Ahsanullah University of Science and Technology

Department of Electrical and Electronic Engineering

Project Report

TITLE: Line following robot using Arduino UNO

Submitted by:

Name :

Nazmus Saqib ID-12-02-05-029

Md. Sifat ID-12-02-05-030

Yafiz Mohammad Saba ID-12-02-05-035

Mir Asif Rahman ID-12-02-05-045

Section – A

Year – 2nd Semester – 2nd

Project Coordinator: **Mr. Hasib Md. Abid Bin Farid**

Designation: Assistant Professor,AUST.

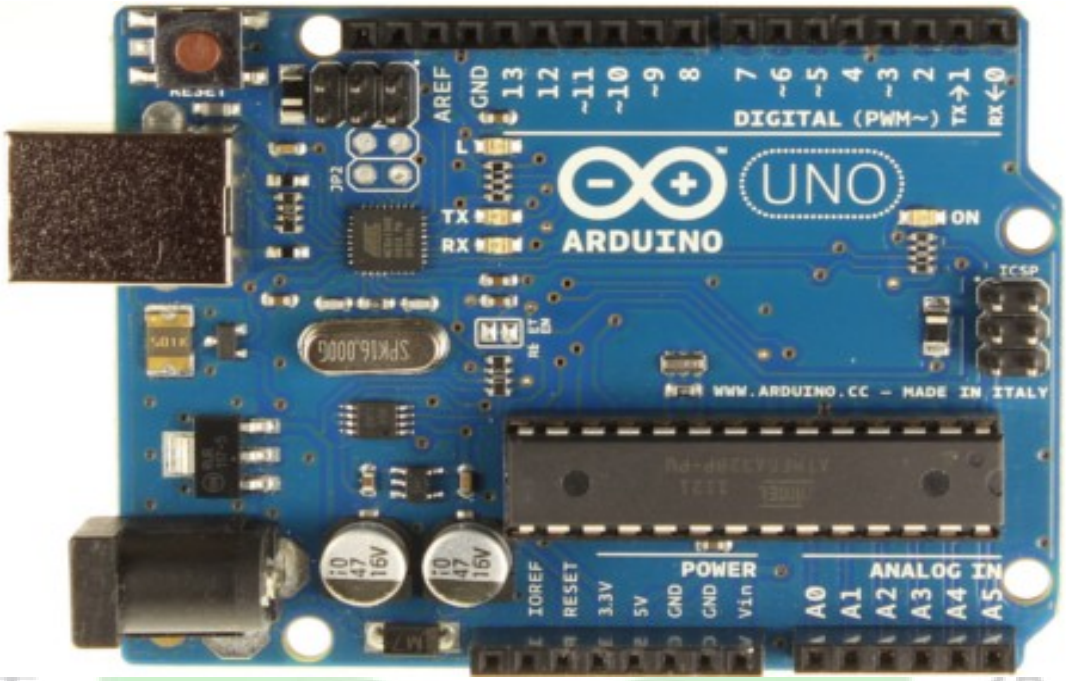
Equipment :

- Arduino Uno R3.
- LED
- LDR
- DC Gear Motor with Wheels.
- DC Power Adapter (9V,2A)
- Ball Castor
- L293D IC
- Breadboard & Jumper wires

Microcontroller:

Microcontrollers are the preferred method for endowing a robot with smarts. The reasons for this include their low costs, simple power requirements (usually 5 to 9 V), and ability of most, to be programmed using software and a simple hardware interface on the PC. Once programmed, the microcontroller is disconnected from the PC and operates on its own. These are programmed either in an assembly language or in a high level language such as C or C++.Here we use Arduino UNO R3.

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the



microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. Besides it is cheap and almost available in electronics hardware stores.

Summary :

Microcontroller	**ATmega328**
Operating Voltage	**5V**
Input Voltage (recommended)	**7-12V**
Input Voltage (limits)	**6-20V**
Digital I/O Pins	**14 (of which 6 provide PWM output)**
Analog Input Pins	**6**
DC Current per I/O Pin	**40 mA**
DC Current for 3.3V Pin	**50 mA**

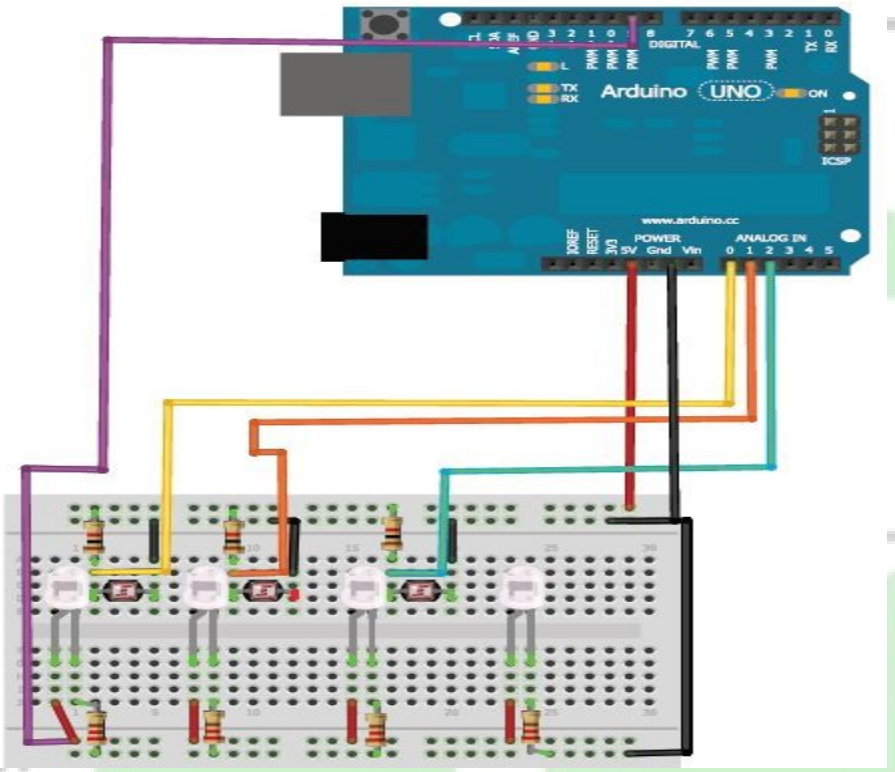
Flash Memory	**32 KB (ATmega328) of which 0.5 KB used by bootloader**
SRAM	**2 KB (ATmega328)**
EEPROM	**1 KB (ATmega328)**
Clock Speed	**16 MHz**

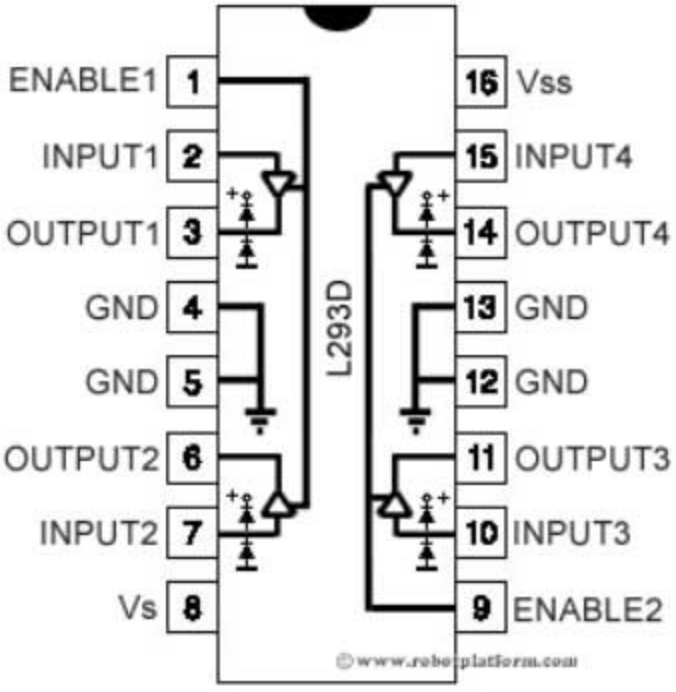
Working Principle:

The main work is based on LED and LDR. Sensor transmits LED. It is known that when light falls on white surface it gets reflected. On the other hand if light falls on black surface it is totally absorbed. This concept is completely followed here.

When LED falls on black line, it is absorbed. So that receiver doesn't get any light . Similarly, when LED falls on white line it is reflected. Thus receiver receives light . This particular concept we used to programme a microcontroller named Arduino UNO r3.

White line gives an ADC value above 600, while black line gives an ADC value ideally below 600. In our project the programme code is written such that for sensor-1 and sensor-2 an ADC value below 600. So that it will make both the motor more forward. So if the sensor-1 and sensor-2 find out black line then the body starts moving.


Arduino
UNO
DIGITAL
POWER
ANALOG IN
www.arduino.cc


ENABLE1 1
INPUT1 2
OUTPUT1 3
GND 4
GND 5
OUTPUT2 6
INPUT2 7
Vs 8
16 Vss
15 INPUT4
14 OUTPUT4
13 GND
12 GND
11 OUTPUT3
10 INPUT3
9 ENABLE2
L293D
©www.robotplatform.com

Code:

```
int sensor1 = A0;
int sensor2 = A1;
int ledpin = 9;
int IN1 = 10;
int IN2 = 11;
void setup()
{
Serial.begin(9600);
pinMode(sensor1, INPUT);
pinMode(sensor2, INPUT);
pinMode(ledpin, OUTPUT);

pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
}
void loop()
{
digitalWrite(ledpin, HIGH);
int sensorvalue1 = analogRead(A0);
int sensorvalue2 = analogRead(A1);
Serial.print("sensor value1 = ");
Serial.println(sensorvalue1);
Serial.print("sensor value2 = ");
Serial.println(sensorvalue2);

if((sensorvalue1 < 600) && (sensorvalue2 < 600) )
{
```

```
digitalWrite(IN2, HIGH);
digitalWrite(IN1, HIGH);
}
if((sensorvalue1 < 600) && (sensorvalue2 > 600) )
{
digitalWrite(IN2, LOW);
digitalWrite(IN1, HIGH);

}
if((sensorvalue1 > 600) && (sensorvalue2 < 600) )
{
digitalWrite(IN2, HIGH);
digitalWrite(IN1, LOW);

}

}
```

Troubleshooting:

1. The most significant problem with this project was integrating the "line sensing" part with the "car driving" part. Individually line sensors work fine and give correct output based on the line. But if we integrate these two modules, then car sometimes skip line sensing. Possible reasons are sensor introduces noise/spike. To solve this problem we

tried different kinds of sensors and emitters. At last the available best effect was achieved by using LED and LDR (receiver).

2. At first we construct the circuit in bread board but we faced some difficulties. That's why we constructed it in Vero board.

3. After sensing the line, light sensors give a voltage. This voltage varies between a range when the line is black. To give binary output, we had to check it using reference voltage. Even different chips should give voltage between same ranges; it does not do it practically . So we had to manually check every sensor to set a reference voltage and made the reference voltage using different resistors.

Application :

1. We can use this robot in a warehouse . It can deliver goods by following lines.
2. Another application is serving food in a restaurant.
3. We can use it to serve medicine to the patient in a hospital.