PS: ADD THIS TO THE REPORT AFTER EXPLAINING THE MAIN PART

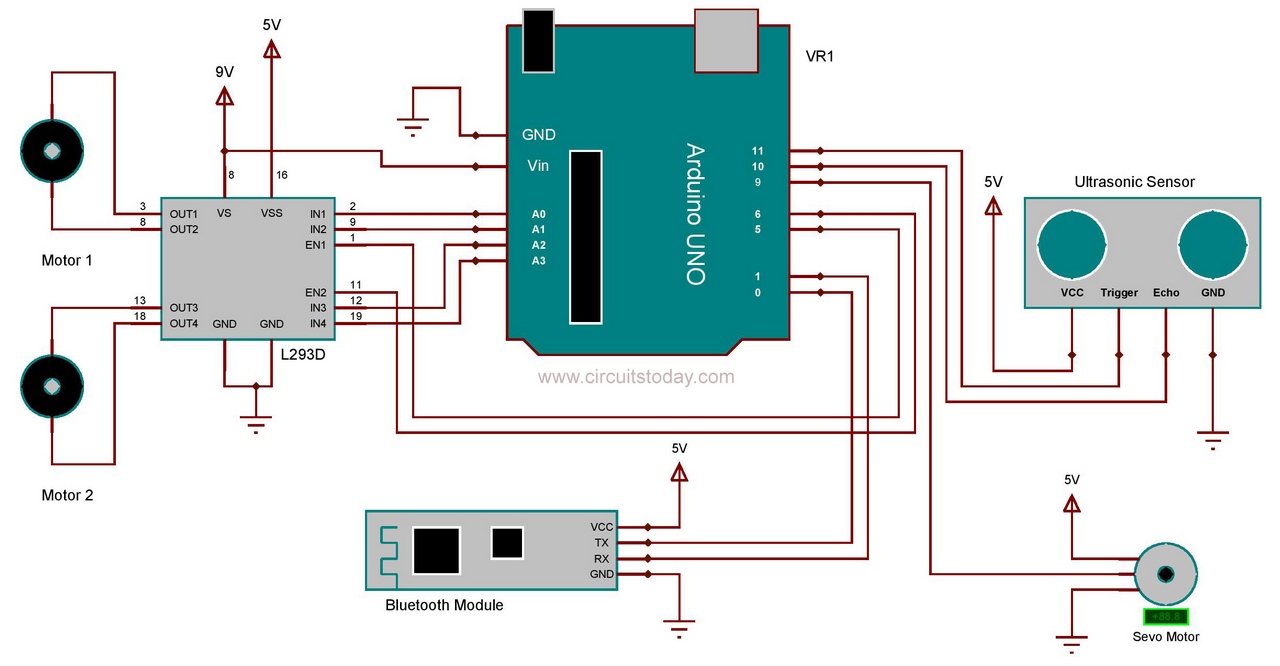
OBSTACLE AVOIDANCE BLUETOOTH CONTROLLED CAR (USING ARDUINO UNO)

AIM OF THE PROJECT : This project is designed to build a robot that automatically detects the obstacle on its path and guides itself whenever an obstacle comes ahead of it. This robotic vehicle is built, using Arduino UNO board. An ultrasonic sensor is used to detect any obstacle ahead of it. A motor driver IC and 2 DC motors are used for controlling the movement of the robot. A servo motor is also used in this project. The ultrasonic sensor is then mounted on the servo and by rotating the servo to different angles we will obtain the readings from the ultrasonic sensor in those angles. This will help the controller to detect the exact path to navigate. A Bluetooth module is also added to the project (which is optional ) in order to control the robot from your android phone when it is in manual mode.

**Objectives of the Project**

* Navigate safely by avoiding obstacles comes ahead.
* Detecting the exact path by checking the sensor readings in different angles.
* Send status of the robot movement (using Bluetooth module) to the nearby android phone when the robot is in automatic mode.
* Navigate in manual mode by receiving signals from the phone.

CIRCUIT DIAGRAM (ONLY CAR)



PROCEDURE FOR MAKING THE CIRCUIT

IMPORTANT CONNECTIONS WITH DEVICES

Assemble the circuit as shown in the diagram! Important connections are explained below.

HC SR04 is the ultrasonic sensor that we are using here. The ultrasonic sensor has 4 pins: Vcc, Trig, Echo and Gnd. Vcc and Gnd are connected to the supply pins of the Arduino. Trig is connected to the 11th pin and Echo is connected to 10th pin of the Arduino.

As mentioned earlier a motor driver IC called L293D is used for controlling the DC motors. It is a 16 pin IC which can drive two motors simultaneously.  1st and 9th pin are the enable pins, which are connected to the 5th and 6th pins of the Arduino board. Pins 2 and 7 are control inputs from microcontroller for first motor. They are connected to pins A0 and A1 of Arduino respectively. Similarly, pins 10 and 15 are control inputs from microcontroller for second motor. They are connected to pins A2 and A3 of Arduino.

When the robot is switched ON, both the motors of the robot will run and the robot moves forward.  During this time, the ultrasonic sensor continuously calculate the distance between the robot and the obstacle in front of it. If the distance between the robot and the obstacle is less than 30cm, the robot will stop moving and rotate the sensor using servo motor to take readings in different angles.  The correct turn which is to be taken can be decided by simply checking the angle in  which sensor gives maximum reading. That will be the path with less obstacle. Servo motor have a signal line , which is connected to the 9th pin of  arduino. The  rotation of the servo is done by generating pwm signal on its signal line.

A Bluetooth module called HC05  is also used in this project (optional). It is connected to the RX and TX pins of Arduino. It is used here for controlling the robot with your android phone. We developed  simple android app for this. The .apk file of the application is attached with this article. You can simply download and install it on your phone. By using the app , you can switch the robot to automatic or manual mode with a single click. And also shows the status of the robot when it is in automatic mode.

ARDUINO CODE

#include <NewPing.h>

#include <Servo.h>

Servo myservo;

#define TRIGGER\_PIN 11

#define ECHO\_PIN 10

#define MAX\_DISTANCE 1000

NewPing sonar(TRIGGER\_PIN, ECHO\_PIN, MAX\_DISTANCE);

#define motor\_aPin1 A0

#define motor\_aPin2 A1

#define motor\_bPin1 A2

#define motor\_bPin2 A3

#define motor\_aEnable 5

#define motor\_bEnable 6

#define OB\_range 30

int i=0, pos = 0,current\_distance=0;

int range0=0, range30=0, range60=0, range85=0,range90=0,range95=0, range120=0, range150=0, range180=0 ;

unsigned long previous\_millis = 0;

char serialData;

void setup()

{

Serial.begin(9600);

myservo.attach(9);

pinMode(motor\_aPin1,OUTPUT);

pinMode(motor\_aPin2,OUTPUT);

pinMode(motor\_bPin1,OUTPUT);

pinMode(motor\_bPin2,OUTPUT);

pinMode(motor\_aEnable,OUTPUT);

pinMode(motor\_bEnable,OUTPUT);

analogWrite(motor\_aEnable,200);

analogWrite(motor\_bEnable,200);

}

void brake()

{

digitalWrite(motor\_aPin1,LOW);

digitalWrite(motor\_aPin2,LOW);

digitalWrite(motor\_bPin1,LOW);

digitalWrite(motor\_bPin2,LOW);

}

void forward()

{

digitalWrite(motor\_aPin1,LOW);

digitalWrite(motor\_aPin2,HIGH);

digitalWrite(motor\_bPin1,LOW);

digitalWrite(motor\_bPin2,HIGH);

}

void forward\_left()

{

digitalWrite(motor\_aPin1,LOW);

digitalWrite(motor\_aPin2,HIGH);

digitalWrite(motor\_bPin1,LOW);

digitalWrite(motor\_bPin2,LOW);

}

void forward\_right()

{

digitalWrite(motor\_aPin1,LOW);

digitalWrite(motor\_aPin2,LOW);

digitalWrite(motor\_bPin1,LOW);

digitalWrite(motor\_bPin2,HIGH);

}

void backward()

{

digitalWrite(motor\_aPin1,HIGH);

digitalWrite(motor\_aPin2,LOW);

digitalWrite(motor\_bPin1,HIGH);

digitalWrite(motor\_bPin2,LOW);

}

void backward\_right()

{

digitalWrite(motor\_aPin1,LOW);

digitalWrite(motor\_aPin2,LOW);

digitalWrite(motor\_bPin1,HIGH);

digitalWrite(motor\_bPin2,LOW);

}

void backward\_left()

{

digitalWrite(motor\_aPin1,HIGH);

digitalWrite(motor\_aPin2,LOW);

digitalWrite(motor\_bPin1,LOW);

digitalWrite(motor\_bPin2,LOW);

}

void left()

{

digitalWrite(motor\_aPin1,LOW);

digitalWrite(motor\_aPin2,HIGH);

digitalWrite(motor\_bPin1,HIGH);

digitalWrite(motor\_bPin2,LOW);

}

void right()

{

digitalWrite(motor\_aPin1,HIGH);

digitalWrite(motor\_aPin2,LOW);

digitalWrite(motor\_bPin1,LOW);

digitalWrite(motor\_bPin2,HIGH);

}

int range(int pos)

{

myservo.write(pos);

delay(300);

current\_distance = sonar.ping\_cm();

if(current\_distance==0)

current\_distance=100;

if(current\_distance > 0 && current\_distance < 15){

Serial.print("B");

if(pos==90){

backward();delay(500);}

if(pos < 90){

backward\_right();delay(500);}

if(pos > 90){

backward\_left();delay(500);}

return current\_distance;

}}

void loop()

{

Automatic: brake();

delay(300);

while(1){

if(Serial.available()>0)

serialData=Serial.read();

if(serialData=='M')

goto Manual;

above: range90=0;

range90=range(90);

delay(10);

while(range90 >= OB\_range ){

if(millis()-previous\_millis>2000){

previous\_millis=millis();

range(180);

delay(50);

range(0);

delay(50);}

range90=range(90);

analogWrite(motor\_aEnable,150);

analogWrite(motor\_bEnable,150);

forward(); Serial.print("F");

if(Serial.available()>0)

serialData=Serial.read();

if(serialData=='M')

goto Manual;

}

brake();

if(range90 <OB\_range)

{

analogWrite(motor\_aEnable,100);

analogWrite(motor\_bEnable,100);

brake();

range60=0;

range60=range(60);

delay(200);

range120=0;

range120=range(120);

delay(200);

if(Serial.available()>0)

serialData=Serial.read();

if(serialData=='M')

goto Manual;

if(range60 >OB\_range || range120 >OB\_range )

{

if(range60 >= range120){

forward\_right();Serial.print("R");

delay(50);goto above;}

else if(range60 < range120){

forward\_left();Serial.print("L");

delay(50);goto above;}

}

if(range60 <OB\_range && range120 <OB\_range)

{

above1: range30=0;

range30=range(30);

delay(200);

range150=0;

range150=range(150);

delay(200);

if(Serial.available()>0)

serialData=Serial.read();

if(serialData=='M')

goto Manual;

if(range30 >OB\_range || range150 >OB\_range )

{

if(range30 >= range150){

right();Serial.print("R");

delay(100);goto above;}

else if(range30 < range150){

left();Serial.print("L");

delay(100);goto above;}

}

if(range30 <OB\_range && range150 <OB\_range)

{

range0=0;

range0=range(0);

delay(200);

range180=0;

range180=range(180);

delay(200);

if(Serial.available()>0)

serialData=Serial.read();

if(serialData=='M')

goto Manual;

if(range0 >OB\_range || range180 >OB\_range )

{

if(range0 >= range180){

backward\_right();Serial.print("R");

delay(200);goto above;}

else if(range0 < range180){

backward\_left();Serial.print("L");

delay(200);goto above;}

}

if(range0 <OB\_range && range180 <OB\_range)

{

backward();Serial.print("B");

delay(200);

goto above1;

}

}

}

}

}

Manual: brake();

delay(300);

while(1){

if(Serial.available()>0)

serialData=Serial.read();

if(serialData=='A')

goto Automatic;

if(serialData=='F')

forward();

if(serialData=='B')

backward();

if(serialData=='S')

brake();

if(serialData=='L')

left();

if(serialData=='R')

right();

}

}

EXPLANATION OF THE CODE

At the beginning ,program will continuously check the reading of the ultrasonic sensor when the servo is at 90 degree. A library called “NewPing.h” is used here for taking the sensor reading. By calling the inbuilt function “sonar.ping\_cm()” we will get distance between the sensor and the obstacle in centimeters. One more library called “Servo.h” is used  in this program for controlling the servo. The function used  is “myservo.write(angle)”.

The position of the servo will be at 90 degree at the beginning.  If the reading from the sensor at 90 degree position is less than 30cm, the robot  will stop moving forward. Then it takes the readings at 60 and 120 degree positions by tilting the servo. These readings are then compared and will turns to the side where the reading is greater. If both these readings are also below 30cm, then the servo will turns to 30 and 150 degree positions. And will turns to the side which gives maximum reading. Again if the readings are still below 30cm, it will check the readings at 0 and 180 degree. If all the readings are less than 30 cm, the robot will moves backward and checks the reading again at 0 and 180 degree positions.  This process will continue until it takes a turn and the jumps to the starting of the program. Then it will checks the reading at 90 degree and the process will repeat again.

The program will continuously check the availability of serial data from the Bluetooth module. If a character ‘M’ is received, the robot will switch to manual mode. Further movement of the robot will be according to the characters received via Bluetooth. If character ‘A’ is received, the robot will again switch back to automatic mode.