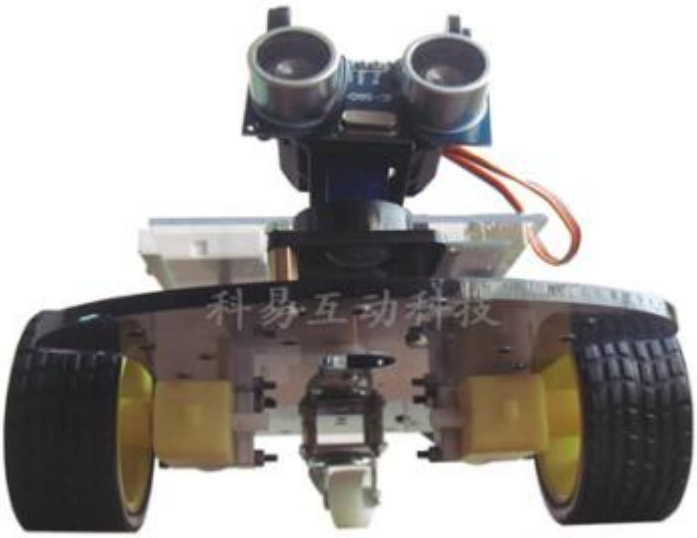
# Installation manual for smart ultrasonic obstacle avoidance car




## I) Introduction

Arduino smart ultrasonic obstacle avoidance car is an application development system for microcontroller learners. It takes arduino atmega-328 as its core to realize ultrasonic obstacle avoidance function. This kit contains large numbers of interesting programs and supports expansion of external circuit modules, thus enhancing the usability of this smart car. Our final goal is to free learners from the boring theoretical knowledge concerning arduino microcontroller and foster learners' ability in microcontroller system development.

## II) Parameters:

1. Motor voltage: 1.5V~12V
2. Length of motor shaft: 10mm
3. Rotate speed of motor: 6.0V 100rpm/min
4. The controlling motor adopts L298N module to realize separation from microcontroller.
5. The obstacle avoidance part adopts HC-SR04 ultrasonic to realize stable performance and precise distance measurement.
6. Supports connection with external voltage of 7V~12V and can carry multiple sensor modules as needed.

## III) Introduction to experimental courses:

1. Application of L298N motor drive board
2. Application of ultrasonic module
3. Smart ultrasonic obstacle avoidance car

## IV) List:

1. Gear motor *2
2. High-quality tire *2
3. Motor fixed part *2
4. Mecanum wheels *1
5. 100*150*2.6MM
6. L298N motor drive board *1
7. Arduino 328 strengthened board *1
8. Holder *1
9. Servo *1
10. Ultrasonic module *1
11. Mini breadboard *1
12. 6 cell AA battery holder *1
13. Dupont cable *12
14. 1M USB cable *1
15. Copper pillar (35mm in length) *3
16. Copper pillar (12mm in length) *4
17. Several 3mm bolts and nuts

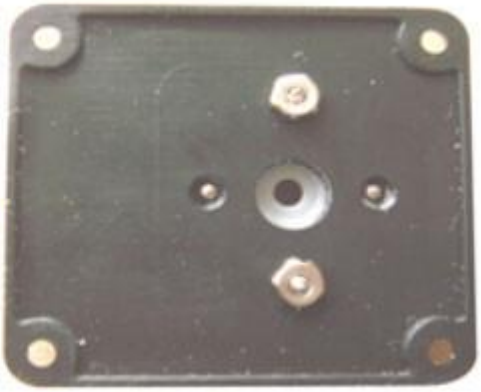
## V) Instruction for installing holder of this smart car

Take out the cross colloid out of the servo accessory bag.

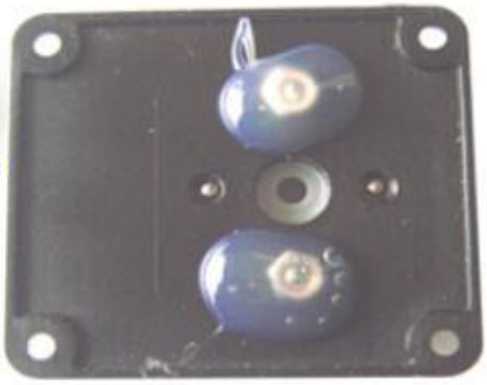
Cut the longer sides of the cross and trim their width until four sides are even

Mount the 2*8mm and 1.2*5mm screws onto the second bore of the cross as below pic. shows, then fix the cross onto bottom plate of the holder

Mount the nuts onto the 2*8mm screw on the bottom plate of the holder



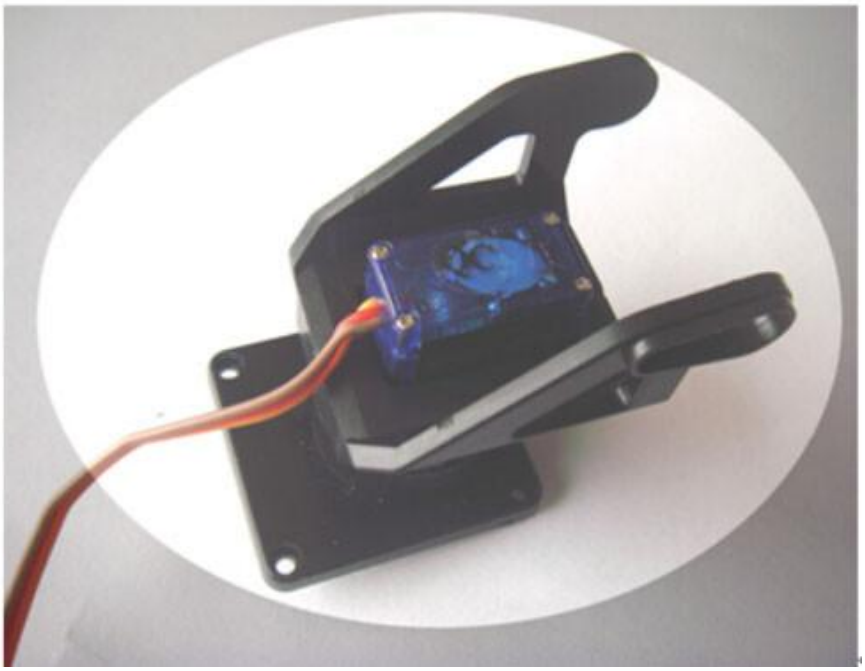
Fix the screw position with hot melt glue



Mount two side wings of the holder onto the servo

After the side wings are mounted, fixed it with screws


Put the well-mounted servo into well-fixed cross colloid and adjust direction

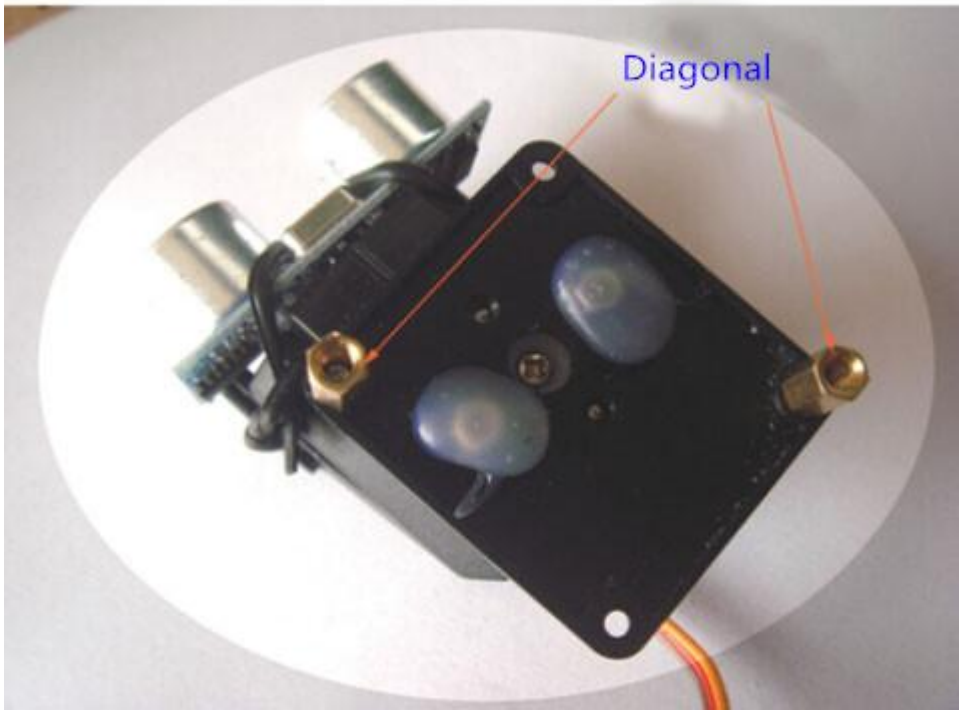


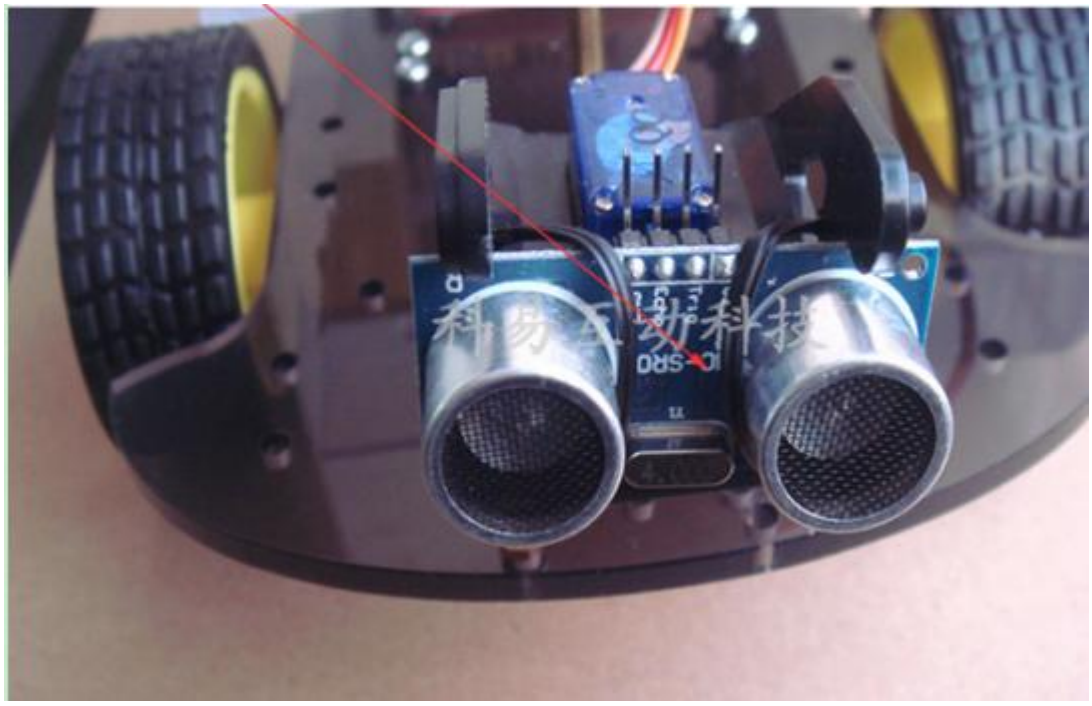
Take out 2*6mm screw from servo accessory bag and mount the screw into the fixing hole of the servo

Fixing hole of the servo

Mount the ultrasonic module in front of the holder with ribbon

Mount the 6mm copper pillar onto the fixing hole of bottom plate of the holder

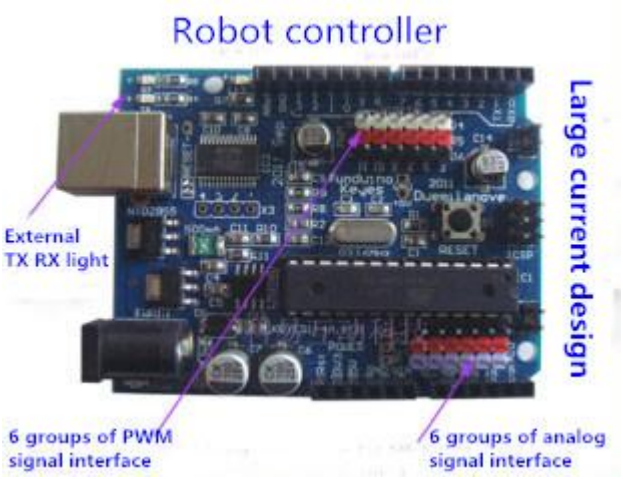


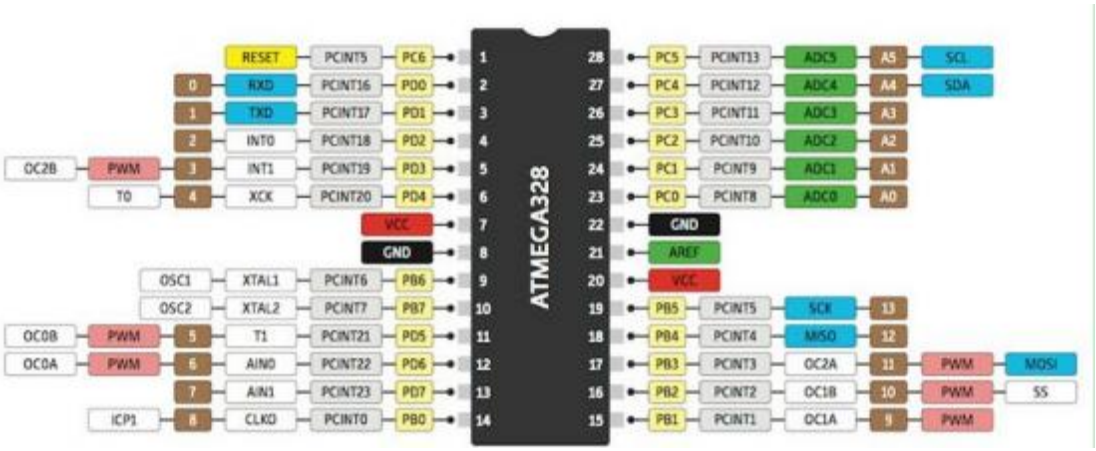
Installation well done! thanks!

## VI) Usage of Arduino microcontroller

1. Introduction




Arduino is a hardware platform with source code open to everyone. This platform is comprised of a circuit board with basic I/O function and a set of program development software. Arduino can be used to develop various interactive products. For example, it can read/write numerous switch and sensor signal and control lights, motors and assorted physical device. In addition, arduino can also be used to develop peripheral device relates to PC and can communicate with PC software during operation. You can DIY Arduino hardware circuit board or get a ready-made module. While the software for program development environment can be downloaded online for free.




Let’s see how arduino team define their product:

Arduino is an electronics prototype platform with source code open for everyone. Arduino is designed for designers, are workers, enthusiasts and those interested in developing interactive devices or interactive environment. Arduino can receive input signal from various sensors and detect the operational environment, then it changes its surroundings via controller power supply,

motor and other drives. The onboard microcontroller uses arduino program language(based on Wiring) and arduino development environment(based on Processing). Arduino can work independently or communicate with PC software(such as Flash, Processing, MaxMSP).

As said before, you can DIY your own arduino hardware circuit board or get a ready-made one, you can get hardware reference design(CAD file) according to open source permit and modify it at will as needed.

2. Installing arduino drive and burn programs

First, download arduino development software on http://arduino.cc/en/Main/Software
The downloaded file is a compress folder named arduino-0023.zip, unzip it onto the hardware.

When connecting arduino strengthened board with Windows via USB, it will remind you that a new USB device named “FT232R USB UART” has been found, then Windows will lead us to the window showing “found new hardware wizard”, select “NO, not now”, and click “Next” button:


Then select drives needed for installing arduino, select“install from list or designated position(advanced)”, then click“Next” button:

The USB driver of arduino should be put in drivers catalogue under Arduino 0021 installation catalogue, we need to indicate that this catalogue is the one to search when installing drivers:

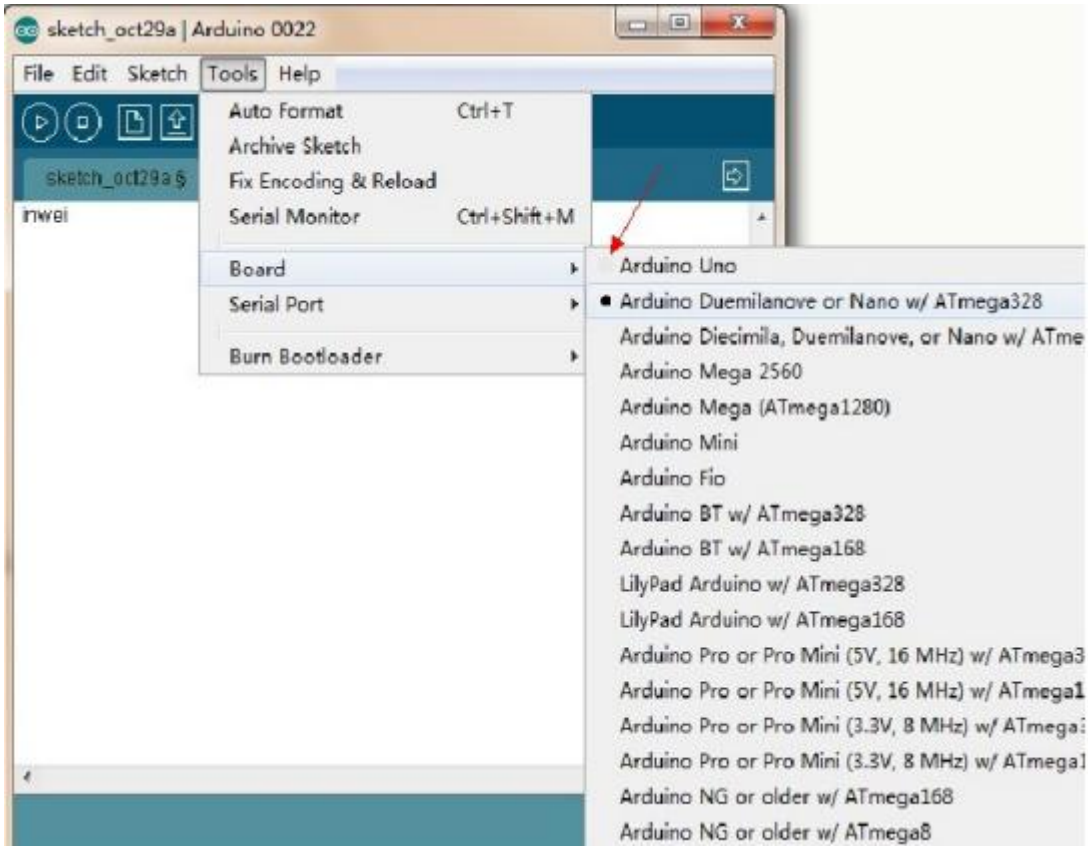

Click “Next” button, Windows will start to search and install USB drive program of Arduino:


If everything goes well, we’ll see below interface:

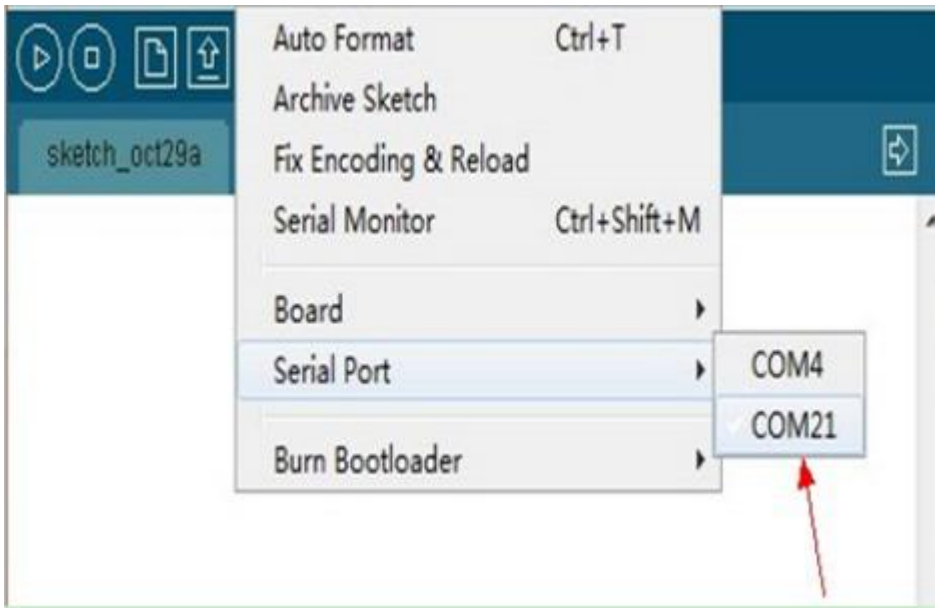

After USB driver of Arduino is installed, we can search for corresponding Arduino serial ports in Windows device manager:


Now let’s see how to burn the first program: light up “L” light.

Click [tools] on Arduino-0023 programming interface, then move the mouse to [board] option on pull-down menu, check whether there is a black point in front of [arduino Duemilanove] in the subsequent sub-menu, if there is no black point, then click[arduino Duemilanove] option.

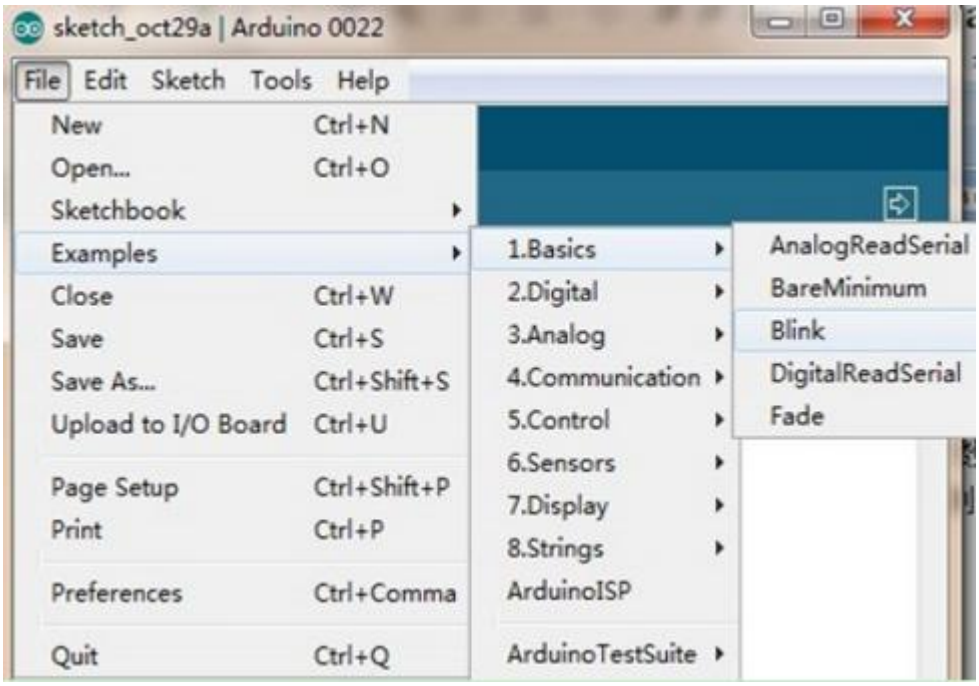



Then, we need to choose the correct COM connection. Do you remember the X value(COMX) you recorded during hardware installation? As the port of Arduino installed just now is 21, click 21.

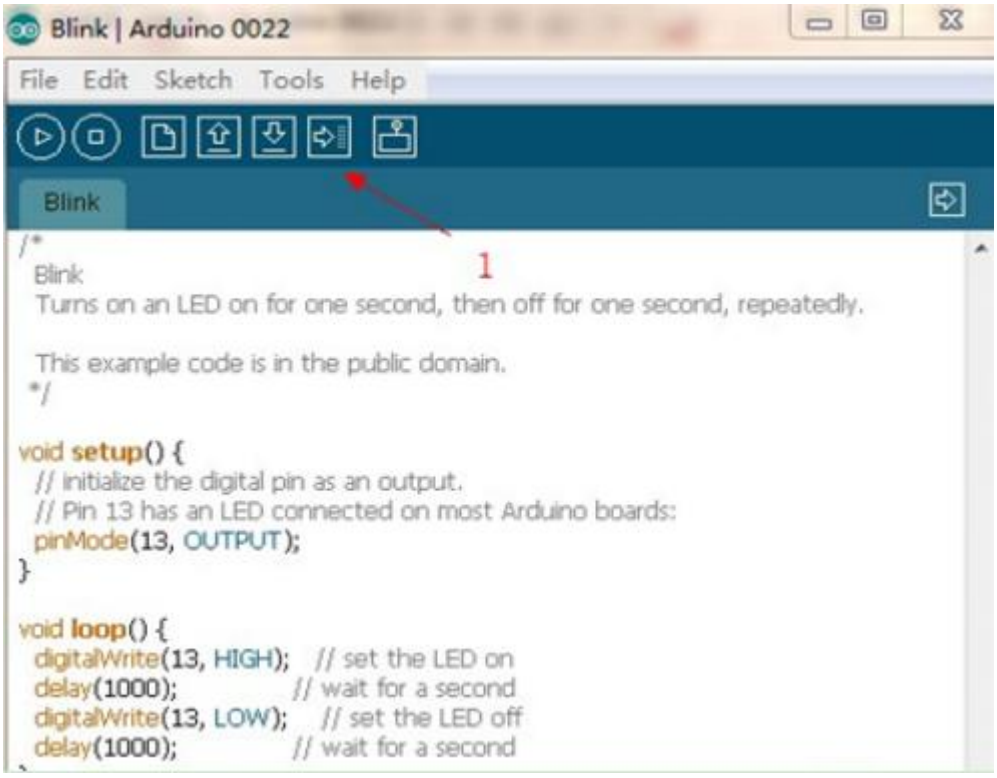



Now let's import a sample program to let "L" light blinker. Click [File] via left mouse button, then move mouse to [Examples] on pull-down menu propped out, the menu will expand rightwards to

[1.Basics], after move mouse to [1.Basics], the menu will continue to expand, find [Blink] and click it with left mouse button.




After click [Blink], an Arduino programming interface will be propped out



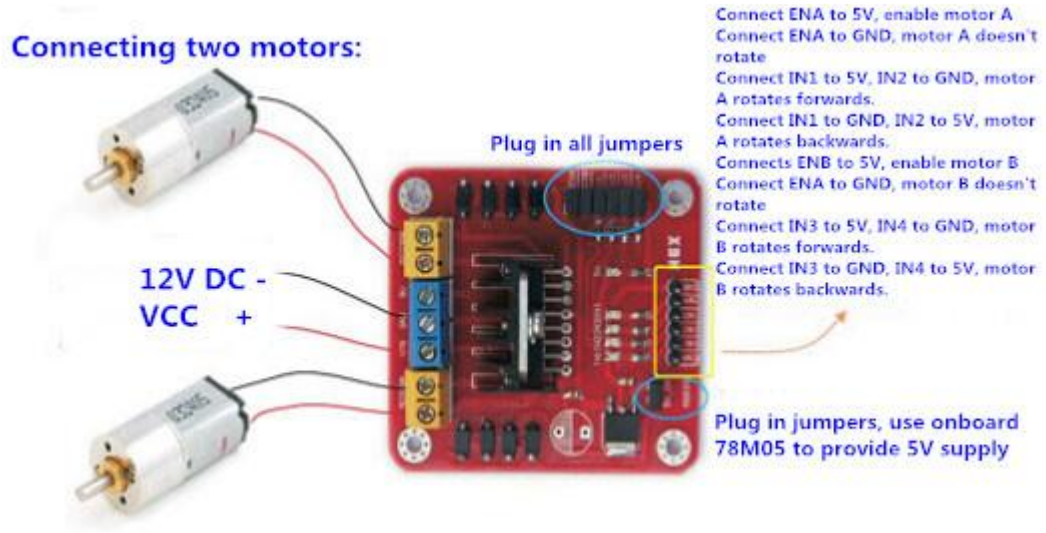

Click the icon pointed by red arrow 1 in the left pic., you'll find that the two yellow lights on

Arduino main board will blinker for a while.
With the two blinkering lights go out, a text tip will be shown under programming frame,
The L light on the main board blinkers at 1 sec interval.
Congratulations! Your first program has been succeeded!

## VII) Detailed introduction to experiment

### 1. Application of L298N motor drive board

Please refer to L298N double H bridge DC motor drive board manual for L298N drive board. As some users are still not clear about how to control double road DC motor, we'll make a detailed instruction on it here:




The power supply of UMS drive part can be provided externally, 9V is appropriate under most conditions; the logic part can take power from within the board, ie. the port can either hang in the air or connect with +5V~+7V. The three pins on both sides of the port are for controlling two road DC motor respectively. EA and EB should be connected with Arduino PWM port for regulating motor speed. I1, I2 and I3 ports are for controlling the forward, backward, turn and brake of two road DC motor respectively, they only need to be plugged in to Arduino digital ports.

Now the preparation work is done, we can move on to write programs. Here, the forward, backward, turn left, turn right and brake functions will all be written into programs for your reference.

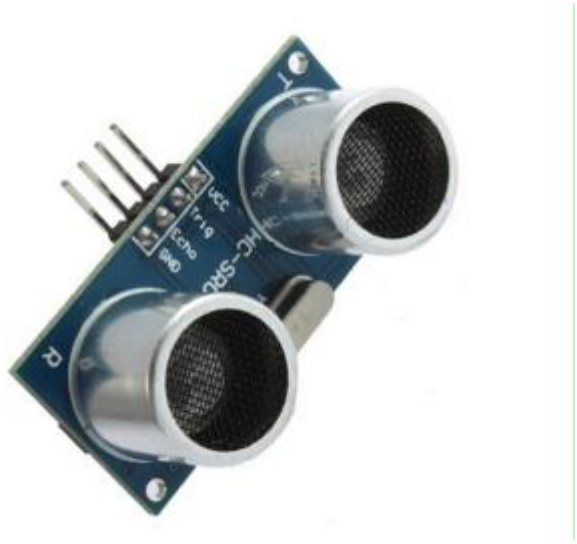
Sample programs are listed as below:

```
int pinI1=8;//define I1 port
int pinI2=9;//define I2 port
int speedpin=11;//define EA(PWM speed adjustment)port
int pinI3=6;//define I3 port
int pinI4=7;//define I4 port
int speedpin1=10;//define EB(PWM speed adjustment)port
void setup()
{
pinMode(pinI1,OUTPUT);
pinMode(pinI2,OUTPUT);
pinMode(speedpin,OUTPUT);
pinMode(pinI3,OUTPUT);
pinMode(pinI4,OUTPUT);
pinMode(speedpin1,OUTPUT);
}
void loop()
{
//go straight forward
analogWrite(speedpin,100);//input analog value and set up speed
analogWrite(speedpin1,100);
digitalWrite(pinI4,LOW);//let DC motor(right) turn anticlockwise
digitalWrite(pinI3,HIGH);
digitalWrite(pinI1,LOW);//let DC motor(left) turn clockwise
digitalWrite(pinI2,HIGH);
delay(2000);
//backward
analogWrite(speedpin,100);//input analog value and set up speed
analogWrite(speedpin1,100);
digitalWrite(pinI4,HIGH);//let DC motor(right) turn clockwise
digitalWrite(pinI3,LOW);
digitalWrite(pinI1,HIGH);//let DC motor(left) turn anticlockwise
digitalWrite(pinI2,LOW);
delay(2000);
//turn left
analogWrite(speedpin,60);//input analog value and set up speed
analogWrite(speedpin1,60);
digitalWrite(pinI4,LOW);//let DC motor(right) turn anticlockwise
digitalWrite(pinI3,HIGH);
digitalWrite(pinI1,HIGH);//let DC motor(left) turn anticlockwise
digitalWrite(pinI2,LOW);
delay(2000);
//turn right
analogWrite(speedpin,60);//input analog value and set up speed
analogWrite(speedpin1,60);
```

```
digitalWrite(pinI4,HIGH);//let DC motor(right) turn clockwise
digitalWrite(pinI3,LOW);
digitalWrite(pinI1,LOW);//let DC motor(left) turn clockwise
digitalWrite(pinI2,HIGH);
delay(2000);
//brake
digitalWrite(pinI4,HIGH);//brake the DC motor(right)
digitalWrite(pinI3,HIGH);
digitalWrite(pinI1,HIGH);//brake the DC motor(left)
digitalWrite(pinI2,HIGH);
delay(2000);
}
```

Note: The turn left and turn right function used in the program is just one way of turning control, you can try other ways of turning control by yourself.

**2. Ultrasonic distance measuring module**




1. Scope of Application

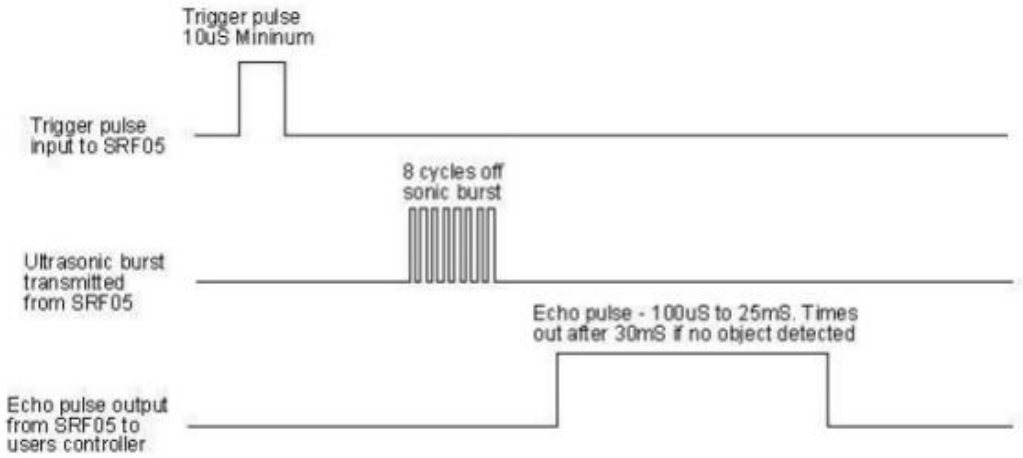
Ultrasonic is frequently used in measuring distance due to its strong directionality, low energy consumption and long spread distance. Ultrasonic distance measurement is usually more efficient and precise, thus it is widely used in mobile robots development.

2. Product introduction

HY-SRF05 ultrasonic distance measuring module can provide 2cm~450cm non-contact detecting distance with precision up to 3mm, which is more than enough to meet our normal needs. This module is comprised of ultrasonic transmitter, receiver and corresponding controlling circuit.

3. Working principle

Let’s first check its working order

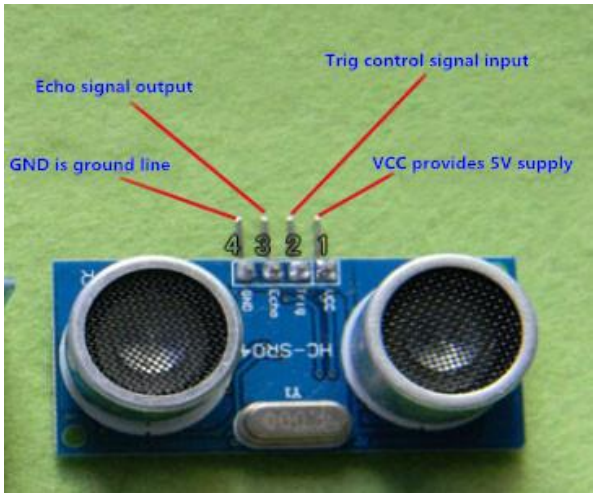



1. First, we lower TRIG, then give at least 10is high level for trigger;
2. After trigger, the module will automatically send 8 40KHZ square waves and detects whether there is returning signal.
3. If there is returning signal, one high level will be output via ECHO, the duration of high level is the time spent from transmission to receipt of ultrasonic. Then the testing distance=duration of high level *340m/s*0.5;

4. Electrical parameter

Working voltage: 0.5V(DC)
Working current: 15mA
Detection distance: 2-450cm
Detection angle:15 degree
Input trigger pulse: 10us TTL level
Output echo signal: output TTL level signal(high), in direct proportion to range

5. Instruction for use

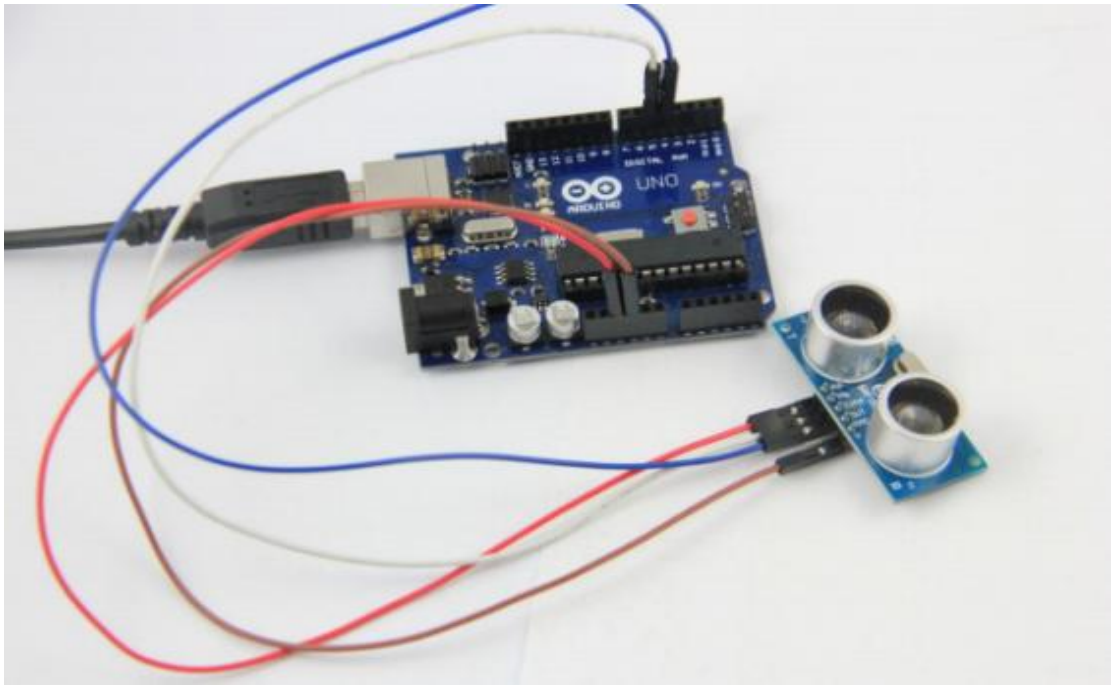



Pin illustration of module is shown above. You only need to control Trig and Echo ports when using Arduino. Conrete speaking, connect these two ports to two digital ports. There will be sample program below to show how to control it. Then you only need to connect with power supply and GND...

6. Module test

Arduino controller *1
USB cable *1
Ultrasonic module *1

Now let’s check the connection part

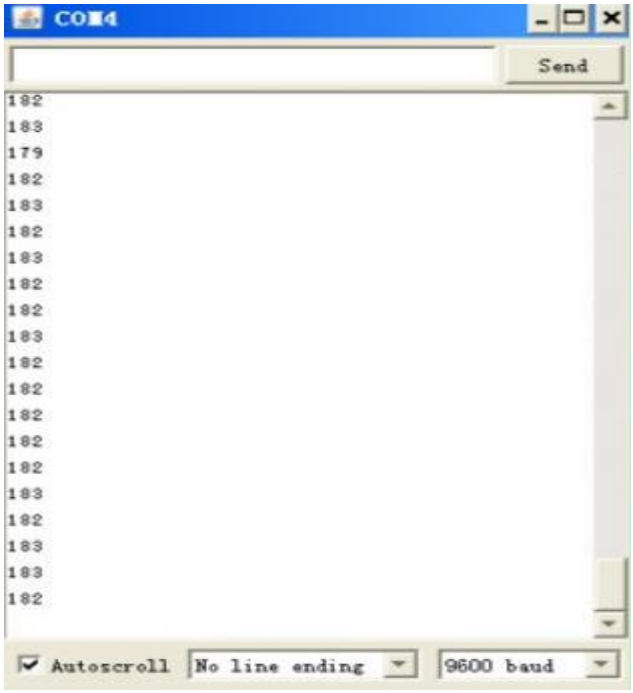


The above D4 and D5 refers to NO.4 and 5 pins of digital ports. Now, we need to learn hwo to use it to measure distance and display it on PC.

Sample code are listed as below:

```
int inputPin=4; // define ultrasonic signal receiver pin ECHO to D4
int outputPin=5; // define ultrasonic signal transmitter pin TRIG to D5
void setup()
{
Serial.begin(9600);
pinMode(inputPin, INPUT);
pinMode(outputPin, OUTPUT);
}
void loop()
{
digitalWrite(outputPin, LOW);
delayMicroseconds(2);
digitalWrite(outputPin, HIGH); // Pulse for 10μ  s to trigger ultrasonic
detection
delayMicroseconds(10);
digitalWrite(outputPin, LOW);
int distance = pulseIn(inputPin, HIGH); // Read receiver pulse time
distance= distance/58; // Transform pulse time to distance
Serial.println(distance); //Output distance
delay(50);
}
```

Compile above sample code and download them to arduino control board, then open Serial Monito window and you'll see a series of data, that's what we want as shown below...

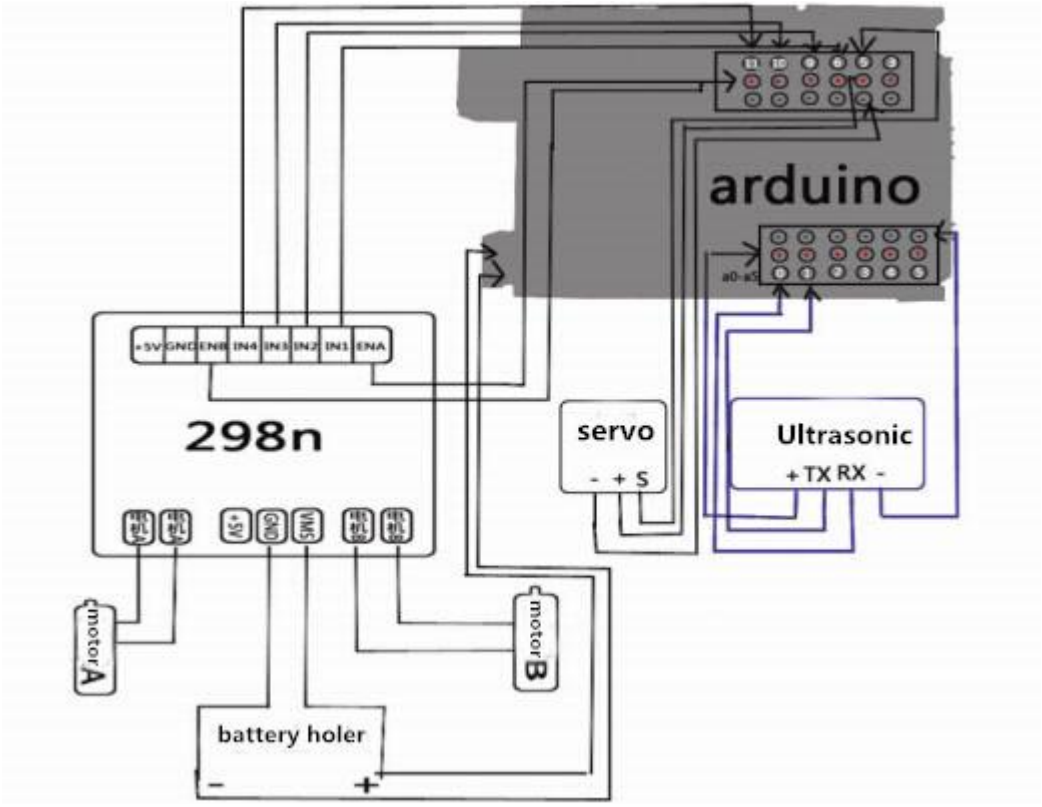



Now, we have test results, we can start to use...

3. Ultrasonic obstacle avoidance

Ultrasonic obstacle avoidance is very easy and convenient to control and its precision is up to our standard, which make it a frequently used way for obstacle avoidance.

Connection diagram of ultrasonic:

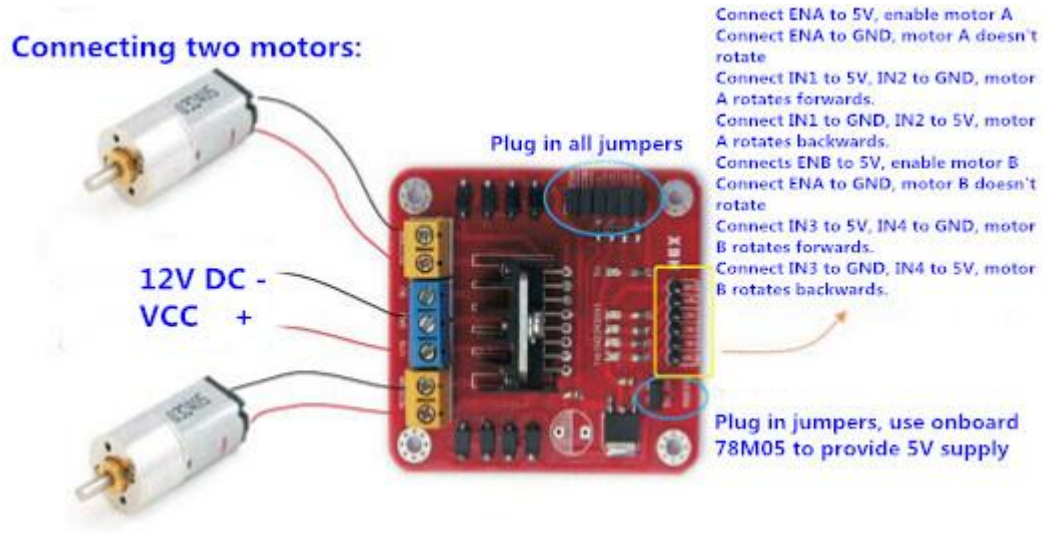



1. Wiring of motor

Connect Motor 1 to MOTOA of L298N
Connect Motor 2 to MOTOB of L298N

2. Power supply of L298N

Power up L298N motor with one road power supply from 6 cell AA battery holder, the other road can be used for arduino main board. Connect '+'of the power supply used for powering L298N motor drive module with VMS port of L298N, and '-' of the power to GND port of L298N, the +5V port on L298N board remains unconnected.




3. Enable motor and turning function(matched with programs)

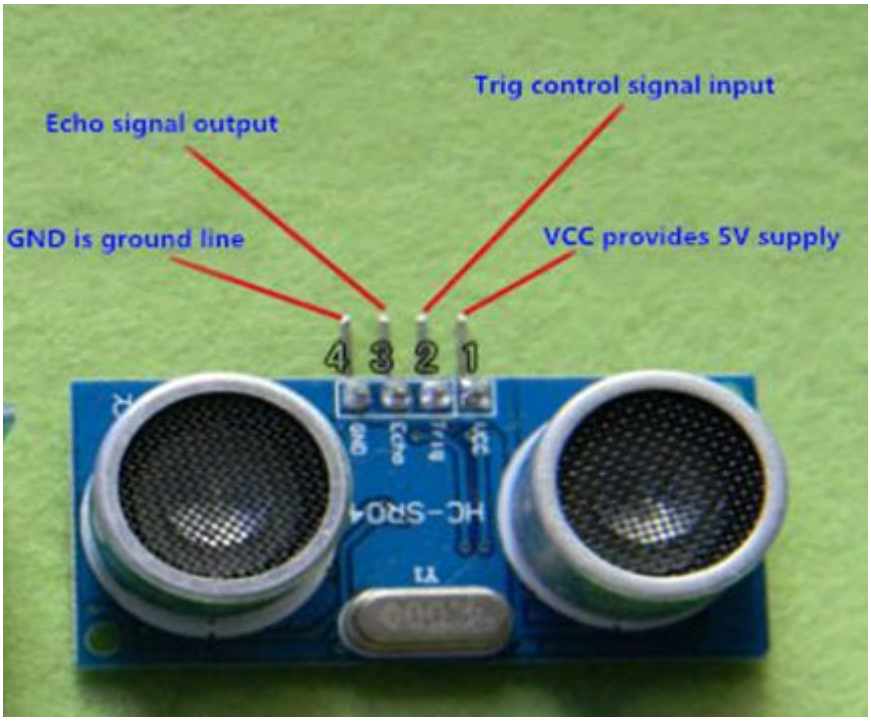
int pinLB=6; // define 6 pin left backward, connect with PWM 6 pin
int pinLF=9; // define 9 pin left backward， connect with PWM 9 pin
int pinRB=10; // define 10 pin left backward， connect with PWM 10 pin
int pinRF=11; // define 11 pin left backward， connect with PWM 11 pin

4. Wiring of servo

myservo.attach(5); // define servo motor as output the fifth pin(PWM)

5. Wiring of ultrasonic sensor

Ultrasonic sensor has four pins
VCC should be connected with +5V
TRIG signal input
Echo signal output
GND should be connected with ground line




Program for smart ultrasonic obstacle avoidance car(ARDUINO)

L=left
R=right
F=forward
B=backward

*/

```
#include <Servo.h>
int pinLB=6; // define 6 pin left backward
int pinLF=9; // define 9 pin left forward
int pinRB=10; // define 10 pin left right backward
int pinRF=11; // define 11 pin left right forward
int inputPin = A0; // define ultrasonic signal receiving pin
int outputPin =A1; // define ultrasonic signal transmission pin
int Fspeedd = 0; // forward speed
int Rspeedd = 0; // turning right speed
int Lspeedd = 0; // turning left speed
int directionn = 0; // forward=8 backward=2 left=4 right=6
Servo myservo; // set myservo
int delay_time = 250; // stabilizing time of servo motor after turning
int Fgo = 8; // forward
int Rgo = 6; // turn right
```

```
int Lgo = 4; // turn left
int Bgo = 2; // backward
void setup()
{
Serial.begin(9600); // define motor output pin
pinMode(pinLB,OUTPUT); // pin 8 (PWM)
pinMode(pinLF,OUTPUT); // pin 9 (PWM)
pinMode(pinRB,OUTPUT); // pin 10 (PWM)
pinMode(pinRF,OUTPUT); // pin 11 (PWM)
pinMode(inputPin, INPUT); // define ultrasonic input pin
pinMode(outputPin, OUTPUT); // define ultrasonic output pin
myservo.attach(5); // define the fifth pin of servo motor(PWM)
}
void advance(int a) // forward
{
digitalWrite(pinRB,LOW); // let motor act(right back)
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,LOW); // let motor act(left back)
digitalWrite(pinLF,HIGH);
delay(a * 100);
}
void right(int b) //turn right(single wheel)
{
digitalWrite(pinRB,LOW); //let motor act(right back)
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,HIGH);
delay(b * 100);
}
void left(int c) //turn left(single wheel)
{
digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,LOW); //let motor act(left back)
digitalWrite(pinLF,HIGH);
delay(c * 100);
}
void turnR(int d) //turn right(double wheel)
{
digitalWrite(pinRB,LOW); //let motor act(right back)
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,LOW); //let motor act(left front)
delay(d * 100);
```

```
}
void turnL(int e) //turn left(double wheel)
{
digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,LOW); //let motor act(right front)
digitalWrite(pinLB,LOW); //let motor act(left back)
digitalWrite(pinLF,HIGH);
delay(e * 100);
}
void stopp(int f) //stop
{
digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,HIGH);
delay(f * 100);
}
void back(int g) //backward
{
digitalWrite(pinRB,HIGH); //let motor act(right back)
digitalWrite(pinRF,LOW);
digitalWrite(pinLB,HIGH); //let motor act(left back)
digitalWrite(pinLF,LOW);
delay(g * 100);
}
void detection() //measure three angles(0.90.179)
{
int delay_time = 250; // stabilizing time of servo motor after turning
ask_pin_F(); // read distance ahead
if(Fspeedd < 10) // if distance ahead is less than 10cm
{
stopp(1); // Clear output data
back(2); // backward for 0.2 sec
}
if(Fspeedd < 25) // if distance ahead is less than 25cm
{
stopp(1); // Clear output data
ask_pin_L(); // read distance on the left side
delay(delay_time); // wait for the servo to stabilize
ask_pin_R(); // ead distance on the right side
delay(delay_time); // wait for the servo to stabilize
if(Lspeedd > Rspeedd) //if the distance on the left side is larger than that of the right side
{
directionn = Rgo; //go rightwards
```

```
}
if(Lspeedd <= Rspeedd) //if the distance on the left side is no more than that of the right side
{
directionn = Lgo; //go leftwards
}
if (Lspeedd < 10 && Rspeedd < 10) //if the distance on both sides is less than 10cm
{
directionn = Bgo; //go backwards

}
}
else //if the distance ahead if no less than 25cm
{
directionn = Fgo; //go forward
}
}
void ask_pin_F() // measure distance ahead
{
myservo.write(90);
digitalWrite(outputPin, LOW); // let ultrasonic transmit low voltage 2μ s
delayMicroseconds(2);
digitalWrite(outputPin, HIGH); // let ultrasonic transmit hight voltage 10μ s, at least 10μ s here
delayMicroseconds(10);
digitalWrite(outputPin, LOW); // keep ultrasonic transmitting low voltage
float Fdistance = pulseIn(inputPin, HIGH); // read time gap
Fdistance= Fdistance/5.8/10; // convert time into distance(unit:cm)
Serial.print("F distance:"); //output distance(unit:cm)
Serial.println(Fdistance); //display distance
Fspeedd = Fdistance; // read distance into Fspeedd(forward speed)
}
void ask_pin_L() // measure distance on the left side
{
myservo.write(5);
delay(delay_time);
digitalWrite(outputPin, LOW); // let ultrasonic transmit low voltage 2μ s
delayMicroseconds(2);
digitalWrite(outputPin, HIGH); // let ultrasonic transmit hight voltage 10μ s, at least 10μ s here
delayMicroseconds(10);
digitalWrite(outputPin, LOW); // keep ultrasonic transmitting low voltage
float Ldistance = pulseIn(inputPin, HIGH); // read time gap
Ldistance= Ldistance/5.8/10; // convert time into distance(unit:cm)
Serial.print("L distance:"); //output distance(unit:cm)
Serial.println(Ldistance); //display distance
Lspeedd = Ldistance; // read distance into Lspeedd(Leftward speed)
```

```
}
void ask_pin_R() // measure distance on the right side
{
myservo.write(177);
delay(delay_time);
digitalWrite(outputPin, LOW); // let ultrasonic transmit low voltage 2μ s
delayMicroseconds(2);
digitalWrite(outputPin, HIGH); // let ultrasonic transmit high voltage 10μ s, at least 10μ s here
delayMicroseconds(10);
digitalWrite(outputPin, LOW); // keep ultrasonic transmitting low voltage
float Rdistance = pulseIn(inputPin, HIGH); // read time gap
Rdistance= Rdistance/5.8/10; // convert time into distance(unit:cm)
Serial.print("R distance:"); //output distance(unit:cm)
Serial.println(Rdistance); //display distance
Rspeedd = Rdistance; // read distance into Rspeedd(rightward speed)
}
void loop()
{
myservo.write(90); //let servo motor return to its ready position, prepared for the next measurement
detection(); //measure angle and decide moving direction
if(directionn == 2) //if directionn(direction) = 2(backward)
{
back(8); // backward(car)
turnL(2); //slightly move leftwards(prevent from being stuck in blind alley)
Serial.print(" Reverse "); //display direction(backward)
}
if(directionn == 6) //if directionn(direction) = 6(rightward)
{
back(1);
turnR(6); // turn right
Serial.print(" Right "); //display direction(turn left)
}
if(directionn == 4) //if directionn(direction) = 4(turn left)
{
back(1);
turnL(6); // turn left
Serial.print(" Left "); //display direction(turn right)
}
if(directionn == 8) /if directionn(direction) = 8(forward)
{
advance(1); // go forward as normal
Serial.print(" Advance "); //display direction(forward)
Serial.print(" ");
```

```
}
}
```