# ANDROCAR

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 UNO

- Standard USB for data and power and programming.
- Power Input connector. o female headers.
- 14 digital I/O ports (of which 6 PWM).
- 6 analog input ports.
- 1 hardware serial port (UART).
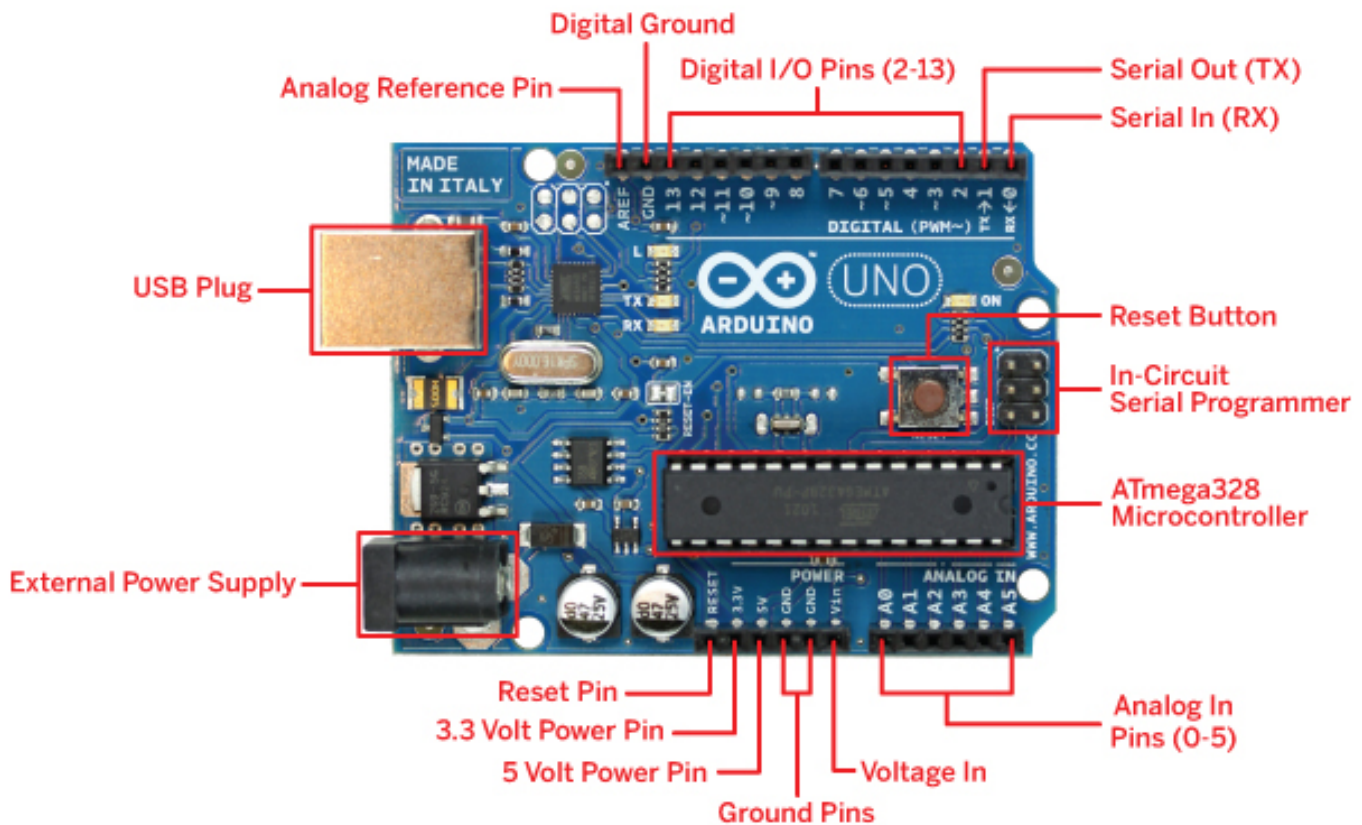
L298N Dual H-Bridge Motor Controller

It is used to drive the motor, Arduino can itself be used to drive forward the motor without L298n but the current supplied is only around 20ma by the Arduino board but the DC motor requires more current for its operation, so we use motor driver IC, which can withstand and supply current as demanded by the DC motor.
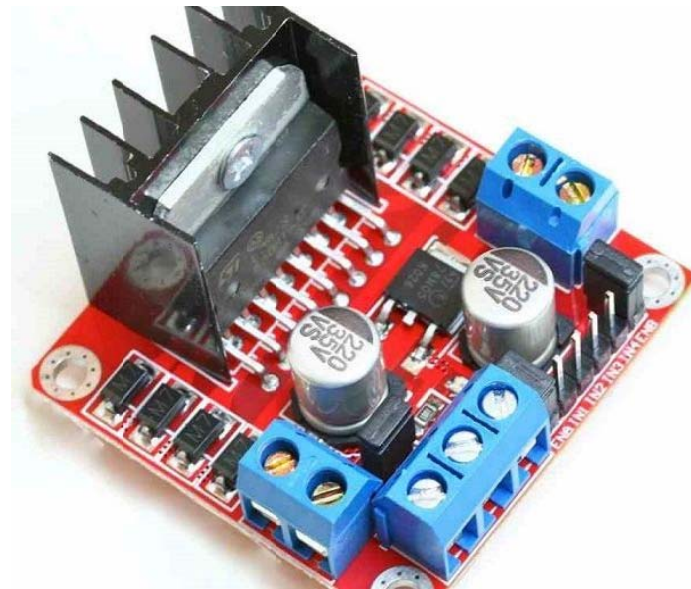
Pins:

- **Out 1: Motor A lead out**
- **Out 2: Motor A lead out**
- **Out 3: Motor B lead out**
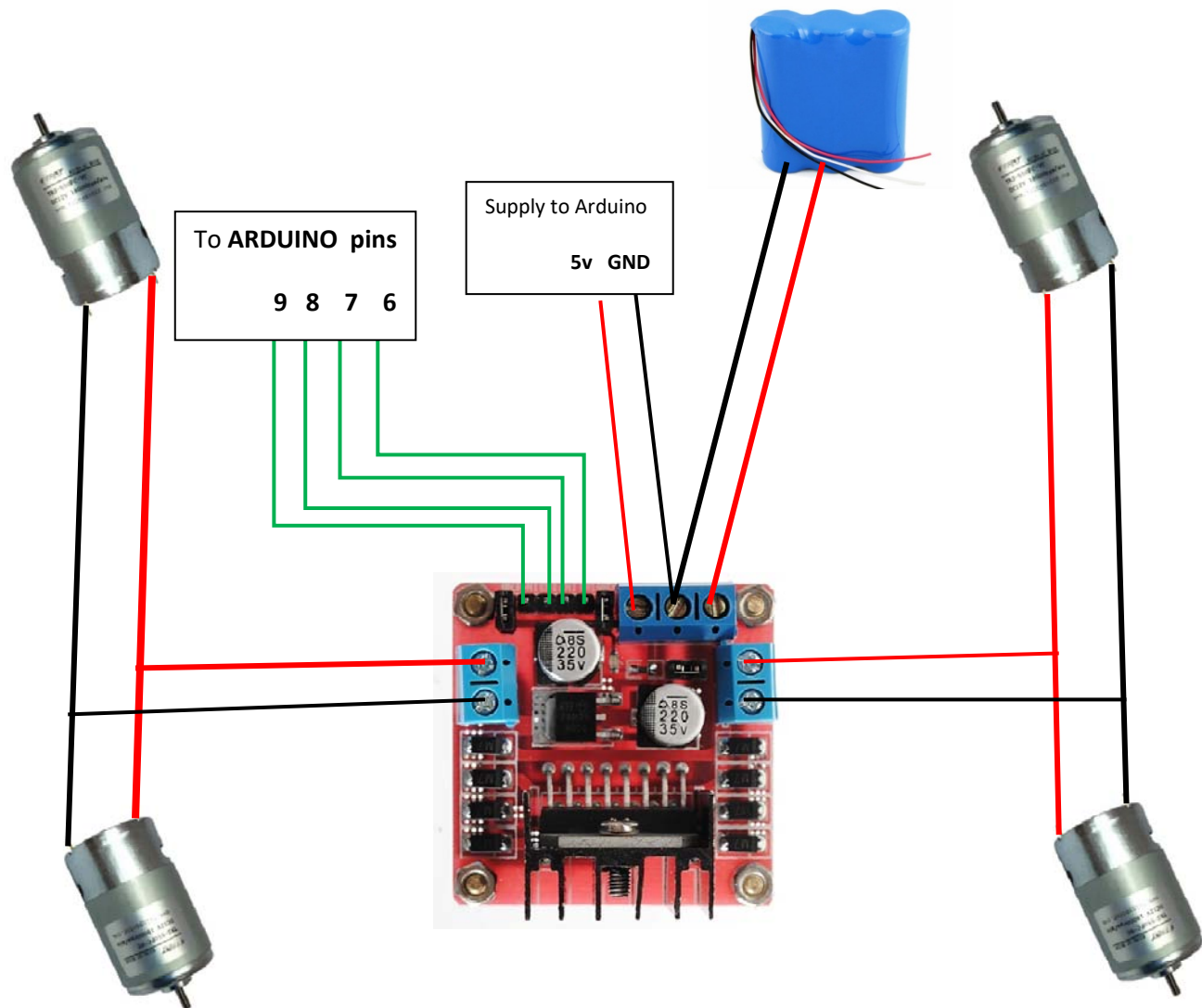- **Out 4: Motor B lead out**
- **GND: Ground**
- **In1: Enable Motor A**

- **In2: Enable Motor A**
- **In3: Enable Motor B**
- **In4: Enable Motor B**

Motor A

Input		Function
EA=H	IN1=H, IN2=L	Forward
	IN1=L, IN2=H	Reverse
	IN1 = IN2	Fast Motor Stop
EA=L	IN1=X, IN2=X	Free Running Motor Stop

L = Low   H = High   X = Don't Care

Same goes for motor B also

BLUETOOTH(HC-05)

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm.

Hardware Features

Typical -80dBm sensitivity

Up to +4dBm RF transmit power

Low Power 1.8V Operation ,1.8 to 3.6V I/O

PIO control

UART interface with programmable baud rate

With integrated antenna

With edge connector

Software Features

Default Baud rate: 38400, Data bits: 8, Stop bit:1,Parity:No parity, Data control: has.

Supported baud rate: 9600, 19200, 38400, 57600, 115200, 230400, 460800.

Given a rising pulse in PIO0, device will be disconnected.

Status instruction port PIO1: low-disconnected, high-connected;

PIO10 and PIO11 can be connected to red and blue led separately. When master and slave

are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.

Auto-connect to the last device on power as default.

Permit pairing device to connect as default.

Auto-pairing PINCODE:"0000" as default

Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.
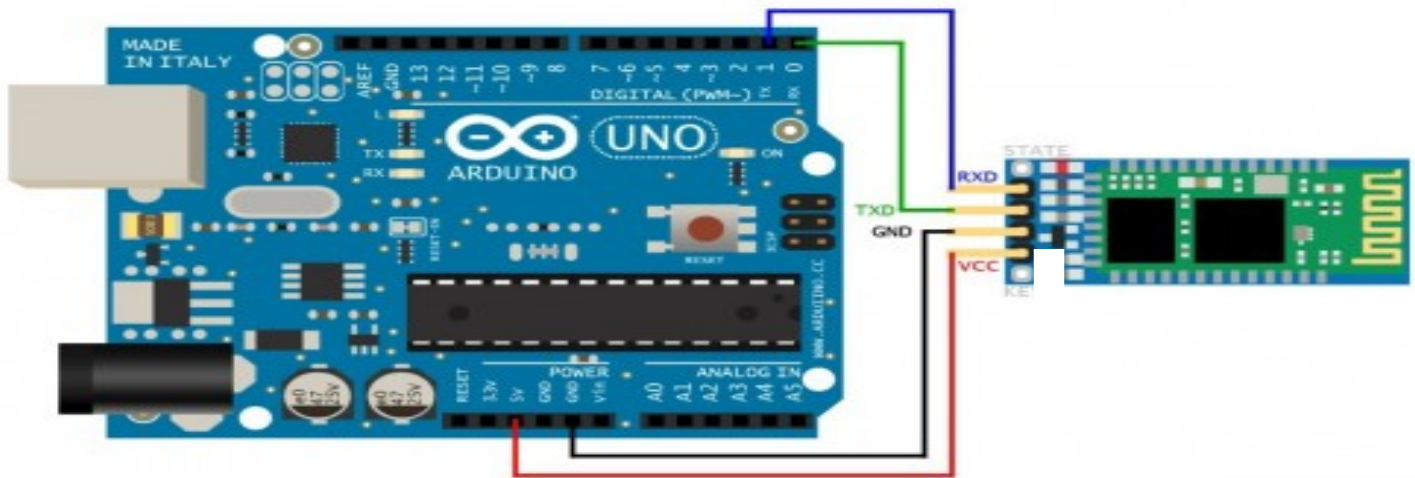
| Pin	Description
State	Can be connected to the Arduino Input in order to know the state of the connection. Paired or disconnected.
Rx	Receive Pin of the module. It is recommended to use a voltage divider as shown in the hook-up.
Tx	Can be connected directly to the Arduino Rx Pin
GND	connected to GND pin of Arduino
5v	This breakout board has a internal 3.3v regulator on board.
EN	Enables or Disables the module. Rarely Used.

ULTRASONIC SENSOR
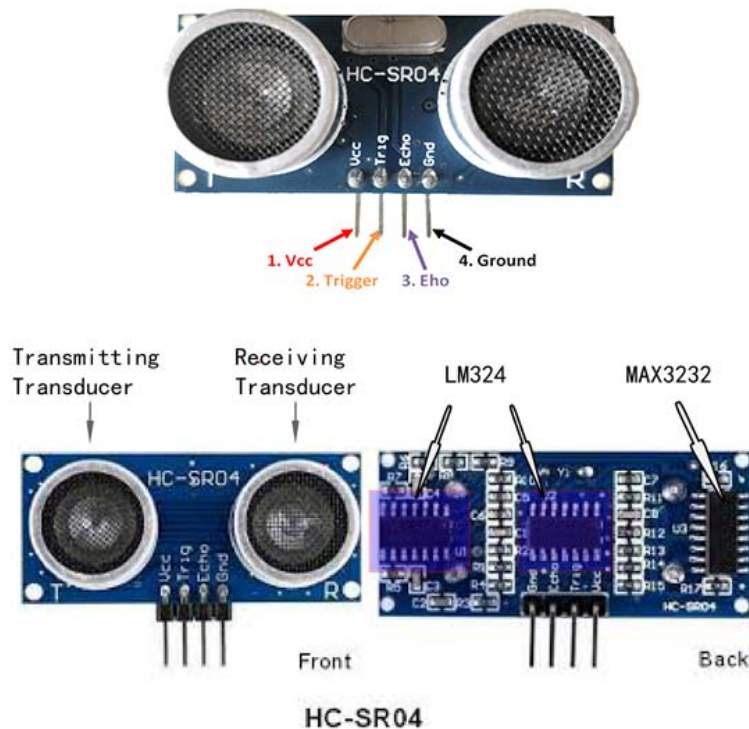
ARDUINO ULTRASONIC CONNECTION
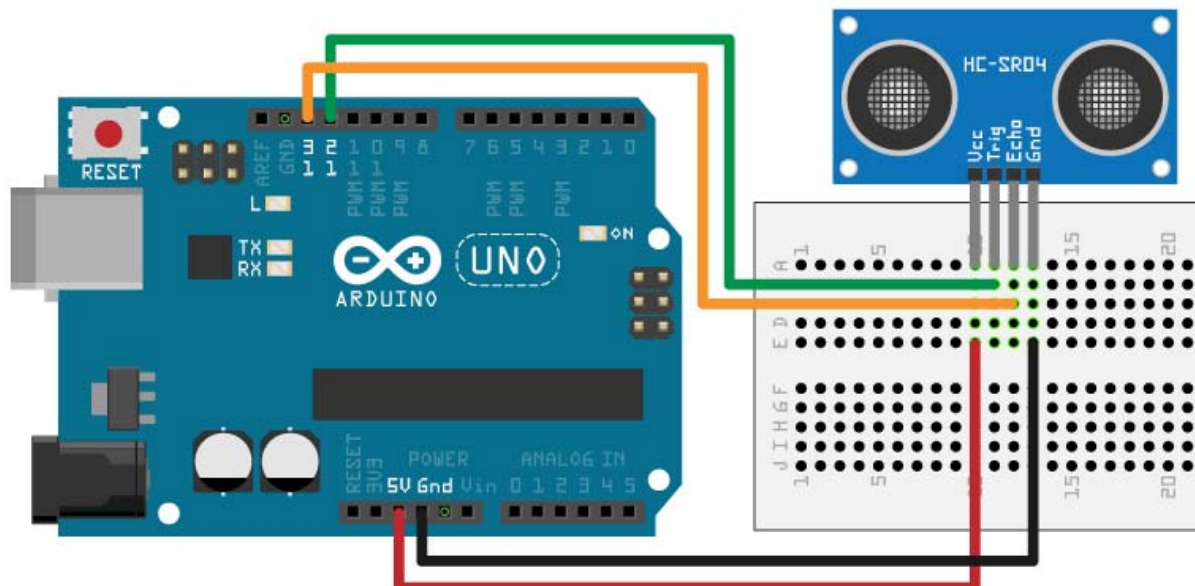
IR sensor

An **infrared sensor** is an electronic device, that emits in order to sense some aspects of the surroundings. An **IR sensor** can measure the heat of an object as well as detects the motion. These types of **sensors** measures only **infrared** radiation, rather than emitting it that is called as a passive **IR sensor**.
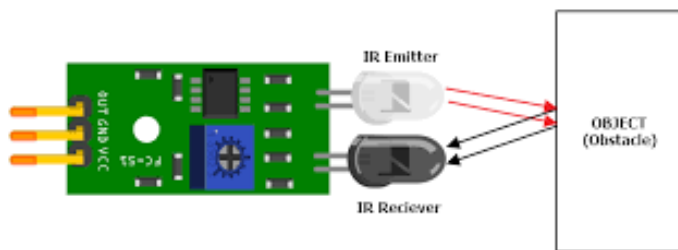
Note: Black surfaces absorbs light naturally. So it will not reflects much light on IR Reciever.
This concept is used in Line follower robot.
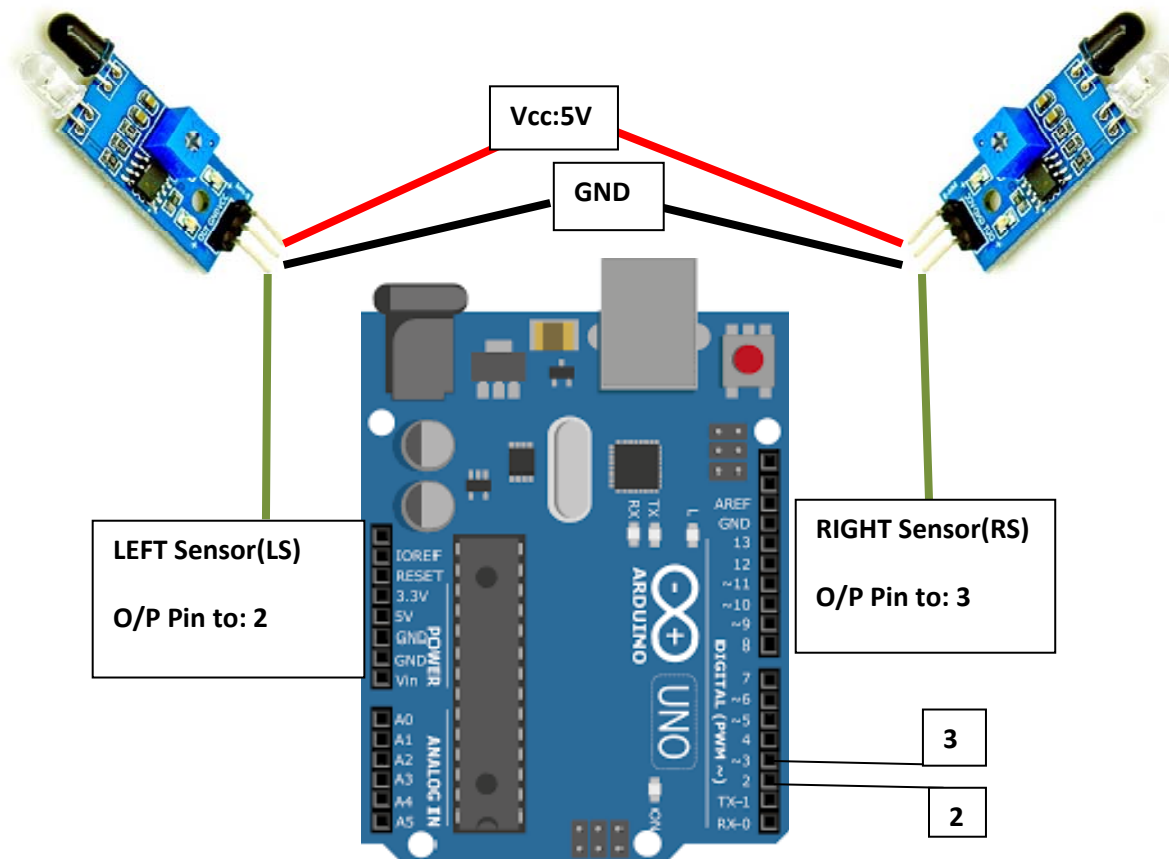
ARDUINO UNO Dip

Digital Pins

It has 14 digital pins. The pins can be configured as an input pin and output pin. **Input** pins accepts **high** when it receives +5V from external supply else **low** when it is at ground state. **Output** pins supply +5V when it is set **high** or **ground** when it is set **low.**

PWM Digital Pins(~)

Pulse Width Modulation or PWM, is a technique for getting analog results with digital pins.

Analog Pins

Analog signal can take any value between 0 to +5V unlike digital signal which can either be +5V or at ground(0V) state. Arduino has 6 analog pins.

The power pins are as follows:

V$_{in}$: A +5V input to the Arduino via this pin can power the Arduino.
USB Jack: It can also power the Arduino and establish a serial communication between Arduino and computer.
Adaptor Jack: It can also power the Arduino. Input voltage through this pin can range between 1 - 13V.
5V: This pin supply +5 to any other device connected.
3.3V: This pin supply +5 to any other device connected.
GND: Ground pins at 0V.
AREF: It is used to change the value of upper limit of analog pin, usually analog pin upper limit is +5V
Serial: 0(RX) is used to receive serial data   1(TX) is used to transmit serial data.

LED(13 pin no.): There is a built-in LED at digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it is off, when pin 13 is declared as output pin.

The Arduino Software includes a serial monitor which allows data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the serial.

Arduino Syntax

pinMode(pin no,INPUT/OUTPUT) Configures the specified pin to behave either as an input or an output pin

digitalWrite(pin no,HIGH/LOW) Write a HIGH or a LOW value to a digital pin. If the pin is high it will supply +5V and if the pin is low it will be at ground state.

delay(milliseconds) Pauses the Arduino for specified time in the brackets.

digitalRead(pin no) Reads the value from a specified digital pin. The result will be either High or Low. If the pin is supplied +5V it will read High, else read LOW.

Serial.begin(9600) Used to begin serial communications, typically at a 9600 baud rate(bits per second)

Serial.print(value) prints data to the serial port, the data printed gets transmitted to serial device connected via Arduino.
Serial.print(78) gives ”78”
Serial.print(1.23456) gives "1.23"
Serial.print("Hello World") gives "HELLO World"
Any data printed gets transferred in string format.

For Creates a loop for repetitive operations.
for (initialization; condition; increment)
                          {   //statement(s);
                                          }

Switch case
Allows you to specify different code that should be executed in various conditions. For example:
switch (var) {
case 1:
//do something when var equals 1
break;

```
case 2:
//do something when var equals 2
break;
default;
//if nothing else matches do default
//default is optional
}
```

Serial.available()
Get the number of bytes (characters) available for reading from the serial port. This is the data that has already arrived and stored in the serial receive buffer.
If Serial.available()>0 some data is present in the serial receive buffer.
If Serial.available()=0 no data is present in the serial receive buffer.

Serial.read()
Reads the data from the serial receive buffer and store it in variable. The data read through this command are the data which are received through serial communication by Arduino.

Arduino Coding

Arduino coding basically consists of two functions:
void setup(): In this function pins are defined to be used either as an input or an output.
void loop(): In this function the main coding is done.
void setup runs only once when the Arduino starts while void loop runs continuously till the Arduino is powered on.

Code for LED ON:
```
void setup()
{
pinMode(13,OUTPUT);
}
void loop()
{digitalWrite(13,HIGH);
}
```

Code for LED Blinking:
```
void setup()
{
```

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