Project 2: Obstacle Avoidance

Objectives:

1. The student will be able to attach the sensors onto the chassis of the car.
2. The student will be able to wire the motor for power and data transfer.
4. The student will be able to wire the Ultra sonic module for power and data transfer.
5. The student will be able to adjust or reposition the sensor for optimum performance.
6. The student will be able to download code.
7. The student will be able to manipulate the code give to change the sensitivity of the sensor.

Time:

2 hours

Introduction:

In this project, the student will attach sensors which, when activated, will allow the car to avoid running into objects in its path. The students will have to wire several parts for both power and data collection as well as upload the given code to the board. The students will then be asked to modify the code so that the car detects objects that are closer or farther away than the original code allows. This will be done by doing mathematical manipulation using the distance formula.

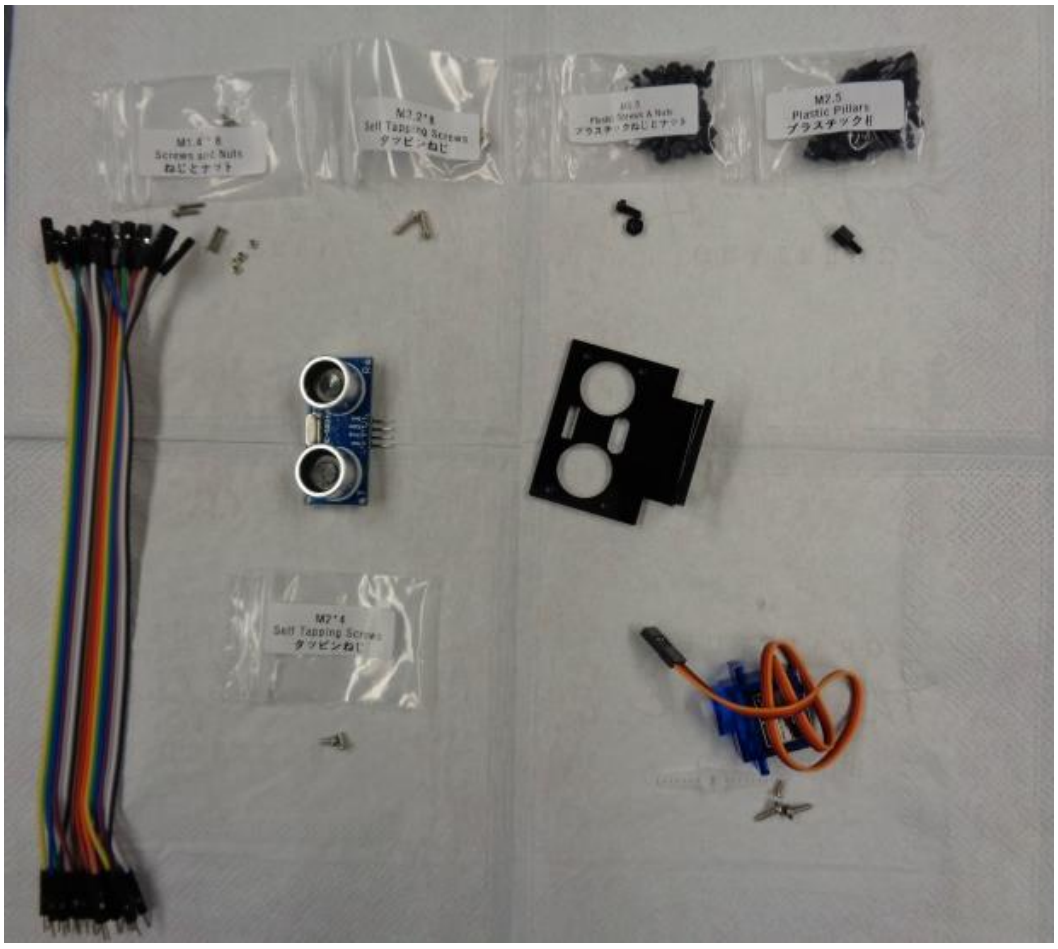
Remove the screws from the copper pillars (stand offs) that connect the upper and lower chassis together so that your car is easier to work with.

Remove the protective film from both sides of the mount.

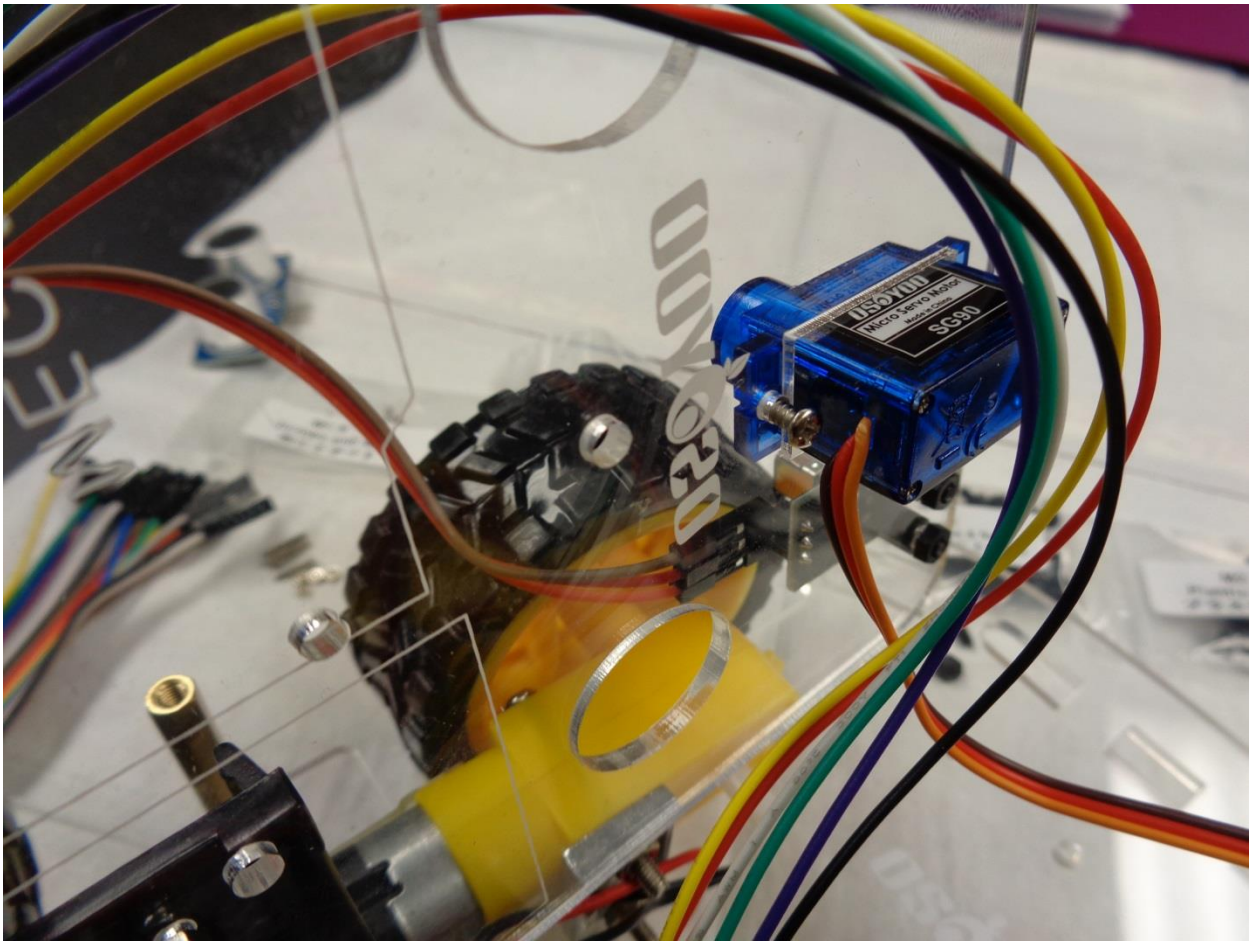
Part 1. Installing Sensors

A. Remove the micro servo motor, mount for the ultrasonic sensor, ultrasonic module, the servomotor arms, wires, and the required bolts and nuts from the box:

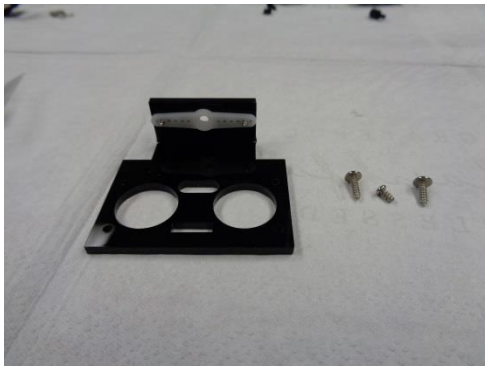
4-M1.4*8 screws	1-1.5*4 screw
4-M1.4 nuts	2.5 plastic screw
2-2.2*8 Self-tapping Screws	2.5 nuts
1-2*4 Self-tapping Screw	2.5 pillar



B. Drop the servo motor into the rectangular hole in the upper chassis with the wires hanging down. Attach the SG90 servo motor to the chassis using the 2.2*8 Self-tapping screw.

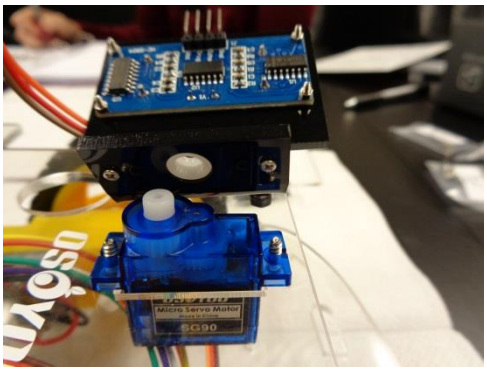


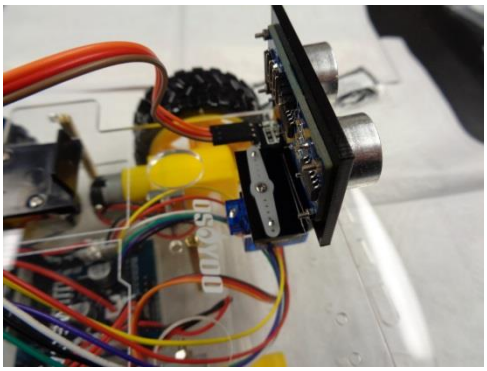
C. Attach the ultrasonic sensor to the mount holder with four M1.4*8 screws and nuts.



D. Place the mount holder on the servo motor horn and attach with an M2*4 self-tapping screw from the servo pack. It should be the smallest one in the pack.

TIP: Rotate the holder on the horn from end stop to end stop to find the approximate center.





Part 2: Wire the motor for power, ground and data transfer

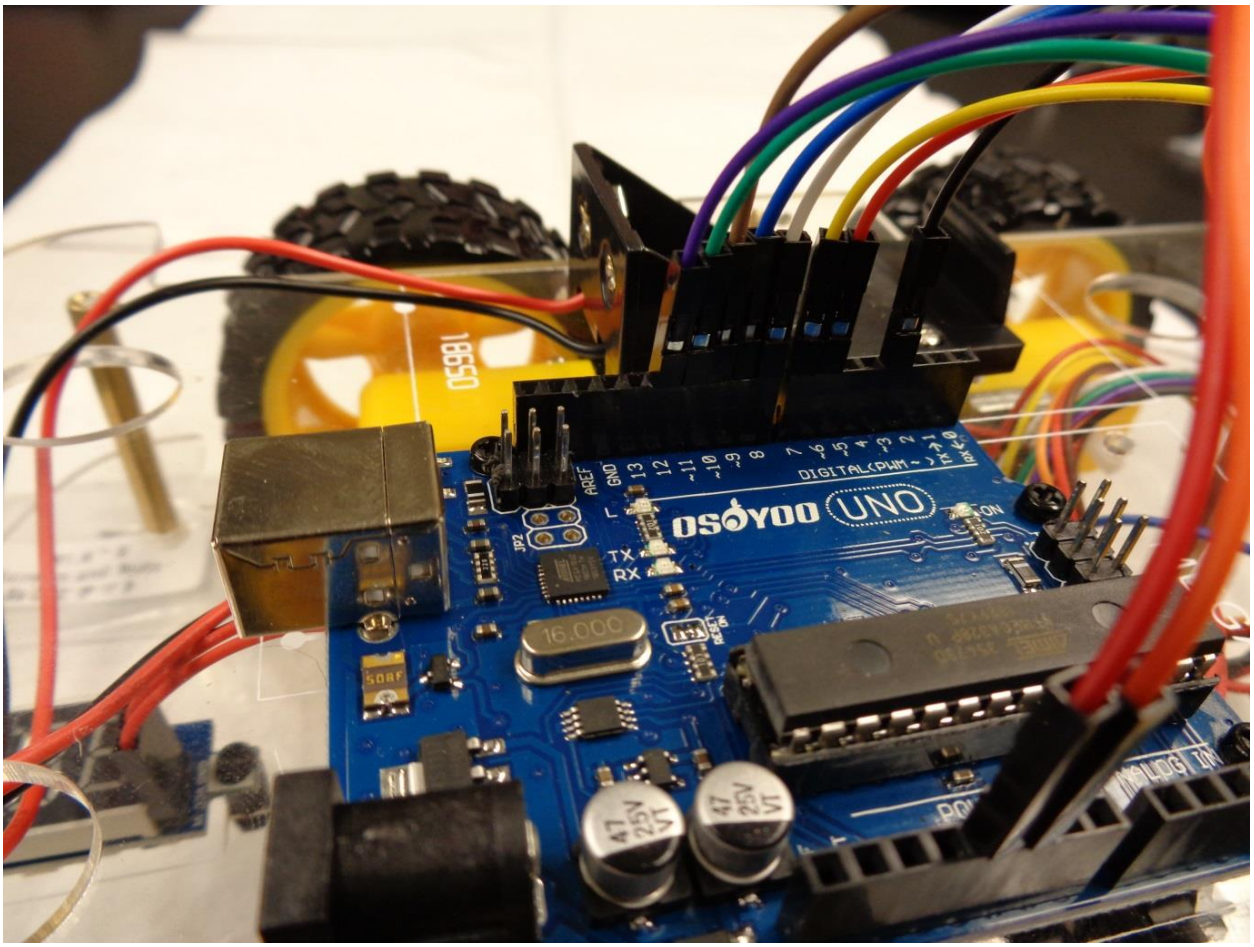
Servo Motor	Wire Color	Model X
GND	Brown	GND
VCC	Red	5V
Signal	Orange	S

A. Connect the brown wire to the GND on the motor driver module. This is the ground wire.
B. Connect the red wire to the 5V on the motor driver module. This is the power.
C. Connect the orange wire to the S on the motor driver module. This is the data transfer wire.

Part 3: Wiring the Ultrasonic module to the Uno board

A. On the Uno board, move the black wire from D9 to D3.
B. Add a male to female wire (green in photo) to D9 and connect it to the Model X Motor Drive on S. Pull this off of the wire bundle.
    a. The rest of the wiring for the Model X motor drive to the Uno board should stay the same:

Model X	Wire Color	Uno Board
ENA	Black	D3
In1	Purple	D12
In2	Green	D11
In3	Yellow	D7
In4	White	D8
ENB	Red	D6
S	Blue	D9



C. You will need four male to female wires to connect the Ultrasonic module to the Uno board. Pull these off of the wire bundle.

Uno Board	Wire Color	Ultrasonic Sensor
GND	Black	GND
D2	Yellow	Echo
D10	Green	Tr1G
5V	Red	VCC

a. Connect the black wire to the Ultra sonic GND and the Uno board GND. (This will be in the section marked digital)
b. Connect the yellow wire to the Ultra sonic “Echo” and the Uno board D2. (This will be in the section marked digital)
c. Connect the green wire to the Ultra sonic “TR1G” and the Uno board D10. (This will be in the section marked digital)
d. Connect the red wire to the Ultra sonic “VCC” and the Uno board 5V. (This will be in the section marked power.

Part 4: Reattach your upper and lower chassis

A. Reconnect your upper and lower chassis using the copper pillars (stand offs).

Part 5: Check the servo motor for proper placement

A. Download the Servo Adjust code for this project from Arduino.nacase.org.
B. Connect your Uno board to your computer using your USB cord.
    a. Verify your code by clicking the check mark in the upper left side of your monitor. You will see "Done Compiling" across the bottom of the screen when complete.
    b. Upload the code by clicking on the arrow next to the checkmark. You will see "Done uploading" when complete.
    c. Turn on your battery box and your Ultra sonic sensor should go to the center. If it doesn't point to center, loosen the center screw on the mount and gently reposition the Ultra sonic sensor.

****Turn off your battery box****

Part 6: Loading the Code

A. Download the code for this project from Arduino.nacase.org.
    a. Click on the file link to begin the download and save to your computer
    b. Once the download is complete, unzip the file or right click—Open With—UnRAR Metro. This will extract the file and save it to your computer.
B. Connect Uno board to PC with USB cord.
C. Upload the code to Arduino IDE
    a. Open Arduino IDE, select File, Open, then select the downloaded code.
D. Verify your code with the checkmark.
E. Upload your code using the arrow next to the check mark. You will see "Done uploading" when complete. Disconnect the USB cord.

Part 7: Try your car

A. Set your car on the ground and turn the battery box on. Your car should move around the room and avoid running into objects in its path. The Ultra sonic sensor will sweep to the left and right before resting back at center. It is sending out a sonar wave that will bounce off of an object close to it and return to the sensor to alert it that something is in the way. This prompts the car to beep, stop, and move to get out of the path of the object.
B. Is your car too sensitive, or not sensitive enough? In Part 9 we will manipulate the code to fix this problem.

Part 8: Manipulate the code

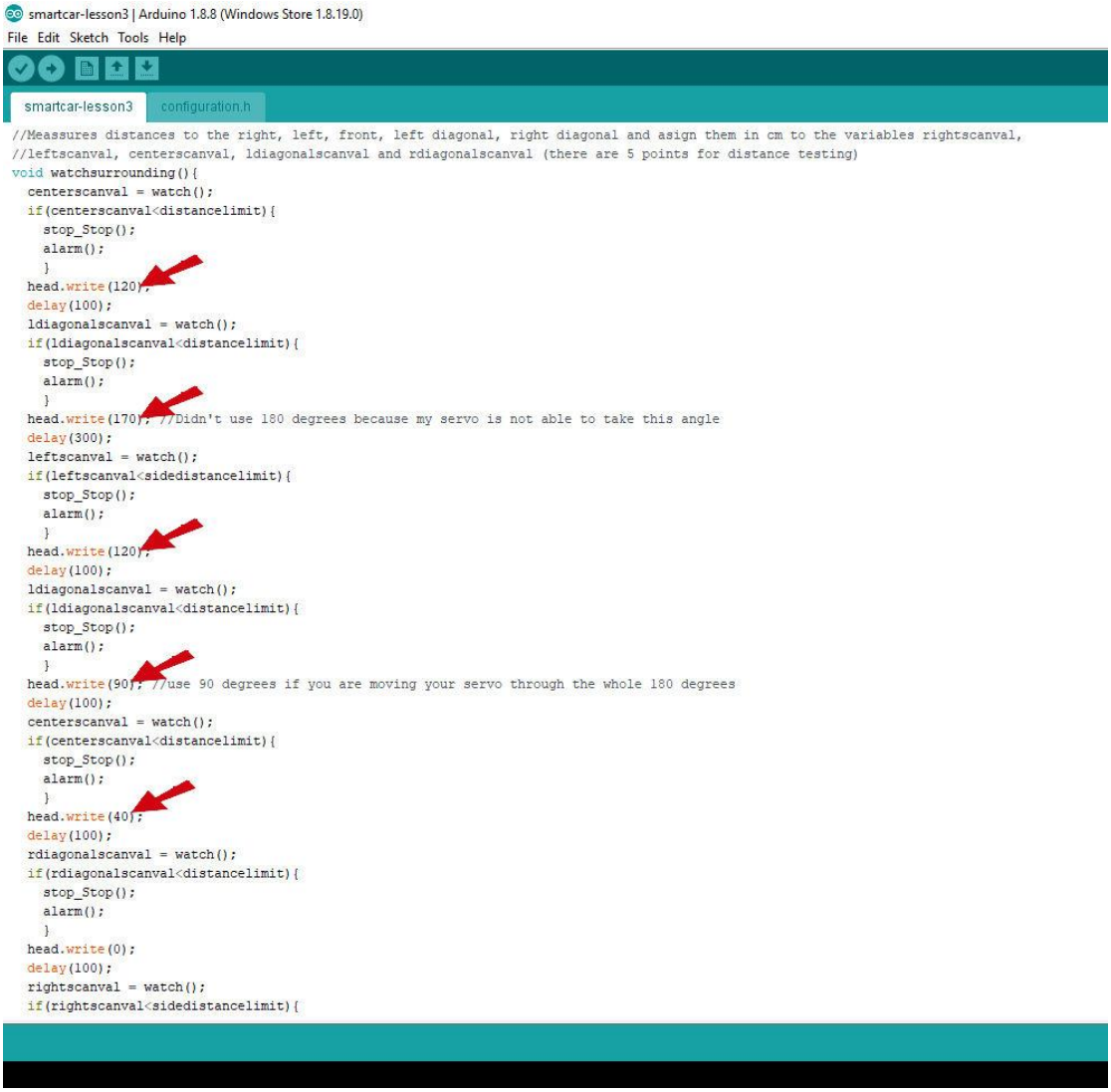
Go back to the code loaded in Arduino IDE and locate the section of code shown below:


```
int watch(){
  long howfar;
  digitalWrite(Trig_PIN,LOW);
  delayMicroseconds(5);
  digitalWrite(Trig_PIN,HIGH);
  delayMicroseconds(15);
  digitalWrite(Trig_PIN,LOW);
  howfar=pulseIn(Echo_PIN,HIGH);
  howfar=howfar*0.01657; //how far away is the object in cm
  //Serial.println((int)howfar);
  return round(howfar);
}
//Meassures distances to the right, left, front, left diagonal, right diagonal and asign them in cm to the variables rightscanval,
//leftscanval, centerscanval, ldiagonalscanval and rdiagonalscanval (there are 5 points for distance testing)
void watchsurrounding(){
  centerscanval = watch();
  if(centerscanval<distancelimit){
    stop_Stop();
    alarm();
    }
  head.write(120);
  delay(100);
  ldiagonalscanval = watch();
  if(ldiagonalscanval<distancelimit){
    stop_Stop();
    alarm();
    }
  head.write(170); //Didn't use 180 degrees because my servo is not able to take this angle
  delay(300);
  leftscanval = watch();
  if(leftscanval<sidedistancelimit){
    stop_Stop();
    alarm();
    }
  head.write(120);
  delay(100);
  ldiagonalscanval = watch();
  if(ldiagonalscanval<distancelimit){
    stop_Stop();
    alarm();
    }
  head.write(90); //use 90 degrees if you are moving your servo through the whole 180 degrees
  delay(100);
  centerscanval = watch();
  if(centerscanval<distancelimit){
    stop_Stop();
    alarm();
```

A. The number shown is calculated using the formula: distance=(time/2) * speed of sound. The speed of sound is 0.0343 cm/s. Using this formula, the code defaults to stopping approximately 1 second before running into an object. This may not be the optimum stopping time. My car would still run into things. You can change the number in your code to change the stopping time of your car. You can find the new number by substituting the stopping time into the equation above. For example, if you want your car to stop 1.5 seconds before it hits an object: distance =(1.5/2)*0.0343 or 0.025725. If you change the number in the code from 0.01657 to 0.025725,

verify the code and reload the new code to your Uno board, your car will stop 1.5 seconds before it runs into something. Feel free to experiment with other times and change your code to see how it affects your car.

B. Open the code in the Arduino software and find the section of code below.



C. In the code you will see degrees listed between 0 and 170 (180 is not used because the servo motor would get stuck if it rotated the entire 180 degrees).
    a. By changing each section marked head.write from 120 to 135 you will allow your servo motor to turn an extra 10 degrees to the right. This allows your scan to cover a bigger area of sweep.
    b. Verify your code again and upload it to the Uno board.
    c. Experiment with your car by changing this number to a new number between 90 and 170 to find the optimum sweep for you. Remember to verify and upload your new code each time.

d. If you would like to manipulate the amount of time the servo motor starts a sweep, you can experiment with the number next to the word delay. It is set to a default of 100.