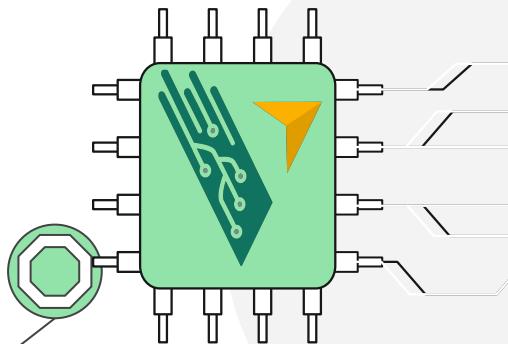
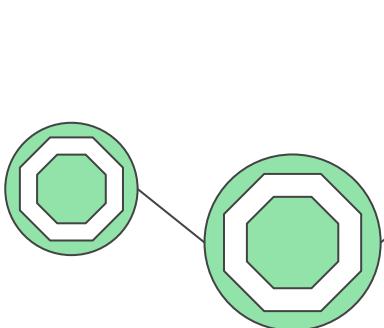


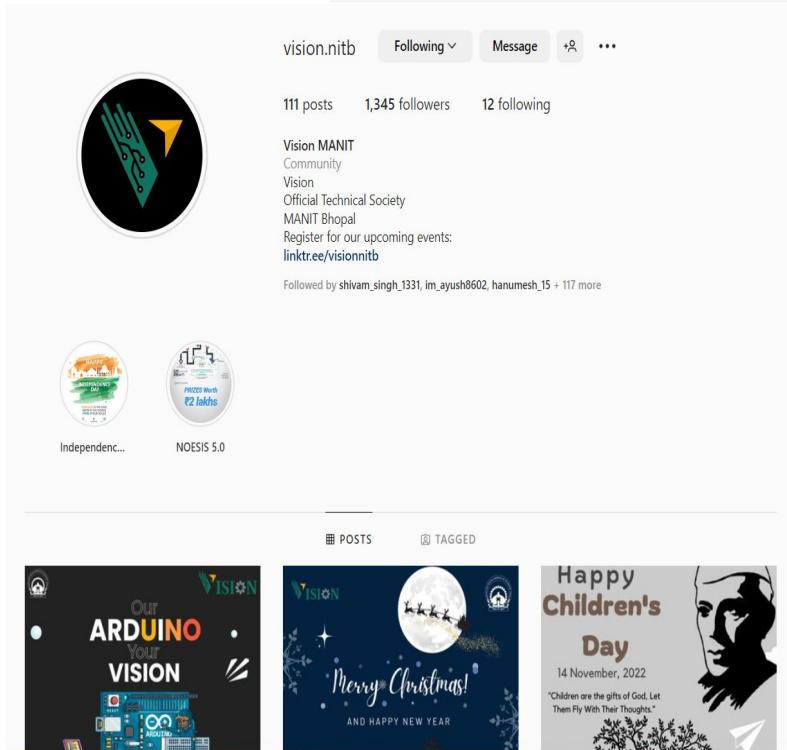
WELCOME

TO A
HORIZON OF
INNOVATIONS
&
LEARNING.



VISION and its **vision**

THE TECHNICAL SOCIETY OF MA-NIT



vision.nitb Following Message +2 ...

111 posts 1,345 followers 12 following

Vision MANIT
Community
Vision
Official Technical Society
MANIT Bhopal
Register for our upcoming events:
linktr.ee/visionnitb

Followed by shivam_singh_1331, im_ayush8602, hanumesh_15 + 117 more

Independenc... NOESIS 5.0

■ POSTS TAGGED

• Our ARDUINO • YOUR VISION • 

Merry Christmas! AND HAPPY NEW YEAR 

Happy Children's Day 14 November, 2022 "Children are the gifts of God. Let Them Fly With Their Thoughts." 



WHAT WE GIVE....



01

NEWER INSIGHTS

BROADENING THE KNOWLEDGE SPECTRUM

02

FUELING INNOVATION

BUILDING IDEAS WITH TECHNOLOGY

03

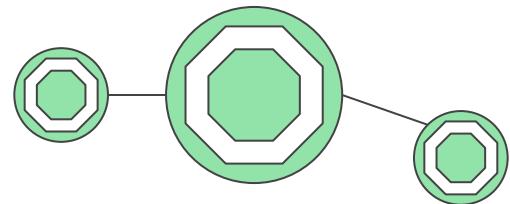
MOULDING MINDS

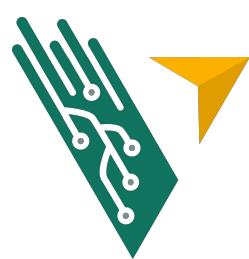
KICKSTARTING CREATIVITY

04

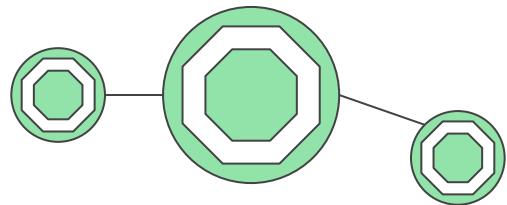
REAL-TIME EXPERIENCE

HANDS-ON EXPOSURE TO HARDWARES

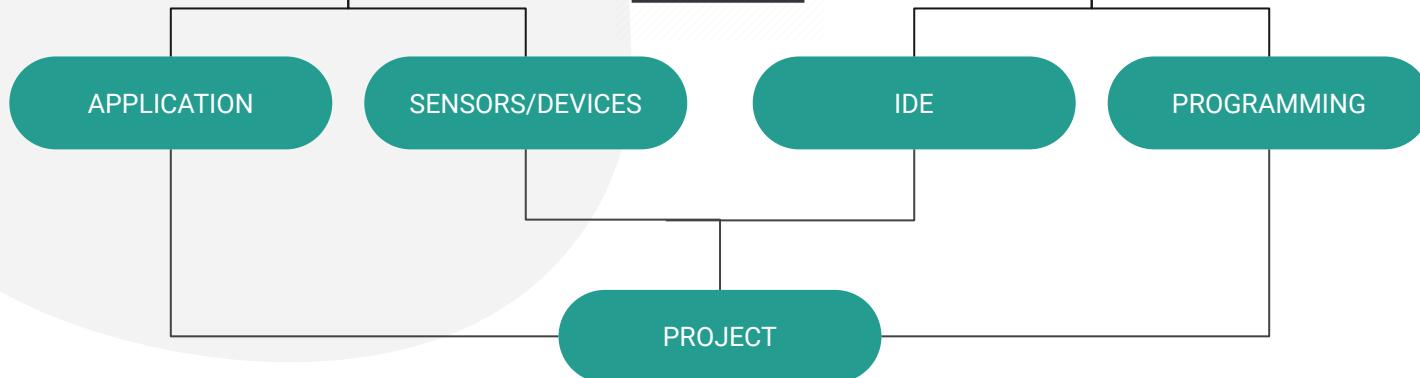
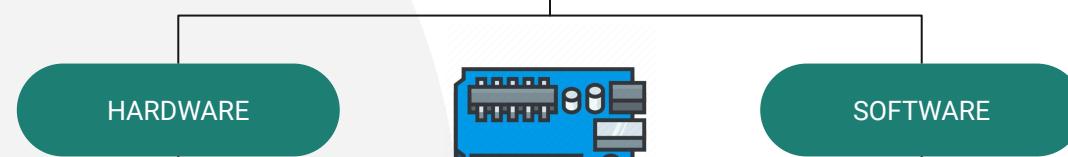


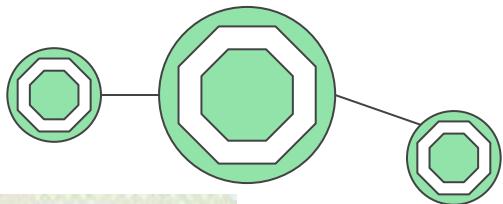


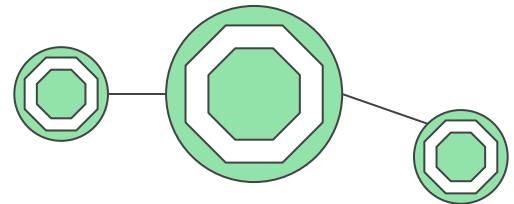
OVERVIEW OF WORKSHOP



ARDUINO





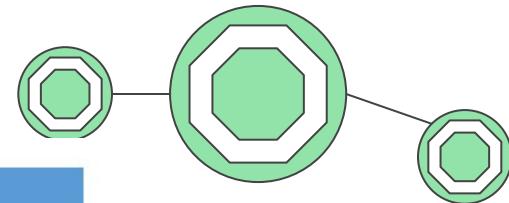


Microcontroller



Microprocessor

Microcontrollers and microprocessors



Summary	Microprocessor	Microcontroller
Applications	Advanced data processing, video, computer vision, personal computers, fast communications, multi-core computation.	Embedded devices, control systems, smartphones, consumer electronics.
Processing Power	Higher	Lower
Memory	External - Flexible	Internal – Limited Size
Power Consumption	Higher	Lower
Size	Larger	Smaller
Price	Expensive	Cheaper
I/O	Need external peripherals with I/O pins	Programmable digital and analog I/O pins

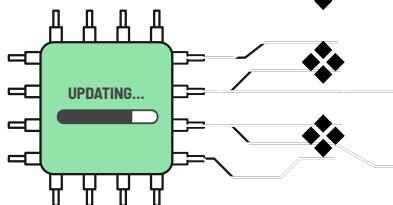
Microcontroller vs Microprocessor



WHAT IS NOT ARDUINO ?

- ❖ It is not only a chip (IC)
- ❖ It is not only a board (PCB)
- ❖ It is not only a programming language
- ❖ It is not only a computer architecture

(although it involves all of these things.....)

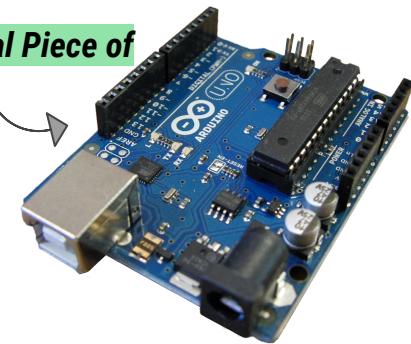


So, What is Arduino ?

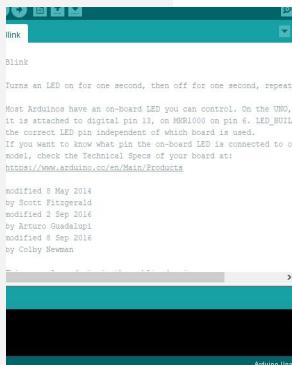
- ★ The Arduino is an open-source electronics platform based on easy-to-use hardware and software used to build electronics projects.
- ★ You can tell your board what to do by sending a set of instructions to the microcontroller on the board.
- ★ All Arduino boards have one thing in common which is a microcontroller.
- ★ An Arduino is based on a microcontroller which is a simple easy to use computer designed for beginners to run 1 program at a time.

ARDUINO means 3 things

1.) A Physical Piece of Hardware



2.) A Programming Environment



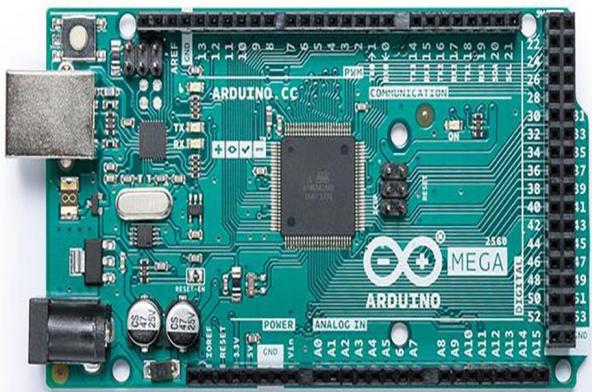
3.) A community and philosophy





VARIANTS OF ARDUINO BOARDS

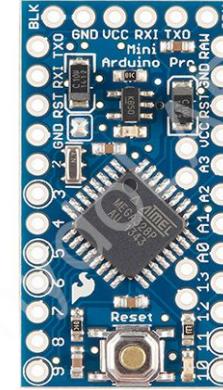
ARDUINO MEGA



ARDUINO UNO



ARDUINO MINI



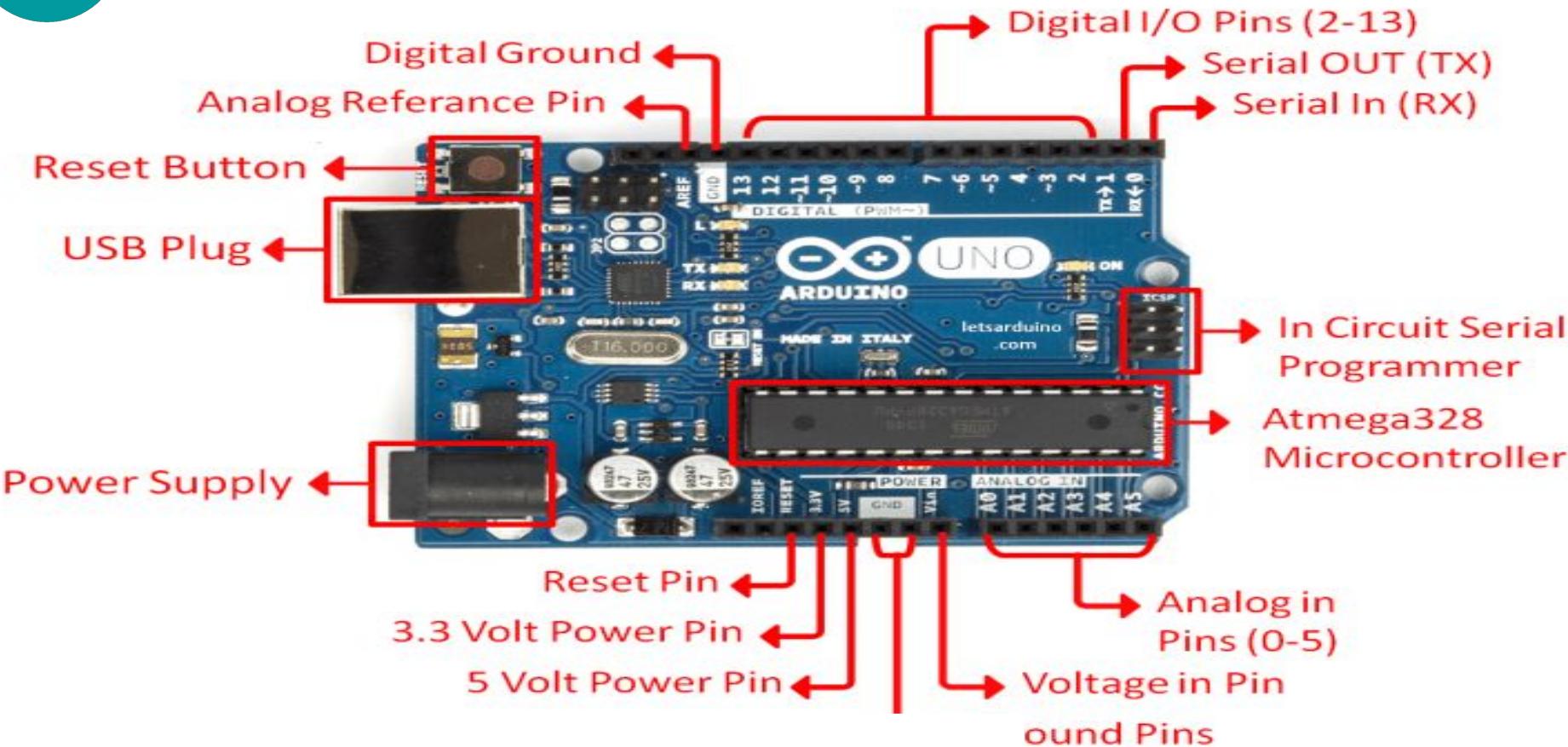
WHY ONLY ARDUINO ?

- For whatever reason, the Arduino microcontrollers have become the de facto standard !
- Strives for the balance between ease of use and usefulness.
 - 1.) Programming Languages seen as a major obstacle .
 - 2.) Arduino C is greatly simplified version of C++.
- It is an open source project, software/hardware is extremely accessible and very flexible to be customized and extended.
- Arduino is Backed up by a growing online community , lots of source code is already available and we can share and post our examples for others to use, too!
- It is inexpensive and comes up with free authoring software.

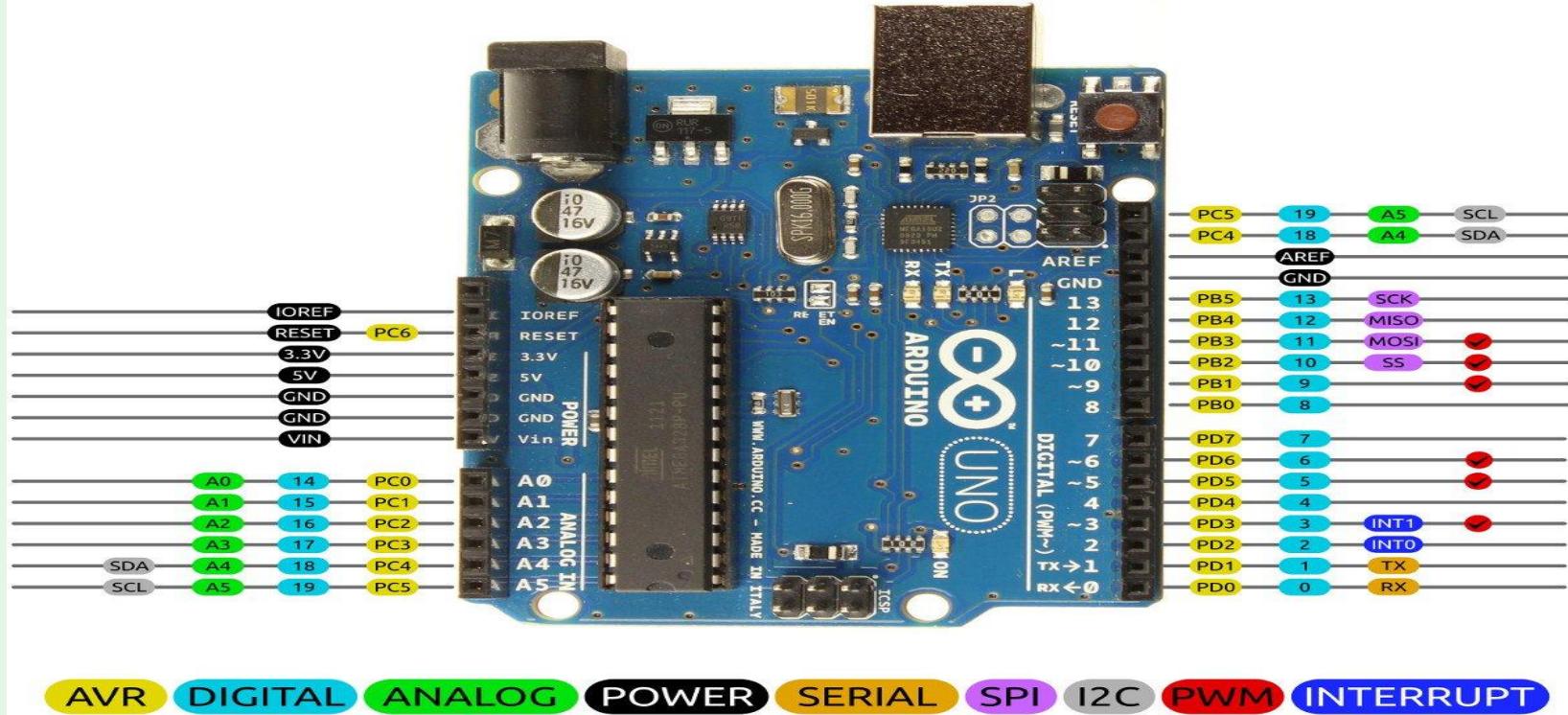




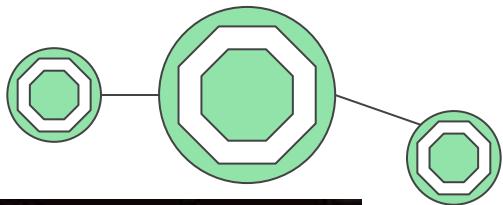
THAT'S ARDUINO UNO R3



Arduino Uno R3 Pinout



2014 by Bouni
Photo by Arduino.cc





BOARD'S DESCRIPTION

- 1. Microcontroller** – The Arduino uno uses ATmega328P as the microcontroller.
- 2. Digital pins** – Arduino uno has 14 digital pins among which 6 pins are pwm pins.
- 3. Analog pins** – There are 6 analog pins(A0-A5). These pins can read the signals from an analog sensor and convert it into a digital value.
- 4. Power pins** – 3.3V, 5V,GND,Vin these are the power pins.
- 5. PWM pins(pulse width modulation)** -These pins generate analog as well as digital output.
- 6. Power port** – Arduino uno can be powered up via power port using a battery or an AC to DC adaptor.
- 7. Crystal Oscillator** - It helps arduino in dealing with time issues.
- 8. USB port** - It helps in data transfer and power supply.
- 9. Voltage regulator.**
- 10. TX/RX LEDs.**
- 11. Reset button.**



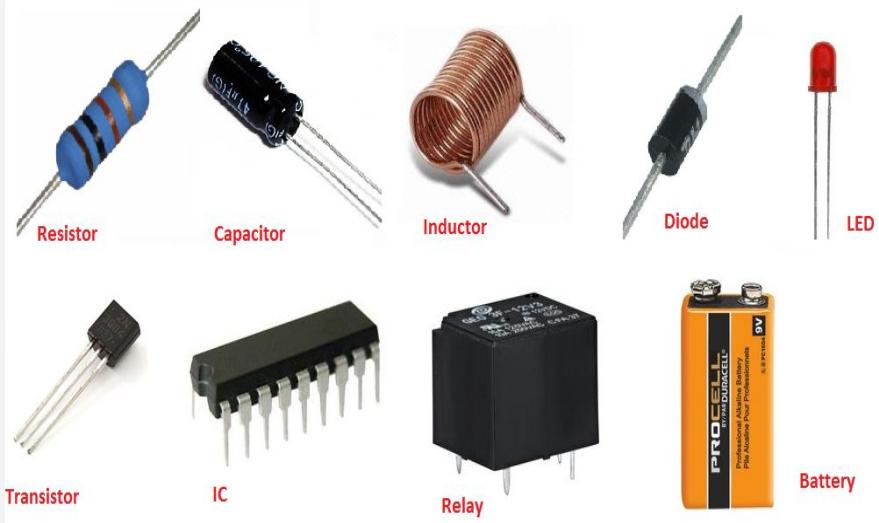
UNO SPECIFICATION'S

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limit): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz



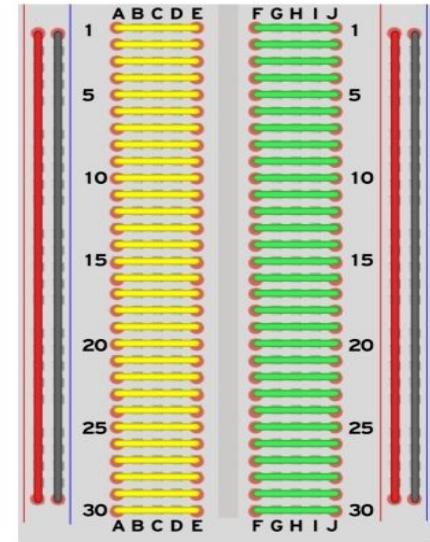
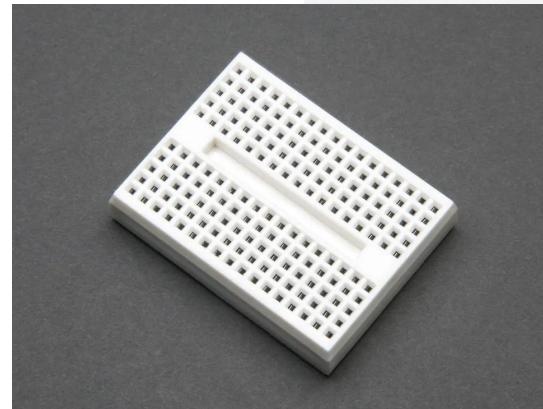
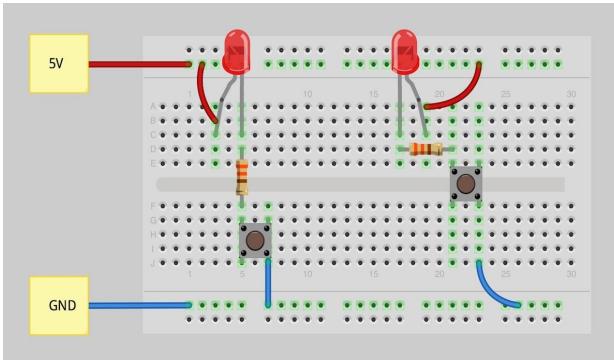
BASIC ELECTRONIC TOOLS

- ❖ BREADBOARD
- ❖ JUMPER WIRE
- ❖ RESISTOR
- ❖ LED
- ❖ POTENTIOMETER



BREADBOARD

- ❑ A breadboard ,sometimes called a proto-board, is a reusable platform for temporarily built electronic circuits.
- ❑ A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).



JUMPER WIRES

- ❑ A **jumper wire** is an **electrical wire** in a cable, with a connector at each end which is used to interconnect the components of a **breadboard** internally or with other components, without soldering.
- ❑ Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard.



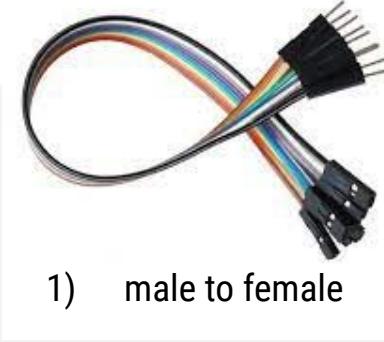
continued.....

JUMPER WIRES

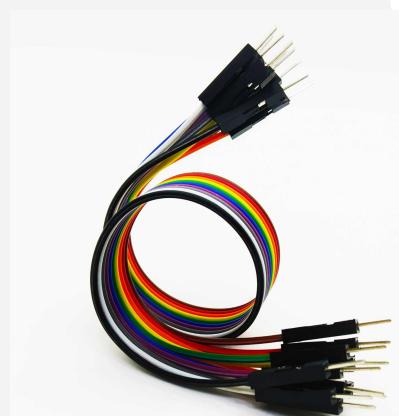
- ❑ Jumper wires typically come in three versions:
 - 1) male-to-male
 - 2) male-to-female
 - 3) female-to-female
- ❑ The difference between each is in the end point of the wire. **Male ends have a pin protruding** and can plug into things, while female ends do not and are used to plug th



3)female-to-female



1) male to female

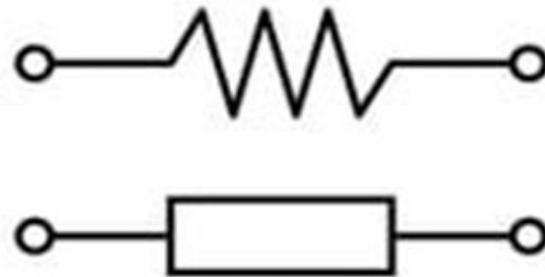
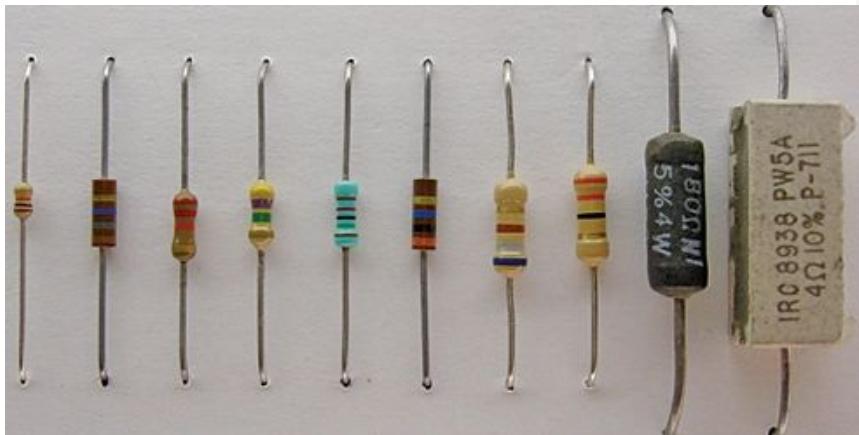


2) male-to-male



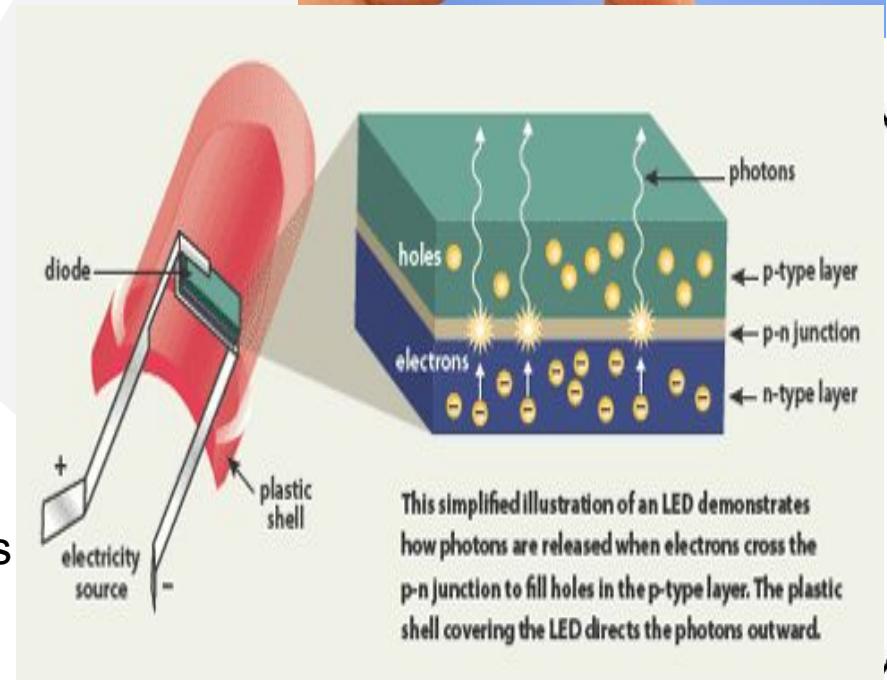
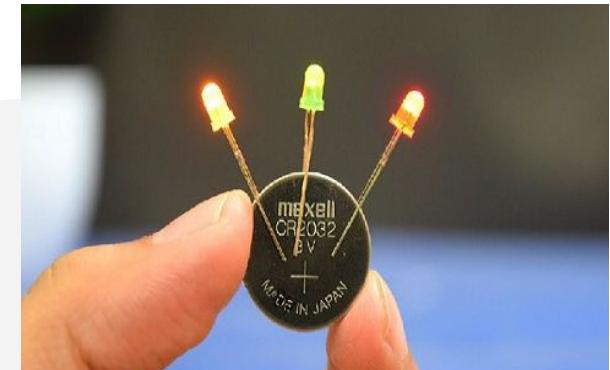
RESISTOR

- ❑ A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element.
- ❑ In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses.



LED

- A light-emitting diode (LED) is a semiconductor device that emits light when current flows through it.
- LEDs work on the principle of **Electroluminescence**. On passing a current through the diode, minority charge carriers and majority charge carriers recombine at the junction. On recombination, energy is released in the form of photons.
- For example – TV remotes , toys , sensors etc....

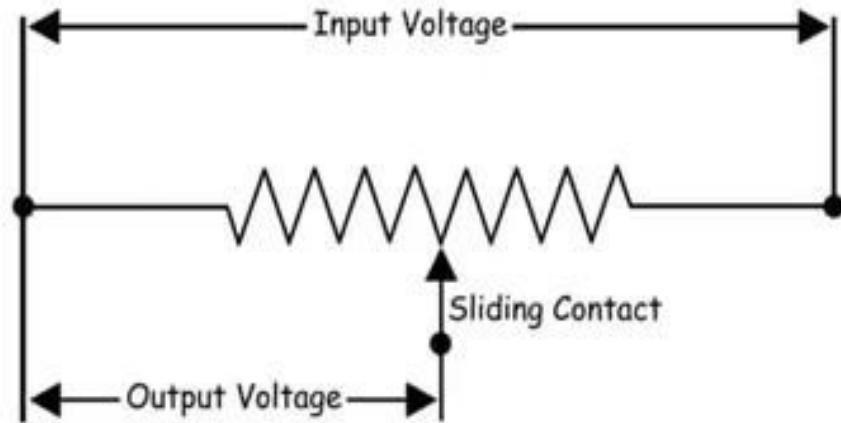
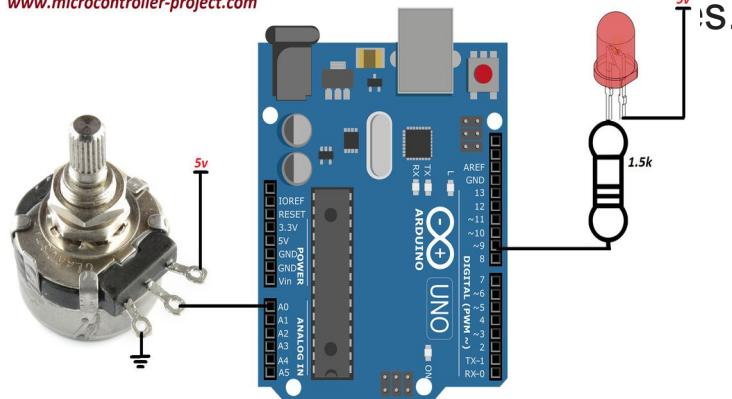


This simplified illustration of an LED demonstrates how photons are released when electrons cross the p-n junction to fill holes in the p-type layer. The plastic shell covering the LED directs the photons outward.

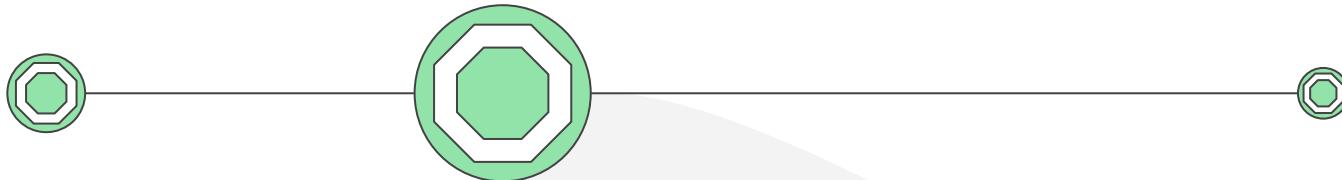
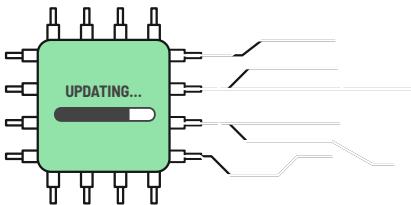
POTENTIOMETER

- ❑ A **potentiometer** is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider
- ❑ Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment.

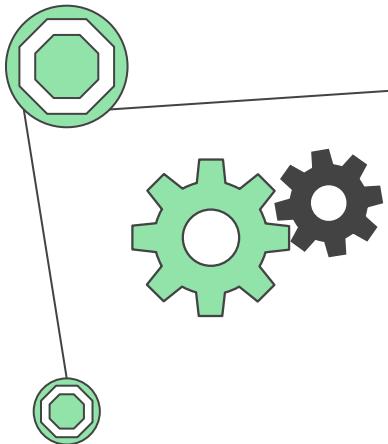
❑ www.microcontroller-project.com



INPUT DEVICES



TYPES OF INPUT DEVICES



**INPUT
DEVICES**

SENSORS



RECEIVER MODULES

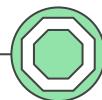


SENSORS

- The sensors are defined as a machine, module, or a device that detect changes in the environment.
- The sensors transfer those changes to the electronic devices in the form of a signal.
- The output signal is easily readable by humans. The sensors are used to measure the physical quantities, such as pressure, temperature, sound, humidity, and light, etc.

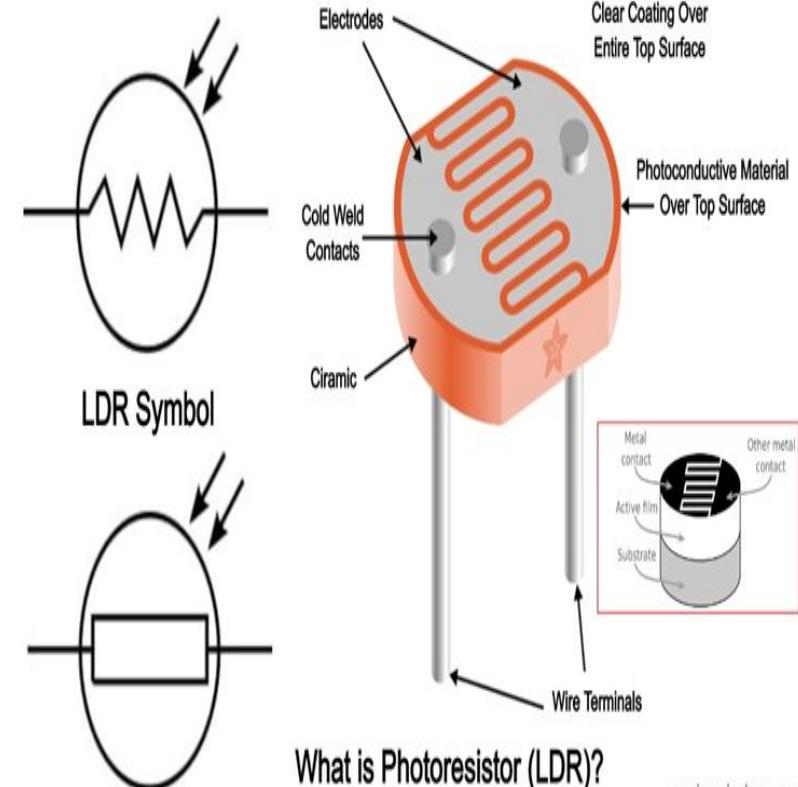
SENSORS

- Infrared (IR) Sensor
- Ultrasonic Sensor (HC-SR04)
- Temperature and Humidity Sensor (DHT11)
- Light Dependent Resistor (LDR)



Light Dependent Resistor (LDR)

1. A Light Dependent Resistor (LDR) is a photocell that works on the principle of photoconductivity.
3. The passive component is basically a resistor whose resistance value decreases when the intensity of light increases.
4. This optoelectronic device is mostly used in light varying sensor circuit, and light and dark activated switching circuits.



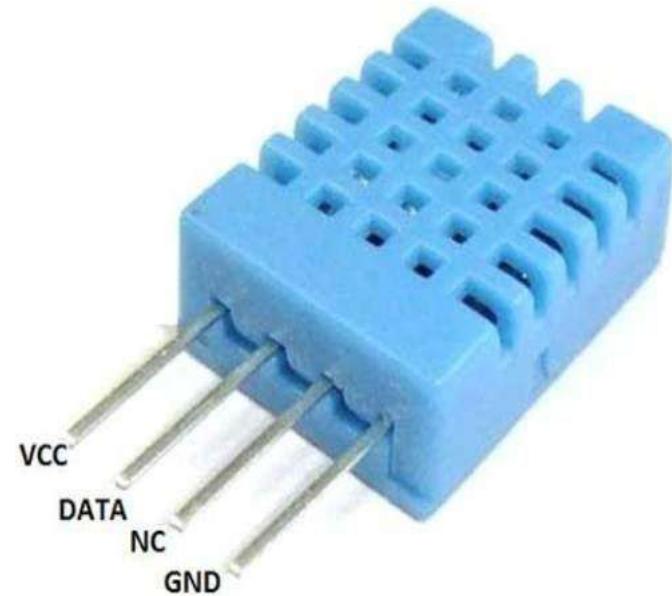
APPLICATIONS OF LDR

- *Automatic Street Light Circuit*
- *Simple Fire Alarm Circuit*
- *Light Activated Switch Circuit*
- *Automatic LED Emergency Light*
- *Night Security Light*



Temperature and Humidity Sensor (DHT11)

1. The DHT11 is a basic digital temperature and humidity sensor.
2. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin.

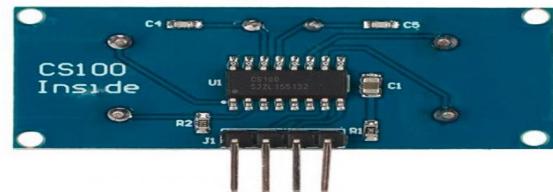


APPLICATIONS OF TEMPERATURE & HUMIDITY SENSOR

- *In designing Temp. and Humidity level-based alarms.*
- *IOT projects based on temperature or humidity levels.*

ULTRASONIC SENSORS (HC-SR04)

- ❑ An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal
- ❑ Ultrasonic sensors have two main components: the transmitter and the receiver (which encounters the sound after it has travelled to and from the target).

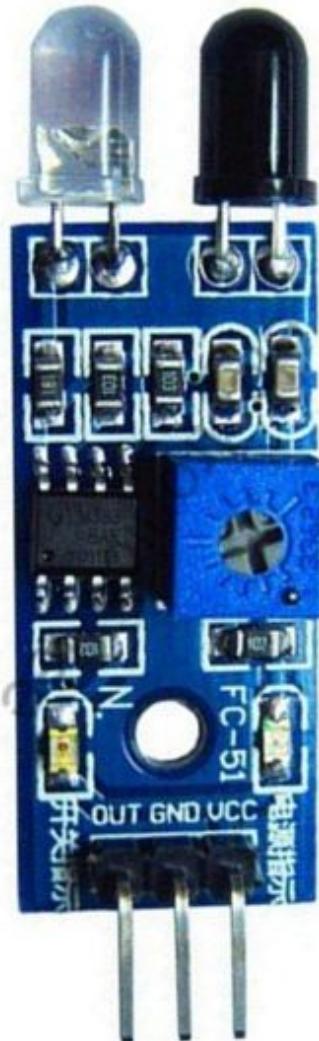


APPLICATIONS OF ULTRASONIC SENSORS

Distance Detection: The ultrasonic sensor is used for the determination of distance between the sensor and obstacle.

The Infrared (IR) Sensor

- ❑ An Infrared light-emitting diode (IR LED) is a special purpose LED emitting infrared rays ranging from 700 nm to 1 mm wavelength
- ❑ The infrared sensor uses two bulbs for detection of obstacles, namely, the transmitter bulb and receiver bulb. The transmitter bulb emits a pulse of infrared wavelength and the receiver gives positive output if it gets back an IR wave.

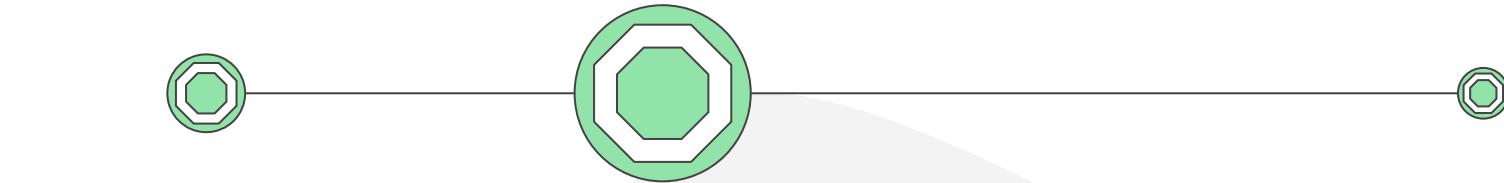


APPLICATIONS OF INFRARED SENSORS

- *Obstacle Detection: The IR sensors can be used to detect obstacles.*
- *Heat Detection: The IR sensors also give a positive output when subjected to heat.*

RECEIVER MODULES

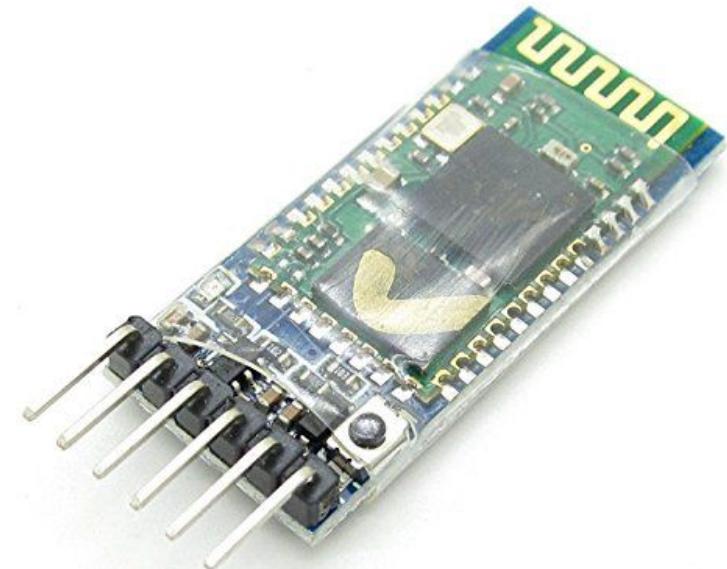
- Bluetooth Receiver Module
- WiFi Receiver Module

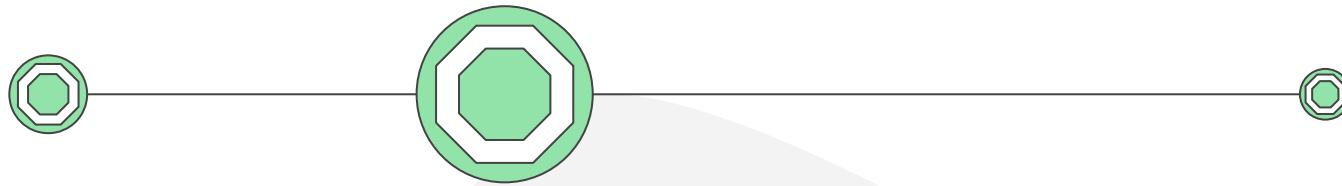


BLUETOOTH RECEIVER MODULE

- The Bluetooth module is a device which is used for short range wireless communication to the respective connected device.

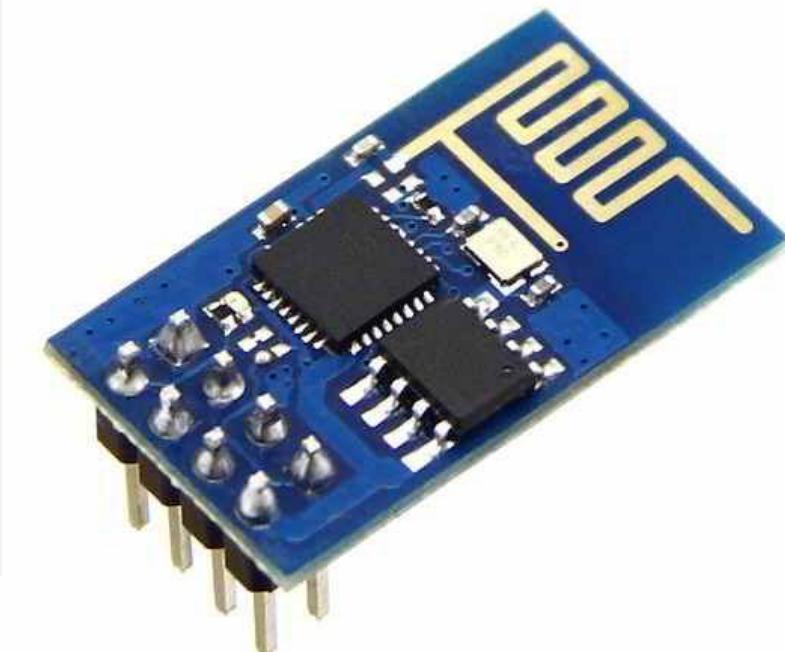
- This module can connect the devices in the range of 10 meters of the Bluetooth and it works on the 2.4GHz frequency band. The Bluetooth module operates on the voltage ranging from 3.3 volts to 5 volts and has 6 pins in total.

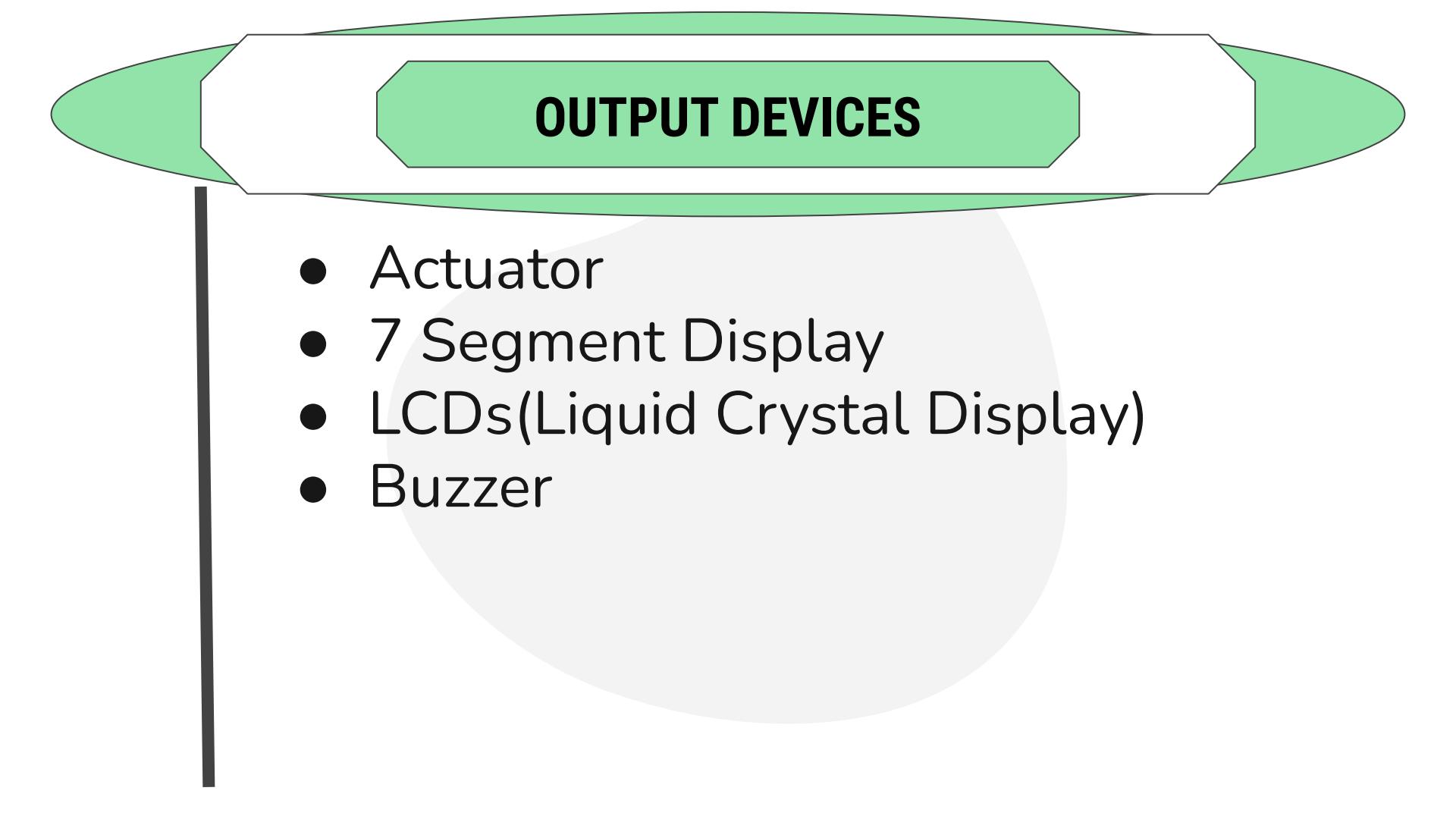




WiFi Receiver Module

- The Wifi module is a complete WiFi network where you can easily connect as a serving WiFi adapter, wireless internet access interface to any microcontroller based design on its simple connectivity.



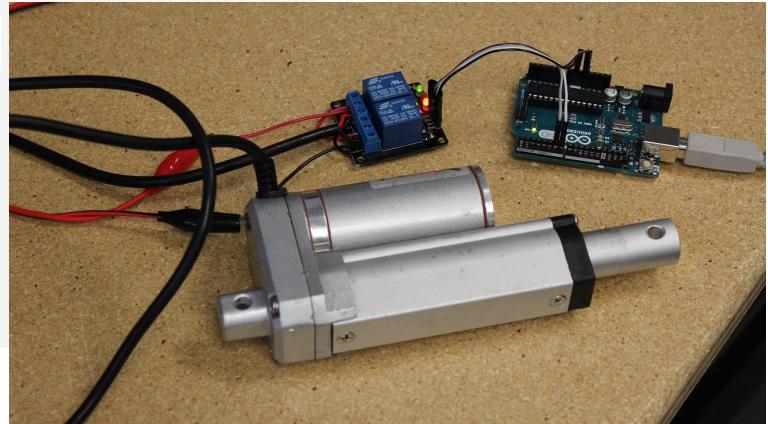


OUTPUT DEVICES

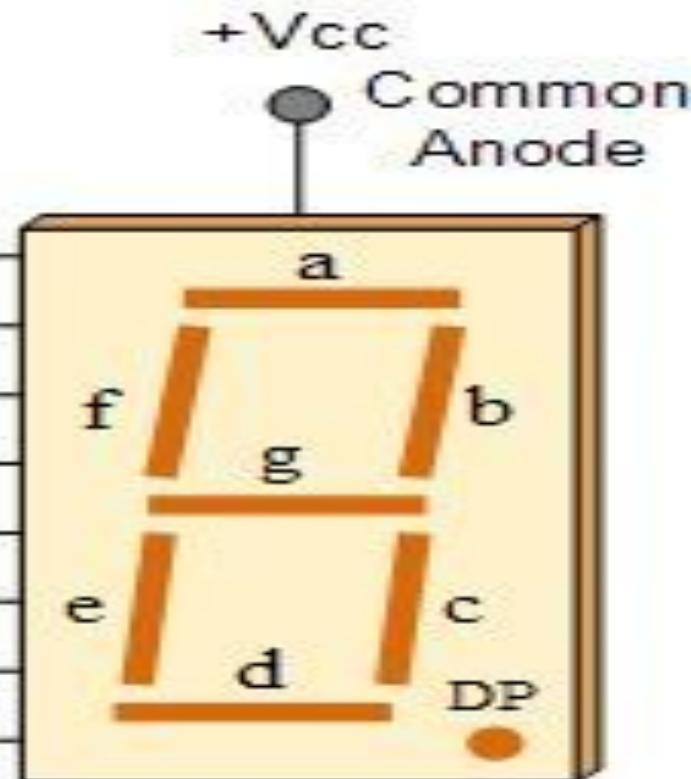
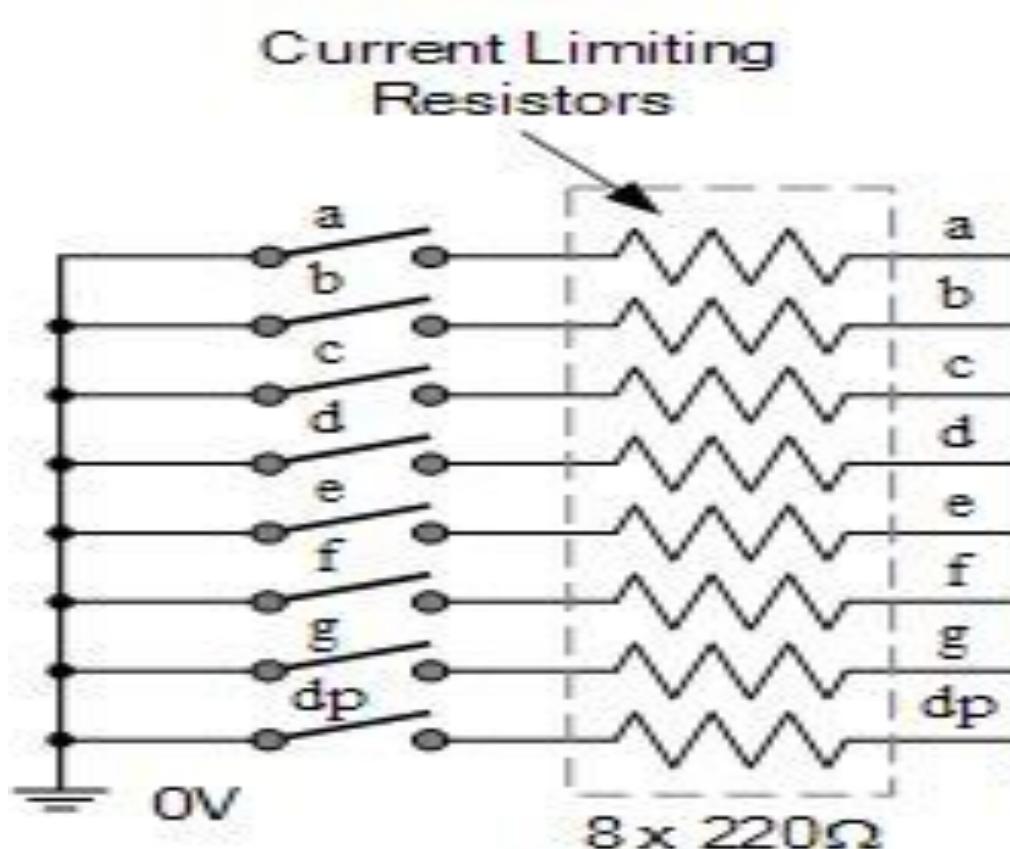
- Actuator
- 7 Segment Display
- LCDs(Liquid Crystal Display)
- Buzzer

ACTUATOR

- ❑ An actuator is a component of a **machine** that is responsible for moving and controlling a mechanism or system, for example by opening a valve
- ❑ What does a ACTUATOR do ?
 - An actuator is a device that produces a motion by converting energy and signals going into the system.
- ❑ The three main types of actuators are :
 - 1)pneumatic (air pressure),
 - 2)hydraulic (fluid pressure)
 - 3)electric.

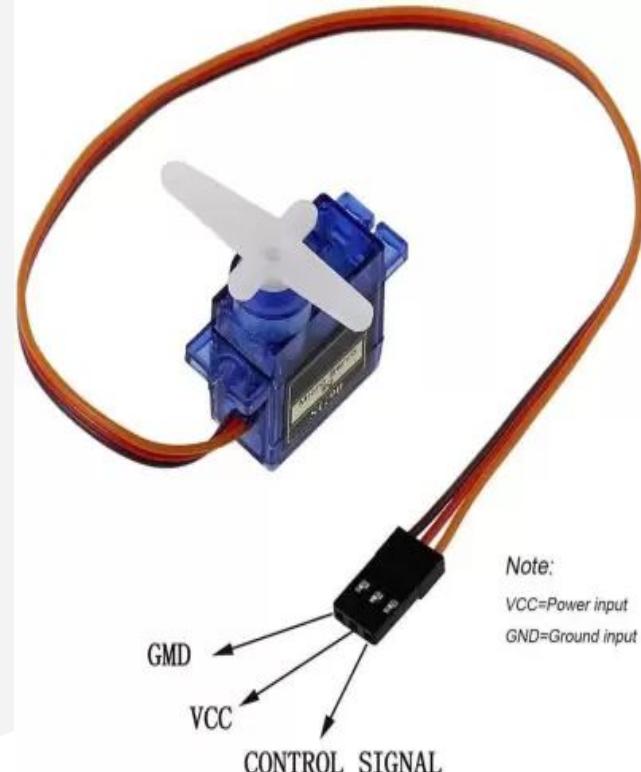


7 SEGMENT DISPLAY



SERVO MOTOR

- ❑ A **servo motor** is a type of motor that can rotate with great precision.
- ❑ If you want to rotate an object at some specific angles or distance, then you use a servo motor
- ❑ The servo motor that is shown is dc servo motor .



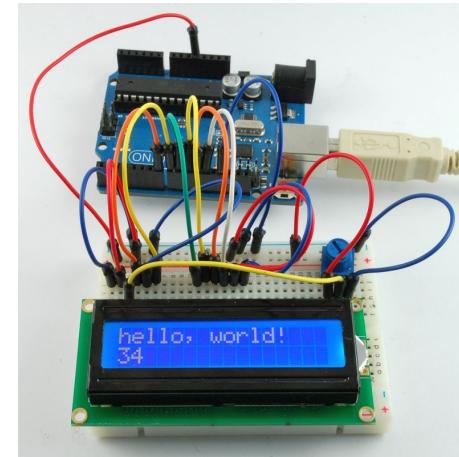
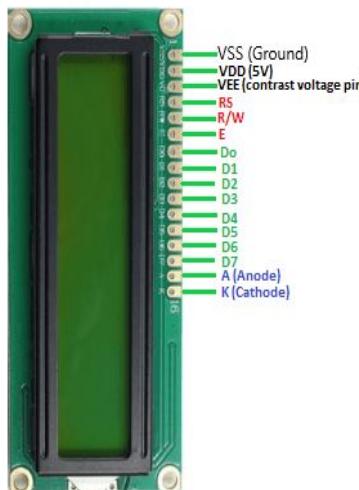
LCDs(Liquid Crystal Display)

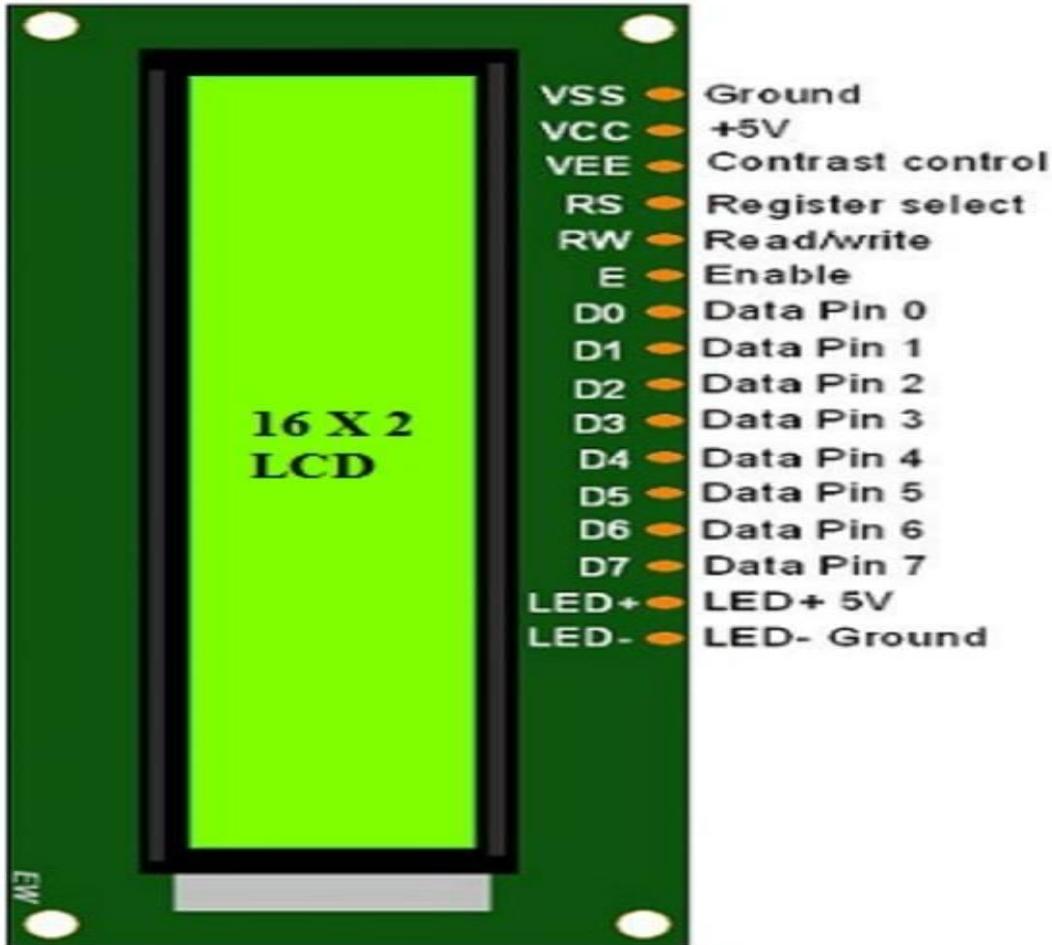
- ❑ LCD (Liquid Crystal Display) is a type of flat panel display which uses liquid crystals in its primary form of operation.
- ❑ The library that allows us to control the **LCD** display is called **Liquid Crystal Library**
 - ❑ LCD Structure

1)The LCD display has a 16-pin interface.

2)The Liquid Crystal Display has a parallel interface. It means that the microcontroller operates several pins at once to control the LCD display.

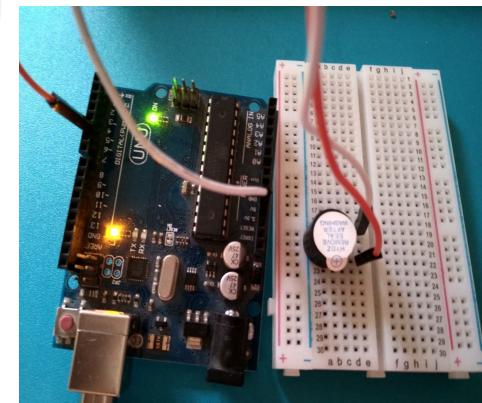
LCD





BUZZER

- ❑ A **buzzer** or **beeper** is an **audio signaling device**
- ❑ How to connect a buzzer to arduino ?
 - Connect the Supply wire (**RED**) of the buzzer to the **Digital Pin 9** of the Arduino through a resistor.
 - Connect the Ground wire (**BLACK**) of the buzzer to any **Ground Pin** on the Arduino





Presents



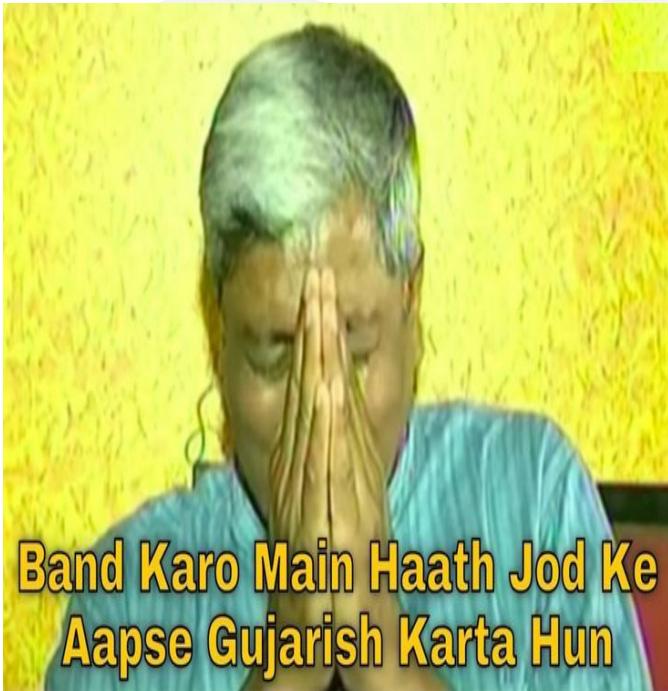
ARDUINO

WORKSHOP

ARDUINO CODE LANGUAGE



ARDUINO KE LIYE AB NAYI LANGUAGE SEEKHNI PADEGI 😢😢😢😢



C LANGUAGE BASICS

The Arduino programming language is based on a very simple hardware programming language called processing, which is similar to the C language.

Hence,



TOPICS TO BE COVERED

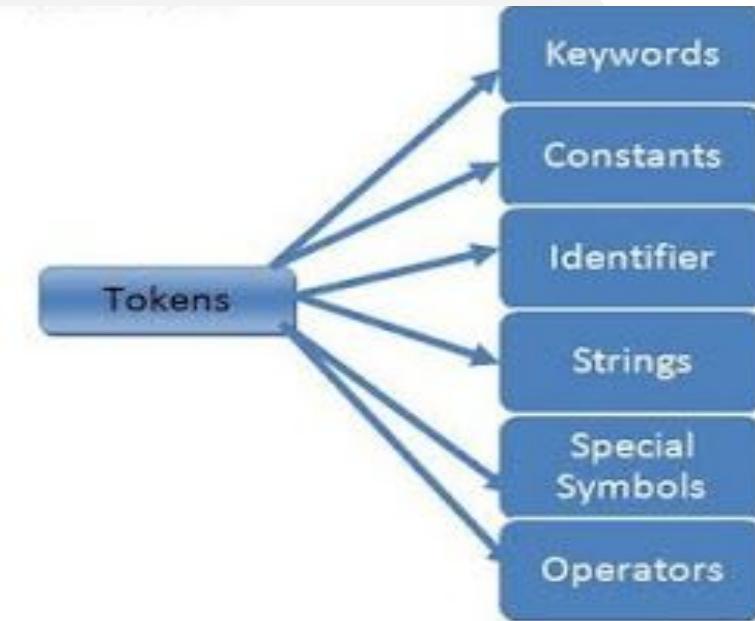
- ❖ TOKENS IN C
- ❖ DATA TYPES & ARRAYS
- ❖ FLOW OF CONTROL & LOOPS
- ❖ FUNCTIONS
- ❖ STRUCTURE OF AN ARDUINO CODE
- ❖ ARDUINO SPECIFIC FUNCTIONS
- ❖ INTRODUCTION TO TINKERCAD

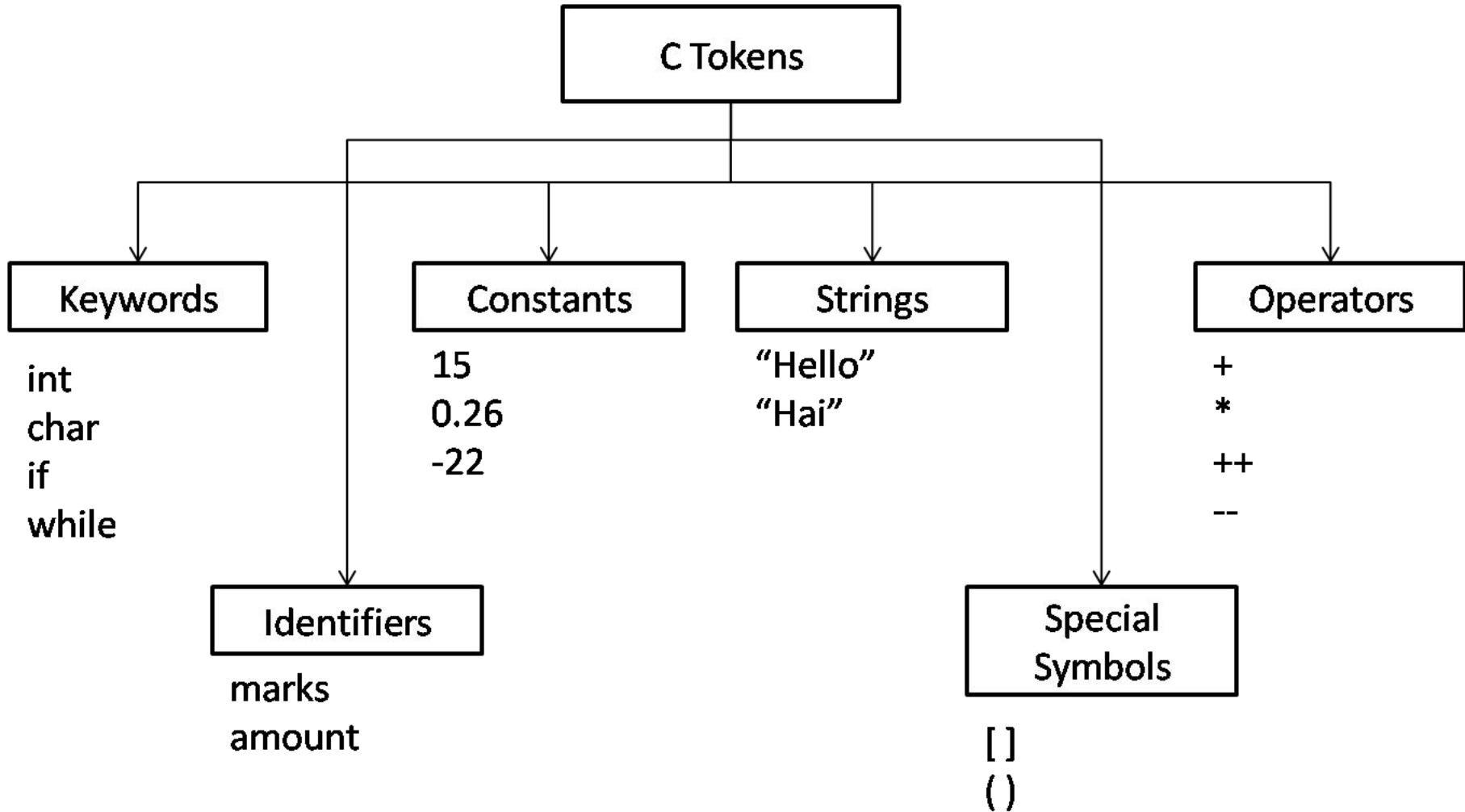


TOKENS

Tokens are the building block of C programming, that means only with the help of tokens a C program can be created.

In C programming, some INDIVIDUAL words and characters are fixed, whose functions are fixed, called tokens.





KEYWORDS

- ◆ Reserve words that carry special meaning to the compiler.
- ◆ keywords can't be used as variables in a program.
- ◆ C language has a total of 32 keywords.
- ◆ Used to perform internal operations.

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

IDENTIFIERS (VARIABLES)

- ❖ Names of variables, functions, arrays, etc.
- ❖ Sequence of letters and digits that is assigned a data type and hence used as a container.
- ❖ User can use any name as identifier except for the 32 keywords.
- ❖ Case sensitive.
- ❖ Precautions while naming identifiers
 - First character of identifier must be an alphabet(letter).
 - Underscore (_) is considered as a letter.
 - Uppercase and lowercase letters are different.

STRINGS

- ◆ A Sequence of characters terminated by a null character '\0'.
- ◆ This terminating character differentiates strings from character arrays.
- ◆ Syntax:

```
char str[]="Vision";
printf("%s\n", str);
```

OPERATORS

- ◆ It is a symbol that operates on a value or a variable.
- ◆ An operator is a symbol that tells the compiler to perform mathematical, bitwise, logical & conditional operations.
- ◆ 3 types:-
 - Unary Operators
 - Binary Operators
 - Ternary Operators

Operators in C

Operators	Type
<code>++ , --</code>	Unary operator
<code>+ , - , * , / , %</code>	Arithmetic operator
<code>< , <= , > , >= , == , !=</code>	Relational operator
<code>&& , , !</code>	Logical operator
<code>& , , << , >> , - , ^</code>	Bitwise operator
<code>= , += , -= , *= , %=</code>	Assignment operator
<code>?:</code>	Ternary or conditional operator

Unary operator ←

Binary operator ←

Ternary operator ←



Operators in C

01

Relational
Operators

02

Assignment
Operators

03

Arithmetic
Operators

04

Bitwise
Operators

05

Conditional
Operators

06

Logical
Operators

07

Misc
Operators

A BASIC C PROGRAM

```
#include <stdio.h>

int main() {

    int a,b,c;
    printf("Enter A and B:");
    scanf("%d%d",&a,&b);

    c=a+b;

    printf("The Sum of A and B is: %d",c);

    return 0;
}
```

DATA TYPES

A data type specifies the type of data that a variable can store such as integer, floating, character, etc.

DataTypes in C

Primary

- Integer
- Character
- Floating Point
- Double Floating Point
- Void

Derived

- Function
- Array
- Pointer
- Reference

User Defined

- Class
- Structure
- Union
- Enum
- Typedef



PRIMARY DATA TYPES WITH SIZE AND RANGES

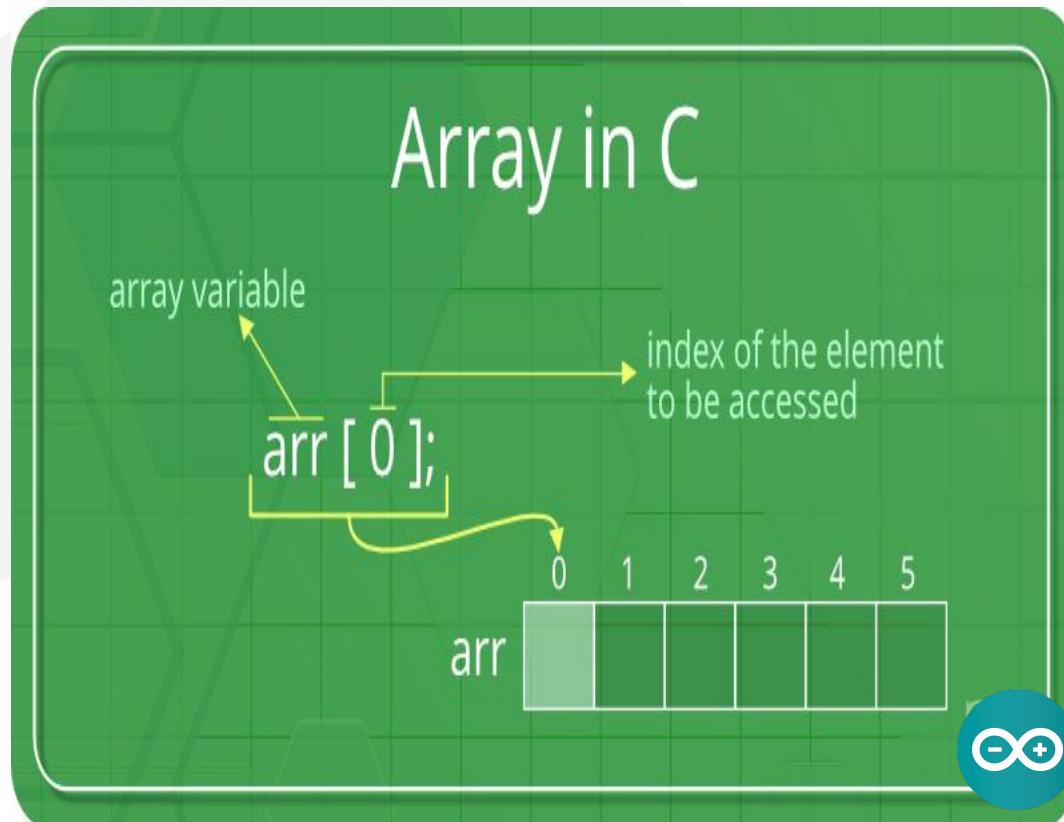
C Basic Data Types	32-bit CPU		64-bit CPU	
	Size (bytes)	Range	Size (bytes)	Range
char	1	-128 to 127	1	-128 to 127
short	2	-32,768 to 32,767	2	-32,768 to 32,767
int	4	-2,147,483,648 to 2,147,483,647	4	-2,147,483,648 to 2,147,483,647
long	4	-2,147,483,648 to 2,147,483,647	8	-9,223,372,036,854,775,808-9,223,372,036,854,775,807
long long	8	9,223,372,036,854,775,808-9,223,372,036,854,775,807	8	9,223,372,036,854,775,808-9,223,372,036,854,775,807
float	4	3.4E +/- 38	4	3.4E +/- 38
double	8	1.7E +/- 308	8	1.7E +/- 308

DERIVED DATA TYPE: ARRAY

An array is:

- Collection of elements
- of the same data type
- placed in contiguous memory locations
- can be individually referenced via index

Clubbing multiple entities of similar type into a larger group.

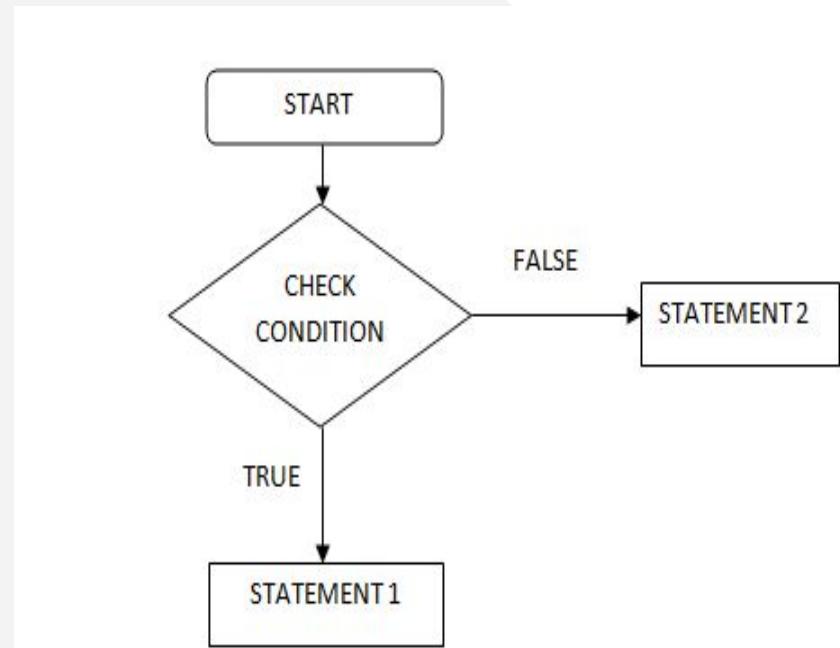


FLOW CONTROL

- ◆ Also known as Conditional iterations.
- ◆ Based on a certain conditions predefined set of instructions are implemented
- ◆ Generally the condition is either true or false.

Examples ::

if else , if , switch cases .





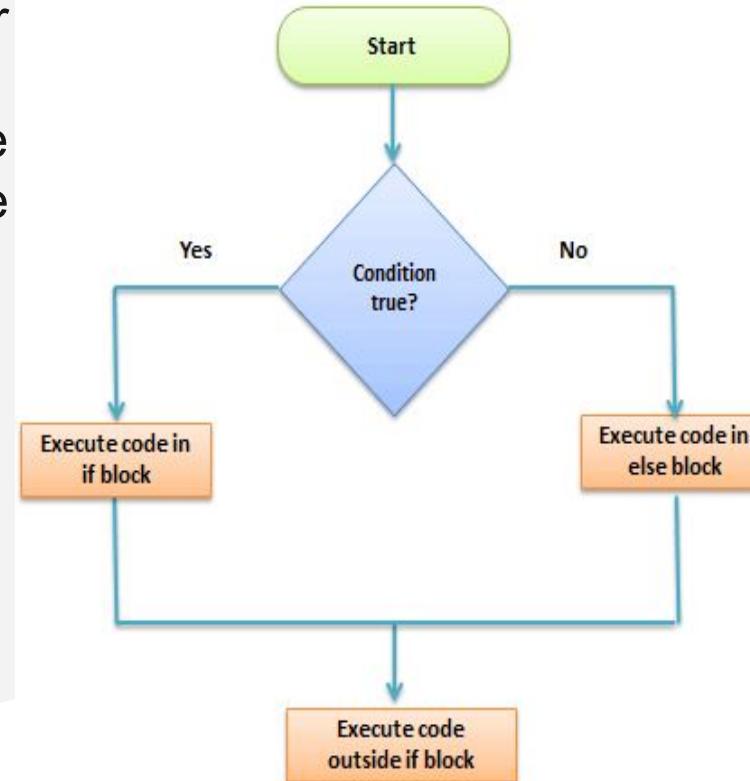
TRUE/
FALSE

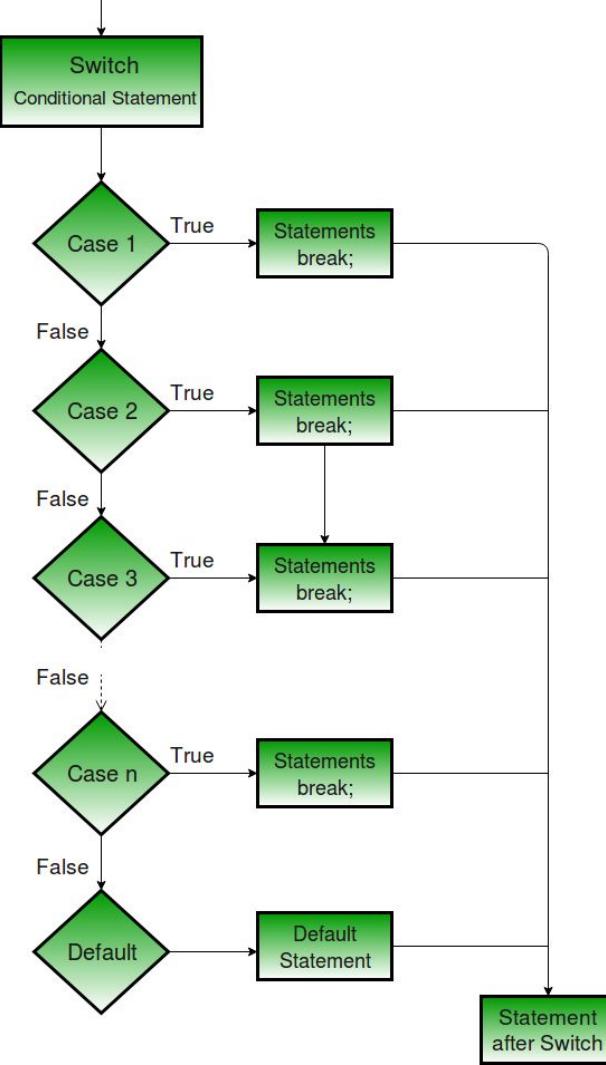
0/1

IF & IF-ELSE STATEMENT:

- ❖ If-else statement is used for decision-making
- ❖ If the given condition is true, then the code inside if block is executed, otherwise else block code is executed
- ❖ Syntax:

```
int a = 10;
if (a < 20) {
    printf("Given Value
is less than 20\n");
}
else {
    printf("Given Value
is greater than 20");
}
```





SWITCH CASE

Switch case statements follow a selection-control mechanism and allow a value to change control of execution.

They are a substitute for long if statements that compare a variable to several integral values.

In a switch statement, the “case value” can be of “char” and “int” type.

```
{
  case value1: statement_1; break;
  case value2: statement_2; break;
  .....
  .....
  case value_n: statement_n; break;
  default: default statement;
}
```

LOOPS

- ❖ Loops in programming are used to repeat a block of code until the specified condition is met.
- ❖ A loop statement allows programmers to execute a statement or group of statements multiple times without repetition of code.
- ❖ THE MOST BASIC PRINCIPLE INVOLVED IS OF TRUE/FALSE.
(0- FALSE & 1- TRUE)

```
printf( "Hello World\n");
..
..
```



TYPES OF LOOPS

There are mainly two types of loops in C Programming

Loops

Entry Controlled

for

```
for( initialization ; condition; updation )  
{  
}
```

while

```
while( condition )  
{  
}
```

Exit Controlled

do-while

```
do  
{  
}  
}while( condition )
```



FOR- LOOP

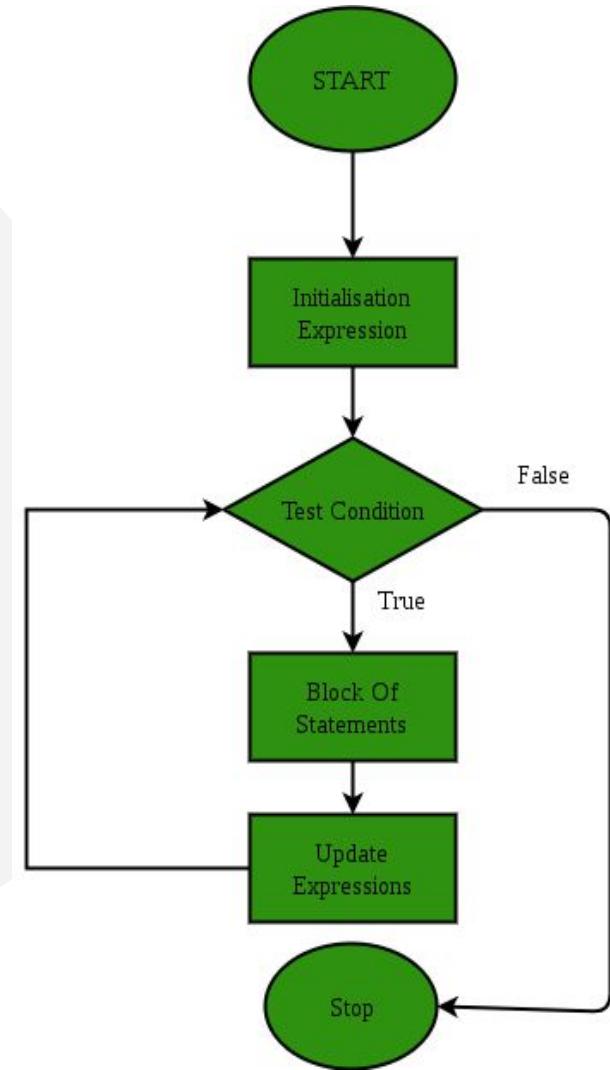
- ❖ For loop in C programming is a repetition control structure that allows programmers to write a loop that will be executed a specific number of times.
- ❖ In for loop the number of iterations is previously known to us.

Syntax:

```
for(initialize expression; test expression; update  
expression)  
{  
    // body of for loop  
}
```

Example:

```
for(int i = 0; i < n; ++i)  
{  
    printf("Body of for loop to execute n times");  
}
```

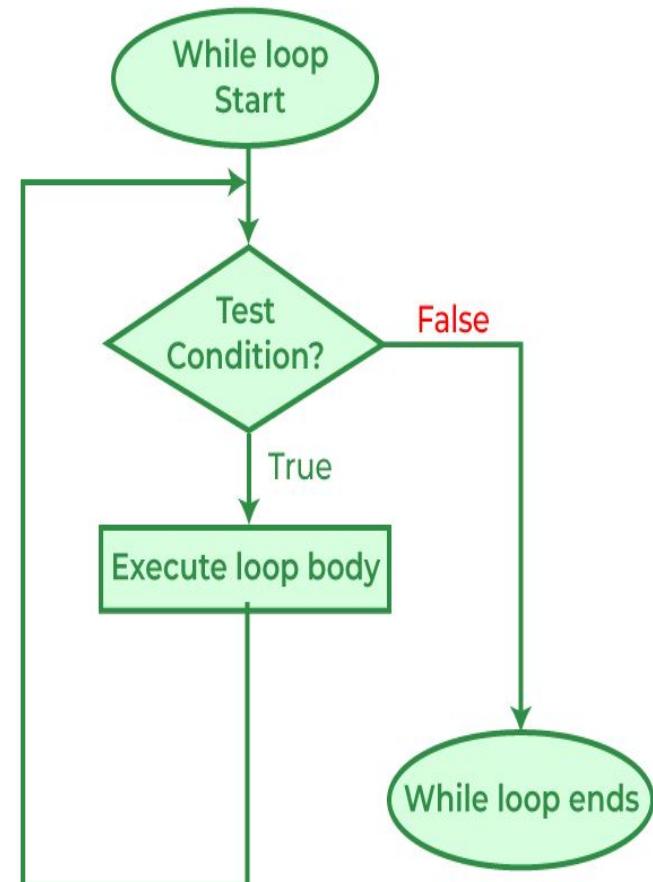


WHILE LOOP

- ❖ In the While loop, the execution is terminated on the basis of the test condition
- ❖ If the test condition will become false then it will break from the while loop else body will be executed.

Syntax:

```
initialization_expression;  
while (test_expression)  
{  
    // body of the while loop  
    update_expression;  
}
```



while($n==1$)

while($n=1$)



FUNCTIONS

- ❖ Functions are sets of statements that take inputs, perform some operations, and produce results
- ❖ A function performs a certain action, and it is important for reusing code
- ❖ 3 parts involved to use a function:-
 - **Function Declaration**:- tell the compiler the number & kinds of parameters it returns, and what types of data it takes.
 - **Function Definition** :- consists function header and a function body.
 - **Function Call**:- A function is called or evoked by providing the name of the function followed by the parameters enclosed in parenthesis.

```
include <iostream>
include <cmath>

using namespace std;

int multiply(int a, int b)
{
    int c;
    c = a * b;
    return c;
}

int main()
{
    int x = 19, y = 21, result;
    result = multiply(x, y);
    cout << result;
}
```



CODE STRUCTURE

The basic structure of Arduino programming language consists of two required functions `setup` and `loop`.



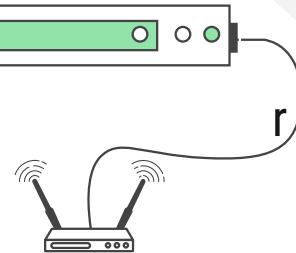
```
void setup(){  
  //code that will run only once
```

```
void loop(){  
  //this contains the bulk of program and runs  
  //consecutively  
  //till the program is stopped  
}
```



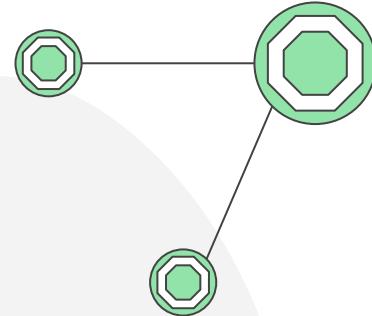
setup()

- ◆ The setup()function is called only once when the program starts.
- ◆ It is used to initialize pin modes we will cover them shortly.
- ◆ This should be included in program even if there is no statement to put here



loop()

- ◆ It starts after the setup()function. it does what its name suggest.
- ◆ It runs the code inside it consecutively till the program is stopped
- ◆ It contains our entire program. here we can also call user defined functions.





Arduino specific Functions

There are some special functions used in Arduino.

They are used to take input and give output through pins.

we don't need to understand their inner working here,

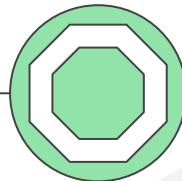
All we need to learn is that how to use them.

we will discuss them one by one.

There are also couple of keywords to define input and output.

HIGH-It simply means ON,1 in binary,5 volts in term of voltage.

LOW-It simply means OFF,0 in binary,0 volts in term of voltage.



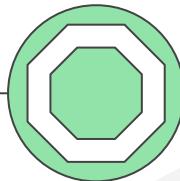
pinMode(pin,mode)

This function is used to configure a specific pin to behave either as input or an output.

syntax:

```
void setup(){  
pinMode(13,OUTPUT); //sets the digital pin 13  
as output  
}
```





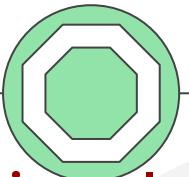
digitalRead(pin)

Read the value from a specified digital pin ,either HIGH or LOW .

syntax:

```
int value=0;  
void setup(){  
pinMode(7,INPUT);  
}  
void loop(){  
value=digitalRead(7);  
//read the value from pin ->7 and assign to the  
//variable "value"  
}
```



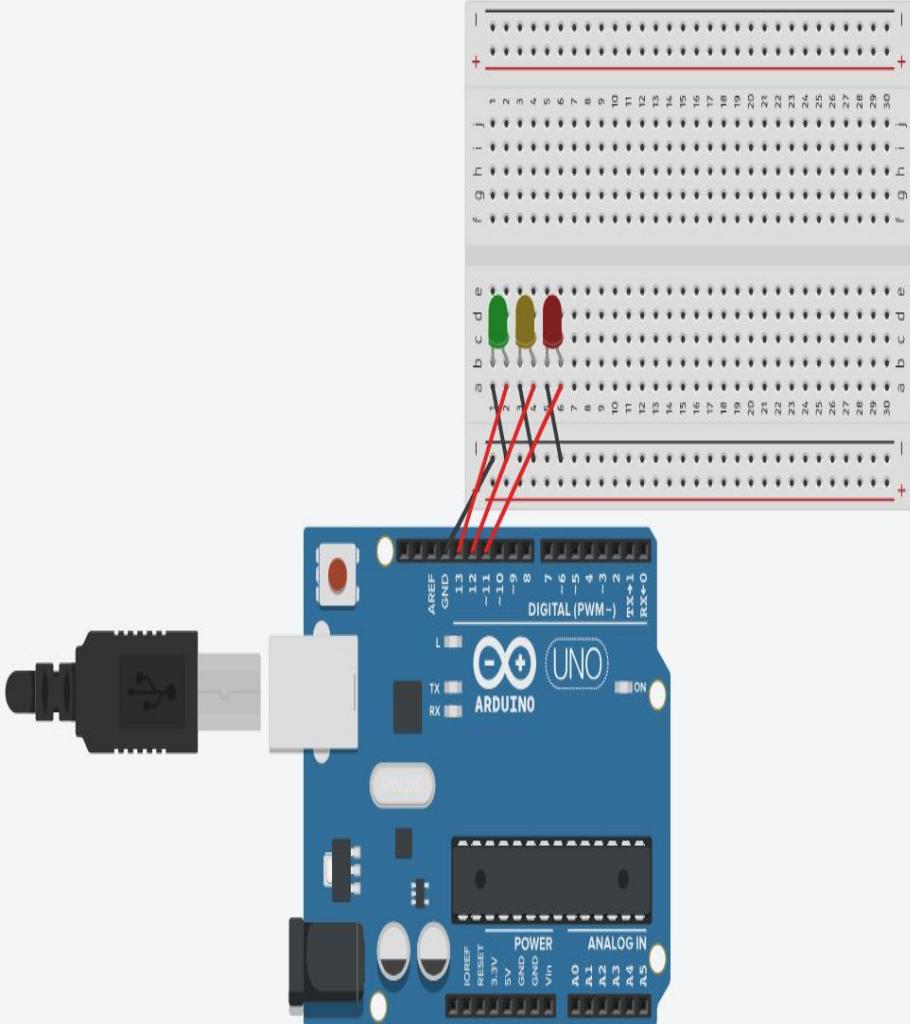


digitalWrite(pin,value)

Used to write value(HIGH or LOW) to a pin.If the pin is configured as an OUTPUT with pinMode(),its voltage will be set to corresponding value : 5V(or3.3V on 3.3V boards)for HIGH , 0V(ground)forLOW.

syntax:

```
void setup(){  
pinMode(13,OUTPUT);  
}  
void loop(){  
digitalWrite(13,HIGH); //sets the digital pin ON  
delay(1000); //waits for a second  
digitalWrite(13,LOW); //sets the digital pin OFF  
delay(1000); //waits for a second  
}
```





analogRead(pin)

It can read value from an analog pin with 10-bit resolution.

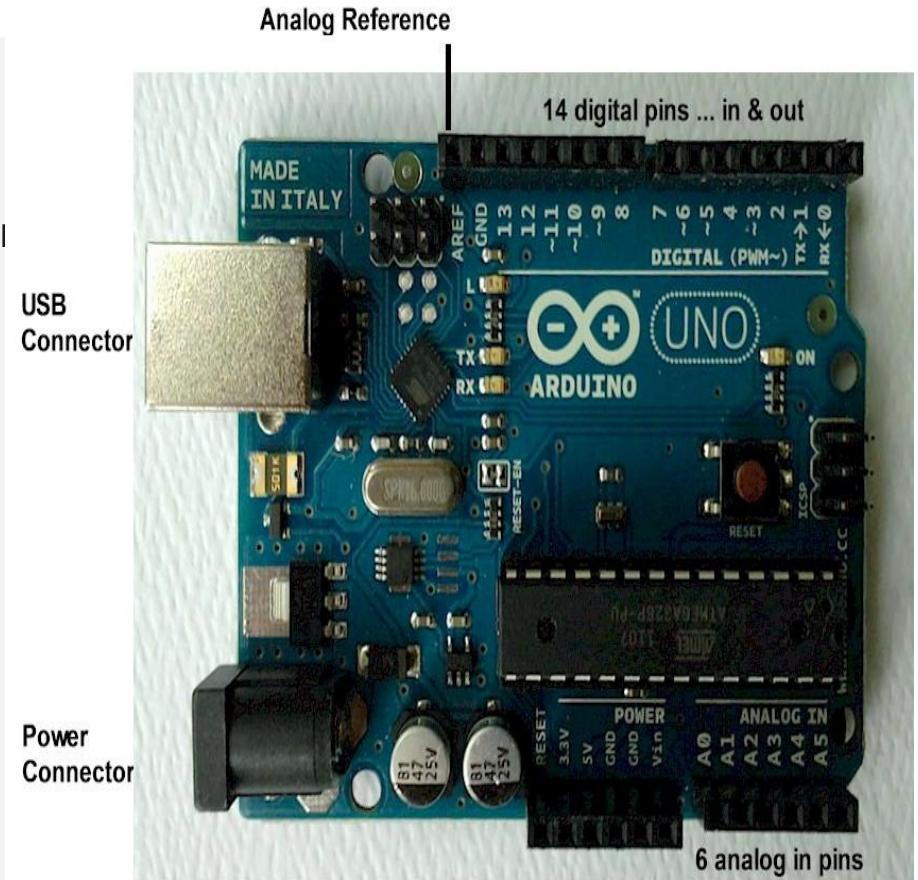
This function can only be used for analog pins

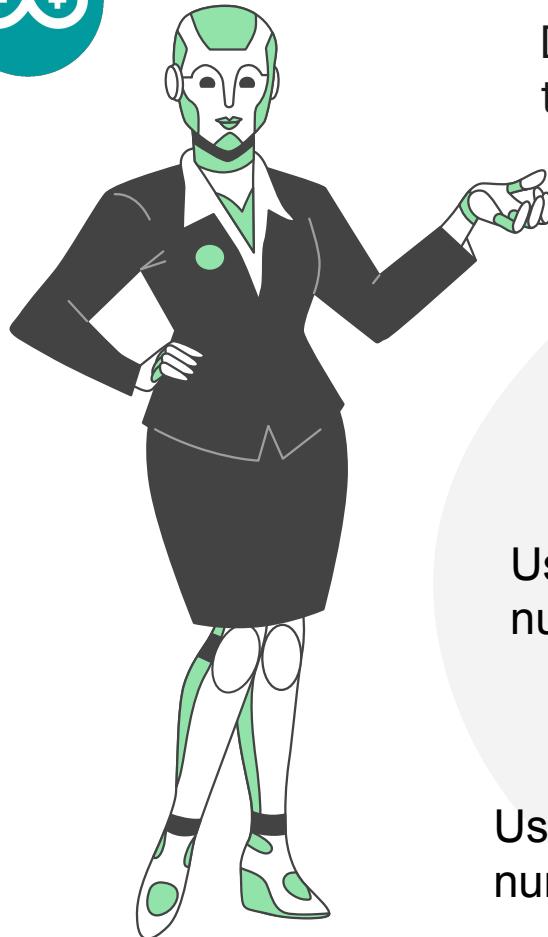
```
value=analogRead(pin);  
//range of " value" from 0 to 1023
```

analogWrite(pin,value)

This function is used to light a LED at varying brightness and to drive a motor at various speed.

It can write an analog value(ranging from 0-255)on a pin . On a newer arduinos this function can work on pins 3,5,6,9,10,11 and on older,it works on pins 9,10,11.





delay(ms)

Delays or pauses the program for amount of time provided in milliseconds.

millis()

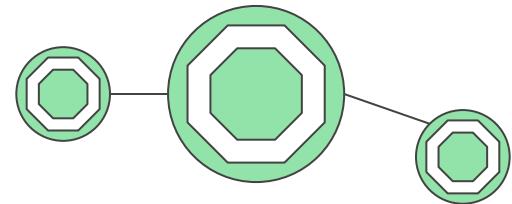
Returns the time since the program is running in milliseconds as 'long' type.

min(x,y)

Used to find the minimum of two numbers, passed in as parameters.

max(x,y)

Used to find the maximum of two numbers, passed in as parameters.



Serial.begin(rate)

Sets the data rate in bits per second (baud) for serial data transmission.

for communicating with serial monitor.

A typical baud rate for communicating with computer is 9600.

It is required when need to communicate with a device using universal serial bus(USB).

Serial.println(data)

It is used to print any data in the console of the device which is connected through USB.

It require Serial.begin(rate)to be executed in order to work.



random(min,max)

This function allow us to get a random number

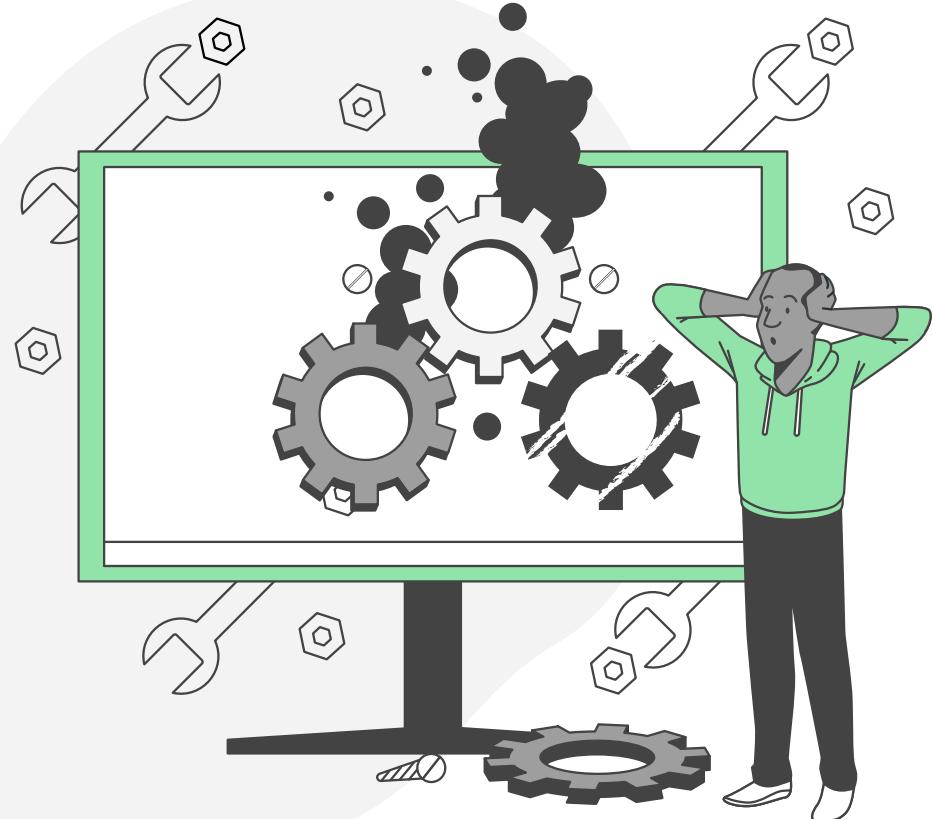
between min and max.if min is not passed ,it is assumed 0.

```
int value = random(0,100)
```

//value becomes a number between 0-100

```
int value =random(100)
```

//same thing as above



ARDUINO LIBRARIES

- The Library is considered as the advanced feature, which extends the capabilities of the Arduino IDE.
- It means that the libraries provide extra functionality to the programming platform of Arduino.
- We could take the analogy of these libraries with a car functions like with different switches we use different functions of car just without taking the tension of that deep functioning of car.



ARDUINO STANDARD LIBRARIES

- The Arduino IDE comes with a set of standard libraries for commonly used functionality.
- These are pre-installed in the "Libraries" folder of the Arduino install.
- If you have multiple versions of the IDE installed, each version will have its own set of libraries. For the most part, it is not a good idea to change the Standard Libraries or install your libraries in the same folder.

The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The Sketch menu is currently active, with "Include Library" highlighted in blue. A dropdown menu is open under "Include Library", listing options like "Manage Libraries...", "Add .ZIP Library...", and a long list of standard Arduino libraries: Arduino libraries, Bridge, EEPROM, Esplora, Ethernet, Firmata, GSM, HID, Keyboard, LiquidCrystal, Mouse, Robot Control, Robot IR Remote, Robot Motor, SD, SPI, Servo, SoftwareSerial, SpacebrewYun, Stepper, and TFT. The main code editor area contains a sketch with basic digitalWrite() calls for an LED on pin 12.

```
void setup() {  
    // initialize pins  
    pinMode(12, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(12, HIGH); // turn the LED  
    delay(1000); // wait  
    digitalWrite(12, LOW); // turn the LED  
    delay(1000); // wait  
}
```

Some Standard Libraries

- ❖ SERVO LIBRARY
- ❖ LCD (LIQUID CRYSTAL)
- ❖ STEPPER

•
..
...

.... and many more !

TinkerCAD

T I N
K E R
C A D

AUTODESK®
TINKERCAD®