



My Arduino Ping Display Robot

by [Cello62](#) on February 8, 2012

Table of Contents

My Arduino Ping Display Robot	1
Intro: My Arduino Ping Display Robot	2
Step 1: List of Materials	2
Step 2: SERVO SHIELD CONSTRUCTION	5
Step 3: LCD SHIELD CONSTRUCTION	6
Step 4: TABLE FOR THE CONNECTIONS	7
Step 5: ASSEMBLING THE ROBOT- FIRST FRAME	7
Step 6: ASSEMBLING THE ROBOT- SECOND FRAME	7
Step 7: Done!- CODE AND THEORY	8
Step 8: Video and If you like it please vote!	10
Related Instructables	10
Comments	10



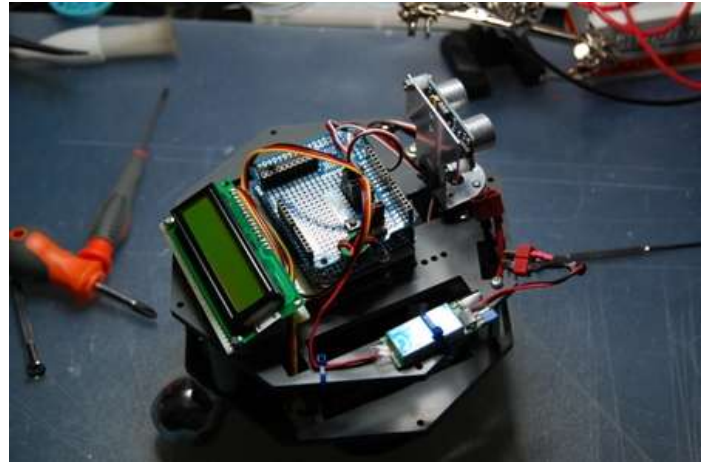
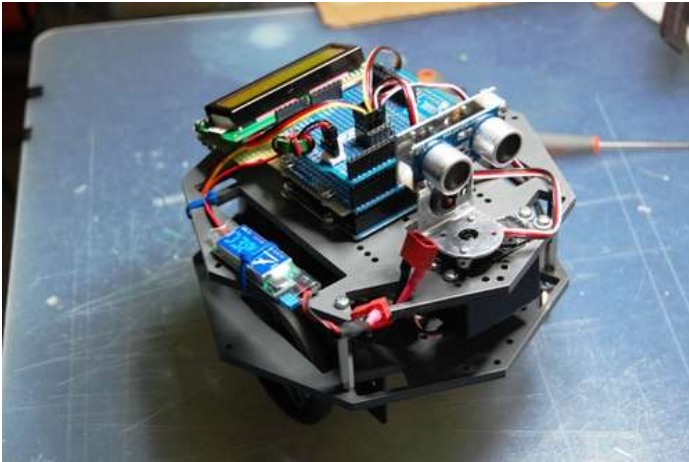
Author: Cello62

There is no joy in possession without sharing

Intro: My Arduino Ping Display Robot

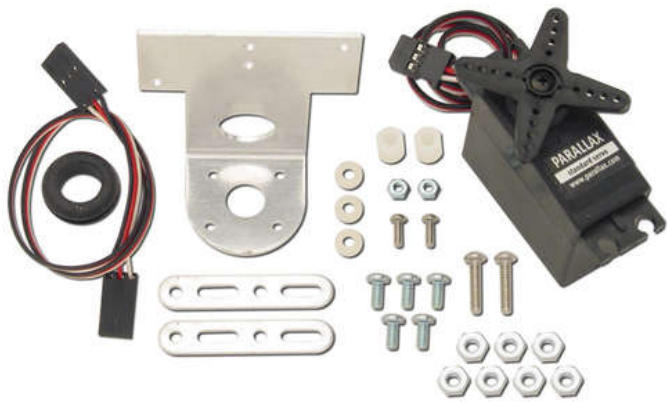
Goals

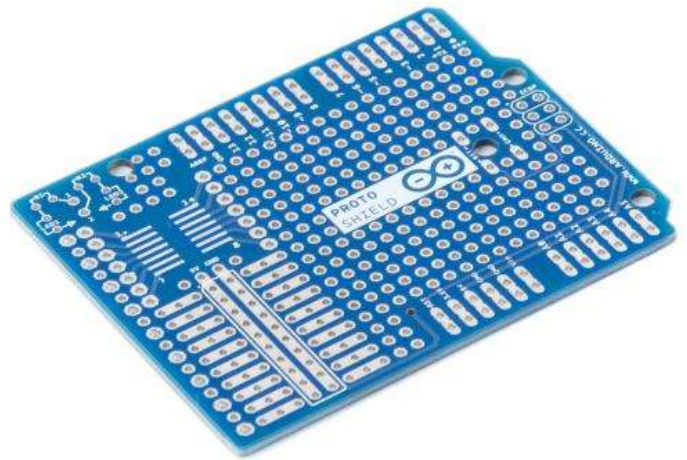
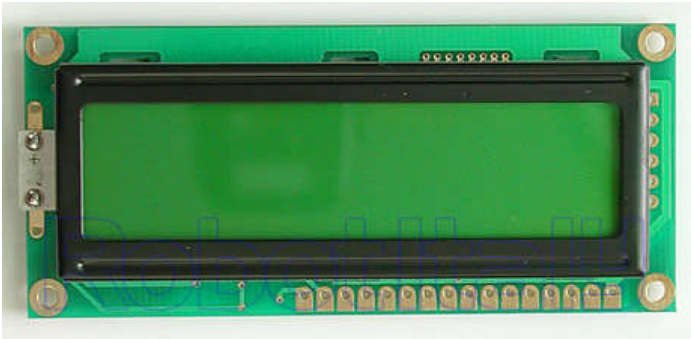
Hello all.
I hope to please share a little robot that I have just finished building.
There are many Ping Boat, perhaps with a tutorials and display less so without pretension, will illustrate what I could do.
I gave myself the goal of realizing a robot that avoids obstacles, hence the use of Ping, who had more autonomy (and less hassle to reload always stylus) and displayed on a LCD display the distance measured obstacle.
The tutorial is directed for beginners but with knowledge of electronics and Arduino programming.



Step 1: List of Materials

- 1 - Arduino UNO or compatible
- 2 - Module OCTAGON GioBlu Robotics http://www.gioblu.com/prodotti?page=shop.product_details&flypage=flypage_images.tpl&product_id=136&category_id=60 or other frame
- 1 - Ping))) Parallax mounting bracket kit <http://www.parallax.com/Store/Robots/AllRobots/tabid/755/ProductID/248/List/0/Default.aspx?SortField=ProductName,ProductName>
- 2 - Servo motors with 360 rotation , in my case HSR-1425SCR
- 2 - 6 cm wheels with rubber no-slip
- 1 - std servo for the Ping (in my case is not required but can be implemented)
- 2 - servo brackets Gioblu Robotics http://www.gioblu.com/prodotti?page=shop.product_details&flypage=flypage_images.tpl&product_id=78&category_id=34
- 4 - PCB spacers from 4 cm http://www.robot-italy.com/product_info.php?cPath=7_135&products_id=1189
- 2 - protoshield for servos, battery and display and added http://www.robot-italy.com/product_info.php?products_id=879 (or other sites)
- 1 - 16x2 HD44780 compatible LCD Display http://www.robot-italy.com/product_info.php?cPath=59_194&products_id=237
- 1 - potentiometer
- 1 - 9v battery holder
- 1 - 2s Lipo 1200mAh 7.4 v
- 1 - UBEC 5.4 / 6V [http://www.cnchelicopter.com/servlet/the-1528/hobbywing-UBEC-dsh-3A-\(2-dsh-6S-Lipo/Detail](http://www.cnchelicopter.com/servlet/the-1528/hobbywing-UBEC-dsh-3A-(2-dsh-6S-Lipo/Detail)
- 1 - pivot wheel
- 1 - piece of matrix board for the display shield
- some strip for shield
- some screw patience and passion..





Step 2: SERVO SHIELD CONSTRUCTION

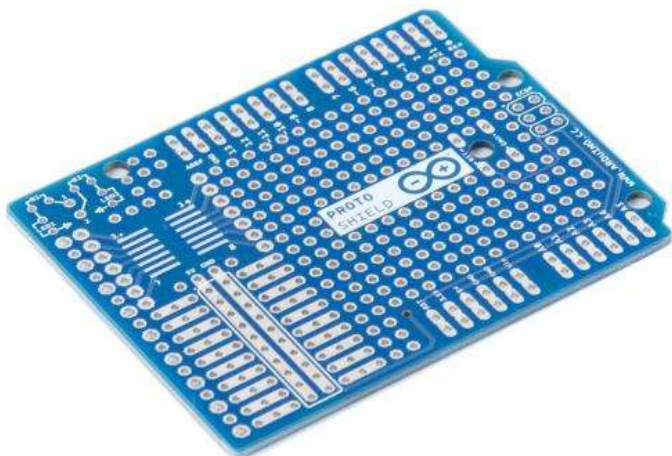
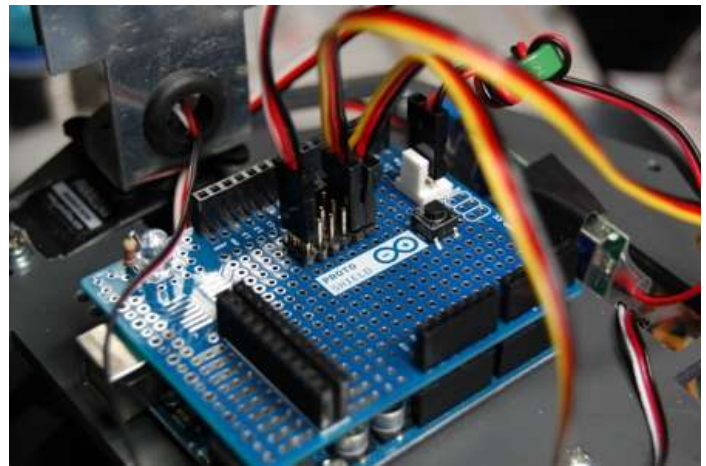
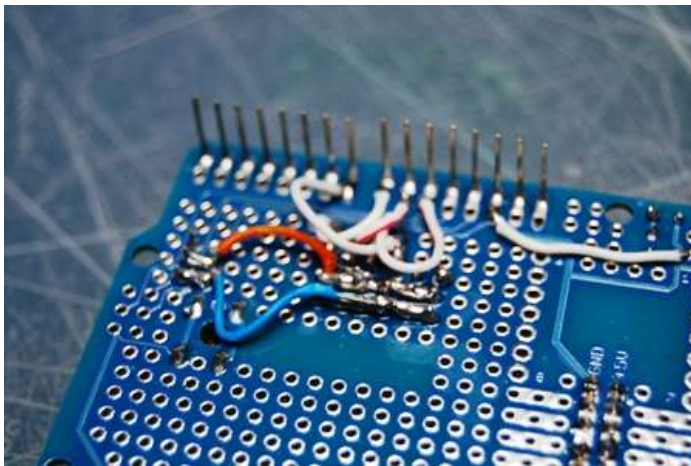
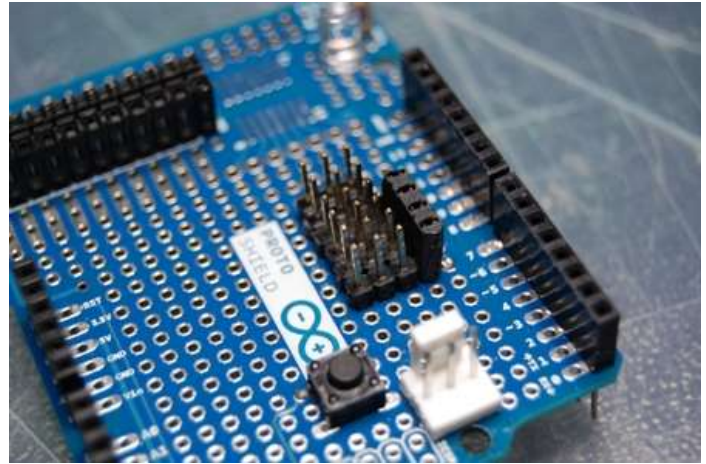
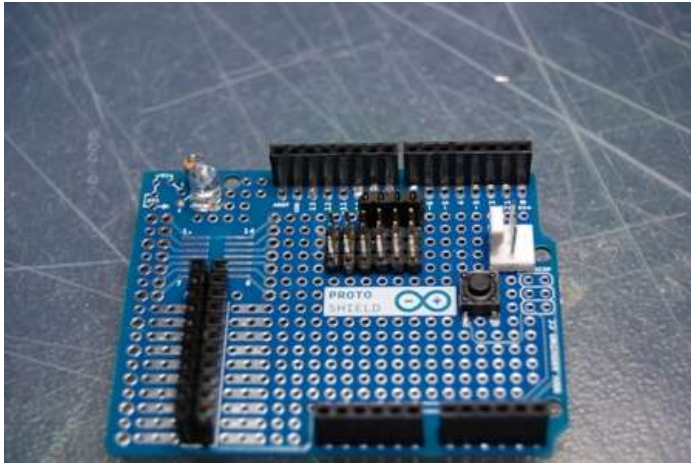
First we begin by shield, why I have chosen this path?. When you need to connect several servo motors, a sensor, batteries etc, of course, there are several PIN signals, power supply and ground to use.

Normally the various little robots are joined by a breadboard to allow you to manage everything at the expense of a lot of threads around that maybe every time you disconnect and still always give a feeling of chaos.

A shield to be attached to Arduino simplifies everything and makes it more convenient connections. As you know for the servos or motors there are already on the market different and functional, eg one of dell'Adafruit or SparkFun, in any case the realization in the house is very simple and inexpensive.

As you can see from the first two photos are soldered several rows of three legs (I have made 6, but you can add several more), each leg corresponds to the poles of the servo and the sensor (signal, 5v, gnd -white/red/black or yellow/red/back). Then sold in parallel (third photo) the 5v (center pin) and GND (pictured blue wire) and the latter connected to the power connector, so all you will connect to the row of pins will be powered together.

For the signal pin, you can decide if you knew at the start (as in my case) corresponds to which pin, solder a wire directly to the corresponding pin, or if you want to manage it from time to time just add (second photo), a row of strip females to connect with a wire to the desired pin. In the above I added a reset button and a LED connected to pin 13 directly to Arduino board.



Step 3: LCD SHIELD CONSTRUCTION

Skipping of course, as you will find dozens of tutorials on how to connect one display to Arduino board eg

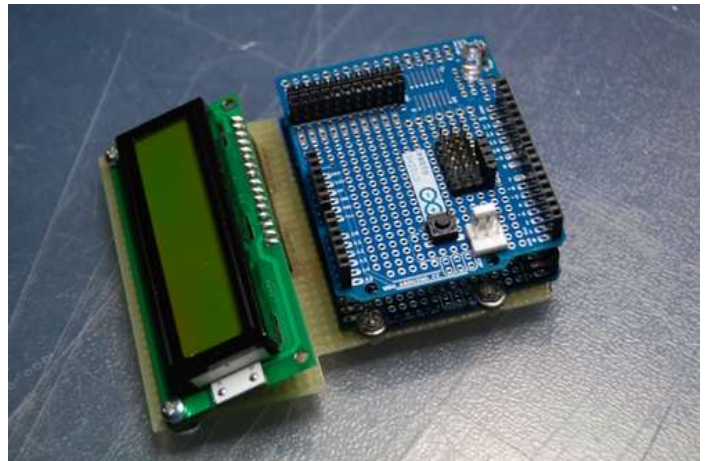
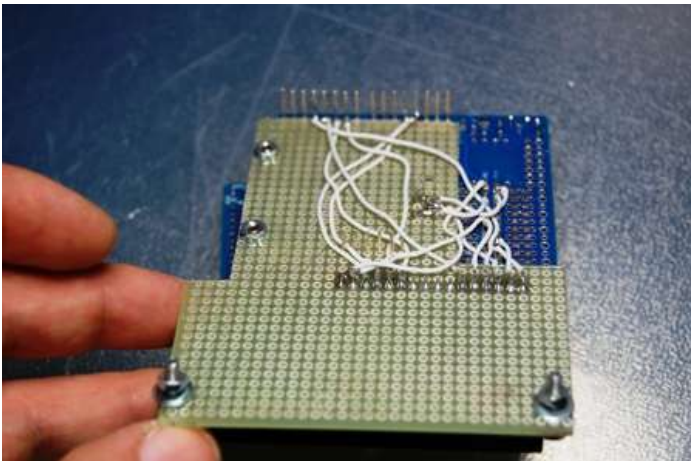
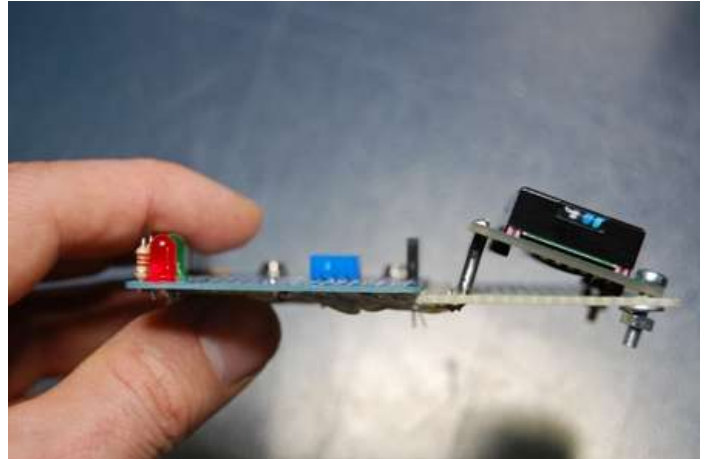
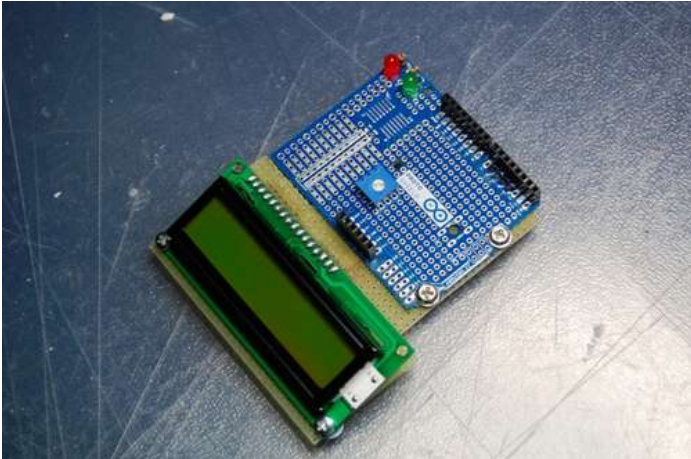
<http://arduino.cc/en/Tutorial/LiquidCrystal>

but I repeat, if you want to take the opportunity to enter into the project, which necessary to reduce the number of wires around that in this case are several, so we can again make a shield that can be useful for other projects with the Arduino.

Of the 16 pins of a display, only 12 are needed for our purpose.

If you look at the first and third photo should crop a breadboard that will serve us well as a support for the display, and like a "bridge" connecting to the Arduino pins. So what should we do, sold (I have soldered slightly inclined, as shown in picture 2, in order to better read the display) a female strip of 16 feet to the matrix board, make from each pin that is used (see the tables below) a wire soldered directly to the corresponding pin of Protoshield, then we're going to overlap to the Arduino board and above this shield (last photo) will get that one made for the servo and sensor **with zero visible wires**.

The last photo is the display in operation.



Step 4: TABLE FOR THE CONNECTIONS

PIN	connect to
1	GND
2	5V
3	(Central Pin of the POT - Add to this one wire to 5v and one to GND)
4 (RS)	3
5 (RW)	GND
6 (EN)	4
11(DB4)	5
12(DB5)	6
13(DB6)	11
14(DB7)	12
15	5V
16	GND

Ping Servo Signal Pin 10
Led to Pin 13
Ping Sensor Signal Pin 7
DX Wheel Pin 9
SX Wheel sx Pin 8

Put the display power under servo shield so it is managed by a battery's capacity.

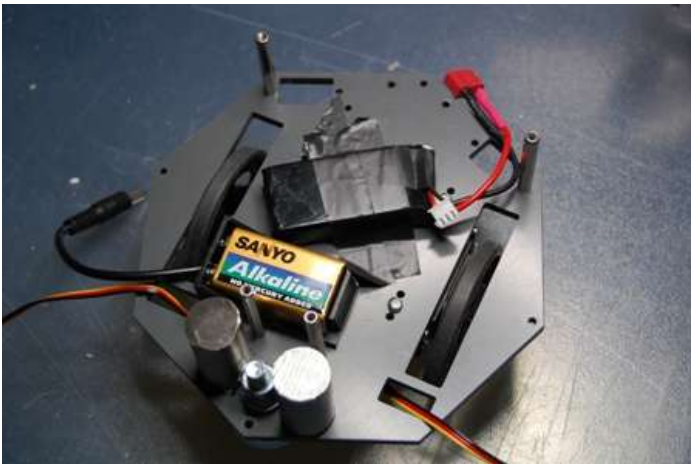
Step 5: ASSEMBLING THE ROBOT- FIRST FRAME

At first frame I have inserted two servos rotated 360, a support for the 9v battery, and I attached, I know ... look like shit, a lipo battery.

I chose this system because, who uses lipo for model knows, I like to take the residual voltage monitored to prevent damage to the battery, with tape so I can remove it easily.

Why two power supplies? Here you can breed different opinions, in any case I think it is better to keep the logic separated (Arduino) to avoid interference from the motors. Where you see the bolt I put a wheel pivoting and two weights because the front was slightly unbalanced.

The third photo is the bottom.



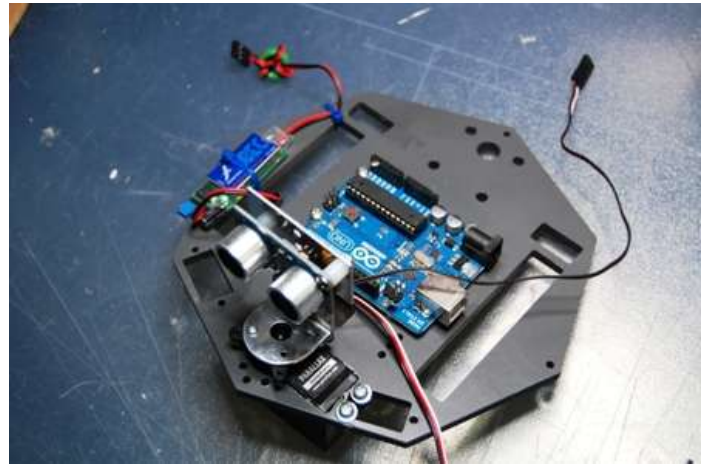
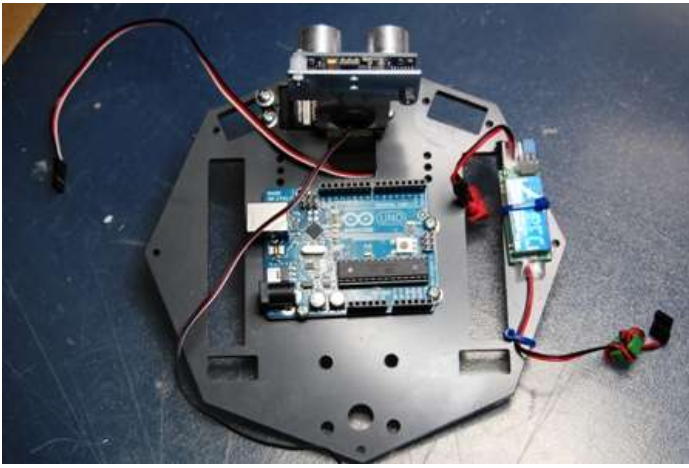
Step 6: ASSEMBLING THE ROBOT- SECOND FRAME

Now prepare the upper frame.

I made a space for the Ping servo, four holes to secure the Arduino and I placed the strap securing it with a UBEC that you see on the left and right of the two. For those not familiar in practice, is used to regulate the output voltage of the battery, because the lipo 2s has a voltage of 7.4 v, while the servo has a conduction voltage range from 4.8 v to 6.

Can also be used diodes or other.

Insert spacers and overlap the first frame to the second.



Step 7: Done!- CODE AND THEORY

Small introduction.

You'll find dozens of variations on how to run the PING sensor, in essence the practical operation of the sensor takes place in the cycle loop () that implements the protocol for the operation of the component PING.

Based on its specific you can control the operation of the sensor through a series of pulses. In summary we can say that you are using a digital pin to measure an analog pin.

Improvement: If you want to make cool, you can also insert a temperature sensor type TMP36 to improve reading, this is because as you know, sound travels at a speed of 343 m / s, this value ment is not absolute because it depends on the air temperature, the bibliography shows this relationship between the speed of sound (C) and the air temperature (t) $C = 331.5 + (0.6 * t)$, knowing the temperature (and here it helps the sensor) we eventually put a few lines of code to implement the reading.

```
#include <Servo.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(3, 4, 5, 6, 11, 12);
Servo PingServo;
Servo LeftServo;
Servo RightServo;
int ultraSoundSignal = 7; // Ultrasound signal pin
int val = 0;
int val2 = 0;
int ultrasoundValue = 0;
int timecount = 0; // Echo counterint
ledPin = 13; // LED connected to digital pin 13
void setup() {
  lcd.begin(16, 2);
  Serial.begin (9600);
  pinMode(8,OUTPUT);
  pinMode(9,OUTPUT);
  LeftServo.attach(8);
  RightServo.attach(9);
  PingServo.attach(10);
  pinMode(ledPin, OUTPUT);
}
void moveServoLeftTo(int angle, int duration){
  // controls the servo to move for a given angle and for a given number of milliseconds
  LeftServo.write(angle);
  for( ; duration > 0; duration -= 20){
  // cycle for the specified number of milliseconds by subtracting 20ms each iteration
```

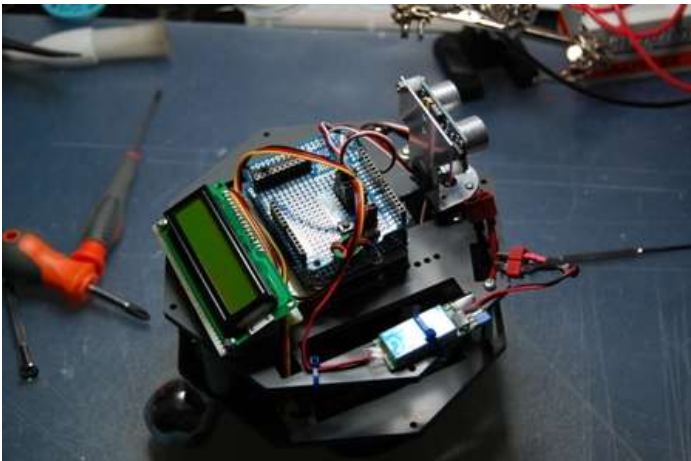
<http://www.instructables.com/id/My-Arduino-Ping-Display-Robot/>


```

delay(20);
}
}
void moveServoRightTo(int angle, int duration){
RightServo.write(angle);
for( ; duration > 0; duration -= 20){
delay(20);
}
}
void loop() {
// Funzioni Ping)))
timecount = 0;
val = 0;
pinMode(ultraSoundSignal, OUTPUT); // Switch signalpin to output
digitalWrite(ultraSoundSignal, LOW); // Send low pulse
delayMicroseconds(2); // Wait for 2 microseconds
digitalWrite(ultraSoundSignal, HIGH); // Send high pulse
delayMicroseconds(5); // Wait for 5 microseconds
digitalWrite(ultraSoundSignal, LOW); // Holdoff
pinMode(ultraSoundSignal, INPUT); // Switch signalpin to input
val = digitalRead(ultraSoundSignal); // Append signal value to val
while(val == LOW) { // Loop until pin reads a high value
val = digitalRead(ultraSoundSignal);
}
while(val == HIGH) { // Loop until pin reads a high value
val = digitalRead(ultraSoundSignal);
timecount = timecount + 1; // Count echo pulse time}
ultrasoundValue = timecount; // Append echo pulse time to ultrasoundValue
lcd.clear();
lcd.setCursor(0,0);
lcd.print("OBSTACLE cm ");
lcd.print(timecount/10);
delay(200);
if(timecount > 0){
digitalWrite(ledPin, HIGH);}

// Servo functions:
if(ultrasoundValue > 500){
moveServoLeftTo(45,50); // servo moves the left of 45 degrees for 500 milliseconds
moveServoRightTo(180,50);
}
if(ultrasoundValue < 500){
moveServoLeftTo(180,40);
moveServoRightTo(180,40);
}
if(ultrasoundValue < 100){
moveServoLeftTo(180,100);
moveServoRightTo(45,100);
}
}
}

```



Step 8: Video and If you like it please vote!

Vote here: <http://www.instructables.com/contest/arduino2012/?show=ENTRIES>

Thanks for watching

Related Instructables



Wall-E's Anti-Social Cousin: Object Avoiding Arduino Controlled Robot! (Photos)
by Brennn10



Arduino 4wd robot with ping sensor "J-Bot"
by JamecoElectronics



How To Make an Obstacle Avoiding Arduino Robot
by generalgeek314



How to Make an Obstacles Avoiding Robot-Arduino Style
by robotkid249



Autonomus Wall Following Obstacle Avoiding Arduino Rescue Bot
by wkuace



Hacking a Rumble Robot (Photos)
by flatlander37

Comments

3 comments

[Add Comment](#)



danilo.bar1 says:

I saw the video.
Is desired that robot tends to curve to the right?

however, very interesting project.

Good Job!

Feb 10, 2012. 3:38 AM [REPLY](#)



Cello62 says:

It is true slightly, depending on the speed of servo rotation, but is not important.
Thanks for your comment

Feb 10, 2012. 7:24 AM [REPLY](#)



danilo.bar1 says:

^_^

Feb 10, 2012. 9:26 AM [REPLY](#)