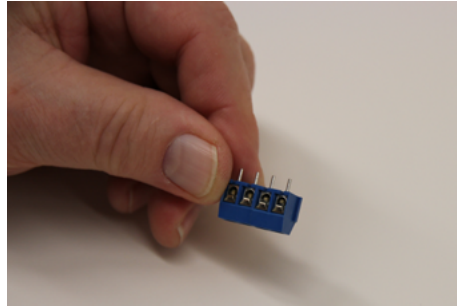


Fig.8

- Solder the 4 position Motor Connection terminal block to the prototype board as shown in Fig. 9, 10.

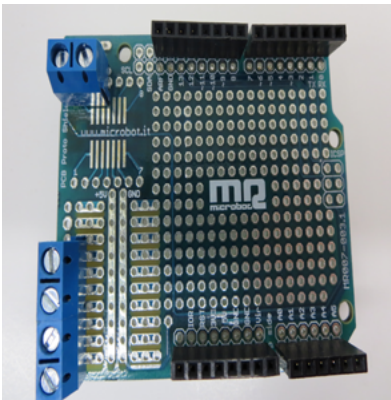


Fig.9

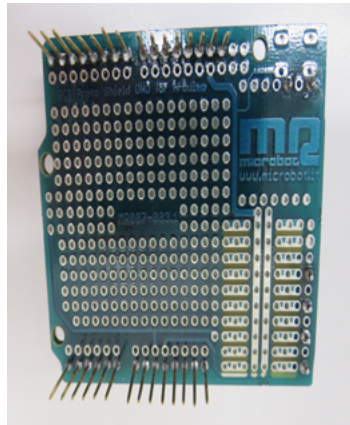


Fig.10

2. Solder the remaining two position terminal block for Motor Power as shown in Fig. 9, 10.
3. Solder the six pin M/F header to the outside strip of A0-A6 as shown in Fig. 9, 10.
Solder the three 8 pin M/F headers to the remaining outside strips as shown in Fig. 9, 10.
4. Plug the Prototype board onto the Arduino Uno with the six pin header aligned with the A0-A5 Arduino Analog input pins as shown in Fig. 11.

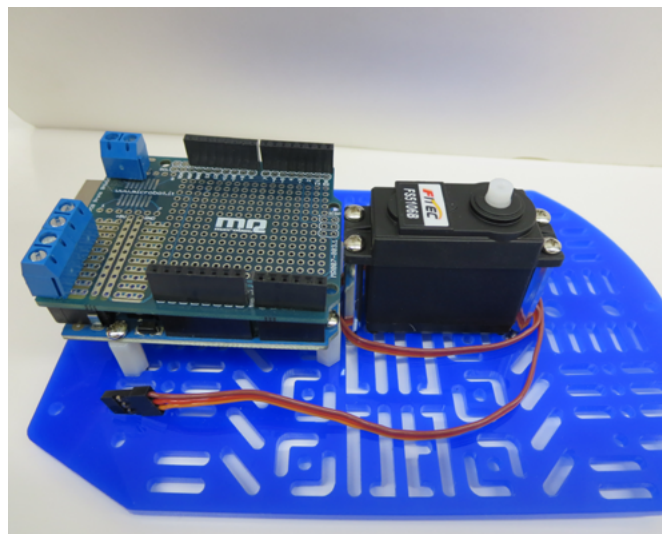


Fig.11

B. MOTOR CONTROLLER (CONTINUED)

5. Mount the Abra-1 blue breadboard onto the Prototype board as shown in Fig. 12. Center the breadboard so that the pin numbers are visible for wiring reference.

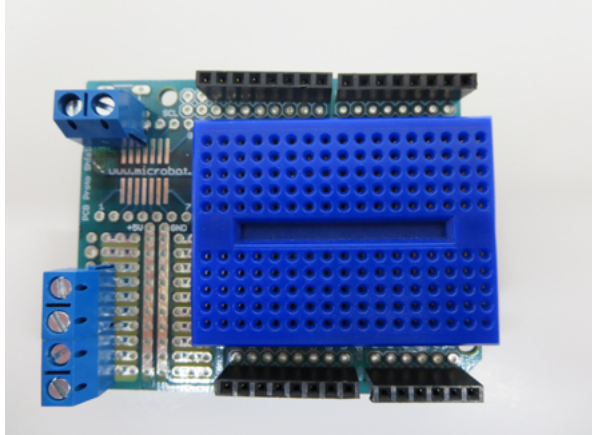


Fig.12

6. Install the L293D Motor Driver IC onto the blue breadboard as shown in Fig. 13. with pin 1 facing the terminal blocks.

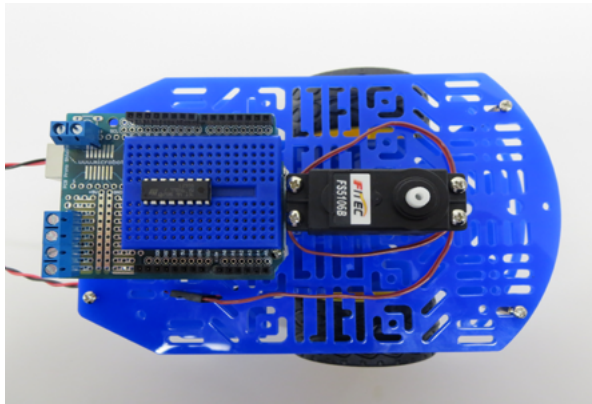


Fig.13

7. Insert the 4AA Battery Pack and connect the Red and Black wires as shown in Fig. 14 to the 2 position Motor Power terminal block.

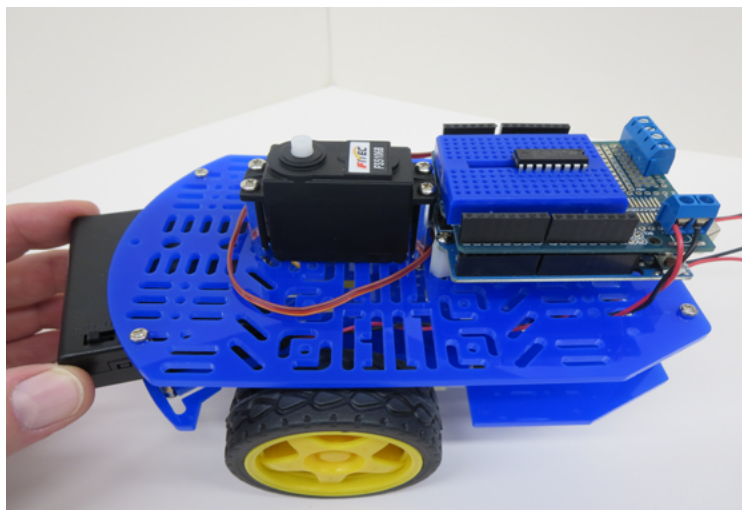


Fig.14

B. MOTOR CONTROLLER (CONTINUED)

8. Motor Power Connections:

See Fig 15:

+6V: Connect a Red wire from the 2 position Motor terminal block Red wire to pin 8 of the L293D.

GND: Connect a Black wire from the 2 position Motor terminal block Black wire to pin 13 of the L293D.

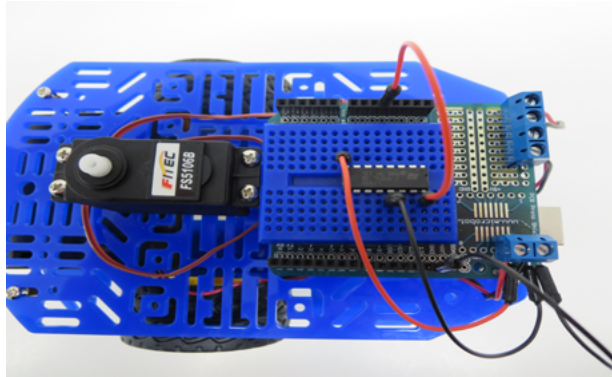


Fig.15

Motor Power Connections:

See Fig. 16:

Jumper pins 13, 12, 5, 4 of the L293D to complete Motor ground connections.

Arduino-Motor GND reference:

Connect a Black wire from the 2 position Motor terminal block Black wire to the Arduino GND pin for GND reference.

Arduino-Motor Driver 5V Reference:

Connect a Red wire from pin 16 of the L293D (Vss) to Arduino 5V Pin.

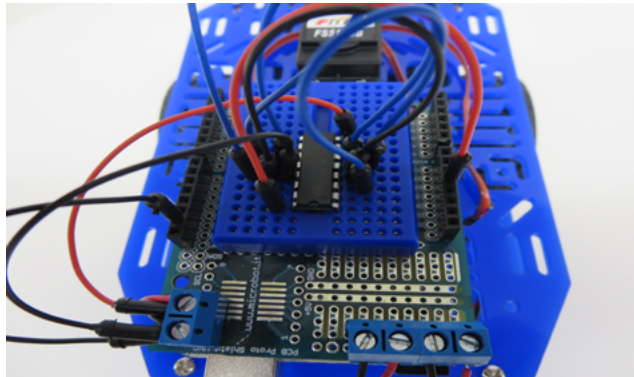


Fig.16

9. Motor Terminal Block Connections:

See Fig. 17

Connect Motor 1 and Motor 2 Red and Black wires to Motor Terminals 1 and 2.

Facing the Motor terminals, Motor 1 and Motor terminal 1 is on the right hand side and Motor 2 and Motor terminal 2 is on the left hand side.

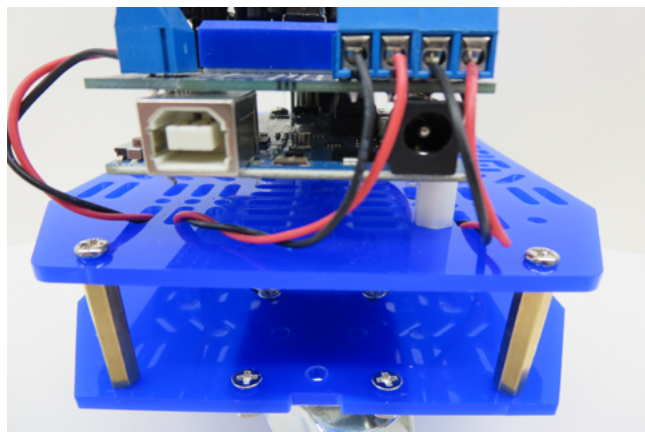


Fig.17

B. MOTOR CONTROLLER (CONTINUED)

10. Motor 1

Connect Pin 1 L293D to Arduino Pin3

Connect Pin 2 L293D to Arduino Pin9

Connect Pin 7 L293D to Arduino Pin8

Connect Pin 3 L293D to Red wire on Motor Terminal 1 (Right Hand Side)

Connect Pin 6 L293D to Black wire on Motor Terminal 1 (Right Hand Side)

11. Motor 2

Connect Pin 9 L293D to Arduino Pin11

Connect Pin 10 L293D to Arduino Pin7

Connect Pin 15 L293D to Arduino Pin6

Connect Pin 14 L293D to the Red wire on Motor Terminal 2 (Left Hand side)

Connect Pin 11 L293D to the Black wire on Motor Terminal 2 (Left Hand Side)

SUMMARY OF L293D CONNECTIONS

Pin	Function	Connection
1	ENABLE1	ARDUINO PIN 3
2	INPUT1	ARDUINO PIN 9
3	OUTPUT1	MOTOR 1
4	GND	MOTOR GND
5	GND	MOTOR GND
6	OUTPUT2	MOTOR 1
7	INPUT2	ARDUINO PIN 8
8	Vs	MOTOR DRIVE SUPPLY (+6V BATTERY NOT ARDUINO +5)
9	ENABLE2	ARDUINO PIN 11
10	INPUT3	ARDUINO PIN 7
11	OUTPUT3	MOTOR 2
12	GND	MOTOR GND
13	GND	MOTOR GND
14	OUTPUT4	MOTOR 2
15	INPUT4	ARDUINO PIN 6
16	Vss	ARDUINO +5

C. SERVO MOTOR CONNECTION

1. Insert the 3 position M/M Header into one end of the Servo Motor Cable as shown in Fig. 18.

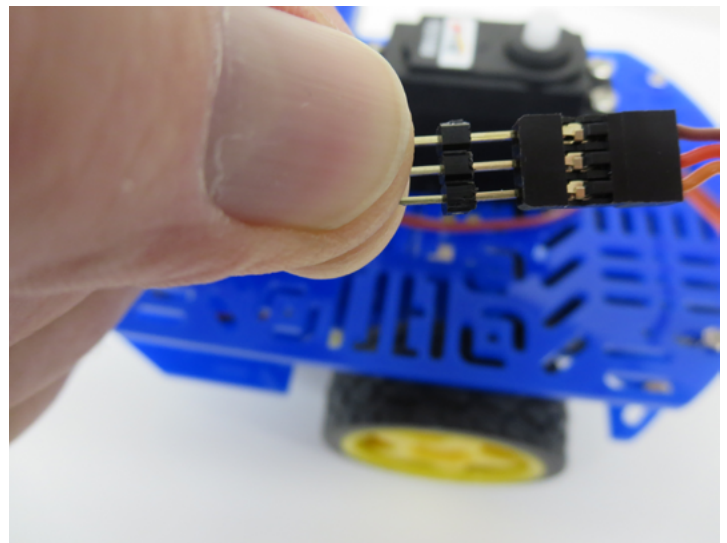


Fig.18

2. Plug the Servo Motor Cable into the breadboard as shown in Fig. 19.

2. Connect the Orange Servo Motor Cable wire to Arduino Pin 5.
Connect the Red Servo Motor Cable wire to the Red wire at the 2 position Motor Power terminal block (+6V).
Connect the Brown servo cable wire to Arduino GND.

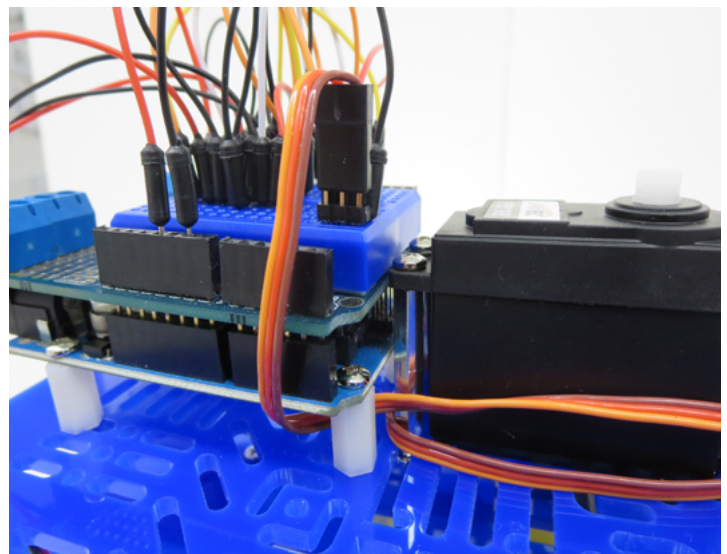


Fig.19

D. ULTRASONIC SENSOR CONNECTION

1. Insert the 4 position M/M Header into one end of the 4 wire Ultrasonic Sensor Cable As shown in Fig.20

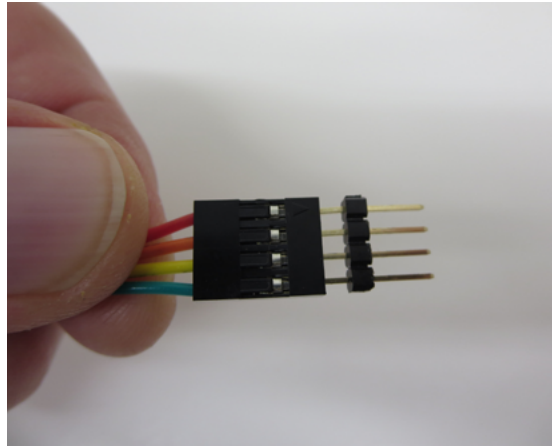


Fig.20

Plug the Ultrasonic Sensor Cable header end into the breadboard as shown in Fig. 21.
Connect the Red wire to pin 16 of the L293D (Arduino +5V).
Connect the Orange wire to Arduino Pin 4 (TRIG).
Connect the Yellow wire to Arduino Pin 2 (ECHO).
Connect the Green wire to Arduino GND (GND).

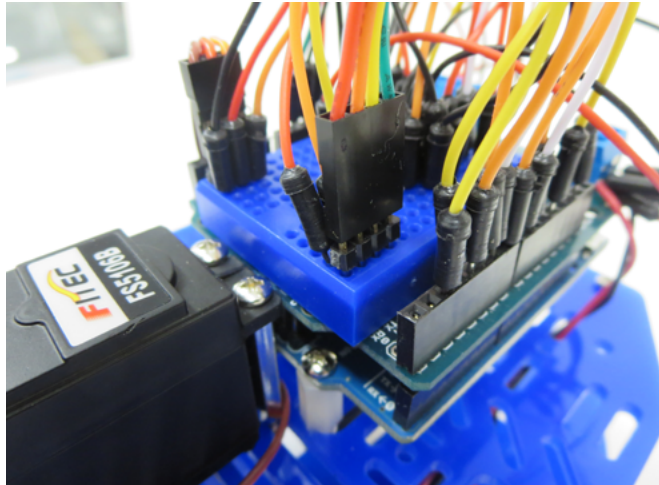


Fig.21

Plug the other end of the 4 wire Ultrasonic Sensor Cable to the Ultrasonic Sensor matching the Green wire to the GND connection on the Sensor as shown in Fig. 22.

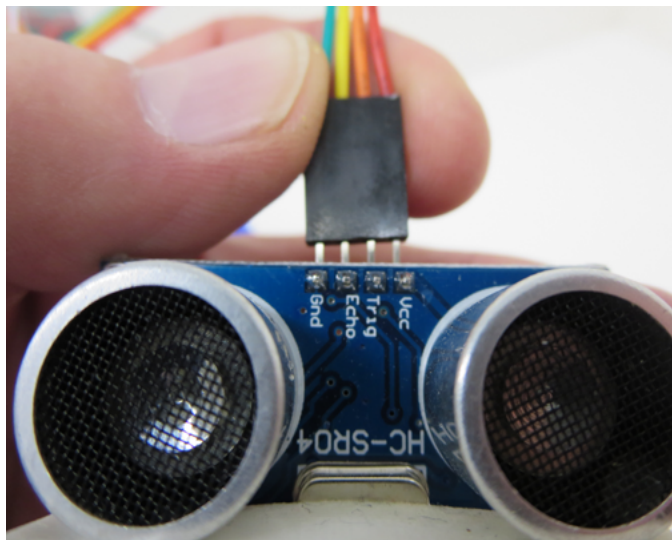


Fig.22

D. ULTRASONIC SENSOR CONNECTION (CONTINUED)

2. Select the two arm servo coupler from the servo motor attachments and push it onto the servo shaft as shown in Fig 23.

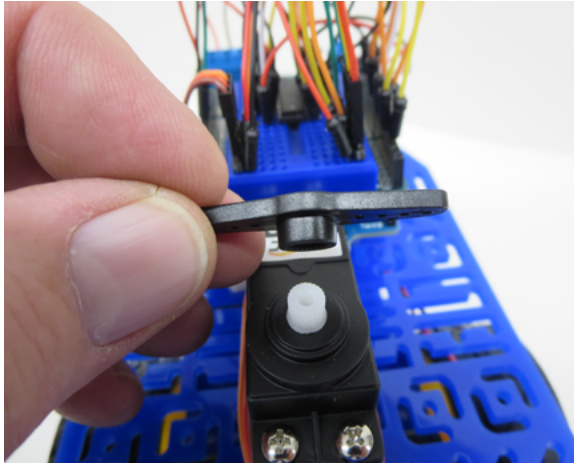


Fig.23

Turn the arm manually and reset its rest position to have the arm parallel to the front with equal 90 degree swing to each side as shown in Fig.24.

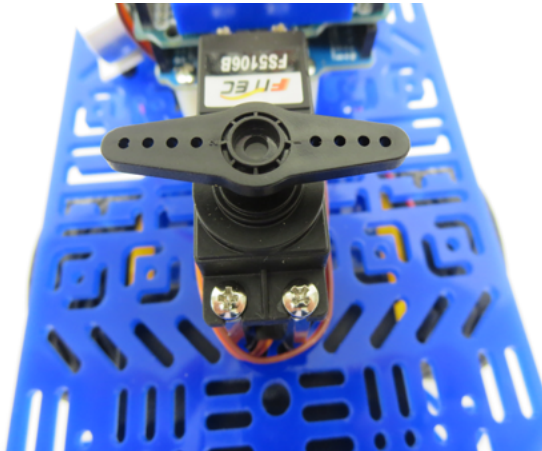


Fig.24

Attach the 1.5 inch double sided tape to the bottom of the two ultrasonic sensor tubes as shown in Fig 25.

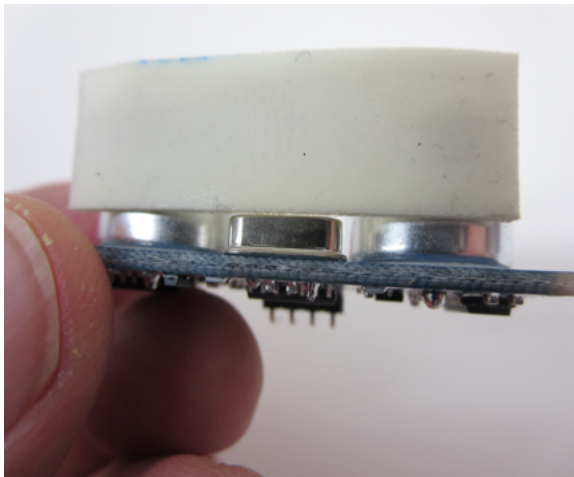


Fig.25

D. ULTRASONIC SENSOR CONNECTION (CONTINUED)

Remove the two arm coupler from the servo motor and attach it to the bottom of the ultrasonic sensor as shown in Fig. 26.

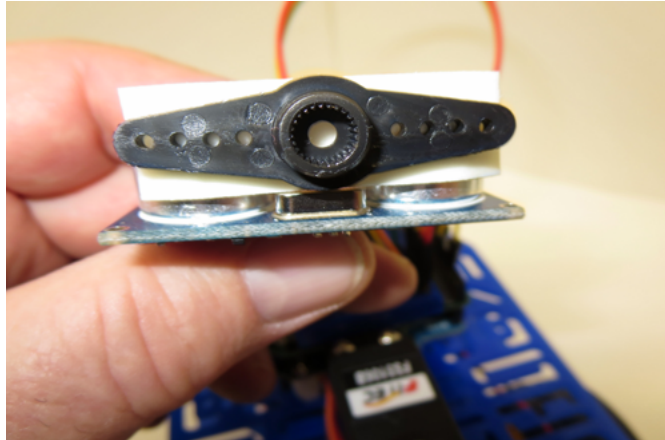


Fig.26

Mount the Ultrasonic sensor onto the servo motor as shown in Fig. 27

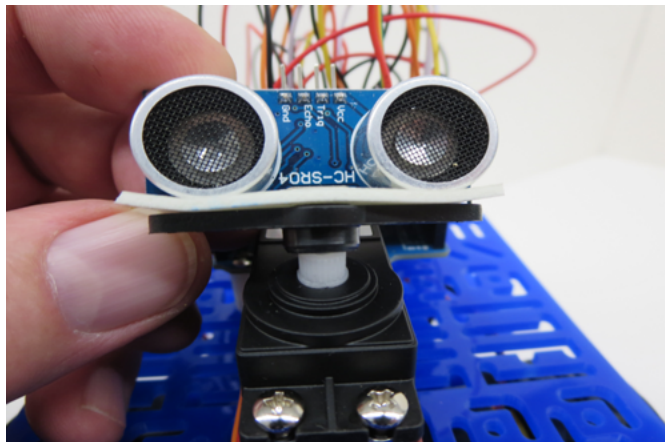


Fig.27

E. PROGRAMMING THE ARDUINO

Download the Arduino IDE from <http://www.arduino.cc/>
Download the "testrobot" sample sketch from www.abra-electronics.com
Connect a USB A-B cable from your computer to the Arduino.
Upload the "testrobot" sketch into your Arduino UNO
Disconnect the USB A-B cable after downloading the program.

Install 4AA batteries into the 4AA Motor Battery holder.
Install a 9 Volt battery into the 9 Volt Arduino Battery holder.
Install the 9 Volt Arduino battery as shown in Fig. 28.
Turn both battery switches on and run the program.
The program is triggered by holding your hand in front of the ultrasonic sensor.
Re-check all your wire connections if any problems occur.

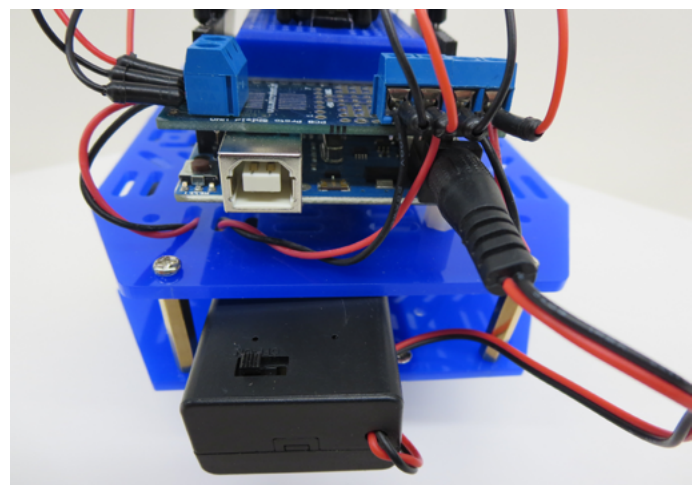


Fig.28

Code for Arduino Robot

```
// Created for Abra's Robot using various sources
// Created on April 8th, 2014
// Created by Oscar and Subash
// Use this code to test your Aduino Robot with the Arduino board.
// Modified Date: 10/04/2014

#include <NewPing.h>          // Header for ping function - to test for the distance from the object
#define TRIGGER_PIN 4        // Trigger Pin on the Arduino Board for ultrasonic range finder
#define ECHO_PIN 2           // Echo Pin the Arduino Board for ultrasonic range finder
#define MAXIMUM_DISTANCE 250 // Maximum distance (range) for the purposes of Ping in centimeters

NewPing sonar (TRIGGER_PIN, ECHO_PIN, MAXIMUM_DISTANCE); // Setup sonar elements

float Thresholddistance = 20.0; // Maximum detection range of the closest object in front, to the left or to // the right, in centimeters

#include <Servo.h>           // Header for the servo motor
#define pinServo 5           // Arduino Pin 5 for the Servo motor

//----- Set up Servo Angles
#define SERVOLEFT 180       // Left - Direction of the servo
#define SERVORIGHT 0        // Right - Direction of the servo
#define SERVOCENTER 90      // Center - Direction of the servo
Servo myServo;              // Create servo object to control a servo
int angleCur = 0;          // Initial direction of the servo

int speed = 255;            // Speed of the motors
// ----- Motors
int motor_left[] = {3,9, 8}; // Initialize the pins for Left Motor connected on Arduino UNO board //Enable1 Input1 Input2
int motor_right[] = {11,7, 6}; // Initialize the pins for Right Motor connected on Arduino UNO board - //Enable2 Input3 Input4
// ----- Setup Module

void setup() {
  Serial.begin(9600);        // Sets the data rate in bits per second (baud) for serial data transmission.

  myServo.attach(pinServo); // Pin 5 for the Servo motor

  // Setup motors
  // Loop to setup the pinModes

  for(int i = 0; i < 3; i++)
  {
    pinMode(motor_left[i], OUTPUT); // Initialize the pins as output for the left motor
    pinMode(motor_right[i], OUTPUT); // Initialize the pins as output for the right motor
  }

  // position servo to center
  servoAngle(SERVOCENTER);

}

// ----- Loop Module
void loop() {
  float forwardDistance = ping();
  drive_forward();
}
```

Code for Arduino Robot (CONTINUED)

```
if (forwardDistance > Thresholddistance)
{
  drive_forward();
}
else
{
  brake();
  servoAngle(SERVOLEFT);    // check for obstacles to the left
  float leftDistance=ping();
  servoAngle(SERVORIGHT);   // check for obstacles to the right
  float rightDistance=ping(); // find the distance to the right
  servoAngle(SERVOCENTER);
  if (leftDistance > rightDistance && leftDistance > Thresholddistance){
    turn_left();
  }
  else if(rightDistance > leftDistance && rightDistance > Thresholddistance){
    turn_right();

  }
  else {
    turn_around();
  }
  delay(1000);
}
}
```

```
// ----- Drive
```

```
// Following Module is for the Robot to stop for 6 seconds
```

```
void motor_stop(){
  int i;
  for(i = 0; i < 3; i++)
  {
    digitalWrite(motor_left[i], LOW);
    digitalWrite(motor_right[i], LOW);
    delay(6000);
  }
}
```

```
// Following Module is for the Robot to slow down
```

```
void brake()
{
  digitalWrite(motor_left[0], HIGH);
  digitalWrite(motor_left[1], LOW);
  digitalWrite(motor_left[2], LOW);

  digitalWrite(motor_right[0], HIGH);
  digitalWrite(motor_right[1], LOW);
  digitalWrite(motor_right[2], LOW);

  delay(25);
}
```

```
// Following Module is for the Robot to go forward
```

```
void drive_forward()
{
```


Code for Arduino Robot (CONTINUED)

```
digitalWrite(motor_left[0], HIGH);  
digitalWrite(motor_left[1], HIGH);  
digitalWrite(motor_left[2], LOW);
```

```
digitalWrite(motor_right[0], HIGH);  
digitalWrite(motor_right[1], LOW);  
digitalWrite(motor_right[2], HIGH);
```

```
}
```

```
// Following Module is for the Robot to go backwards  
void drive_backward()
```

```
{  
  digitalWrite(motor_left[0], HIGH);  
  digitalWrite(motor_left[1], LOW);  
  digitalWrite(motor_left[2], HIGH);
```

```
  digitalWrite(motor_right[0], HIGH);  
  digitalWrite(motor_right[1], HIGH);  
  digitalWrite(motor_right[2], LOW);
```

```
}
```

```
// Following Module is for the Robot to turn left  
void turn_left()
```

```
{  
  digitalWrite(motor_left[0], HIGH);  
  digitalWrite(motor_left[1], HIGH);  
  digitalWrite(motor_left[2], LOW);
```

```
  digitalWrite(motor_right[0], HIGH);  
  digitalWrite(motor_right[1], HIGH);  
  digitalWrite(motor_right[2], LOW);
```

```
  delay(300);
```

```
}
```

```
// Following Module is for the Robot to turn right  
void turn_right()
```

```
{  
  digitalWrite(motor_left[0], HIGH);  
  digitalWrite(motor_left[1], LOW);  
  digitalWrite(motor_left[2], HIGH);
```

```
  digitalWrite(motor_right[0], HIGH);  
  digitalWrite(motor_right[1], LOW);  
  digitalWrite(motor_right[2], HIGH);
```

```
  delay(300);
```

```
}
```

Code for Arduino Robot (CONTINUED)

```
// Following Module is to make a u-turn if there are obstacles to the right, to the left and in front
void turn_around()
{
  digitalWrite(motor_left[0], HIGH);
  digitalWrite(motor_left[1], LOW);
  digitalWrite(motor_left[2], HIGH);

  digitalWrite(motor_right[0], HIGH);
  digitalWrite(motor_right[1], LOW);
  digitalWrite(motor_right[2], HIGH);

  delay(600);
}

// Module for the Servo
void servoAngle(int angleNew)
{
  // change direction when limits are reached

  if (angleNew > angleCur)
  {
    for (int angle = angleCur; angle < angleNew; angle += 1)
    {
      myServo.write(angle);
      delay(20);
    }
    angleCur = angleNew;
  }
  else if (angleNew < angleCur)
  {
    for (int angle = angleCur; angle > angleNew; angle -= 1)
    {
      myServo.write(angle);
      delay(20);
    }
    angleCur = angleNew;
  }
}

// Following Module is to test for ping using ultrasonic range finder
float ping()
{
  delay(50);
  unsigned int uS = sonar.ping();
  if (uS == 0)
  {
    return MAXIMUM_DISTANCE;
  }
  else
    return (uS / US_ROUNDTRIP_CM);
}
```