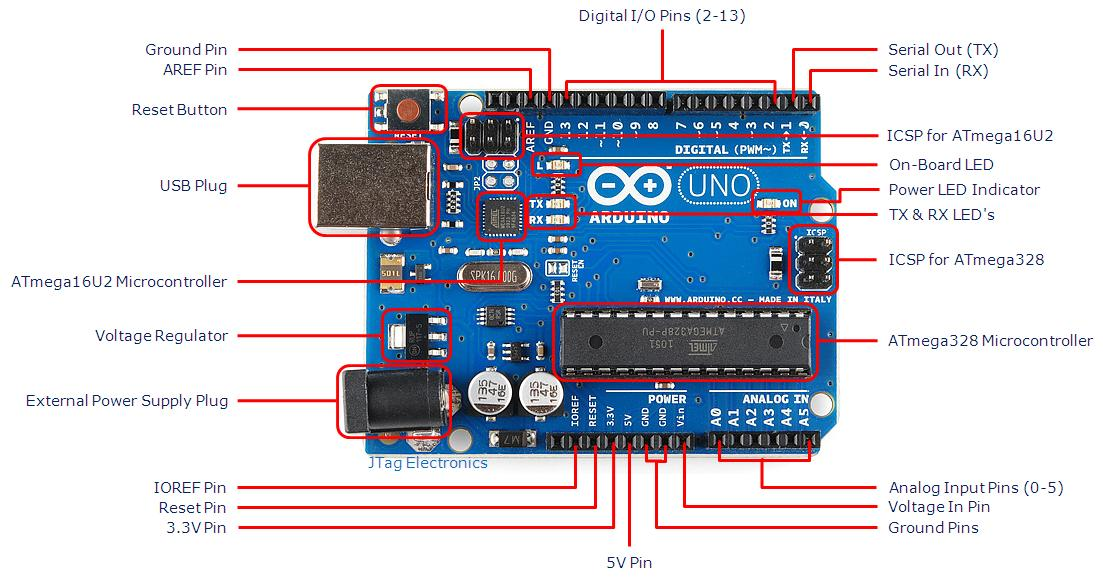
**Obstacle detection Robot using Ultrasonic Sensor and Arduino UNO**

**Arduino UNO:**

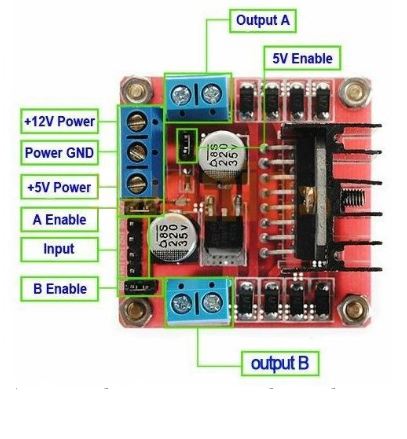
Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

**

**Ultrasonic Sensor(HC-SR04 )**

Ultrasonic range sensor( HC - SR04) provides 2cm - 400cm distance measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit.



**Motor Driver Module (L298D)**

The L298N H-bridge module can be used with motors that have a voltage of between 5 and 35V DC. With the module used in this tutorial, there is also an onboard 5V regulator, so if your supply voltage is up to 12V you can also source 5V from the board.

1. DC motor 1 “+” or stepper motor A+
2. DC motor 1 “-” or stepper motor A-
3. 12V jumper – remove this if using a supply voltage greater than 12V DC. This enables power to the onboard 5V regulator
4. Connect your motor supply voltage here, maximum of 35V DC. Remove 12V jumper if >12V DC
5. GND
6. 5V output if 12V jumper in place, ideal for powering your Arduino (etc)
7. DC motor 1 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
8. IN1
9. IN2
10. IN3
11. IN4
12. DC motor 2 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
13. DC motor 2 “+” or stepper motor B+
14. DC motor 2 “-” or stepper motor B-

**Controlling DC Motors**

To control one or two DC motors is quite easy with the L298N H-bridge module. First connect each motor to the A and B connections on the L298N module. If you’re using two motors for a robot (etc) ensure that the polarity of the motors is the same on both inputs. Otherwise you may need to swap them over when you set both motors to forward and one goes backwards!

Next, connect your power supply – the positive to pin 4 on the module and negative/GND to pin 5. If you supply is up to 12V you can leave in the 12V jumper (point 3 in the image above) and 5V will be available from pin 6 on the module. This can be fed to your Arduino’s 5V pin to power it from the motors’ power supply. Don’t forget to connect Arduino GND to pin 5 on the module as well to complete the circuit.

Now you will need six digital output pins on your Arduino, two of which need to be PWM (pulse-width modulation) pins. PWM pins are denoted by the tilde (“~”) next to the pin number, for example:

****

Finally, connect the Arduino digital output pins to the driver module. In our example we have two DC motors, so digital pins D9, D8, D7 and D6 will be connected to pins IN1, IN2, IN3 and IN4 respectively. Then connect D10 to module pin 7 (remove the jumper first) and D5 to module pin 12 (again, remove the jumper).

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.

However the motors will not turn until a HIGH is set to the enable pin (7 for motor one, 12 for motor two). And they can be turned off with a LOW to the same pin(s). However if you need to control the speed of the motors, the PWM signal from the digital pin connected to the enable pin can take care of it.

**Project Description:**

Here we are going to implement a bot which is going to detect obstacle and according to that, it going to change its direction.

For this we require

• Arduino UNO

• Ultrasonic Sensor

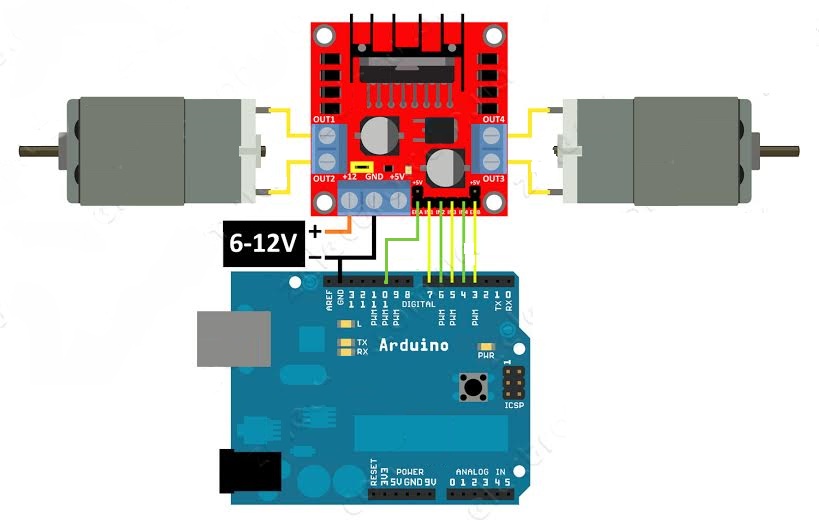
• L298d

• Dc gear Motor

Now first we are going to connect all ultrasonic sensor with our Arduino board. Then according to program our all ultrasonic sensor going generate a trigger signal which is going to receive by our echo pin of Ultrasonic Sensor.

Then we are going to run an algorithm according which we are going to manipulate our desire distance for obstacle detection then we are going to control our motor rotation direction for movement of our bot.

50CM(Front),15CM (Each Side).



**CODE**

int Echo1 = A4; //LEFT\_SENSOR ECHO

int Trig1 = A5; //LEFT\_SENSOR TRIG

int Echo2 = A2; //MID\_SENSOR ECHO

int Trig2 = A3; //MID\_SENSOR TRIG

int Echo3 = A0; //RIGHT\_SENSOR ECHO

int Trig3 = A1; //RIGHT\_SENSOR TRIG

int in1 = 8;

int in2 = 9;

int in3 = 6;

int in4 = 7;

int ENA = 5;

int ENB = 11;

int ABS = 200;

intLeft\_Distance = 0,Right\_Distance = 0,Middle\_Distance = 0 ;

void \_mForward()

{

digitalWrite(in1,LOW);

digitalWrite(in2,HIGH);

digitalWrite(in3,LOW);

digitalWrite(in4,HIGH);

analogWrite(ENA,ABS);

analogWrite(ENB,ABS);

Serial.println("ROBOT\_MOVING\_FORWARD");

}

void \_mBack()

{

digitalWrite(in1,HIGH);

digitalWrite(in2,LOW);

digitalWrite(in3,HIGH);

digitalWrite(in4,LOW);

analogWrite(ENA,ABS);

analogWrite(ENB,ABS);

Serial.println("ROBOT\_MOVING\_BACKWARD");

}

void \_mleft()

{

digitalWrite(in1,LOW);

digitalWrite(in2,HIGH);

digitalWrite(in3,LOW);

digitalWrite(in4,LOW);

analogWrite(ENA,ABS);

analogWrite(ENB,ABS);

Serial.println("ROBOT\_MOVING\_LEFT");

}

void \_mright()

{

digitalWrite(in1,LOW);

digitalWrite(in2,LOW);

digitalWrite(in3,LOW);

digitalWrite(in4,HIGH);

analogWrite(ENA,ABS);

analogWrite(ENB,ABS);

Serial.println("ROBOT\_MOVING\_RIGHT");

}

void \_mStop()

{

digitalWrite(in1,LOW);

digitalWrite(in2,LOW);

digitalWrite(in3,LOW);

digitalWrite(in4,LOW);

digitalWrite(ENA,LOW);

digitalWrite(ENB,LOW);

Serial.println("ROBOT\_STOP");

}

/\*Ultrasonic distance measurement Sub function\*/

intLeft\_Distance\_test()

{

digitalWrite(Trig1, LOW);

delayMicroseconds(2);

digitalWrite(Trig1, HIGH);

delayMicroseconds(20);

digitalWrite(Trig1, LOW);

floatFdistance = pulseIn(Echo1, HIGH);

delay(10);

Fdistance= Fdistance/ 29 / 2;

return (int)Fdistance;

}

intMiddle\_Distance\_test()

{

digitalWrite(Trig2, LOW);

delayMicroseconds(2);

digitalWrite(Trig2, HIGH);

delayMicroseconds(20);

digitalWrite(Trig2, LOW);

floatFdistance = pulseIn(Echo2, HIGH);

delay(10);

Fdistance= Fdistance/ 29 / 2;

return (int)Fdistance;

}

intRight\_Distance\_test()

{

digitalWrite(Trig3, LOW);

delayMicroseconds(2);

digitalWrite(Trig3, HIGH);

delayMicroseconds(20);

digitalWrite(Trig3, LOW);

floatFdistance = pulseIn(Echo3, HIGH);

delay(10);

Fdistance= Fdistance/ 29 / 2;

return (int)Fdistance;

}

void setup()

{

Serial.begin(9600);

pinMode(Echo1, INPUT);

pinMode(Trig1, OUTPUT);

pinMode(Echo2, INPUT);

pinMode(Trig2, OUTPUT);

pinMode(Echo3, INPUT);

pinMode(Trig3, OUTPUT);

pinMode(in1,OUTPUT);

pinMode(in2,OUTPUT);

pinMode(in3,OUTPUT);

pinMode(in4,OUTPUT);

pinMode(ENA,OUTPUT);

pinMode(ENB,OUTPUT);

\_mStop();

}

void loop()

{

Left\_Distance = Left\_Distance\_test();

delay(10);

Middle\_Distance = Middle\_Distance\_test();

delay(10);

Right\_Distance = Right\_Distance\_test();

delay(10);

Serial.println("Left\_Distance\tMiddle\_Distance\tRight\_Distance\tStatus\n");

Serial.print(Left\_Distance);

Serial.print("cm\t\t");

Serial.print(Middle\_Distance);

Serial.print("cm\t\t");

Serial.print(Right\_Distance);

Serial.print("cm\t\t");

if(Middle\_Distance<=50)

{

if(Right\_Distance>Left\_Distance)

{

if((Right\_Distance<=30)&&(Left\_Distance<=30))

{\_mStop();

delay(100);

\_mBack();

delay(600);

}

else{

\_mright();

delay(300);}

}

else if(Right\_Distance<Left\_Distance)

{

if((Right\_Distance<=30)&&(Left\_Distance<=30))

{\_mStop();

delay(100);

\_mBack();

delay(700);

}

else{

\_mleft();

delay(300);

}

}

}

else if(Right\_Distance<=20)

{

\_mleft();

delay(300);

}

else if(Left\_Distance<=20)

{

\_mright();

delay(300);

}

else

{

\_mForward();

}

}