



Efficient Mechanisms using ARDUINO to Control Robots

Karthik Balasubramanian¹, Arokia Jerald Praveen.D², Aswin Chandrasekaran³, Nawin Pranav.M.V⁴,
Kiran.R⁵

Student, School of Electrical Engineering, VIT University, Vellore-632014, India^{1, 2, 5}

Student, School of Information and Technology, VIT University, Vellore-632014, India³

Student, School of Mechanical and Building Sciences, VIT University, Vellore-632014, India⁴

Abstract: Microcontrollers are used in automatically and autonomously controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems. This paper introduces the controlling of robot using joystick, mobile and laptop using the arduino board. The robot consists of arduino board, an L293D interface circuit and a motor driving system. The controlling devices are connected to the arduino board. The arduino board sends signals to the interfacing board L293D which controls the motor driving system. The robot can also be used as an obstacle avoiding robot and a line follower robot. Using a robot as an obstacle avoider or line follower is made possible simply by using sensors. The sensors used here are proximity sensors. The proximity sensors use variable resistor(s). If the resistance varies the sensor sends different signal to the arduino board and accordingly the robot moves independently. The robot works on combinational logic design and gates.

Keywords: Joystick, DTMF(Dual Tone Multiple Frequency), proximity sensors, arduino board, L293D, DC motors.

INTRODUCTION

In comparison to microprocessors that are primarily used in personal computers and other general purpose applications, microcontrollers are designed for embedded applications. A microcontroller can be considered as a self-contained system with a processor, memory and peripherals and can be used as an embedded system. The majority of microcontrollers in use today are embedded in other machinery, such as automobiles, telephones, appliances, and peripherals for computer systems. The microcontroller used here is derived from the ARDUINO platform.

Platform Arduino

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, and MaxMSP).

Specifications of Arduino

On the far left is the Universal Serial Bus (USB) connector. This connects the board to your computer for three reasons: to supply power to the board, to upload your instructions to the Arduino, and to send data to and receive it from the computer. On the right is the power connector. Through this connector, you can power the Arduino with a standard mains power adapter. The microcontroller is the “brains” of the Arduino. It is a tiny computer that contains a processor to execute instructions, includes various types of memory to hold data and instructions from our sketches, and provides various avenues of sending and receiving data. The first row offers power connections and the ability to use an external RESET button. The second row offers six analog inputs that are used to measure electrical signals that vary in voltage. Furthermore, pins A4 and A5 can also be used for sending data to and receiving it from other devices. Sockets (or pins) numbered 0 to 13 are digital input/output (I/O) pins. They can either detect whether or not an electrical signal is present or generate a signal on command. Pins 0 and 1 are also known as the serial port, which is used to send and receive data to other devices, such as a computer via the USB connector circuitry. The pins labeled with a tilde (~) can also generate a varying electrical signal, which can be useful for such things as creating lighting effects or controlling electric motors. Next are some very useful devices called light-emitting diodes (LEDs); these very tiny devices light up when a current passes through them.

LED's in the Arduino

The Arduino board has four LEDs: one on the far right labeled ON, which indicates when the board has power, and three in another group. The LEDs labeled TX and RX light up when data is being transmitted or received between the Arduino and attached devices via the serial port and USB. The L LED is for your own use (it is connected to the digital I/O pin number 13). The little black square part to the left of the LEDs is a tiny microcontroller that controls the USB interface that allows your Arduino to send data to and receive it from computer.

Reset Button

As with a normal computer, sometimes things can go wrong with the Arduino, and when all else fails, you might need to reset the system and restart your Arduino. This simple RESET button on the board (Figure 2-7) is used to restart the system to resolve these problems.

Motor Driving Table

The L293D motor driver is used in every single robot. Based on the digital output received by the motors, they move in a particular direction. The given table provides the combinations of four possible output values, where A,B,C and D are the output pins of the L293D. Irrespective of the robot behavior, using either combinational logic design or Arduino development, the outputs from the L293D to the motors have to follow the given table.

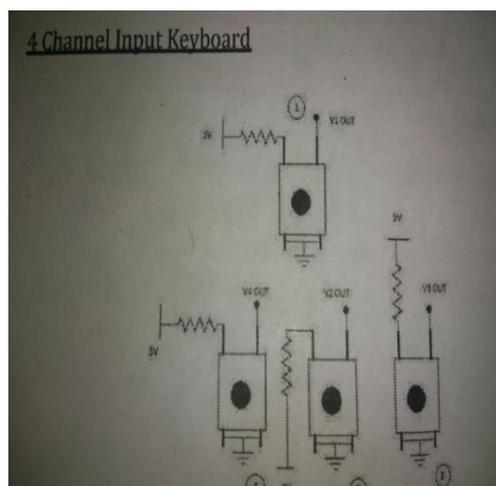
A	B	C	D	O/P
1	0	1	0	FRONT
0	1	0	1	BACK
1	0	0	1	RIGHT
0	1	1	0	LEFT

Manual Robot

The stages involved in a manual robot:



The first stage is the input stage where in we use a 4 channel keypad to provide the required control signals. Next stage is the combinational logic circuit that acts as the interface circuit to convert these signals into the required ones for controlling the motor. The last stage is the motor driving unit that receives the signals and accordingly controls the direction of the rotation of the motors



The diagram below shows one such module/keyboard. Here the switches are connected in active low configuration, such that if a switch is pressed, the corresponding output will turn low. The switches are numbered 1,2,3 and 4 as shown.

The output conditions for the inputs are as follows

1	2	3	4	O/P
0	1	1	1	FRONT
1	0	1	1	BACK
1	1	0	1	RIGHT
1	1	1	0	LEFT

Interface circuit

Since L293D needs input like (1010),(0101),(0110) we need an interface circuit. Let 1,2,3,4 be the inputs and A,B,C,D be the outputs. Based on the combinational logic design procedure we can get the following table.

1	2	3	4	A	B	C	D
0	1	1	1	1	0	1	0
1	0	1	1	0	1	0	1
1	1	0	1	1	0	0	1
1	1	1	0	0	1	1	0

Mobile Controlled (DTMF based) Wireless Robot



DTMF is Dual Tone Multiple Frequency. In a mobile controlled robot the range is longer limited. Such a robot can be operated via the keypad of a mobile phone. The

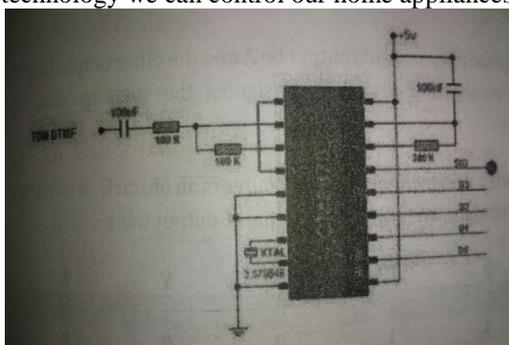
keypad of the mobile phone transmits the pressed key on the keypad via a form of signal known as DTMF. The key pressed on the keypad is generally heard as a sound on the receiver's end so if we did connect a compactable audio jack we would have the audio signal as output from the audio jack. next step is to process this signal and convert this audio signal into some form of binary signal such that the output is different binary combinations for different keys pressed on the keypad. The task is done using an encoder and the output is the binary form of the number pressed. CM8870 is the encoder used for this purpose.

DTMF and Working of the system

DTMF signal is used over analog telephone lines in the voice frequency band between telephone handsets and other communication devices and the switching center. DTMF is a method of instructing switching system of the telephone numbers to be dialed or to issue commands to the switching systems. Robotic control using the DTMF is achieved when the user at the transmitting side presses the mobile phone keypad buttons and at the remote location receiver receives the signal so that the robot takes corresponding actions.

DTMF Decoder IC-CM8870

It is 18 pin IC. The operating voltage is 2.5V-5.5V. One wire of the headphone is connected to the wire of the IC and another to ground. The cell phone (receiver) to a 3.5mm jack is connected at the other end. The cell phone must be kept at auto answer mode. Now call on that phone from another phone (transmitter). Press the keys on the remote phone and the robot moves forward (press 2), backward (press 8), left (press 4) and right (press 6). The four output pins (D0,D1,D2,D3) of the IC is connected to any part of the microcontroller. DTMF technology is very useful in home automation i.e., with the help of this technology we can control our home appliances.



OBSTACLE AVOIDING ROBOT

The robot is designed based on IR-photodiode proximity sensors. The circuit consists of three main parts. They are:

1. Proximity sensor based robot
2. Interface circuit based on combinational logic
3. Motor driver circuit

The output of the proximity sensors will be used to determine the direction of motion of the robot. This is achieved by suitably converting the output into proper control signals for the motor driver unit.

Working Mechanism

The left proximity sensor output is A and the other is B. In the following design, if the robot encounters an obstacle, it

turns right, else it goes straight. The input-output table is shown below:

A	B	W	X	Y	Z
0	0	1	0	1	0
0	1	1	0	0	1
1	0	1	0	0	1
1	1	1	0	0	1

W, X, Y and Z are the outputs from the combinational logic interface which acts as the inputs of the motor driver unit. By the same approach of SOP and POS, we obtain equations as:

$$W=1$$

$$X=0$$

$$Y=(A.B)'$$

$$Z=A+B$$

The following circuit can be easily implemented using logic gates. We require only one NOT gate and one OR gate.

The two things to be noted here are that the range of the proximity sensor is kept maximum that the robot has sufficient distance to stop and turn and a smooth turn will not be made by the robot. As soon as it encounters an obstacle, it turns till the sensor outputs turn low. Then again it moves slightly ahead, till the outputs turn high. Again the robot turns. This cycle repeats itself.

Line Follower Robot

The line follower robot uses the same proximity sensors in a slightly different manner. Here we make use of the fact that the sensor gives a high output when placed in front of a white object and gives a low output in front of a black object. We design the interface circuit and we make use of two sensors placed in line with each other, facing the bottom. We use a black line thick enough to come in proximity of both sensors when in straight line.

The sensors are placed at the bottom of the chassis. When the black path is straight, both the sensors have low output. Hence the robot keeps moving straight. In the next case, when there is a right turn, the left proximity sensor will go out of the black path at a certain point of time as the robot was initially moving straight. We want the robot to turn right in this case. Similarly for the left turn. In case both the inputs are high the robot is out of path, therefore we set it to come backwards as there are chances the robot may overshoot out of the track at some steep turns. The input-output table is shown below:

A	B	W	X	Y	Z
0	0	1	0	1	0
0	1	0	1	1	0
1	0	1	0	0	1
1	1	0	1	0	1

By using SOP or POS approach, we got the various outputs:

$$W=B'$$

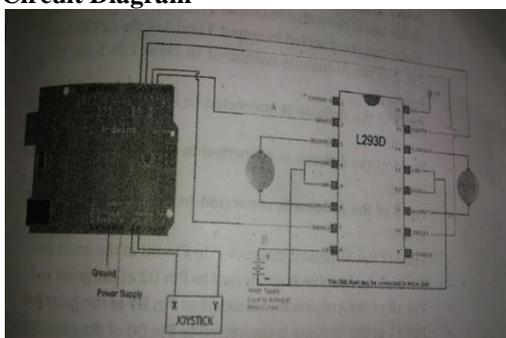
X=B
Y=A'
Z=A

We thus develop the interface circuit using logic gates and the outputs of this stage act as inputs to the motor drive system. We require only two NOT gates to design the following logic. Here we must note that the distance between the proximity sensors is critical in designing a robot. The closer the sensors the faster the robot travels on straight lines. If the distance between the sensors is kept more the robot turns faster but takes time on straight line paths.

JOYSTICK CONTROLLED ROBOT

It is a terrestrial robot whose motion can be controlled using a joystick. This robot uses analog read function which is the core of the robot.

Circuit Diagram



CONNECTIONS

- The L293D is connected to power supply which supplies power to the arduino board.
- Motor one is connected to output 1 and 2 of the L293D board.
- Motor two is connected to output 3 and 4 of the L293D board.
- Pin 2 of the arduino is connected to input PIN 1 in the L293D board.
- Pin 3 of the arduino is connected to input PIN 2 in the L293D board.
- Pin 4 of the arduino is connected to input PIN 3 in the L293D board.
- Pin 5 of the arduino is connected to input PIN 4 in the L293D board.
- PIN A0 of the arduino is connected to PIN X of the joystick.
- PIN A1 of the arduino is connected to PIN Y of the joystick.

Program for Joystick Controlled Robot

```
int motor_left[] = {2, 3};
int motor_right[] = {4, 5}; //pins 4 and 5 are addressed
using motor right
int switch1=A1; //pins A0 to A1 are addressed to the 2
switches
int switch2=A2;
int p=A0;
int g=A3;
int r1; //r1 to r4 are variables
int r2;
//----- Setup
void setup() //loop runs only once
```

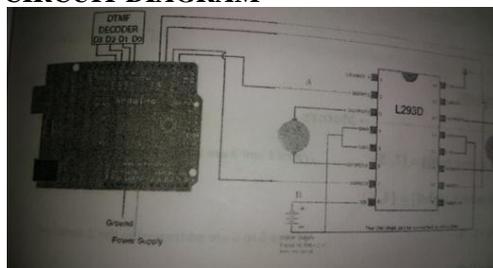
```
{
// Setup motors
inti;
for (i = 0; i< 2; i++)
{
pinMode(motor_left[i], OUTPUT); //initializing the motor
as an output port
pinMode(motor_right[i], OUTPUT);
}
pinMode(switch1,INPUT); //initializing the switch as an
input port
pinMode(switch2,INPUT);
pinMode(p,OUTPUT);
pinMode(g,OUTPUT);
r1=0;
r2=0;
}
//----- Loop
void loop() //runs infinitely
{
analogWrite(p,225);
analogWrite(g,0);
r1=analogRead(switch1); //r1 to r4 get the values of s1 to
s4 respectively
r2=analogRead(switch2);
if(r1>700)
turn_right();
if(r1<90)
turn_left();
if(r2<90)
drive_forward();
if(r2>700)
drive_backward();
if(r1>90&& r1<700&& r2>90&& r2<700)
motor_stop();
}
// ----- Drive
void motor_stop(){
digitalWrite(motor_left[0], LOW); //Assigns the value of
motor left to LOW
digitalWrite(motor_left[1], LOW);
digitalWrite(motor_right[0], LOW);
digitalWrite(motor_right[1], LOW);
}
void drive_forward(){
digitalWrite(motor_left[0], HIGH); //Assigns the value of
motor left to HIGH
digitalWrite(motor_left[1], LOW);
digitalWrite(motor_right[0], HIGH);
digitalWrite(motor_right[1], LOW);
}
void drive_backward(){
digitalWrite(motor_left[0], LOW);
digitalWrite(motor_left[1], HIGH);
digitalWrite(motor_right[0], LOW);
digitalWrite(motor_right[1], HIGH);
}
void turn_left()
{
digitalWrite(motor_left[0], LOW);
```

```
digitalWrite(motor_left[1], HIGH);
digitalWrite(motor_right[0], HIGH);
digitalWrite(motor_right[1], LOW);
}
void turn_right()
{
digitalWrite(motor_left[0], HIGH);
digitalWrite(motor_left[1], LOW);
digitalWrite(motor_right[0], LOW);
digitalWrite(motor_right[1], HIGH); }
```

Mobile Controlled Robot

This can be operated by keypad of a mobile phone by using a form of signal known as DTMF (Dual Tone Multi Frequency).

CIRCUIT DIAGRAM



Connections

- The L293D is connected to power supply which supplies power to the arduino board.
- Motor one is connected to output 1 and 2 of the L293D board.
- Motor two is connected to output 3 and 4 of the L293D board.
- Pin 2 of the arduino is connected to input PIN 1 in the L293D board.
- Pin 3 of the arduino is connected to input PIN 2 in the L293D board.
- Pin 4 of the arduino is connected to input PIN 3 in the L293D board.
- Pin 5 of the arduino is connected to input PIN 4 in the L293D board.
- PIN 8 of the arduino is connected to PIN D3 of the gamepad.
- PIN 9 of the arduino is connected to PIN D2 of the gamepad.
- PIN 10 of the arduino is connected to PIN D1 of the gamepad.
- PIN 11 of the arduino is connected to PIN D0 of the gamepad.

PROGRAM FOR MOBILE CONTROLLED ROBOT

```
int motor_left[] = {2, 3}; //pins 2 and 3 are
addressed using motor left
int motor_right[] = {4, 5};
int s1=8; //pins 8 to 11 are addressed to the 4
switches
int s2=9;
int s3=10;
```

```
int s4=11;
int p=12;
int g=13;
int r1; //r1 to r4 are variables
int r2;
int r3;
int r4;
void setup() //loop runs only once
{
// Setup motors
inti;
for (i = 0; i < 2; i++)
{
pinMode(motor_left[i], OUTPUT);
//initializing the motor as an output port
pinMode(motor_right[i], OUTPUT);
}
pinMode(s1, INPUT); //initializing the switch as an
input port
pinMode(s2, INPUT);
pinMode(s3, INPUT);
pinMode(s4, INPUT);
pinMode(p, OUTPUT);
pinMode(g, OUTPUT);
r1=0;
r2=0;
r3=0;
r4=0;
}
void loop()
//runs infinitely
{
digitalWrite(p, HIGH);
digitalWrite(g, LOW);
r1=digitalRead(s1); //r1 to r4 get the values of
s1 to s4 respectively
r2=digitalRead(s2);
r3=digitalRead(s3);
r4=digitalRead(s4);
if(r1==HIGH&&r2==HIGH&&r3==LOW&&r4==LOW)
{
motor_stop();
}
if(r1==LOW&&r2==LOW&&r3==HIGH&&r4==LOW)
{
turn_left();
}
if(r1==HIGH&&r2==LOW&&r3==HIGH&&r4==LOW)
{
drive_backward();
}
if(r1==LOW&&r2==HIGH&&r3==LOW&&r4==LOW)
{
drive_forward();
}
if(r1==LOW&&r2==HIGH&&r3==HIGH&&r4==LOW)
{
turn_right();
}
}
```

```
//----- Drive
voidmotor_stop(){
digitalWrite(motor_left[0],LOW);

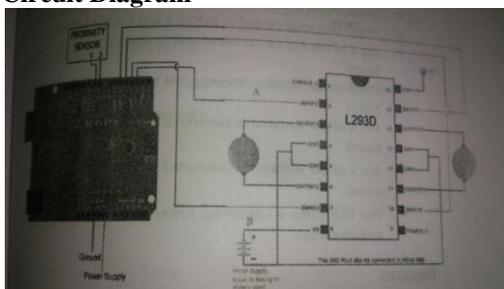
//Assigns the value of motor left to LOW
digitalWrite(motor_left[1], LOW);
digitalWrite(motor_right[0], LOW);
digitalWrite(motor_right[1], LOW);
}
voiddrive_forward(){
digitalWrite(motor_left[0],HIGH);

//Assigns the value of motor left to HIGH
digitalWrite(motor_left[1], LOW);
digitalWrite(motor_right[0], HIGH);
digitalWrite(motor_right[1], LOW);
}
voiddrive_backward(){
digitalWrite(motor_left[0], LOW);
digitalWrite(motor_left[1], HIGH);
digitalWrite(motor_right[0], LOW);
digitalWrite(motor_right[1], HIGH);
}
voidturn_left()
{
digitalWrite(motor_left[0], LOW);
digitalWrite(motor_left[1], HIGH);
digitalWrite(motor_right[0], HIGH);
digitalWrite(motor_right[1], LOW);
}
voidturn_right()
{digitalWrite(motor_left[0], HIGH);
digitalWrite(motor_left[1], LOW);
digitalWrite(motor_right[0], LOW);
digitalWrite(motor_right[1], HIGH); }
```

OBSTACLE AVOIDING ROBOT

The working principle of the robot is based on infrared photo diode proximity sensors. The principle of line following robot.

Circuit Diagram



CONNECTIONS

- This L293D is connected to power supply which supplies power to the arduino board.
- Motor one is connected to output 1 and 2 of the L293D.
- Motor 2 is connected to output 3 and 4 of the L293D board.
- PIN 2 of the arduino is connected to input PIN 1 in the L293D board

- PIN 3 is connected to the input PIN 2 of the L293D board
- PIN 4 is connected to the input PIN 3 of the L293D board.
- PIN 5 is connected to the input PIN 4 of the L293D board.
- The output from the proximity sensor 1 is connected to pin 8 of Arduino.
- The output from the proximity sensor 2 is connected to pin 9 of Arduino.

PROGRAM FOR OBSTACLE AVOIDING ROBOT

```
intmotor_left[] = {2, 3}; //pins 2 and 3 are addressed using motor left
intmotor_right[] = {4, 5};
int s1=6; //pins 5 to 6 are addressed to the 2 switches
int s2=7;
int p1=8;
int p2=10;
int g1=9;
int g2=11;
int read1; //read1 and read2 are variables
int read2;
//----- Setup
void setup()
//loop runs once
{
// Setup motors
inti;
for(i = 0; i< 2; i++)
{
pinMode(motor_left[i], OUTPUT);
//initializing the motor as an output port
pinMode(motor_right[i], OUTPUT);
}
pinMode(s1,INPUT); //initializing the switch as an input port
pinMode(s2,INPUT);
pinMode(p1,OUTPUT);
pinMode(p2,OUTPUT);
pinMode(g1,OUTPUT);
pinMode(g2,OUTPUT);
read1=0;
read2=0;
}
//----- Loop
void loop()
//loop runs infinitely
{
digitalWrite(p1,HIGH);
digitalWrite(p2,HIGH);
digitalWrite(g1,LOW);
digitalWrite(g2,LOW);
read1=digitalRead(s1); //read1 and read2 get the values of s1 and s2 respectively
read2=digitalRead(s2);
```

```

if(read1==LOW&&read2==LOW)
{
drive_forward();
}
if(read1==HIGH&&read2==HIGH)
{
turn_right();
}
if(read1==HIGH&&read2==LOW)
{
turn_right();
}
if(read1==LOW&&read2==HIGH)
{
turn_right();
}
//----- Drive
voidmotor_stop()
{
digitalWrite(motor_left[0], LOW);
//Assigns the value of motor left to LOW
digitalWrite(motor_left[1], LOW);
digitalWrite(motor_right[0], LOW);
digitalWrite(motor_right[1], LOW);
delay(25);
}
voiddrive_forward()
{
digitalWrite(motor_left[0],HIGH); //Assigns the value of
motor left to HIGH
digitalWrite(motor_left[1], LOW);
digitalWrite(motor_right[0], HIGH);
digitalWrite(motor_right[1], LOW);
}
voiddrive_backward()
{
digitalWrite(motor_left[0], LOW);
digitalWrite(motor_left[1], HIGH);
digitalWrite(motor_right[0], LOW);
digitalWrite(motor_right[1], HIGH);
}
voidturn_left()
{
digitalWrite(motor_left[0], LOW);
digitalWrite(motor_left[1], HIGH);
digitalWrite(motor_right[0], HIGH);
digitalWrite(motor_right[1], LOW);
}
voidturn_right()
{digitalWrite(motor_left[0], HIGH);
digitalWrite(motor_left[1], LOW);
}

```

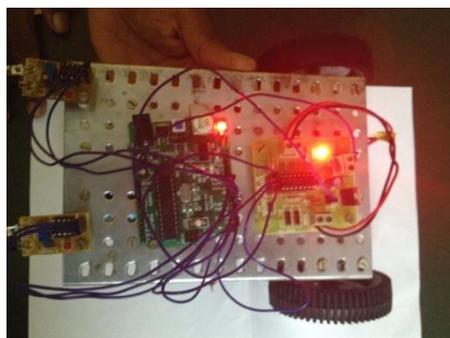
```

digitalWrite(motor_right[0], LOW);
digitalWrite(motor_right[1], HIGH); }

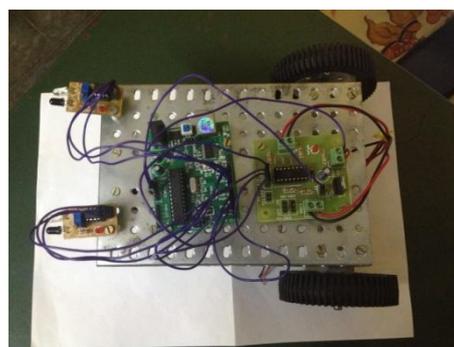
```

Hardware Implementation

Robot in ON condition (From Top View)



Robot in OFF condition (From TOP view)



CONCLUSION

This paper describes the different methods of controlling a robot. The joystick controlled robot, mobile controlled robot and laptop controlled robot are manually controlled robot while the obstacle avoider robot and line follower robot are automatically controlled using proximity sensors. All the robots discussed above has same arduino board, L293D and motor driving unit but only the types of controlling the robot varies.

5. Yun Chan Cho ; Jae Wook Jeon “Remote robot control system based on DTMF of mobile phone”.

REFERENCES

1. Pakdaman, M , Sanaatiyan, M.M, Ghahroudi, M.R “A line follower robot from design to implementation: Technical issues and problems”
2. Hasan, K.M,AI-Nahid, A. ; Reza, K.J,Khatun, S,Basar, M.R “Sensor based autonomous color line follower robot with obstacle avoidance”
3. Jong-Ho Han ; Kyung-Wook Noh ; Jang-Myung Lee “PD-Fuzzy control for the velocity of mobile robot using the haptic joystick”.
4. Nagata, F. ; Watanabe, K. ; Hashino, S. ; Tanaka, H. ; Matsuyama, T. ; Hara, K “Polishing robot using a joystick controlled teaching system”.