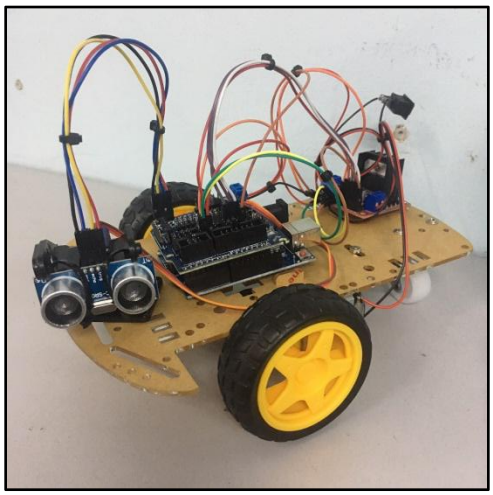
SYNACORP TRADING & SERVICES
No.9, 1st Floor, Lorong 1/SS2, Bandar Tasek Mutiara, 14120 Simpang Ampat, S.Prai (S), Penang
Tel : +604.502.1726 Hunting Line : 012.403.3474 Fax : +604.502.1726
(Website) http://www.synacorp.my (Email) sales@synacorp.com.my

Arduino Uno R3 2WD Smart Robotics Robot Car Chassis Kit Set

Introduction:



This Arduino Uno R3 2WD Smart Robotics Robot Car Chassis Kit Set consists of set of chasis body, DC gear motor, Arduino Uno R3, Ultrasonic sensor, motor driver, servo motor and other components. This robot car is used for detecting obstacles and avoiding the collision.

Its ability to operate autonomously is based on the bot's ultrasonic sensor. The ultrasonic sensor emits high-frequency sound waves (these waves can't be detected by human ears because they are too high), and waits for those sound waves to reflect off of an object, and calculates how long it takes for the sound to return to the sensor. The microcontroller controls the motors left, right, back, front, based on ultrasonic signals. In order to control the speed of each motor pulse width modulation is used (PWM). This set includes servo motor which helps the robot car find another way by turning 180° when the ultrasonic sensor detects an obstacle in front of it.

Objective:

The robot will detects an obstacle in front of it and find the best way to move.

SYNACORP TRADING & SERVICES
No.9, 1st Floor, Lorong 1/SS2, Bandar Tasek Mutiara, 14120 Simpang Ampat, S.Prai (S), Penang
Tel : +604.502.1726 Hunting Line : 012.403.3474 Fax : +604.502.1726
(Website) http://www.synacorp.my (Email) sales@synacorp.com.my

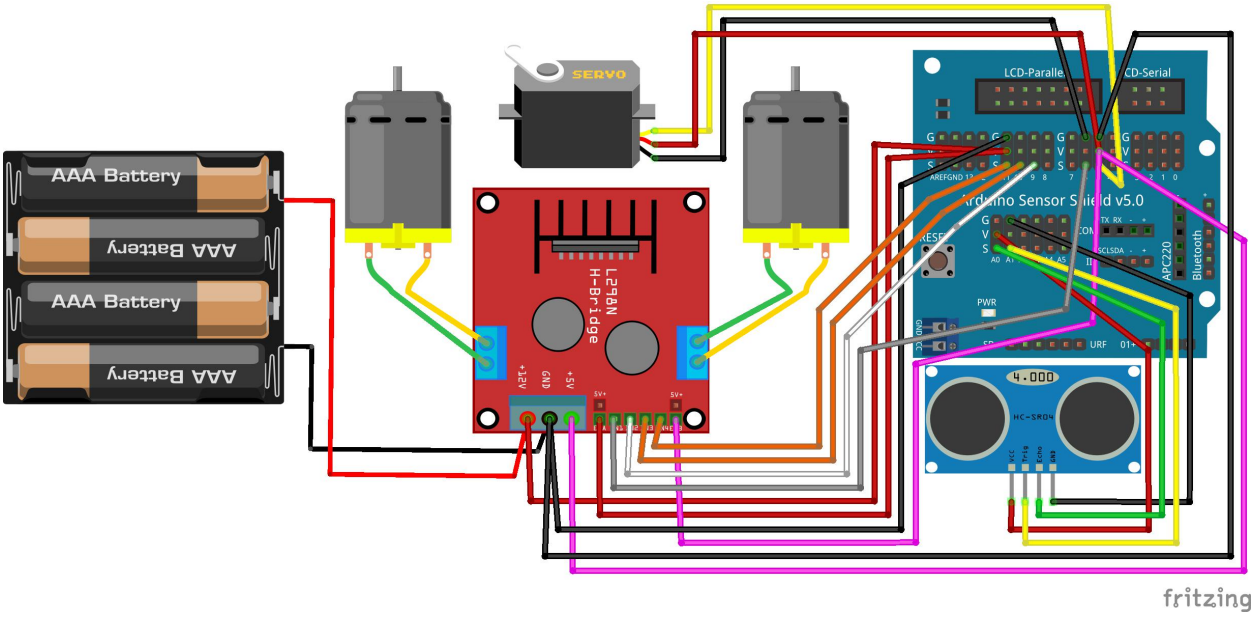
Components Needed:

- Car chassis x1
- Car Wheels x2
- DC Gear Motor x2
- 20 line gun code disk x2
- Fasteners (high intensity black acrylic) x4
- Caster x1
- Four battery box x1
- Quality Rocker Switch x1
- Assembly drawing x1
- Several screw nut
- V5 shield x1
- UNO R3 x1
- SG90 x1
- FPV x1
- L298N x1

Procedures:

Step 1- Connect the wiring of the components as shown in diagram below.

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



Step 2 – Connect the cables to power supply (i.e. Battery Box), UNO board, ultrasonic sensor and Servo. The following table shows the connection map.

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

UNO board Sensor Shield	**L289N**	**Battery Box**	**Motor left**	**Motor Right**	**Servo**	**Ultrasonic sensor**
	GND	GND				
	VMS	VMS				
	+(Left)		+(red)			
	-(Left)		-(black)			
	+(Right)			+(red)		
	-(Right)			-(black)		
V(pin11)	ENA	VMS				
S(pin6)	IN1					
S(pin9)	IN2					
S(pin10)	IN3					
S(pin11)	IN4					
V(pin5)	ENB					
G(pin5)	GND					
V(pin5)	5V+					
S(pin5)					S	
V(pin5)					+	
G(pin6)					-	
V(pinA0)						+
S(pinA1)						Trig
S(pinA0)						Echo
G(pinA1)						-

Step 3 – Connect USB of the Arduino UNO to the laptop. Turn the power on and upload the following code.

```
#include <Servo.h>
int pinLB=6; // define pin6 as left back connect with IN1
int pinLF=9; // define pin9 as left forward connect with IN2
int pinRB=10; // define pin10 as right back connect with IN3
int pinRF=11; // define pin11 as right back connect with IN4
int inputPin = A0; // define ultrasonic receive pin (Echo)
int outputPin =A1; // define ultrasonic send pin(Trig)
int Fspeedd = 0; // forward distance
int Rspeedd = 0; // right distance
int Lspeedd = 0; // left distance
int directionn = 0; //
```

```
Servo myservo; // new myservo
int delay_time = 250; // set stable time
int Fgo = 8;
int Rgo = 6;
int Lgo = 4;
int Bgo = 2;
// forward
// turn right
// turn left
// back

void setup()
{
Serial.begin(9600);
pinMode(pinLB,OUTPUT);
pinMode(pinLF,OUTPUT);
pinMode(pinRB,OUTPUT);
pinMode(pinRF,OUTPUT);
pinMode(inputPin, INPUT);
pinMode(outputPin, OUTPUT);
myservo.attach(5); // define the servo pin(PWM)
}
void advance(int a) // forward
{
digitalWrite(pinRB,LOW);
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,LOW);
digitalWrite(pinLF,HIGH);
delay(a * 40);
}
void turnR(int d) //turn right
{
digitalWrite(pinRB,LOW);
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,LOW);
delay(d * 100);
}
void turnL(int e) //turn left
{

digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,LOW);
digitalWrite(pinLB,LOW);
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) //back
{
digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,LOW);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,LOW);
delay(g * 100);
}
void detection() //test the distance of different direction
{
int delay_time = 250; //
ask_pin_F(); // read forward distance
if(Fspeedd < 10) // if distance less then 10
{
stopp(1);
back(2);
}
if(Fspeedd < 10) // if distance less then 10
{
stopp(1);
ask_pin_L();
delay(delay_time);
ask_pin_R();
delay(delay_time);
ask_pin_F();
if(Lspeedd > Rspeedd) //if left distance more than right distance
{
directionn = Rgo;
}
if(Lspeedd <= Rspeedd)//if left distance not more than right
//distance
{
directionn = Lgo;
}
//if left if (Lspeedd < 10 && Rspeedd < 10) distance and right
//distance both less than 10
{
directionn = Bgo;
}
}
else
{
directionn = Fgo; // forward go
}
}
void ask_pin_F() // test forward distance
{
myservo.write(90);
digitalWrite(outputPin, LOW);
delayMicroseconds(2);
digitalWrite(outputPin, HIGH);
```

```
delayMicroseconds(10);
digitalWrite(outputPin, LOW);
float Fdistance = pulseIn(inputPin, HIGH);
Fdistance= Fdistance/5.8/10;
Serial.print("F distance:");
Serial.println(Fdistance);
Fspeedd = Fdistance;
}
void ask_pin_L() // test left distance
{
myservo.write(5);
delay(delay_time);
digitalWrite(outputPin, LOW);
delayMicroseconds(2);
digitalWrite(outputPin, HIGH);
delayMicroseconds(10);
digitalWrite(outputPin, LOW);
float Ldistance = pulseIn(inputPin, HIGH);
Ldistance= Ldistance/5.8/10;
Serial.print("L distance:");
Serial.println(Ldistance);
Lspeedd = Ldistance;
}
void ask_pin_R() // test right distance
{
myservo.write(177);
delay(delay_time);
digitalWrite(outputPin, LOW);
delayMicroseconds(2);
digitalWrite(outputPin, HIGH);
delayMicroseconds(10);
digitalWrite(outputPin, LOW);
float Rdistance = pulseIn(inputPin, HIGH);
Rdistance= Rdistance/5.8/10;
Serial.print("R distance:");
Serial.println(Rdistance);
Rspeedd = Rdistance;
}
void loop()
{
myservo.write(90);
detection();
if(directionn == 2)
{
back(8);
turnL(2);
Serial.print(" Reverse ");
}
if(directionn == 6)
{
back(1);
turnR(6);
Serial.print(" Right ");
```

```
}
if(directionn == 4)
{
back(1);
turnL(6);
Serial.print(" Left ");
}
if(directionn == 8)
{
advance(1);
Serial.print(" Advance ");
Serial.print(" ");
}
}
```

Step 4 –Collect data and see the result at Serial Monitor tab in the Arduino IDE.