विज्ञान एवं
प्रौद्योगिकी मंत्रालय
MINISTRY OF
SCIENCE AND
TECHNOLOGY

सत्यमेव जयते

Miranda House
University of Delhi

INSPIRE INTERNSHIP PROGRAMME 2024

Innovation in Science Pursuits for Inspired Research
An Initiative of DST, Govt of India

8-12 JULY 2024

Robotics: Impacting Life

Offered by: Physics Department

ROBOTICS: IMPACTING LIFE

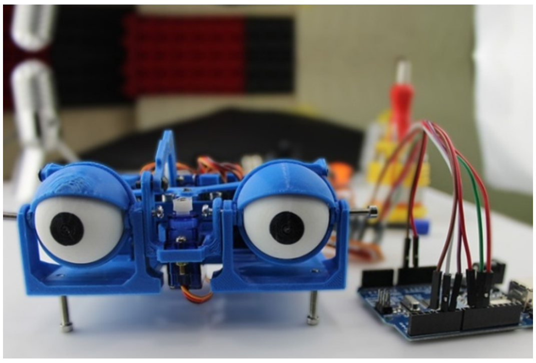
In simple terms, robotics combines science, engineering, and technology to design, construct, operate, and use machines programmed to replicate, substitute, or assist humans in completing tasks of varying complexity. These machines are known as robots.

Basic aspects of the robotics

The basic aspects of the robotics to create a robot are given below:

- **Electrical/electronic components-** The robotics requires electrical and electronic components as power supply, sensors, and microcontroller and motors circuits.
- **Mechanical equipment-** The robotics requires mechanical equipment for giving shape or designing the body of a robot.
- **Computer programs-** The robotics also includes computer programs to provide the instructions to the robot as what type of task, when it should be done, how it should be done, etc. Robo ML, ROBOFORTH, XRCL, and visual programming are the programming languages which are used in the robotics.

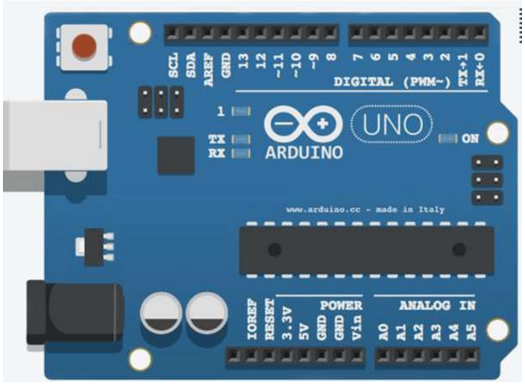
Robots in Lab using Arduino

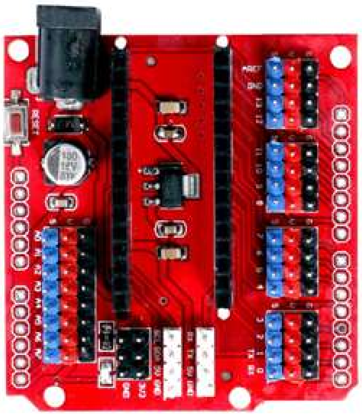


Preliminary Questions

1. What is Arduino?
2. What are the real-life applications of Arduino?
3. Which Arduino is mostly used?
4. Does Arduino play any role in Robotics?

WHAT IS ARDUINO?





Arduino is a microcontroller-based open source electronic prototyping board which can be programmed with an easy-to-use software Arduino IDE. There are various boards available such as Arduino Lilypad, Arduino Mini, Arduino Mega, and Arduino Nano. In this experiment, Arduino Uno board is utilized. Here UNO means 'one' in Italian. It was named as UNO to label the first release of Arduino Software. It was also the first USB board released by Arduino. It is very much popular compared to other boards in the family because it has documentation that is much more detailed. This led to its increased adoption for making interactive electronic projects.

The board consists of digital and analog Input/Output pins (I/O), shields, and other circuits. The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a USB connector, a power jack, and an ICSP (In-Circuit Serial Programming) header. The major components of Arduino UNO board are shown in figure 1 and described in detail as follows:

- USB connector - This is a printer USB port used to load a program from the Arduino IDE onto the Arduino board. The board can also be powered through this port.

- Power port - The Arduino board can be powered through an AC-to-DC adapter or a battery. The power source can be connected by plugging in a plug into the power jack of the board.

- Microcontroller - It is the most prominent black rectangular chip with 28 pins. Think of it as the brains of your Arduino. The microcontroller used on the UNO board is Atmega328P by Atmel (a major microcontroller manufacturer).

- Analog Input pins - The Arduino UNO board has 6 analog input pins, labeled “Analog 0 to 5.” These pins can read the signal from an analog sensor like a temperature sensor and convert it into a digital value so that the system understands. These pins just measure voltage and not the current because they have very high internal resistance. Hence, only a small amount of current flows through these pins.

Although these pins are labeled analog and are analog input by default, these pins can also be used for digital input or output.

- Digital pins - Arduino has digital pins from "Digital 0 to 13." These pins can be used as either input or output pins. When used as output, these pins act as a power supply source for the components connected to them. When used as input pins, they read the signals from the component connected to them.

  When digital pins are used as output pins, they supply 40 milliamps of current at 5 volts, which is more than enough to light an LED.
  Some of the digital pins are labeled with tilde (~) symbol next to the pin numbers (pin numbers 3, 5, 6, 9, 10, and 11). These pins act as normal digital pins but can also be used for Pulse-Width Modulation (PWM), which simulates analog output like fading an LED in and out.

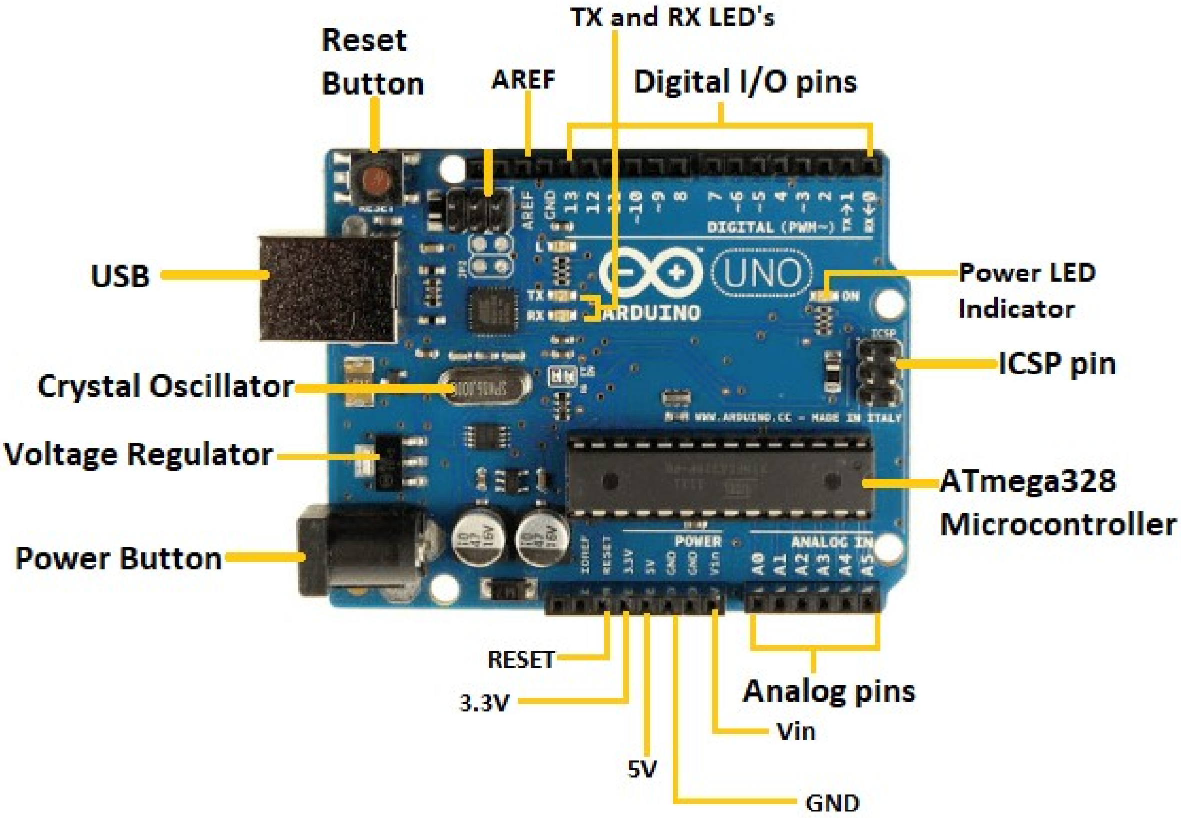


Fig. 1 Pictorial representation of Arduino Uno.

- Reset switch - When this switch clicked, it sends a logical pulse to the reset pin of the Microcontroller, and now runs the program again from the start. This can be very useful if the code doesn't repeat, but we want to test it multiple times.

- Crystal oscillator - This is a quartz crystal oscillator which ticks 16 million times a second. On each tick, the microcontroller performs one operation, for example, addition, subtraction, etc.

- USB interface chip - Think of this as a signal translator. It converts signals in the USB level to a level that an Arduino UNO board understands.

- TX RX LEDs - TX stands for transmit, and RX for receive. These are indicator LEDs which blink whenever the UNO board is transmitting or receiving data.

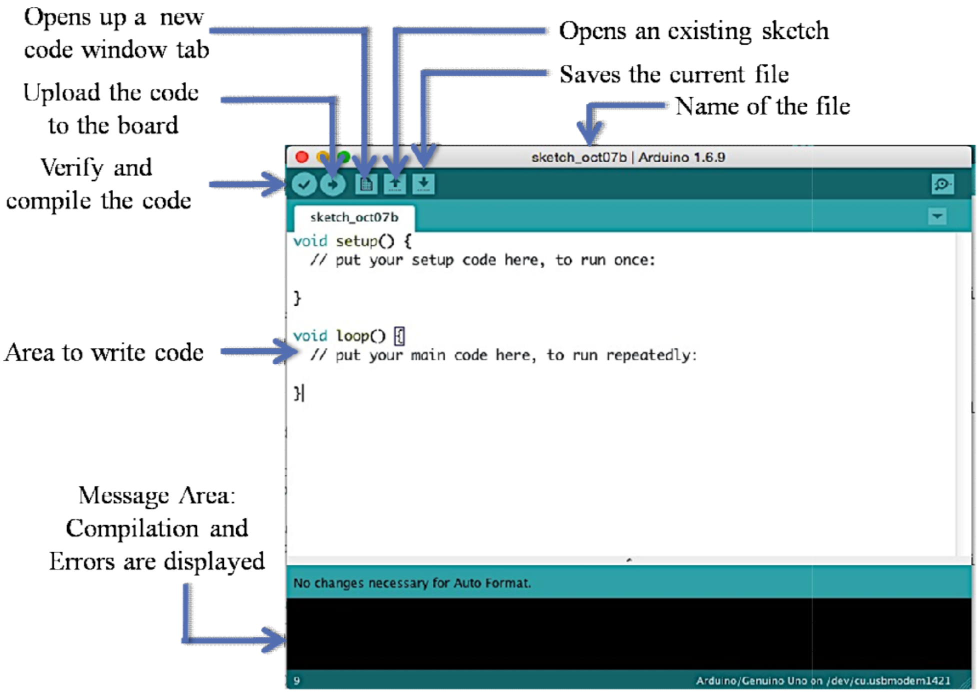
Arduino senses the environment by receiving inputs from many sensors, and affects its surroundings by controlling lights, motors, and other actuators.

Arduino Software

Arduino is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms. The IDE is common to all available boards of Arduino. User can write code to instruct the hardware to work. The link to download the Arduino software is given below:

https://www.arduino.cc/en/software

Download the software according to your operating system and install the Arduino driver. The IDE window is shown in figure 2.


Opens up a new code window tab
Opens an existing sketch
Saves the current file
Name of the file
Upload the code to the board
Verify and compile the code
Area to write code
Message Area: Compilation and Errors are displayed
sketch_oct07b | Arduino 1.6.9
sketch_oct07b

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

No changes necessary for Auto Format.

9 Arduino/Genuino Uno on /dev/cu.usbmodem1421

Fig. 2 Arduino IDE software window.

ARDUINO UNO FOR GLOWING LED

Objective:

Design a circuit for automatic blinking of LED using Arduino Uno.

Apparatus Required: Arduino Uno board, Bread board, Arduino IDE software, LED, Resistor, Laptop etc.

Circuit Diagram

This is one of the simple examples which can show the physical output (blinking the on-board LED) with an Arduino. The circuit diagram for the automatic blinking of LED using Arduino Uno is shown in figure 3. As shown in Fig. 3, a digital pin (for example pin 13) is connected to the LED. LED is connected to a resistor in series which is then grounded.

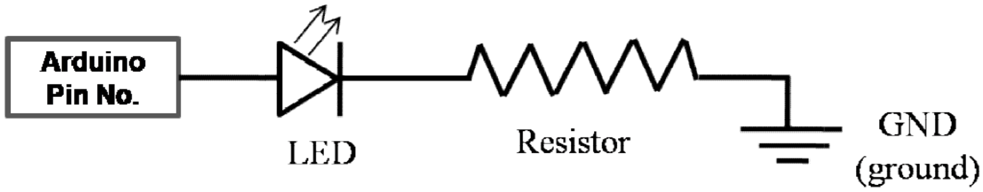


Fig. 3. The circuit diagram for the automatic blinking of LED using Arduino Uno.

A pictorial representation of the circuit connected with Arduino is shown in Fig. 4.

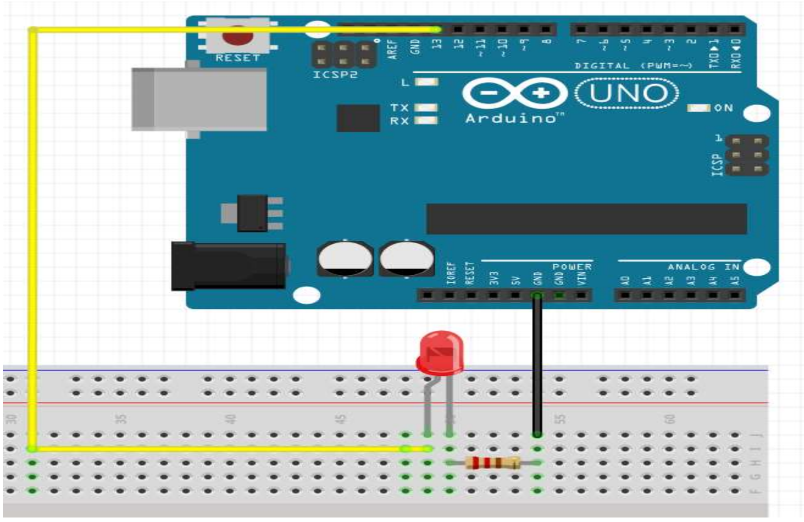


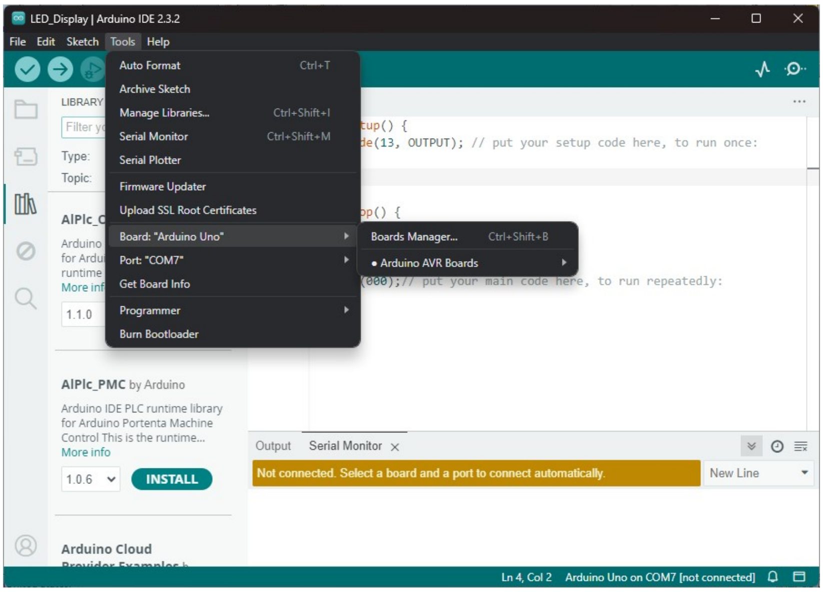
Fig. 4 A pictorial representation of the circuit.

The circuit is connected with a computer using the Arduino connector (This has one end as USB connector). The power LED indicator will glow as soon the Arduino is connected to the computer. A code is written in Arduino IDE for the aforementioned circuit. The code is shown below:

/* Blink: Turns on an LED for one second then off for one second, repeatedly. */

```
void setup() {
pinMode(13, OUTPUT);
}
void loop() {
digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);
}
```

After writing the code a board (Arduino Uno) is selected by clicking Boards Manager tab of the Tools tab of the IDE software as shown in Fig. 5.



```
tup() {
de(13, OUTPUT); // put your setup code here, to run once:

op() {
(000);// put your main code here, to run repeatedly:
```

Not connected. Select a board and a port to connect automatically.

Board: "Arduino Uno"

Port: "COM7"

Ln 4, Col 2   Arduino Uno on COM7 [not connected]

Fig. 5 A pictorial representation of the IDE software for selecting the Arduino Uno board.

The code is then verified by clicking the Verify Button and uploaded to the board by using uploading button. An error message will be displayed in Message Area if the code has any error or the board is not connected or selected properly. For example an error message is displayed in Message Area of the Fig. 5 “Not connected Select a board and a port to connect automatically”. As soon all the errors will be rectified the LED will start blinking according to set time period.

Post Questions:

1. How can the blinking time of LED be controlled?
2. Why does the resistor necessary in the circuit?

SELF DRIVING ARDUINO CAR

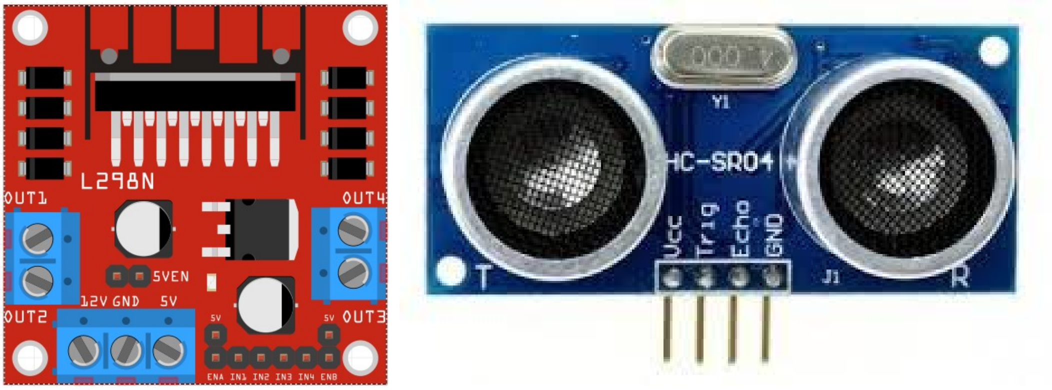
Objective:

Design a circuit for self driving car using Arduino Uno.

Apparatus Required: Arduino Uno board, Bread board, Arduino IDE software, 1 Ultrasonic Sensor, 4 Gear Motor, 4 Wheel, Battery, Switch, Capacitors (5 no. of 100 uF and 2 no. of 0.1uF), 1 Motor Drive L298N.

What is an L298N motor driver?

The L298N is a dual H-Bridge motor driver which allows the speed and the direction control of DC motors at the same time. L298N motor driver is shown in Fig. 6(a).



(a) (b)

Fig. 6 (a) L298N motor driver, (b) Ultrasonic Sensor.

Ultrasonic Sensor

An ultrasonic sensor transmits pressure waves of sound energy at frequencies between 25 and 50 KHz, which is above the human audible range. Most of the ultrasonic sensors operate with pulse waveforms, which detect the presence of an object and measure its speed. These also measure the distance of the object from the sensor by detecting the portion of the transmitted energy that is reflected towards the sensor from an area defined by the transmitter's beam width. A picture of ultrasonic sensor is shown in figure 6 (b).

Circuit Diagram

A pictorial representation of the circuit for self driving car is shown in Fig. 7. As shown in figure the motor driver is connected to four DC motors. Both the right motors are connected with pin Out1 and Out2 and the other two are connected with pin Out3 and Out4. The motor driver is connected with Arduino Uno at digital pins. Similarly Ultrasonic sensor is connected by analog pins given on Arduino board.

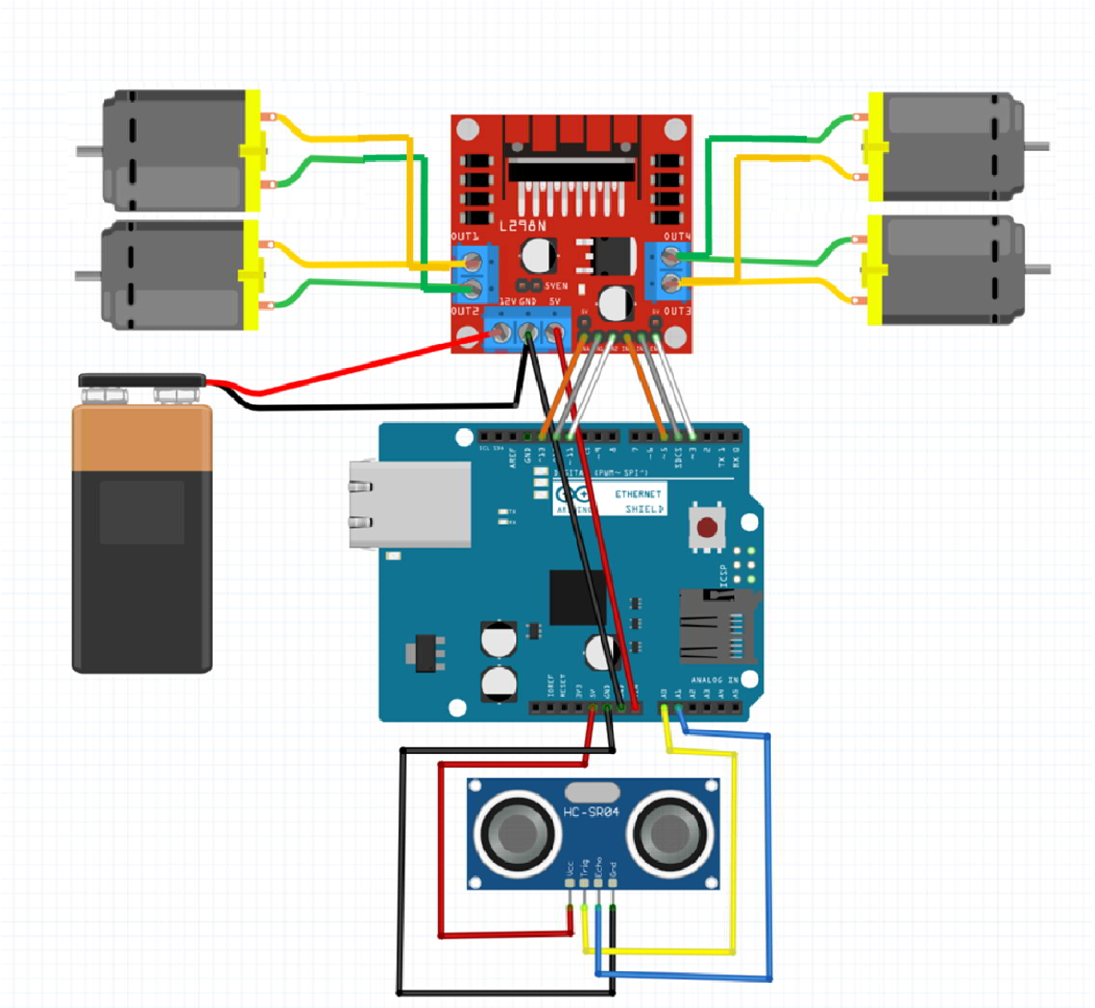


Fig. 7 A pictorial representation of the circuit of the self driving car.

Pin Connections

Pin	Usage
Out 1 and Out 2	The Out 1 and Out 2 pins are connected to motor A and B
Out 3 and Out 4	The Out 3 and Out 4 pins are connected to motor C and D
+12V(Vcc)	The 12V pin supplies power to the motor.
GND	The **GND** pin is for the ground and also needs to be connected to the Arduino board ground.
5V	Conneted to Vin of Arduino board.
ENA	Will enable the right side motors, connected to digital pins on Arduino board.

IN1 and **IN2**	The **In1** and the **In2** pins are both used to control the direction of motor A and B. If the In1 pin is HIGH, and In2 pin is LOW, the motors will spin forward. If the In1 pin is LOW, and In2 pin is HIGH, the motors will spin backward. If both pins are HIGH or if both pins are LOW, the motor will stop.
ENB	Will enable the left side motors, connected to digital pins on Arduino board.
IN3 and **IN4**	The **In3** and the **In4** pins are both used to control the direction of motor C and D. If the In3 pin is HIGH, and In4 pin is LOW, the motor will spin forward. If the In3 pin is LOW, and In4 pin is HIGH, the motor will spin backward. If both pins are HIGH or if both pins are LOW, the motor will stop.

A code is written in Arduino IDE for the aforementioned circuit. The code is shown below:

```
#include <NewPing.h>
int In1 = 7;  // control right motors, control direction
int In2 = 8;  // control right motors, control direction
int ENA = 5;  // enable right motors, control speed
int ENB = 11; // enable left motors, control speed
int In3 = 12; // control left motors, control direction
int In4 = 13; // control left motors, control direction
int SPEED = 150;  //initialize speed for all motors
// for sensor pins
#define trig_pin A1 //analog input 1
#define echo_pin A2 //analog input 2
#define maximum_distance 200
int distance;
long duration;
boolean turn = false;
NewPing sonar(trig_pin, echo_pin, maximum_distance); //sensor
function
void setup() {
  pinMode(In1,OUTPUT); // input to right motors
  pinMode(In2,OUTPUT); // input to right motors
  pinMode(ENA,OUTPUT);  // enable right motors
  analogWrite(ENA, SPEED);  //Feeding speed to right motors
  pinMode(In3,OUTPUT);  // input to left motors
  pinMode(In4,OUTPUT);  // input to left motors
  pinMode(ENB,OUTPUT);  // enable left motors
  analogWrite(ENB , SPEED); //Feeding speed to left motors
  }
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(trig_pin, LOW);
  delayMicroseconds(2);
  digitalWrite(trig_pin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig_pin, LOW);
  duration = pulseIn(echo_pin, HIGH);
  distance = duration/58.2;
  delay(100);
  if(distance > 45){
    moveForward();
    }
  else{
    stopHere();
    delay(500);
    moveBackward();
    delay(500);
    if (turn == false){
        turnLeft();
        turn = true;
        }
    else{
      turnRight();
      turn = false;
      }
    }
  }
void moveForward(){    // move motors in forward direction
    digitalWrite(In1, HIGH);
    digitalWrite(In2, LOW);
    digitalWrite(In3, HIGH);
    digitalWrite(In4, LOW);
    }
void moveBackward(){  // move motors in backward direction
    digitalWrite(In1, LOW);
    digitalWrite(In2, HIGH);
    digitalWrite(In3, LOW);
    digitalWrite(In4, HIGH);
    }
void stopHere(){      //Stop the motros
  digitalWrite(In1, LOW);
    digitalWrite(In2, LOW);
    digitalWrite(In3, LOW);
    digitalWrite(In4, LOW);
    }
void turnLeft(){     //Move to left
  digitalWrite(In4, HIGH);
  digitalWrite(In2, HIGH);
  digitalWrite(In3, LOW);
  digitalWrite(In1, LOW);
  delay(250);
  digitalWrite(In3, HIGH);
  digitalWrite(In2, HIGH);
  digitalWrite(In4, LOW);
  digitalWrite(In1, LOW);
  }
void turnRight(){   // Move to right
  digitalWrite(In4, LOW);
  digitalWrite(In2, LOW);
  digitalWrite(In3, HIGH);
  digitalWrite(In1, HIGH);
  delay(250);
  digitalWrite(In3, LOW);
  digitalWrite(In2, LOW);
  digitalWrite(In4, HIGH);
  digitalWrite(In1, HIGH);
  }
void stopHere()
{
  digitalWrite(In1, LOW);
    digitalWrite(In2, LOW);
    digitalWrite(In3, LOW);
    digitalWrite(In4, LOW);
}
void turnLeft(){
 // digitalWrite(LeftLED, HIGH);
  digitalWrite(In4, HIGH);
  digitalWrite(In2, HIGH);
  digitalWrite(In3, LOW);
  digitalWrite(In1, LOW);
  delay(250);
  digitalWrite(In3, HIGH);
  digitalWrite(In2, HIGH);
  digitalWrite(In4, LOW);
  digitalWrite(In1, LOW);
}
```

Post Questions

1. How do self-driving cars stay on the road?
2. How do vehicles with autonomous or driver-assist features automatically brake, steer around obstacles, or perform tasks like adaptive cruise control?