Introduction to Arduino Programming for Robotic Application

IEEE RAS - VIT CHENNAI

COORDINATORS:

SUMIT
HARIPRASAD
SHIVARAM K K
KEERTI M

IEEE Robotics & Automation Society

AIM:

This Workshop aims to provide :

1.The opportunity to gain theoretical knowledge of Arduino.

2.Understand the structure, mechanism and working of Obstacle Avoidance Robot.

3.Understand the underlying principles of obstacle avoidance.

4.Visualize the live demonstration of the working model.

5.To understand the basics on how to create/ simulate an actual robot.

IEEE Robotics & Automation Society

OUTCOMES:

By the end of this workshop :

1.We will have a comprehensive understanding of various components.

2.Understand the basics of Arduino.

3.Understand the process of Circuit Simulation in TinkerCad software.

IEEE Robotics & Automation Society

- Through this workshop we will see the process of building an obstacle avoiding robot step by step.

  1.Introduction to basic components.

  2.TinkerCad simulation

  3.Programming of Arduino Uno.

  4.Live demonstration of the working model.

IEEE Robotics & Automation Society

Components Used

1.ARDUINO UNO BOARD

2.L298N/293D MOTOR DRIVER

3.BATTERY

4.TWO DC MOTORS

5.JUMPER CABLES

6.THREE HC-SR04 SENSORS

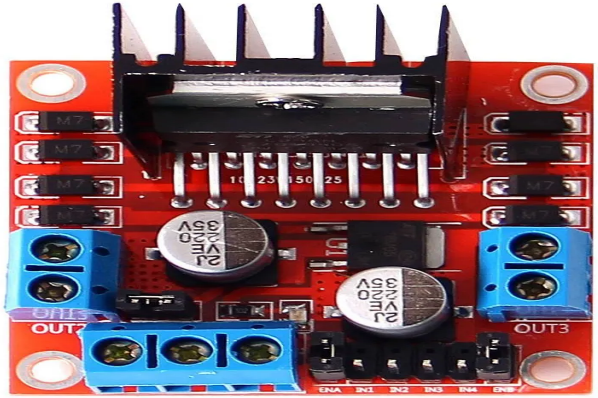
7.WHEELS

8.SWITCH

IEEE Robotics & Automation Society

L298N/293D - Motor Driver

- The L298N is a dual H-Bridge motor driver.
- It allows speed and direction control of two DC motors or one stepper motor at the same time.
- We can only have full control over a DC motor if we can control its speed and spinning direction.

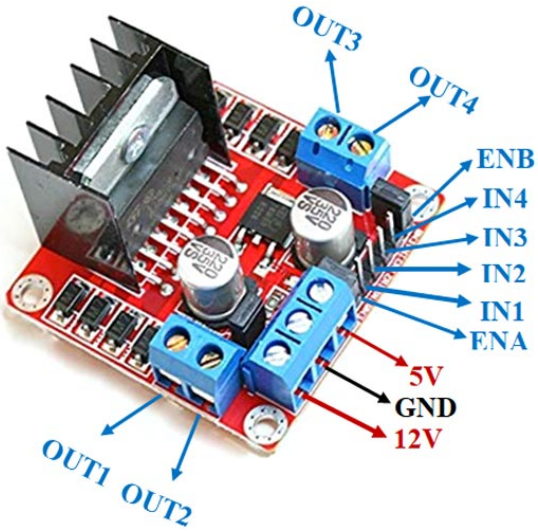


  - PWM – to control speed
  - H-Bridge – to control the spinning direction

IEEE Robotics & Automation Society

Pins:

VCC	Used to Supply power to the motor.Input voltage between 5V to 35V.
GND	GND is a ground pin.
IN1, IN2	These pins are input pins of **Motor A**.Used to control the rotating direction of motor A.
IN3, IN4	These pins are input pins of **Motor B**. Used to control the rotating direction of Motor B.
ENA	ENA pin is used to control the speed of **Motor A**.
ENB	ENB pin is used to control the speed of **Motor B**.

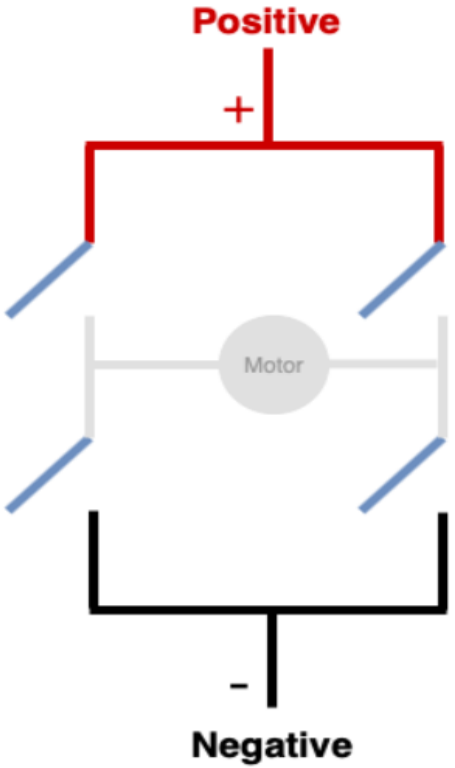


IEEE Robotics & Automation Society

Working:

Case 1:

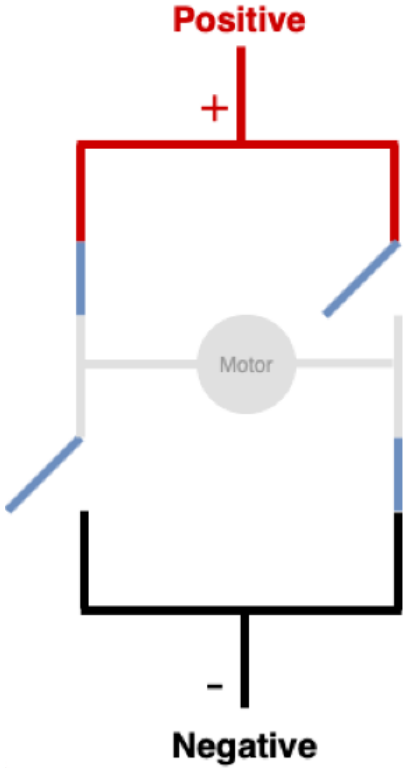
When all switches are open then no current goes to the Motor terminals. So, in this condition, the motor is stopped (not working).



IEEE Robotics & Automation Society

Case 2:

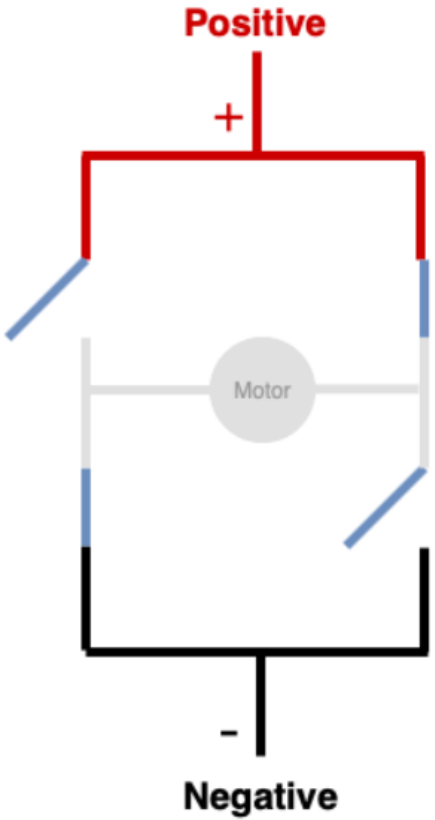
When the switch S1 and S4 are closed. In this condition motor start rotating in a particular direction **(clockwise).**



IEEE Robotics & Automation Society

Case 3:

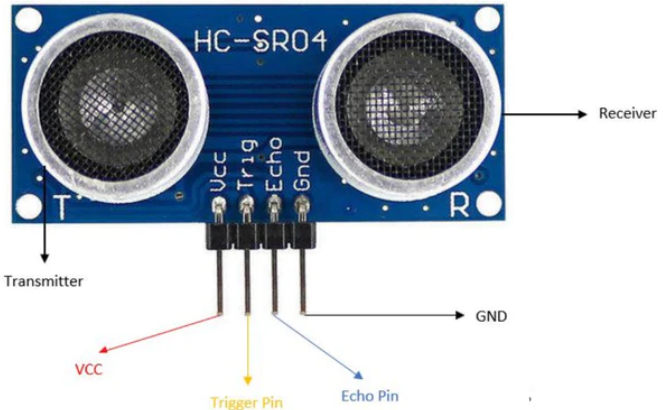
When S2 and S3 switches are closed. In this condition motor start rotating in a particular direction **(anticlockwise).**



IEEE Robotics & Automation Society

Ultrasonic Sensor

- An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves.
- Uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity.
- Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing.
- The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse.

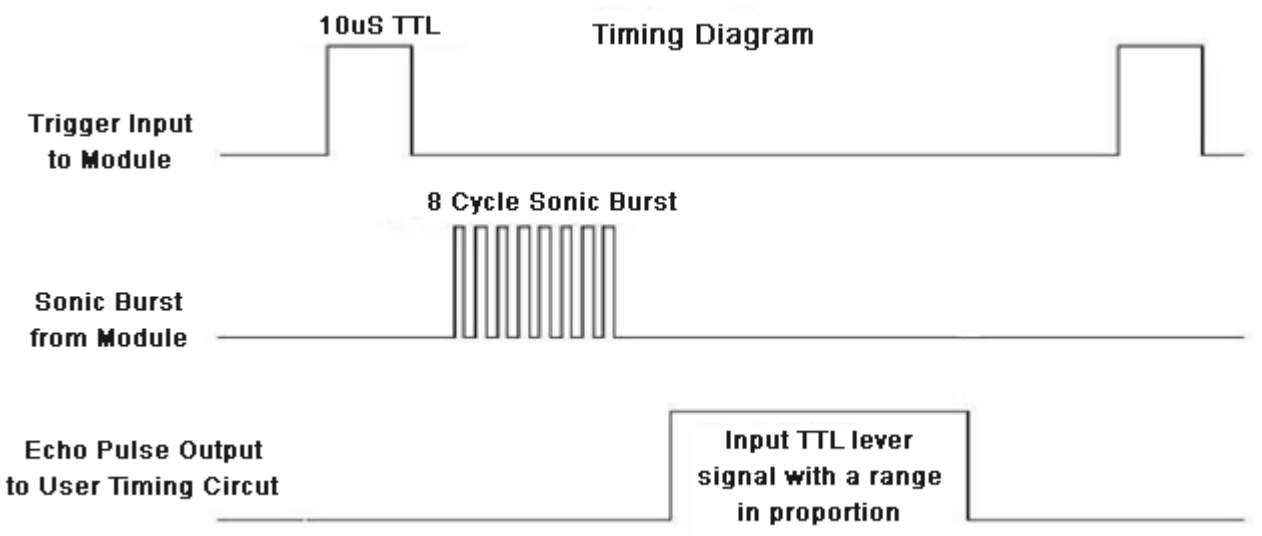


IEEE Robotics & Automation Society

Pin Name	Description
Vcc	The Vcc pin powers the sensor, typically with +5V
Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
Ground	This pin is connected to the Ground of the system.

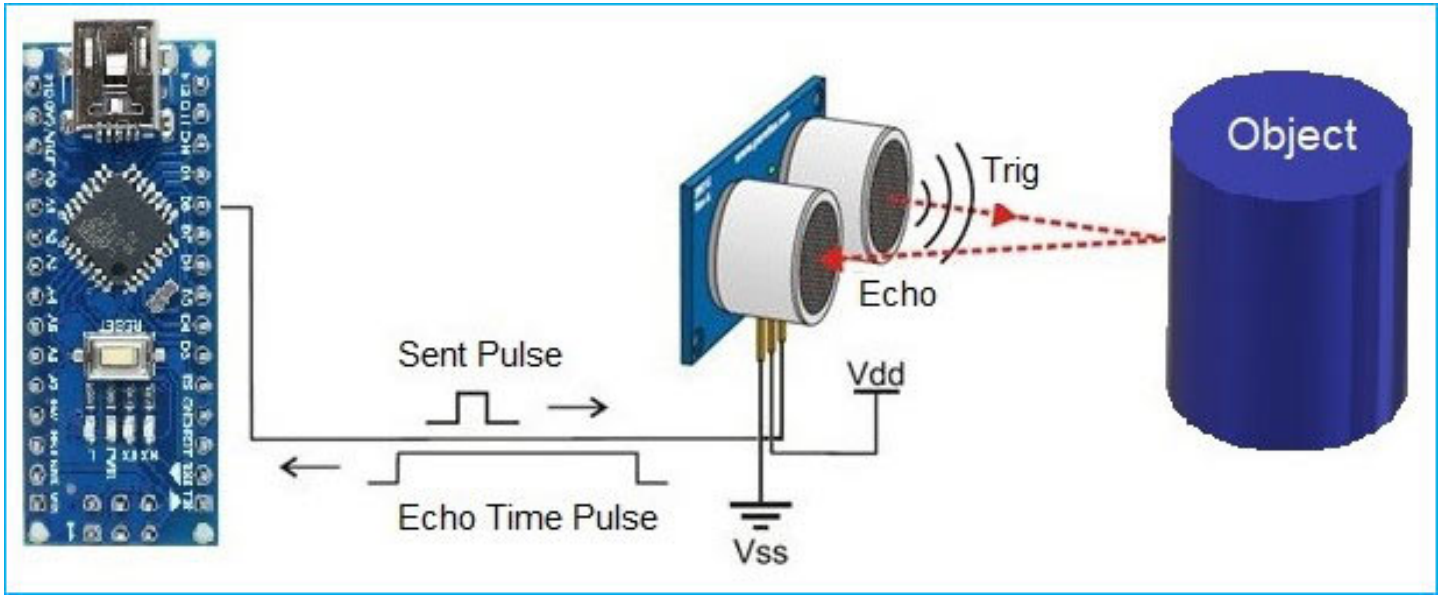
Working:

The Trig pin of HC-SR04 is made high for at least 10 us. A sonic beam is transmitted with 8 pulses of 40KHz each.



IEEE Robotics & Automation Society

The signal then hits the surface and return back and captured by the receiver Echo pin of HC-SR04. The Echo pin had already made high at the time sending high.



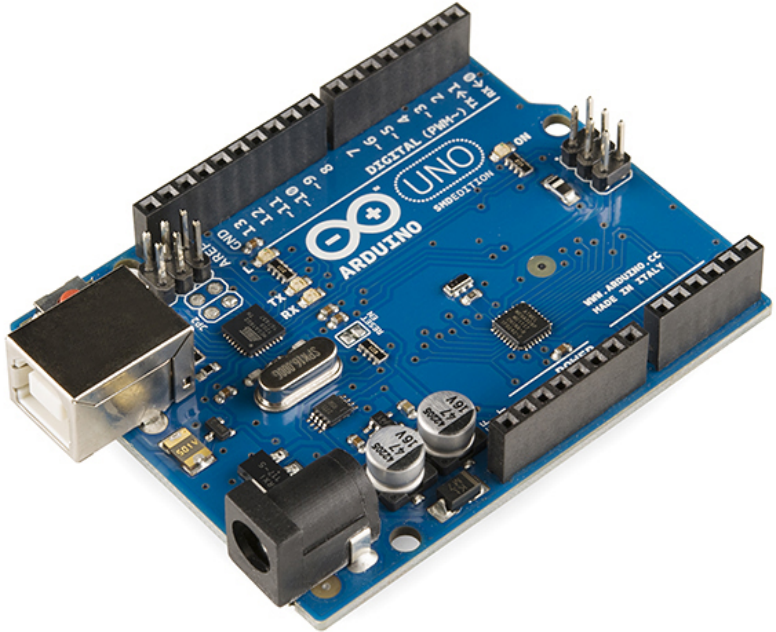
IEEE Robotics & Automation Society

Design and Working of Obstacle Avoidance Robot

- In this robot, we will be using an Arduino Uno microcontroller along with 3 ultrasonic sensors.
- The Arduino Uno will be programmed to use the data from the sensors to determine the robot's position.
- The robot will move forward until it detects an obstacle, then it will use the sensors to determine which direction to turn to avoid the obstacle. This process will continue until the robot reaches the end point.
- The robot is programmed to move forward until it detects an obstacle using the ultrasonic sensors. The sensors emit ultrasonic waves and measure the time taken for the waves to bounce back from the obstacle. This data is used to calculate the distance between the robot and the obstacle.
- The Arduino Uno microcontroller is responsible for controlling the robot's movements and processing the sensor data. The program for the robot is written using the Arduino IDE and uploaded to the microcontroller.

Arduino

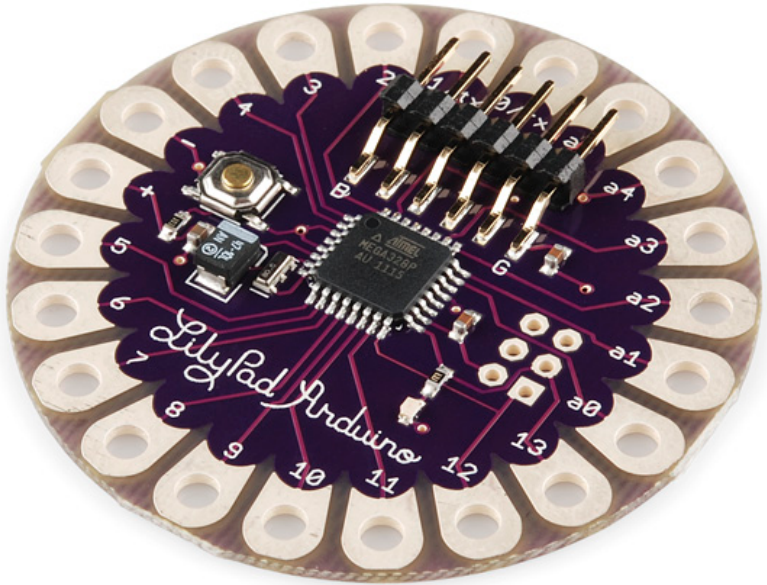
- microcontroller board that can be programmed to interact with the physical world through various sensors



IEEE Robotics & Automation Society

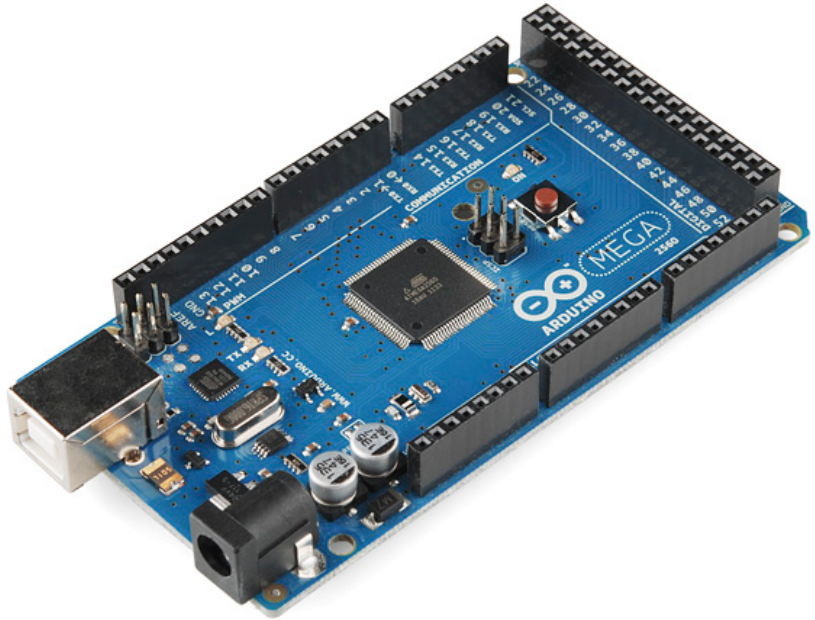
LilyPad Arduino

The LilyPad Arduino is considered as other Arduino board type that is designed for integrating with wearable projects and e-textile projects. This board comes in round shape that helps to decrease the snagging and can be easily connected to other devices. This board uses the Atmega328 microcontroller and Arduino bootloader in it. This board uses very less external component in it that makes the design easy and compatible.



Arduino Mega

This boards is considered as the microcontroller that uses the Atmega2560 in it. There are total 54 input pins and output pins in it in which 14 pins are of PWM output, 4 pins are of hardware port, 16 pins as analog inputs. The board also contain one USB connection, ICSP header, power jack and one REST pin. Used in home automation appliances

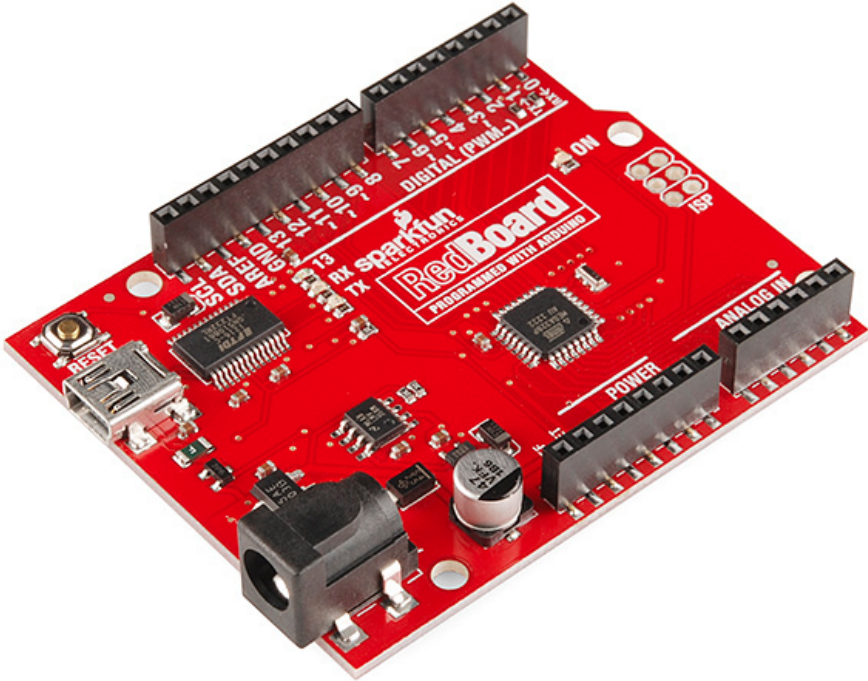


Arduino Leonardo

This board is considered as the microcontroller that uses the Atmega32u4 in it. There are total 20 digital input pins and output pins in it, in that 7 pins are used As PWM and 12 pins used as analog inputs. The board also contain one micro USB connection, power jack, and one RESET button fit in it. There are additional pins which act as crystal oscillator of frequency 16 MHz. used in making of keyboards, mouse, joystick and other appliances.

Arduino Red Board

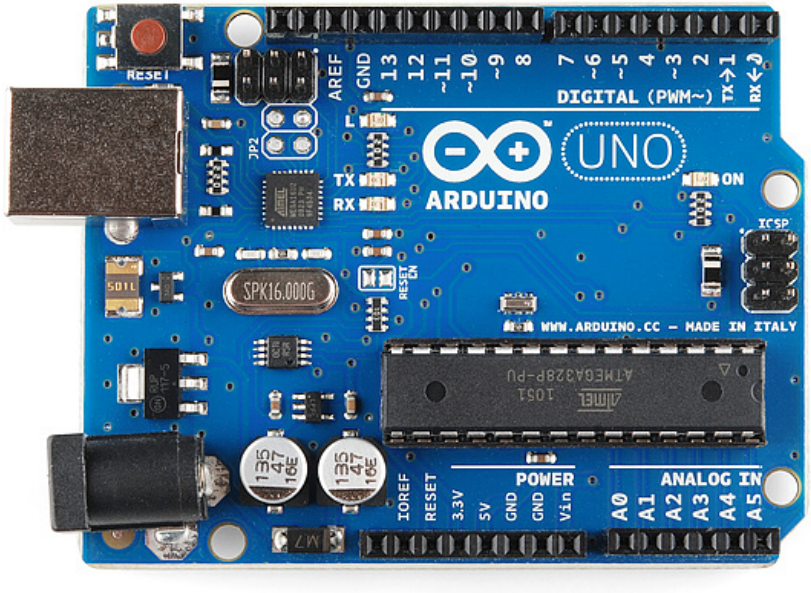
The Arduino Red board is another type of Arduino board that uses the mini USB cable for getting programmed and the Arduino IDE is used for this purpose. This board is compatible with Windows 8 operating system and there is no need to change the security settings to make this board working. The Red board uses the FTDI chip and USB chip for the connection to other device. As the design of red board is very simple it can be easily integrate with other projects. The only requirement is to plug the red board and select appropriate option and can upload program in no time. Used in IoT projects



IEEE Robotics & Automation Society

Arduino UNO

The development of Arduino UNO board is considered as new compared to other Arduino boards. This board comes up with numerous features that helps the user to use this in their project. The Arduino UNO uses the Atmega16U2 microcontroller that helps to increase the transfer rate and contain large memory compared to other boards. No extra devices are needed for the Arduino UNO board like joystick, mouse, keyboard and many more. The Arduino UNO contain SCL and SDA pins and also have two additional pins fit near to RESET pin.

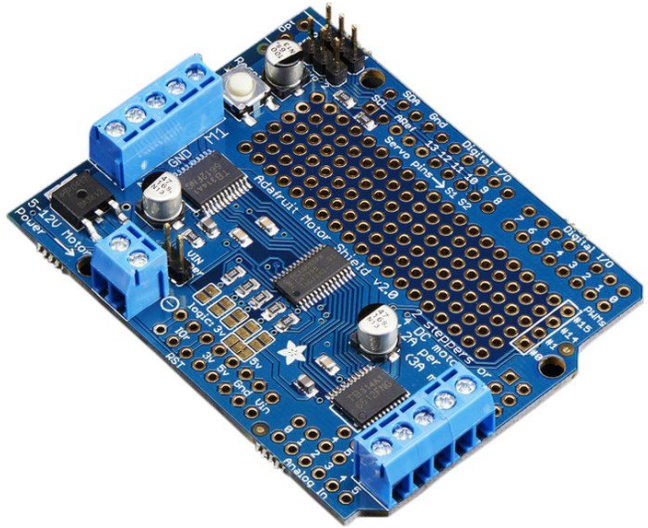


IEEE Robotics & Automation Society

- Boards plugged over the Arduino board to expand its functionalities.
- Different varieties of shields used for various tasks, such as Arduino motor shields, Arduino communication shields, etc.
- Increases the capabilities of the projects.
- Makes our work easy.
- The pin position of the shields is similar to the Arduino boards.
- We can also connect the modules and sensors to the shields with the help of the connection cable.

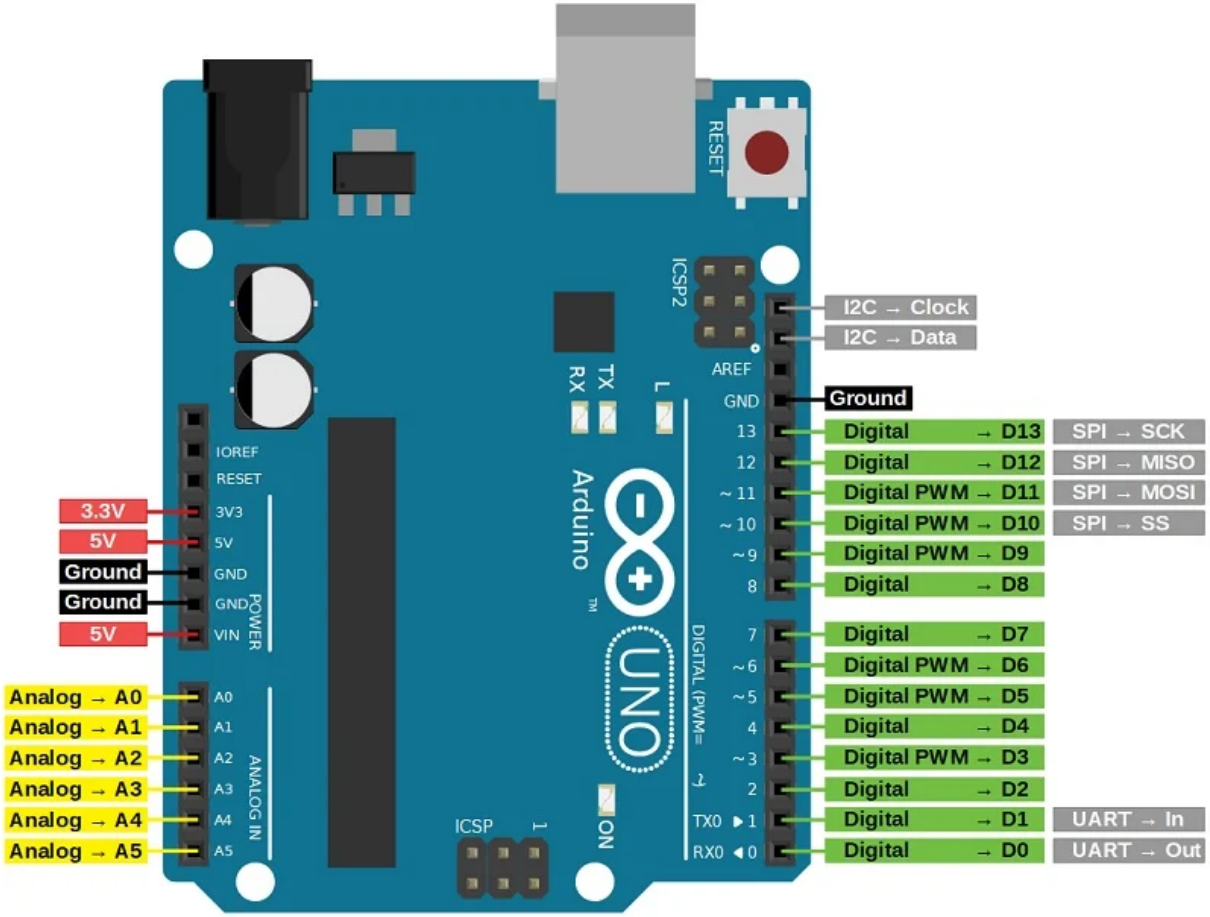
Arduino Motor Shield

- Allows you to easily control motor direction and speed using an Arduino
- By allowing you to simply address Arduino pins, it makes it very simple to incorporate a motor into your project
- allows you to be able to power a motor with a separate power supply of up to 12v.



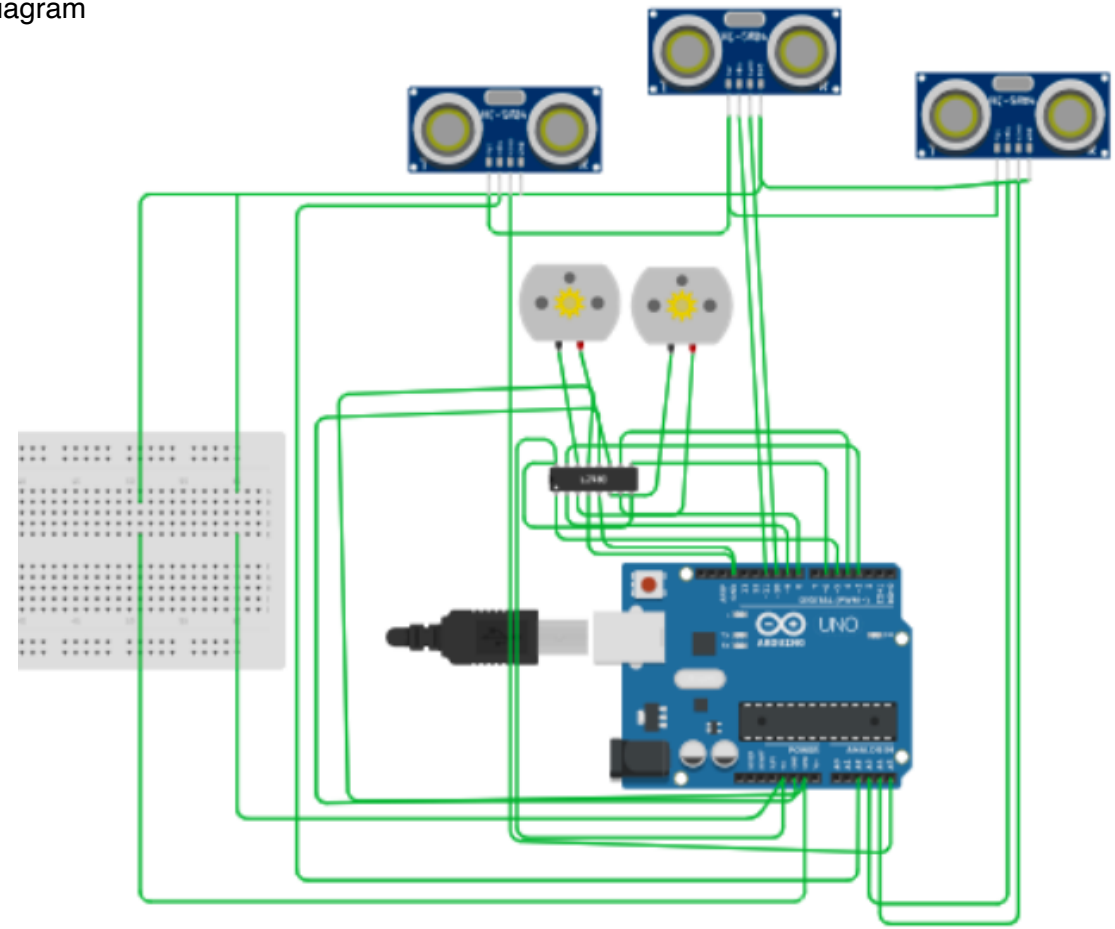
IEEE Robotics & Automation Society

Arduino UNO pin diagram



IEEE Robotics & Automation Society

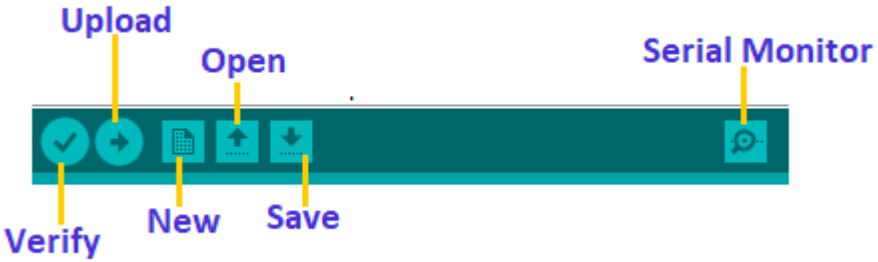
Obstacle detection pin diagram



IEEE Robotics & Automation Society

CODING IN ARDUINO

1. IDE - The Arduino software (IDE) is open-source software.
2. Code online or download it for your PC.



IEEE Robotics & Automation Society

CODING IN ARDUINO

Setup contains an initial part of the code to be executed. The pin modes, libraries, variables, etc., are initialized in the setup section. It is executed only once during the uploading of the program and after reset or power up of the Arduino board.

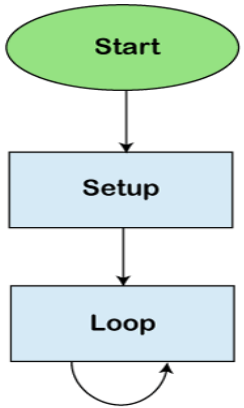
The loop contains statements that are **executed repeatedly**. The section of code inside the curly brackets is repeated depending on the value of variables.

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

}

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

}
```



IEEE Robotics & Automation Society

CODING IN ARDUINO - SYNTAX

pinMode (pin,value)

Mode: We can set the mode as INPUT or OUTPUT

digitalWrite (pin, value)

digitalWrite (13, HIGH);

High : value as high depending on the board voltage

Low : 0 volt

delay (time)

Time must be in milli-seconds

analogWrite(pin, value)

value: the duty cycle: between 0 (always off) and

255 (always on). Allowed data types: int.

IEEE Robotics & Automation Society

CODING IN ARDUINO - LED BLINK

```
void setup (){
pinMode ( 13, OUTPUT);
}
void loop (){
digitalWrite (13, HIGH);   // led connected to pin 13.
delay (5000);                    // delay 5 sec
digitalWrite (13, LOW);   // turn off led
delay (1000);                                // delay 1 sec
}
```

IEEE Robotics & Automation Society

CODING IN ARDUINO - OBSTACLE AVOIDANCE ROBOT

```
const int trigPin1 = 11; //sends waves

const int echoPin1 = 10; //if receive send1

const int trigPin2 = A3; // Analog 3;

const int echoPin2 = A4;

const int trigPin3 = A2;

const int echoPin3 = A5;

const int in1 = 9; //lm1

const int in2 = 8; //lm2

const int in3 = 4; //rm1

const int in4 = 3; //rm2

const int enA = 5;

const int enB = 6;

#define PWM 200 //pulseWidthMod

#define DIS 25 //threshold dist
```

IEEE Robotics & Automation Society

CODING IN ARDUINO - OBSTACLE AVOIDANCE ROBOT

```
void setup()
{
 pinMode(trigPin1, OUTPUT);
 pinMode(echoPin1, INPUT);
 pinMode(trigPin2, OUTPUT);
 pinMode(echoPin2, INPUT);
 pinMode(trigPin3, OUTPUT);
 pinMode(echoPin3, INPUT);
 pinMode (in1, OUTPUT);
 pinMode (in2, OUTPUT);
 pinMode (in3, OUTPUT);
 pinMode (in4, OUTPUT);
 pinMode (enA, OUTPUT);
 pinMode (enB, OUTPUT);
}
```

IEEE Robotics & Automation Society

CODING IN ARDUINO - OBSTACLE AVOIDANCE ROBOT

```
void loop() {

if ( FrontSensor() < DIS && RightSensor () <DIS && LeftSensor ()<DIS)    // obstacle in front of all 3 sides
{

      turn_right ();

      delay(3000);

}

else if (FrontSensor() <DIS && RightSensor () <DIS && LeftSensor ()>DIS)  // obstacle on right and front sides

{

turn_left ();

delay(3000);



}
```

IEEE Robotics & Automation Society

CODING IN ARDUINO - OBSTACLE AVOIDANCE ROBOT

```
else if (FrontSensor() <DIS && RightSensor () >DIS && LeftSensor() < DIS) // obstacle on left and front sides {

turn_right ();

delay(3000);

}

else if (FrontSensor() <DIS && RightSensor () >DIS && LeftSensor ()>DIS) //obstacle on front sides

{

turn_right ();

delay(3000);}

else if (FrontSensor() >DIS && RightSensor () >DIS && LeftSensor ()<DIS)  // obstacle on left sides

{

turn_right ();
delay(3000);
}
```

IEEE Robotics & Automation Society

CODING IN ARDUINO - OBSTACLE AVOIDANCE ROBOT

```
else if (FrontSensor() >DIS && RightSensor () <DIS && LeftSensor()>DIS) // obstacle on right sides
 {
turn_left ();
delay(3000);
}
else{
 forward();
}
}
```

IEEE Robotics & Automation Society

CODING IN ARDUINO - OBSTACLE AVOIDANCE ROBOT

```
void forward (){
 digitalWrite(in1, HIGH);  //dir1
 digitalWrite(in2, LOW);  //dir2
 digitalWrite(in3, HIGH);
 digitalWrite(in4, LOW);
 analogWrite(enA, PWM); //speed
 analogWrite(enB, PWM);
}
void turn_left (){
 digitalWrite(in1, HIGH);
 digitalWrite(in2, LOW);
 digitalWrite(in3, LOW);
 digitalWrite(in4, HIGH);
 analogWrite(enA, PWM);
 analogWrite(enB, PWM);
}
```

```
void turn_right ()
{
 digitalWrite(in1, LOW);
 digitalWrite(in2, HIGH);
 digitalWrite(in3, HIGH);
 digitalWrite(in4, LOW);
 analogWrite(enA, PWM);
 analogWrite(enB, PWM);

}
```

IEEE Robotics & Automation Society

CODING IN ARDUINO - OBSTACLE AVOIDANCE ROBOT

```
long FrontSensor ()
{
long dur;
digitalWrite(trigPin1, LOW);                    // dont send signal
delayMicroseconds(5);
digitalWrite(trigPin1, HIGH);                   // send UltraSonic signal
delayMicroseconds(10);
digitalWrite(trigPin1, LOW);                    // stop sending US
dur = pulseIn(echoPin1, HIGH);                  // returns duration
return (dur/60);                                // convert the time to dist
}

//speed in air: 343m/s => 1cm in 30 sec.
```

IEEE Robotics & Automation Society

CODING IN ARDUINO - OBSTACLE AVOIDANCE ROBOT

```
long RightSensor ()
{
long dur;

digitalWrite(trigPin2, LOW);
delayMicroseconds(5);

digitalWrite(trigPin2, HIGH);
delayMicroseconds(10);

digitalWrite(trigPin2, LOW);
dur = pulseIn(echoPin2, HIGH);

return (dur/60);

}
```

IEEE Robotics & Automation Society

CODING IN ARDUINO - OBSTACLE AVOIDANCE ROBOT

```
long LeftSensor ()
{
long dur;
digitalWrite(trigPin3, LOW);
delayMicroseconds(5);
digitalWrite(trigPin3, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin3, LOW);
dur = pulseIn(echoPin3, HIGH);
return (dur/60);                    // convert the distance to centimeters.

}
```

IEEE Robotics & Automation Society

Explanation for movement

1. digitalWrite(in1, HIGH);: This sets the in1 pin to HIGH, which means the corresponding terminal of the motor driver is connected to the positive terminal of the battery.
2. digitalWrite(in2, LOW);: This sets the in2 pin to LOW, which means the corresponding terminal of the motor driver is connected to the negative terminal of the battery.
3. digitalWrite(in3, HIGH);: This sets the in3 pin to HIGH, which means the corresponding terminal of the second motor driver is connected to the positive terminal of the battery. This causes the second motor to rotate in the same direction as the first motor.
4. digitalWrite(in4, LOW);: This sets the in4 pin to LOW, which means the corresponding terminal of the second motor driver is connected to the negative terminal of the battery. This completes the circuit and allows the second motor to rotate in the same direction as the first motor.
5. analogWrite(enA, PWM);: This sets the enA pin to a specific pulse width modulation (PWM) value, which controls the speed of the first motor. The higher the PWM value, the faster the motor will spin.
6. analogWrite(enB, PWM);: This sets the enB pin to the same PWM value as enA, which controls the speed of the second motor.

Time For Some Hands on Experience

LED + UltraSonic code : https://github.com/Shivaram35144/IEEERAS

Connections : https://www.tinkercad.com/things/18js3FMGefD

Connections for Obstacle Avoidance : https://www.tinkercad.com/things/cFO8Mf9D4Ix

IEEE Robotics & Automation Society

Time For Demonstration

THANK YOU !

IEEE Robotics & Automation Society