CSE233
Embedded Systems and IoT

Introduction To Embedded Systems

- Embedded System (ES) is a combination of HW and SW which is designed to perform a specific task by a given time.

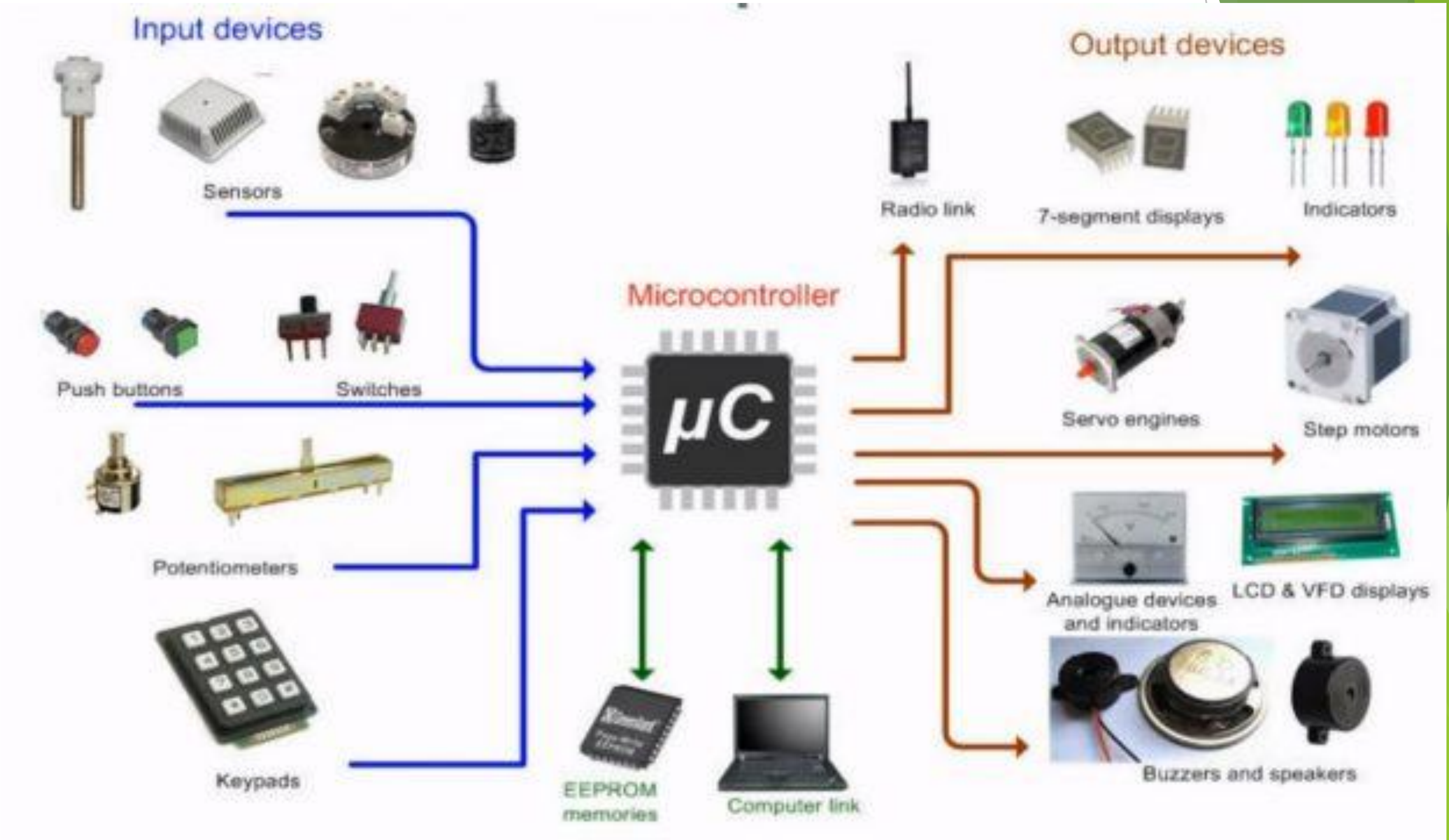
- E.g.- Washing machine, Digital Camera, Microwave oven etc.

- In modern cars there are about 100 ES
- In Aeroplanes there are about 1500 ES

Components of Embedded Systems

- The heart of Embedded Systems (ES) is Microcontrollers.
- There are also Input and Output sensors and Actuators.
- So, at first we will discuss the Basic of Electrical and Electronics, as we already have studied electrical circuits, electronic devices and circuits and physics.

Components of Embedded Systems



Components of Embedded Systems

Inputs	Outputs
1. Sensors	1. LEDs
2. Touch panel	2. LCD Monitor
3. Buttons	3. Buzzer
4. IR	4. IR
5. Bluetooth etc	5. Bluetooth etc

IoT

- IoT refers to a network of interconnected devices that communicate and exchange data.
- A smart refrigerator that connects to the internet to order groceries is an IoT device.

Thank You

Department of Computer Science and Engineering
Daffodil International University

Introduction to Embedded Systems and IoT

Course Title: Embedded Systems and IoT
Course Code: CSE233

Prepared By:

Jul Jalal Al-Mamur Sayor(JMS)
Lecturer, Dept. of CSE, DIU

Today's Lecture

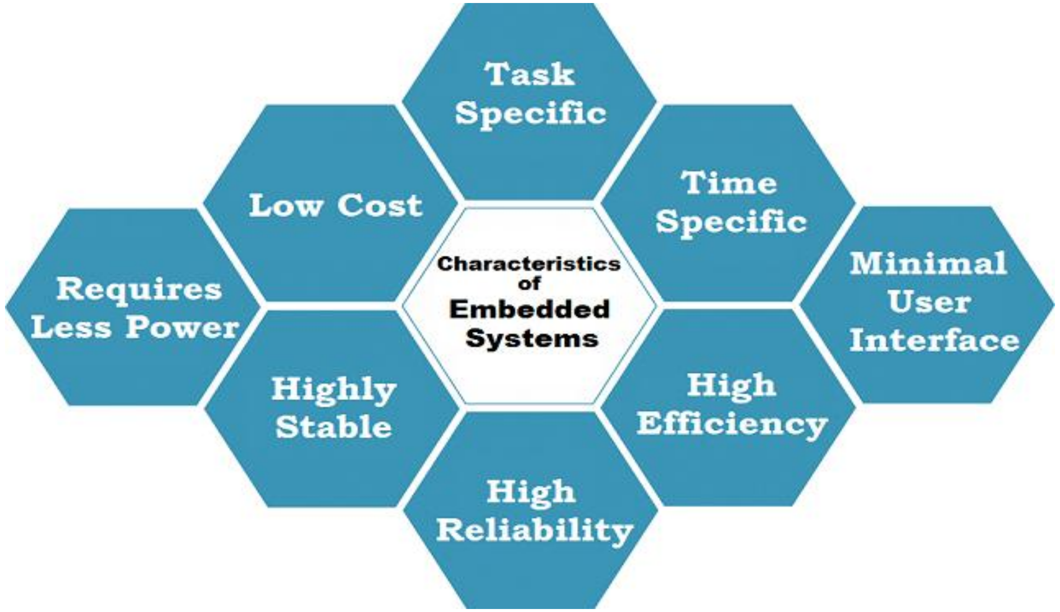
1. **Introduction to Embedded System**
   1. What is an Embedded System?
   2. Characteristics of Embedded Systems
   3. Embedded Systems Applications
   4. Embedded Systems Examples
   5. Microprocessors vs Microcontroller
2. **Introduction to IoT**
   1. What is IoT?
   2. IoT- How it works?
   3. IoT applications
   4. Benefits of IoT
   5. Challenges in IoT

What is an Embedded System?

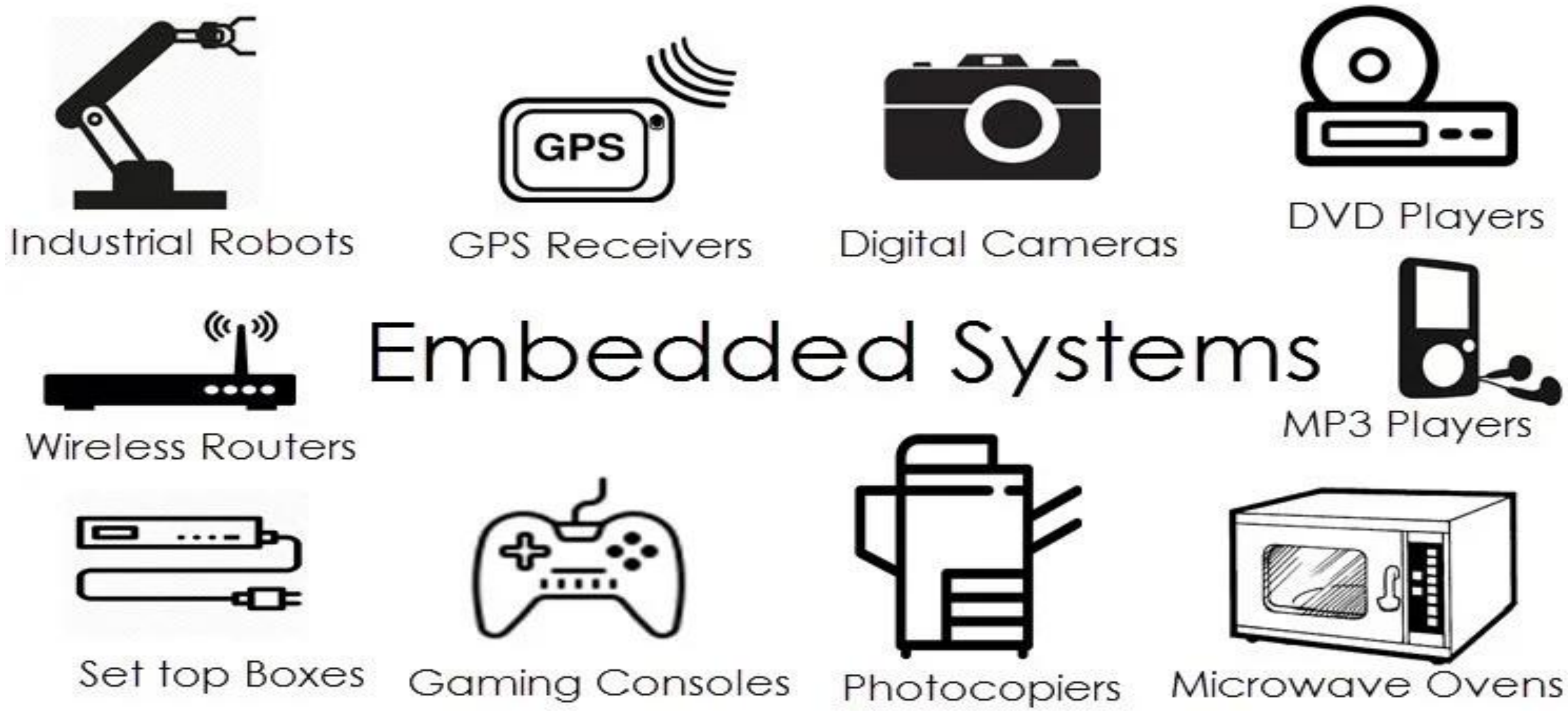
- An embedded system is a combination of hardware and software, where the hardware typically includes a microcontroller or microprocessor, and the software is designed to control the hardware.
- This system is designed to perform a specific or limited set of tasks efficiently.
- It is designed to operate autonomously without human intervention.
- It operates under real-time performance constraints that must be strictly met.

Characteristics of Embedded Systems

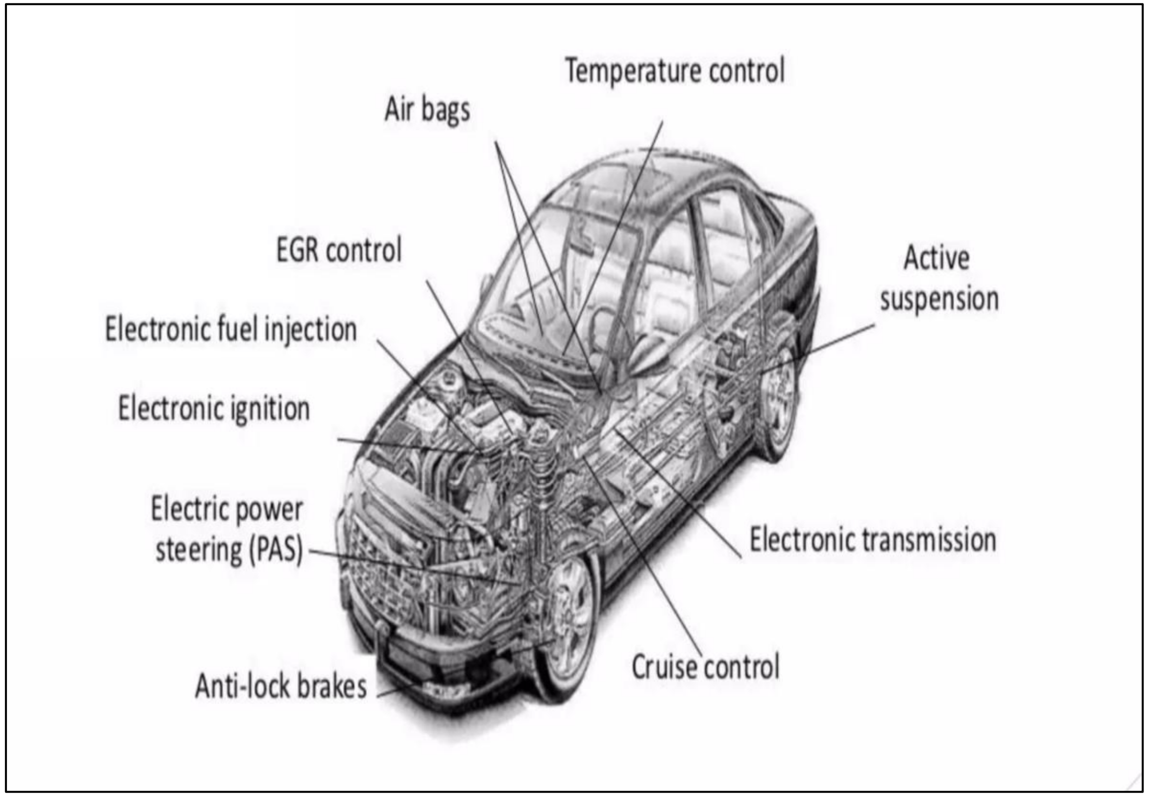
- ❑ **The system must meet the following characteristics:**
  - ❑ Performance
  - ❑ Cost/size
  - ❑ Real time requirements
  - ❑ Power consumption
  - ❑ Reliability

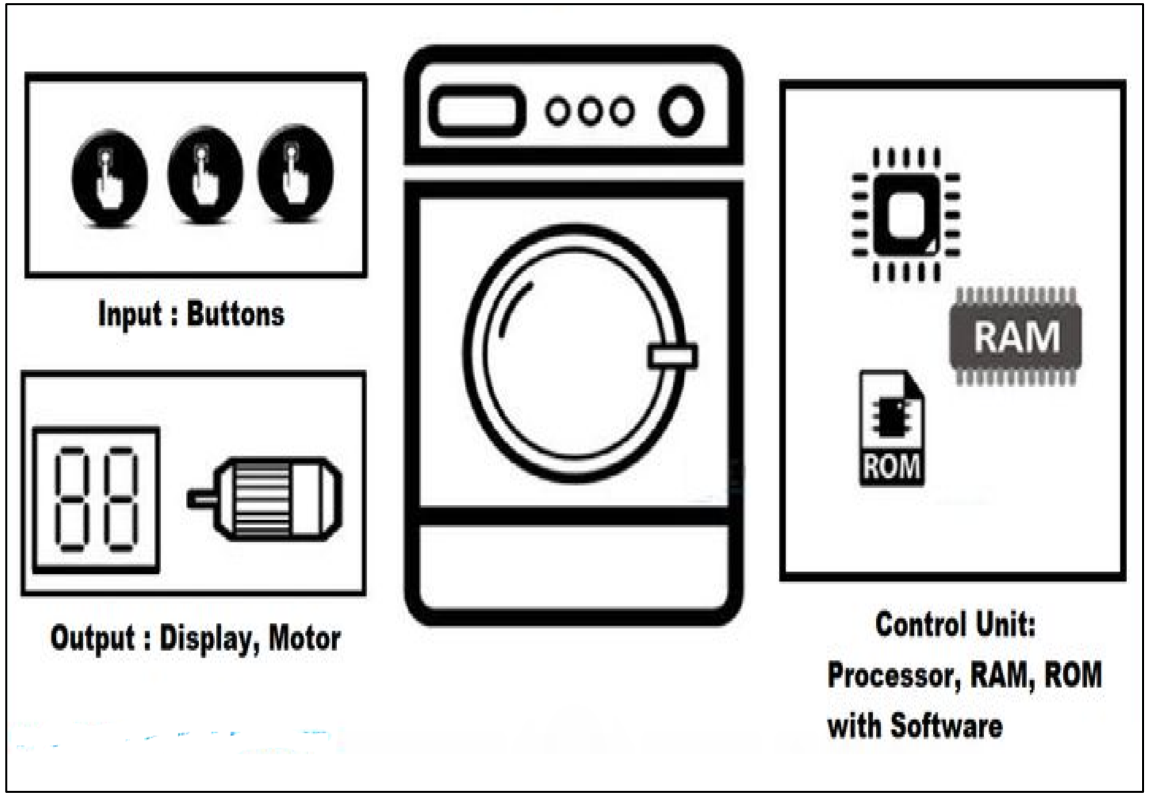


Embedded Systems Applications



Embedded Systems Examples

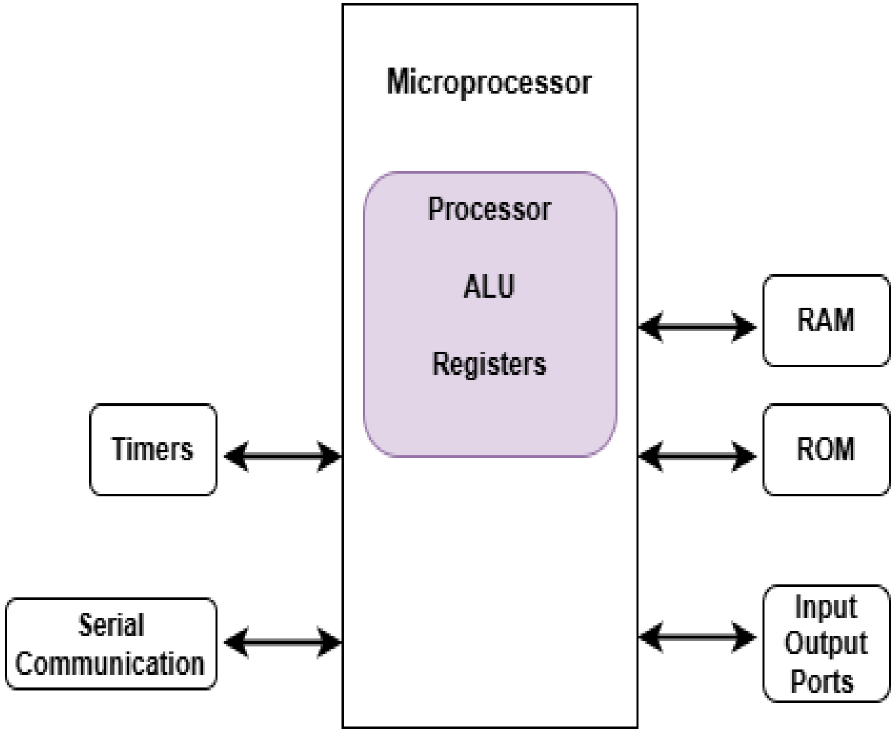




Microprocessors vs Microcontroller

❑ What is Microprocessor?

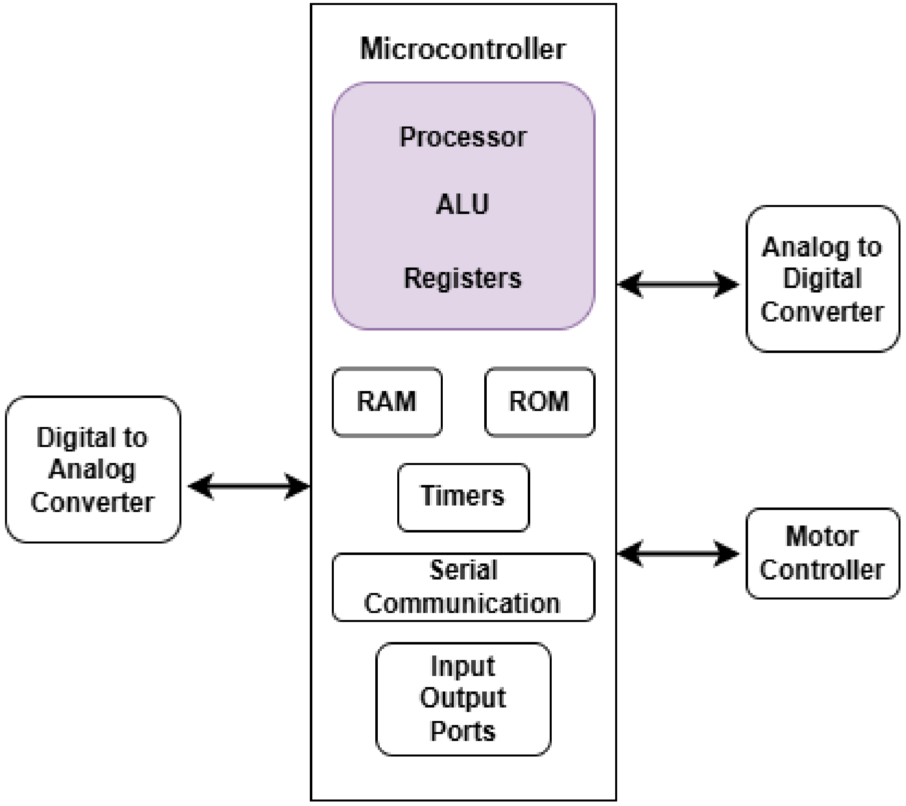
- A microprocessor is a compact, integrated circuit (IC) that serves as the central processing unit (CPU) of a computer system.
- It executes instructions from programs by performing arithmetic, logic, control, and input/output operations.
- Unlike microcontrollers, microprocessors lack built-in peripherals like memory or I/O ports.
- Examples: Intel Core i7, AMD Ryzen, ARM Cortex (used in mobile devices)



Microprocessors vs Microcontroller(Cont.)

❑ What is Microcontroller?

- A microcontroller is a compact, integrated circuit designed to control specific tasks in embedded systems.
- It integrates a CPU, memory (RAM and ROM), and peripherals like I/O ports, timers, and ADCs into a single chip, making it perfectly suited for self-contained applications.
- Examples: Arduino, ESP32, ESP8266

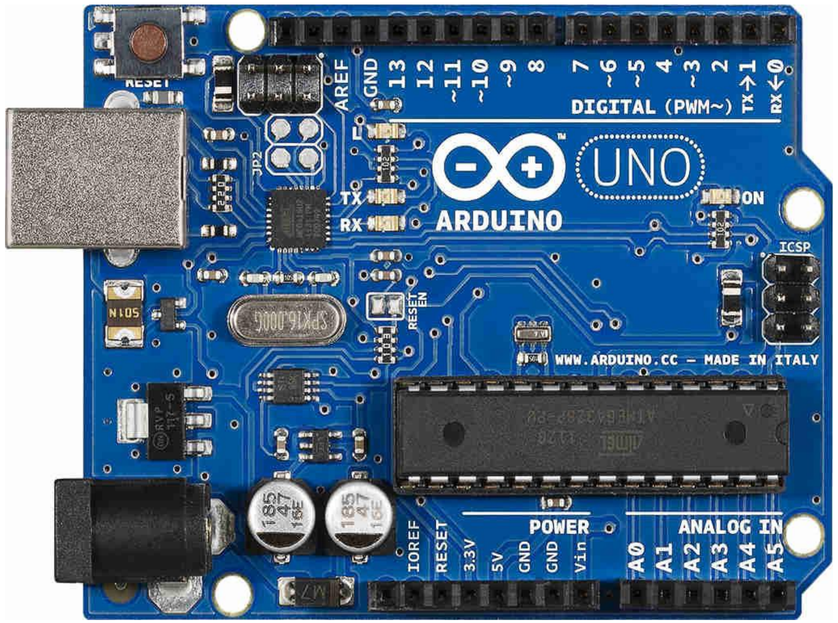


Microcontroller Example

❑ What is Arduino?

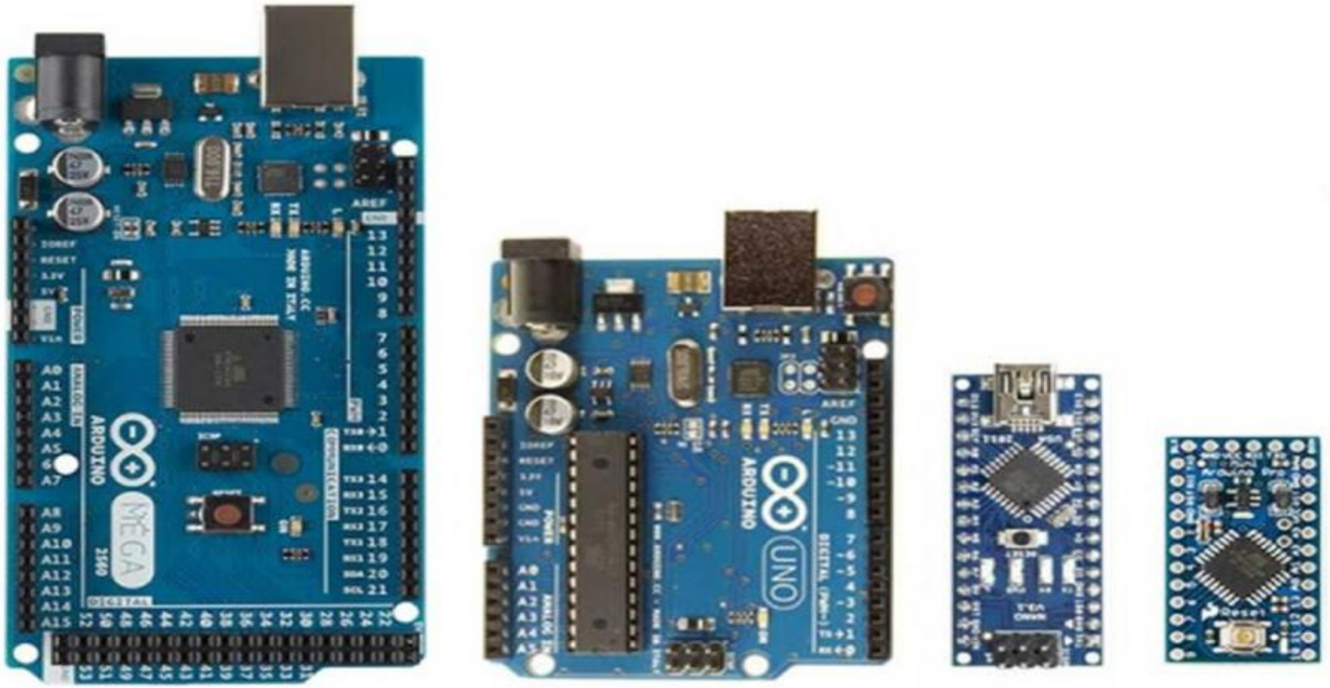
- Arduino is an open-source platform designed for building and prototyping electronic projects. It includes both hardware (microcontroller boards) and software (Arduino IDE) for programming.

- It is beginner-friendly, with a simple programming interface and a wide range of boards. It is suitable for tasks like automation, IoT, robotics, and more.



Microcontroller Example(Cont.)

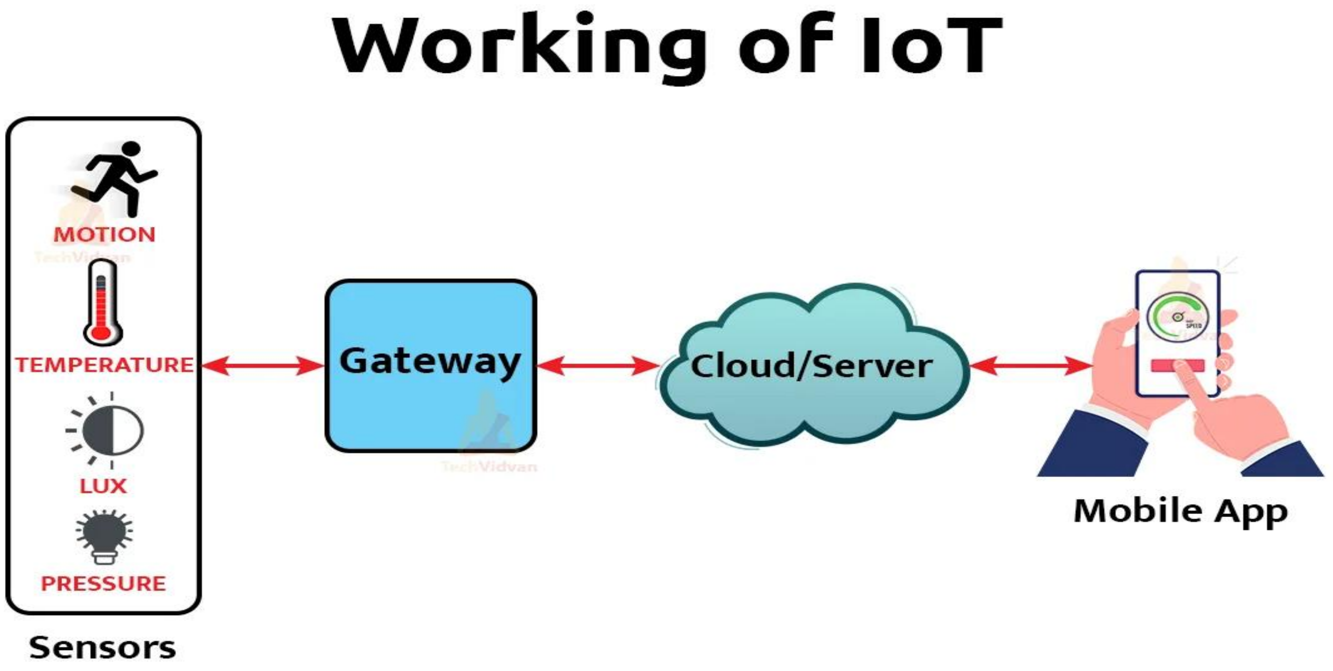
- Some versions of Arduino board



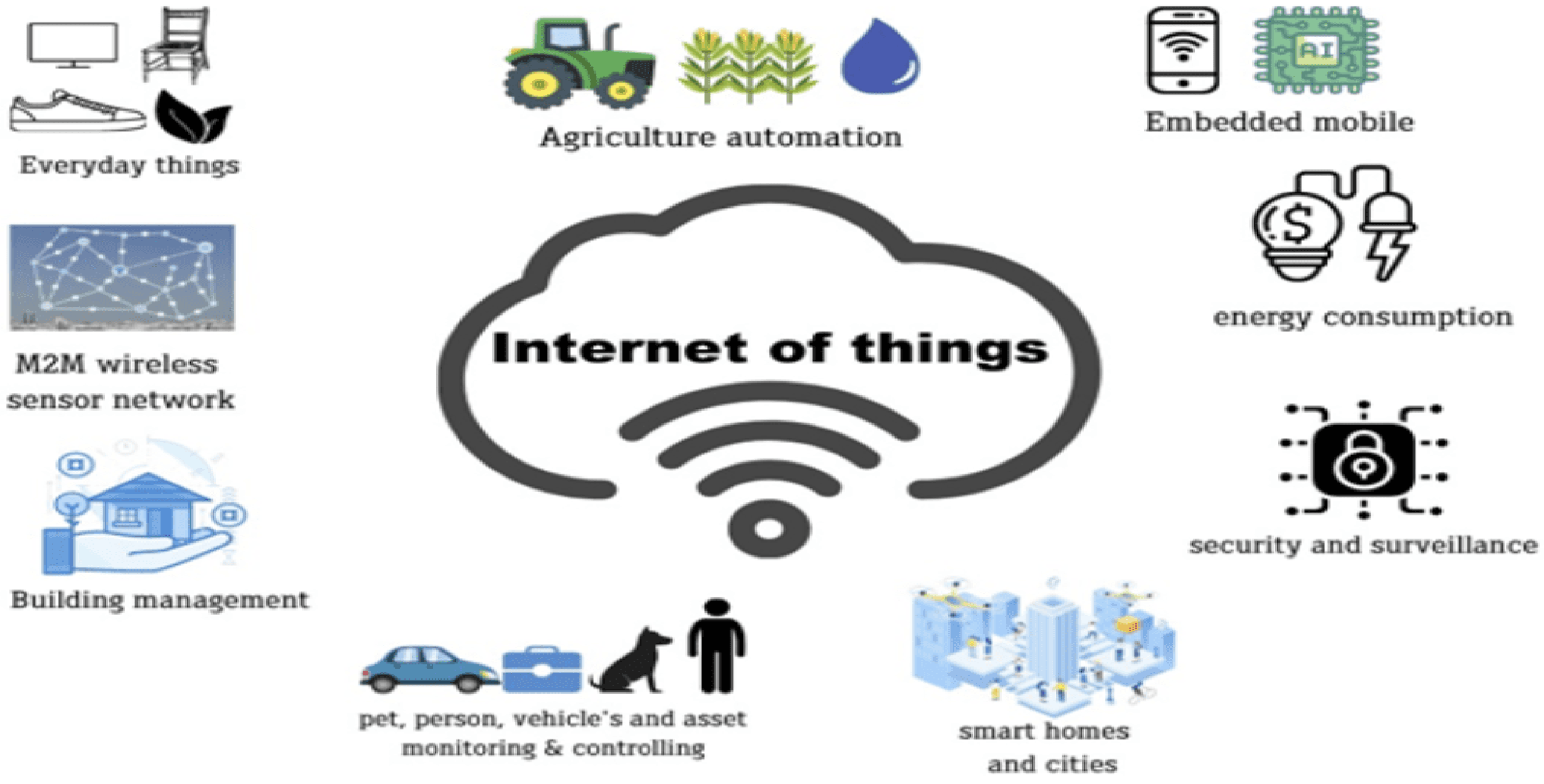
What is IoT?

- The Internet of Things (IoT) refers to a network of physical devices embedded with sensors, software, and connectivity that enable them to collect, exchange, and act on data over the internet.
- Sensors collect and transmit real-time data for analysis. Devices communicate with each other and cloud systems.
- IoT enables smart decision-making and automated actions.
- Examples: Fitness tracker, Smart Home Devices

IoT- How it works?

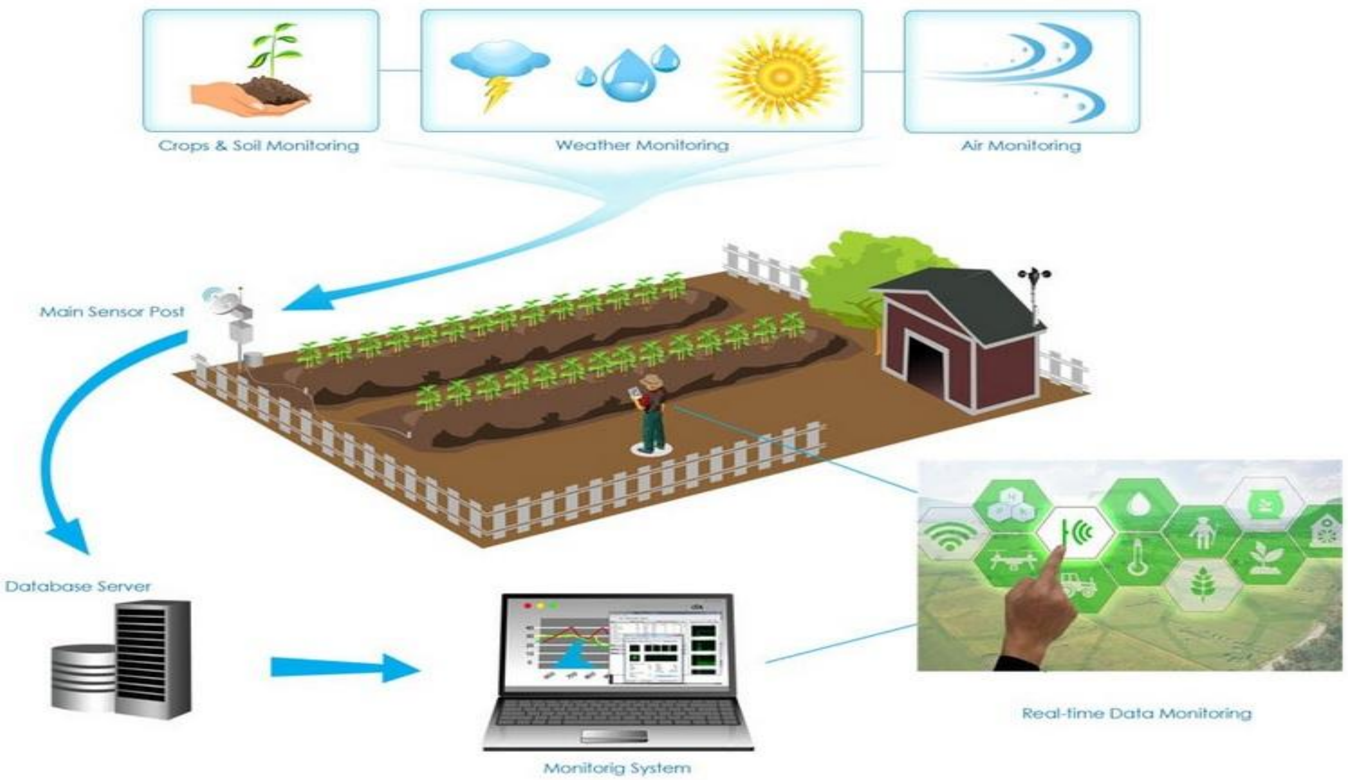


Applications of IoT



Benefits of IoT

- The Internet of Things (IoT) offers several benefits. They are-
  - Improved efficiency
  - Real-time monitoring
  - Cost savings
  - Better decision-making
  - Enhanced automation



Challenges in IoT

❑ **The Internet of Things (IoT) faces several challenges, including:**

- **Security and Privacy Risks:** IoT devices collect sensitive data, making them prime targets for hackers. Weak security can lead to breaches and unauthorized access.
- **Data Management and Analysis:** IoT creates a lot of data, which can overload storage and analysis systems, needing advanced tools to process and understand it.
- **High Implementation Costs:** IoT deployment involves high initial costs for devices, connectivity, and infrastructure, which can deter small businesses or underfunded sectors.
- **Interoperability of Devices:** IoT devices from different manufacturers and protocols struggle to communicate seamlessly.

Embedded System vs IoT

Aspect	Embedded System	IoT (Internet of Things)
Definition	Dedicated system designed for specific tasks.	Network of connected devices sharing data over the internet.
Scope	Operates within a single device or for a specific task.	Expands embedded systems by enabling internet connectivity.
Connectivity	Operates offline; may connect within the device only.	Requires internet connectivity for data sharing and communication.
Components	Microcontroller, sensors, actuators, memory, and interfaces.	Embedded systems, internet modules, cloud storage, IoT platforms.
Functionality	Performs predefined tasks with limited or no interaction.	Collects, processes, and shares data for smart, adaptive tasks.
Examples	Washing machine, pacemaker, digital camera.	Smart thermostat, wearable fitness tracker, smart home systems.
Dependency on Internet	Independent of internet; may work offline.	Heavily dependent on internet connectivity for functionality.
Adaptability	Static and task-specific functionality.	Dynamic, interactive, and remotely controllable.

An IoT device = Embedded System + Internet Connectivity + Cloud Integration

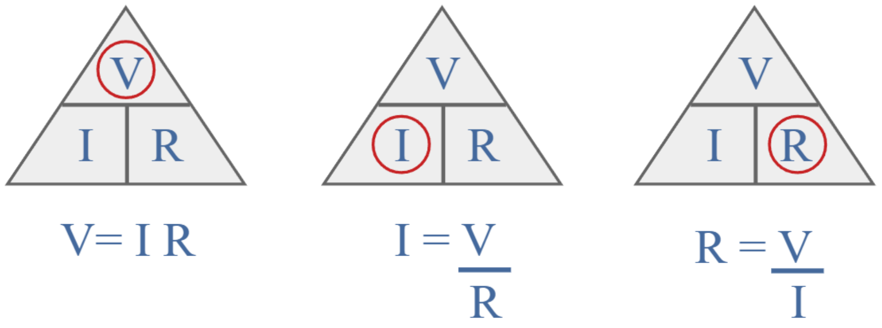
❑ **Ohm's Law :** Ohm's Law states that the current flowing through a conductor between two points is directly proportional to the voltage across the two points, provided the temperature remains constant.

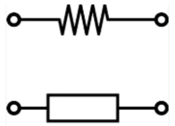
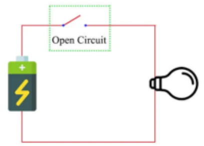
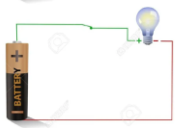
Formula: $V = I \times R$

Where: V = Voltage (Volts)

I = Current (Amperes)

R = Resistance (Ohms, Ω)



Component/Concept	Definition	Key Use / Note	Symbol
Resistor	A passive component that limits or regulates the flow of electric current.	Protects components from excessive current.	
Capacitor	A device that stores and releases electrical energy in an electric field.	Used for filtering, timing, and energy storage.	
Open Circuit	A broken path in which no current flows.	Occurs when a wire is cut or a switch is off.	
Short Circuit	A low-resistance connection allowing excessive current flow.	Can cause overheating or damage.	

Thank you for hearing with patience

Department of Computer Science and Engineering
Daffodil International University

Introduction to Basic Electronics

Course Title: Embedded Systems and IoT

Course Code: CSE233

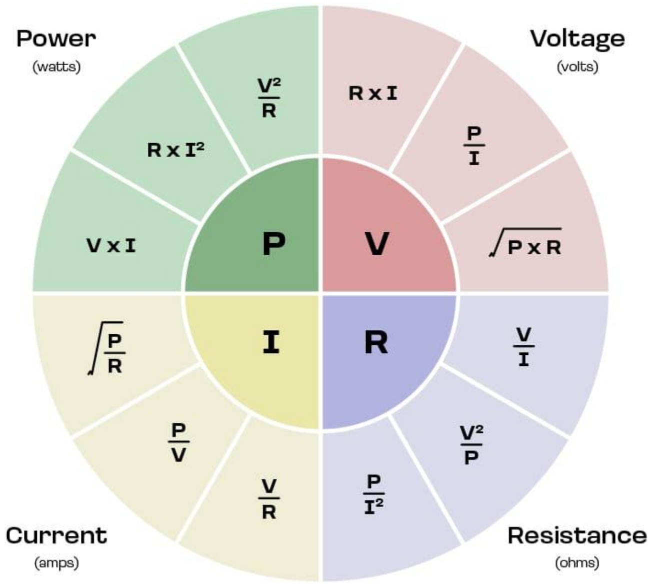
Prepared By:

Jul Jalal Al-Mamur Sayor(JMS)

Lecturer, Dept. of CSE, DIU

Today’s Lecture

1. **Introduction Basic Electronics**
   i. Ohm's Law
   ii. Basic Components (Resistors, Capacitors, LEDs)



Ohm’s Law

- Ohm’s Law states that the current (I) flowing through a conductor between two points is directly proportional to the voltage (V) across the two points and inversely proportional to the resistance (R) of the conductor. It is mathematically expressed as: $V=IR$.

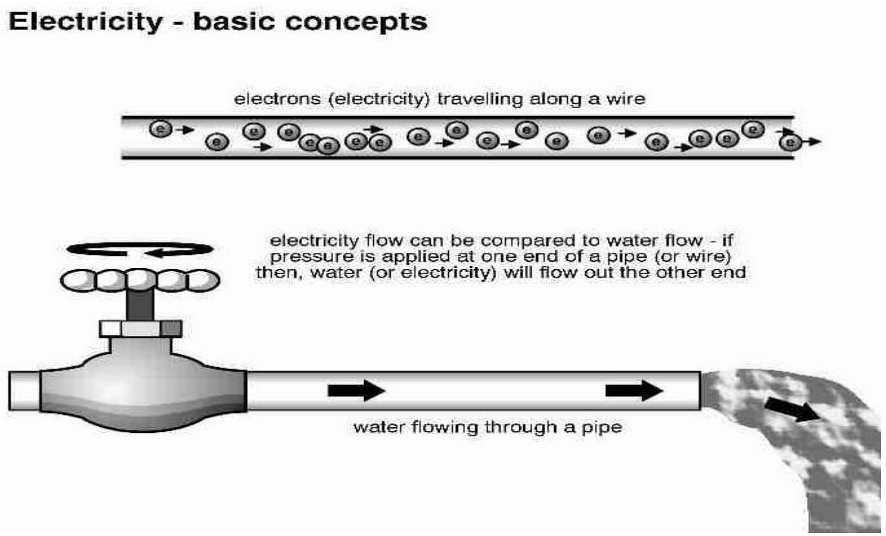
- Voltage is the amount of energy per charge move electrons in one point to another in a circuit. It is measured in volts (v).

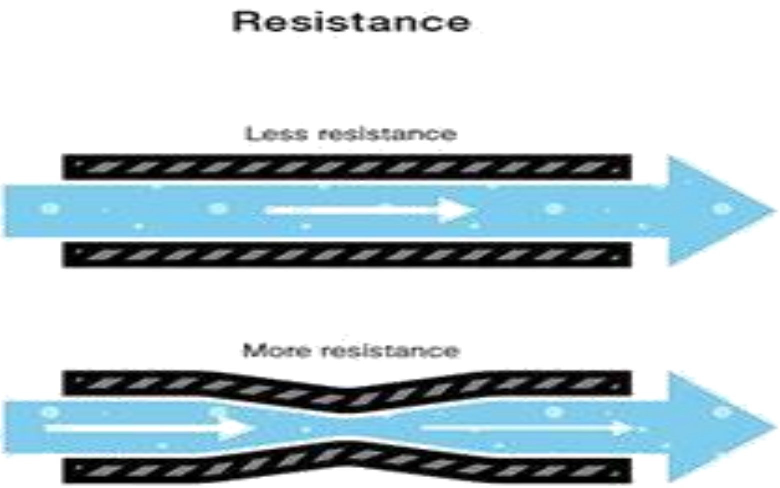
- Current is the rate of charge flow and it is measured in amperes (A).

- Resistance is the opposition of current and it is measured in ohms (Ω). 

Resistor

- R=V/I mathematical formula. Here, R is resistance V is volt and I is current. The unit of resistance is ohm (Ω) as memorial to its inventor.
- Resistance is directly proportional to voltage (V) and inversely proportional to current (I).

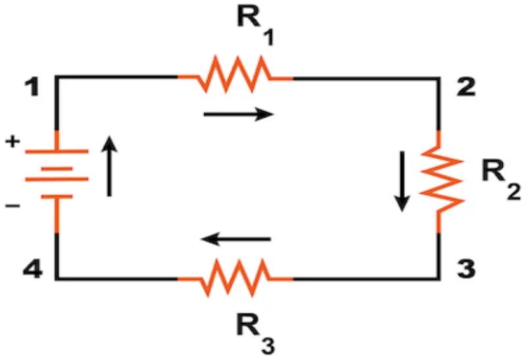


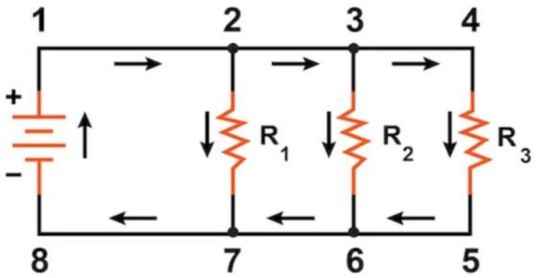


Resistor (Cont.)

- There are mainly two types of Resistors. One is Fixed Resistor and another one is Variable Resistor.

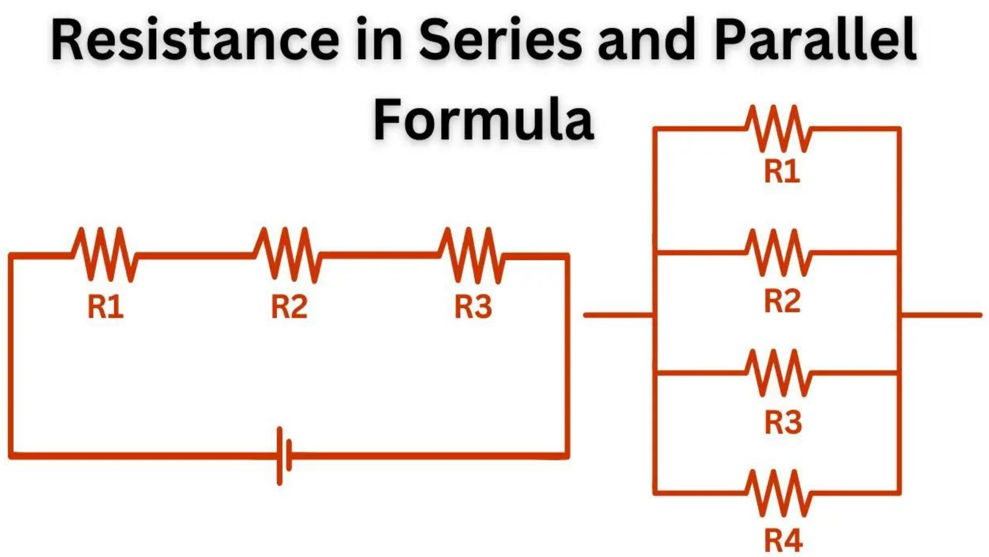
- Resistors can be connected in two configurations depending on the desired electrical properties.
  I. Series connection
  II. Parallel Connection





Resistor (Cont.)

- In series connection,
  I. The current is the same through all resistors.
  II. The voltage divides across the resistors.
- In parallel connection,
  I. The voltage is the same across all resistors.
  II. The current divides among the resistors.

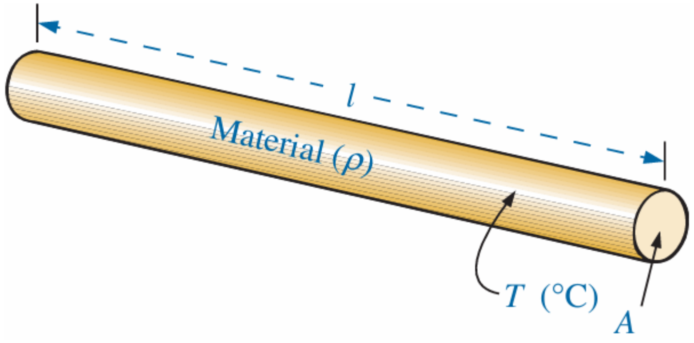


$$R_{eq} = R_1 + R_2 + R_3 + \ldots + R_n$$

$$1/R_{eq} = 1/R_1 + 1/R_2 + 1/R_3 \ldots + 1/R_n$$

Resistor (Cont.)

- The resistance of any material is due primarily to four factors:
  1. Material
  2. Length
  3. Cross-sectional area
  4. Temperature of the material



- The first three elements are related by the following basic equation for resistance:
  R=ρ*(l/A)

- Conductance, or electrical conductance, measures a material's ability to conduct electricity, essentially showing how easily an electrical current can flow through it. It is the opposite of electrical resistance, mathematically expressed as 1/R.

Examples

EXAMPLE 15 Determine the resistance of 100 ft of #28 copper telephone wire if the diameter is 0.0126 in.

"#28" refers to the American Wire Gauge (AWG)

Solution: Unit conversions:

$$l = 100\ \cancel{\text{ft}}\left(\frac{12\ \cancel{\text{in.}}}{1\ \cancel{\text{ft}}}\right)\left(\frac{2.54\ \text{cm}}{1\ \cancel{\text{in.}}}\right) = 3048\ \text{cm}$$

1 ft = 30.48cm

$$d = 0.0126\ \text{in.}\left(\frac{2.54\ \text{cm}}{1\ \text{in.}}\right) = 0.032\ \text{cm}$$

1 inch= 2.54cm

Therefore,

$$A = \frac{\pi d^2}{4} = \frac{(3.1416)(0.032\ \text{cm})^2}{4} = 8.04 \times 10^{-4}\ \text{cm}^2$$

$$R = \rho\frac{l}{A} = \frac{(1.723 \times 10^{-6}\ \Omega\text{-cm})(3048\ \text{cm})}{8.04 \times 10^{-4}\ \text{cm}^2} \cong \mathbf{6.5\ \Omega}$$

Using the units for circular wires and Table 2 for the area of a #28 wire, we find

$$R = \rho\frac{l}{A} = \frac{(10.37\ \text{CM-}\Omega/\text{ft})(100\ \text{ft})}{159.79\ \text{CM}} \cong \mathbf{6.5\ \Omega}$$

$$d = 0.0126\ \text{in} \times 1000\frac{\text{mils}}{\text{in}} = 12.6\ \text{mils}$$

The area (A) in circular mils is the square of the diameter in mils: A=d^2=(12.6 mils)2=158.76 CM

EXAMPLE 13

a. Determine the conductance of a 1 Ω, a 50 kΩ, and a 10 MΩ resistor.

b. How does the conductance level change with increase in resistance?

Solution: Eq. (12):

a. 1Ω: $G = \frac{1}{R} = \frac{1}{1\ \Omega} = \mathbf{1\ S}$

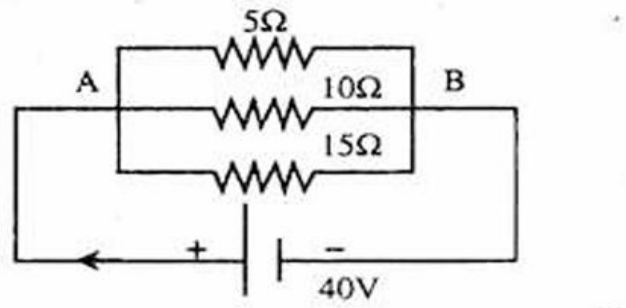
$50\ \text{k}\Omega$: $G = \frac{1}{R} = \frac{1}{50\ \text{k}\Omega} = \frac{1}{50 \times 10^3\Omega} = 0.02 \times 10^{-3}\text{S} = \mathbf{0.02\ mS}$

$10\ \text{M}\Omega$: $G = \frac{1}{R} = \frac{1}{10\ \text{M}\Omega} = \frac{1}{10 \times 10^6\Omega} = 0.1 \times 10^{-6}\text{S} = \mathbf{0.1\ \mu S}$

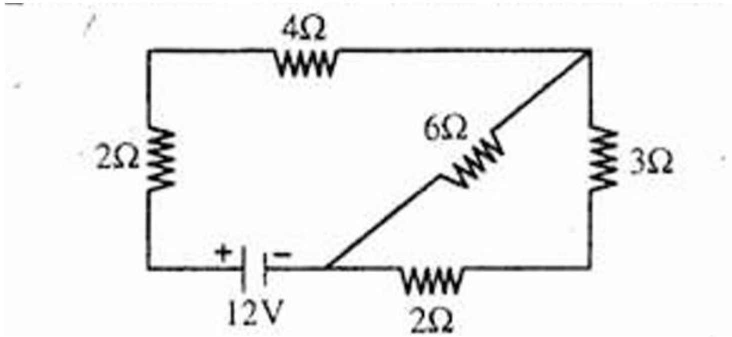
b. The conductance level decreases rapidly with significant increase in resistance levels.

Examples (Cont.)

- ❑ Find out the resistance of the metal wire of 2 m length and 0.6 mm in the diameter if its resistivity happens to be 7.065 × 10^-8Ωm.
- ❑ Find the resistivity of the material of the conductor of resistance 5Ω, area of cross-section 60cm^2 and length equal to 50 cm.
- ❑ Solve this and calculate the current in R3.

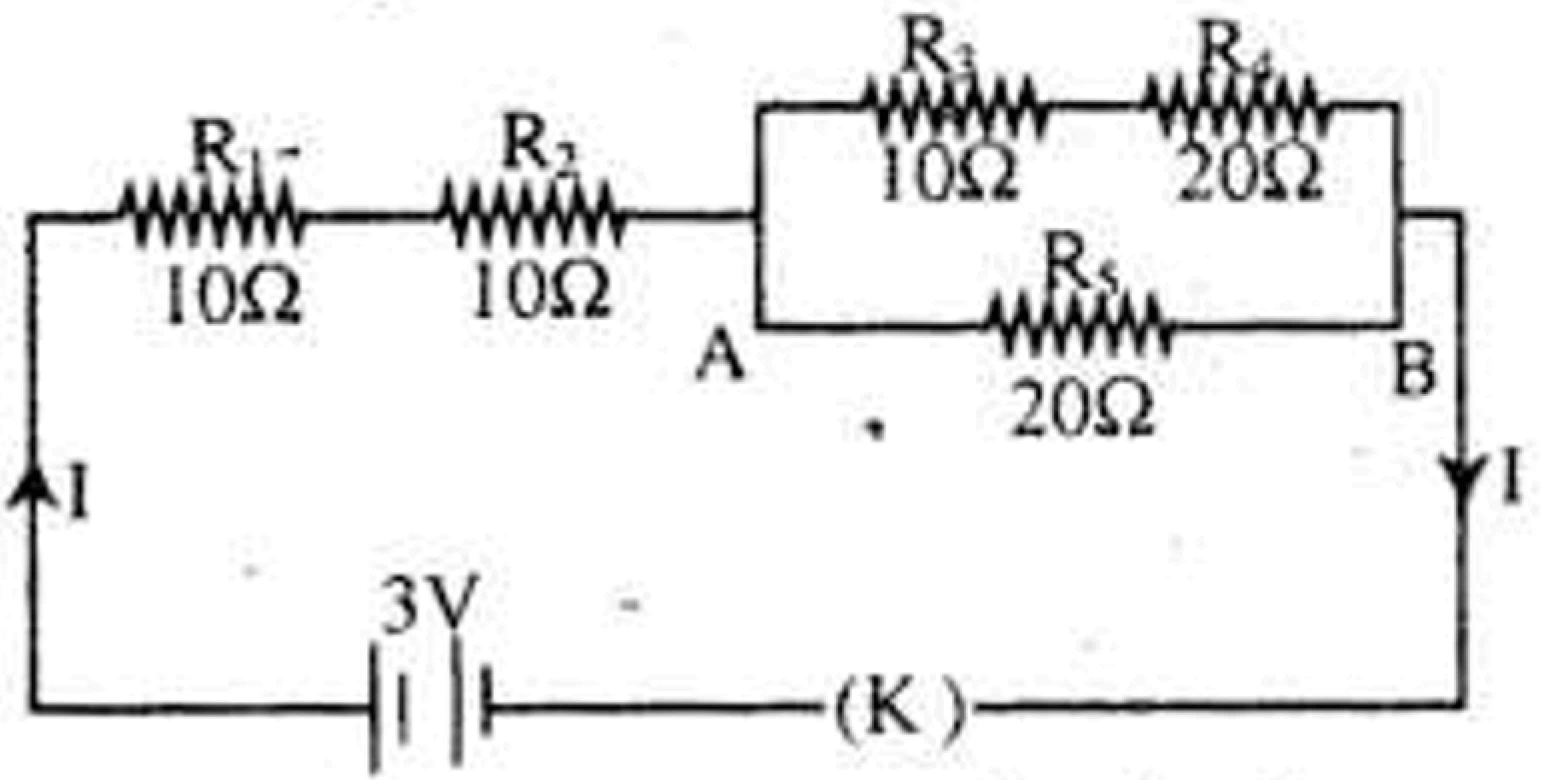


- ❑ Solve this and calculate the current value in 6Ω.

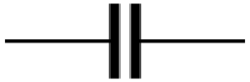


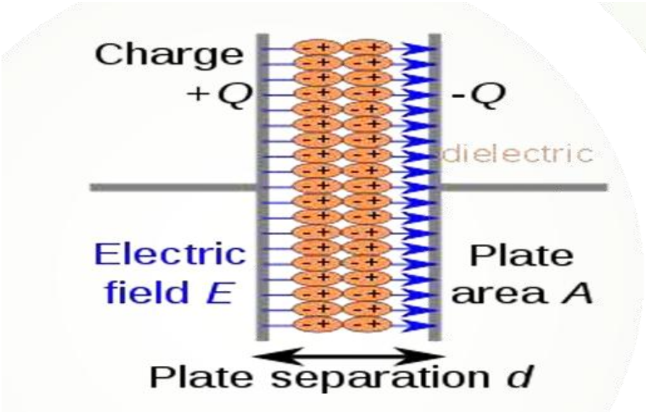
Examples (Cont.)

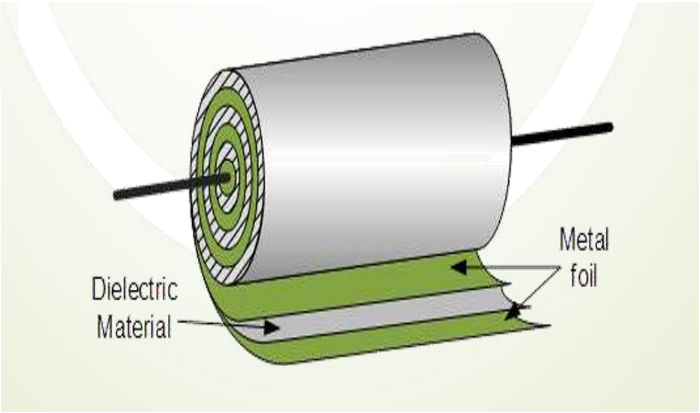
- **Solve this and calculate the current value in R5.**



Capacitor

- Device that store electric charges.
- Consists of two plates.
- Separated by an insulator. Non-conductive region can either be a vacuum or an electrical insulator known as dielectric. Such as glass, air, paper, semi-conductor.
- Variety of sizes and shapes.
- Symbol: 
- Types of Capacitor:
  a. Parallel plate capacitor.
  b. Cylindrical capacitor.





Capacitor (Cont.)

- Ability to store charge.
- Unit of capacitance is Farad(F).
- Capacitance values are normally smaller, such as µF, nF or pF.
- $C=Q/V$
- Capacitance of capacitor is directly proportional to Charge store on that capacitor and inversely proportional to Voltage.

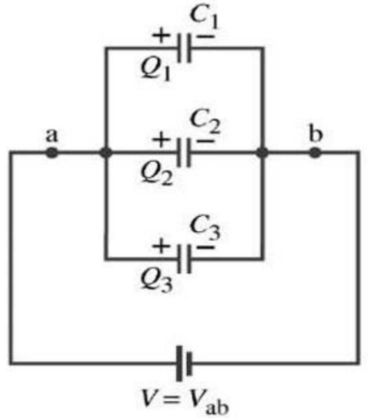
Capacitor (Cont.)

- **Capacitors In Parallel:**
  1. Voltage remain same.
  2. The equivalent capacitance of two capacitors connected in parallel is the sum of the individual capacitances.
- **Capacitors In Series:**
  1. Charge remain same.
  2. The reciprocal of the equivalent capacitance of two capacitors connected in series is the sum of the reciprocals of the individual capacitances.

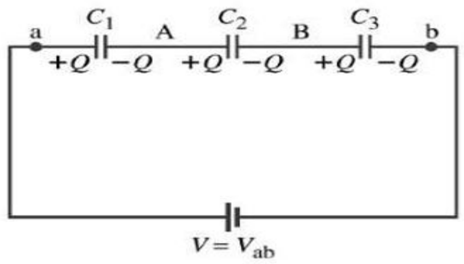
$$Q = Q_1 + Q_2 + Q_3$$
$$= C_1V + C_2V + C_3V$$
$$= (C_1 + C_2 + C_3)V$$
$$= C_{eq}V$$

Capacitors in Parallel:

$$C_{eq} = C_1 + C_2 + C_3 + \ldots$$



$$V = V_1 + V_2 + V_3$$
$$= \frac{Q}{C_1} + \frac{Q}{C_2} + \frac{Q}{C_3}$$
$$= Q\left(\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3}\right)$$
$$= \frac{Q}{C_{eq}}$$



For n capacitors in series:

$$\frac{1}{C_{eq}} = \frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \ldots$$

Examples

A parallel plate capacitor has square plates of side 5 cm and separated by a distance of 1 mm. (a) Calculate the capacitance of this capacitor. (b) If a 10 V battery is connected to the capacitor, what is the charge stored in any one of the plates? (The value of $\varepsilon_o = 8.85 \times 10^{-12}$ Nm^2 C^{-2})

Solution

(a) The capacitance of the capacitor is

$$C = \frac{\varepsilon_0 A}{d} = \frac{8.85 \times 10^{-12} \times 25 \times 10^{-4}}{1 \times 10^{-3}}$$

$= 221.2 \times 10^{-13}$ F

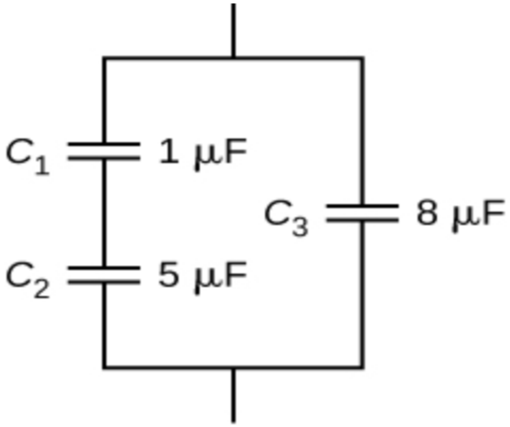
C = $22\,.\,12 \times 10^{-12}$ F = 22 .12 pF

(b) The charge stored in any one of the plates is Q = CV, Then

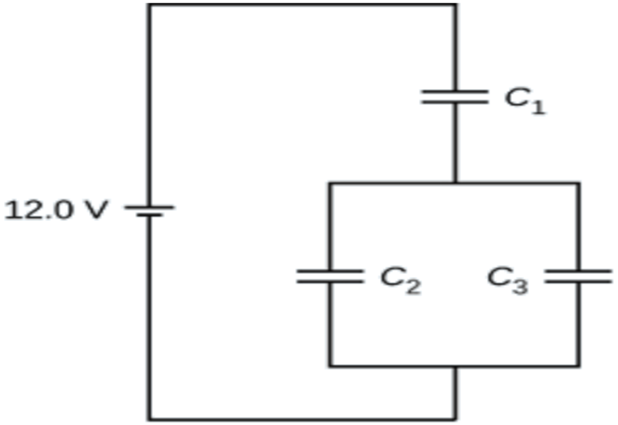
$= 22\,.\,12 \times 10^{-12} \times 10 = 221.2 \times 10^{-12}$ C = 221.2 pC

Examples (Cont.)

- Solve this



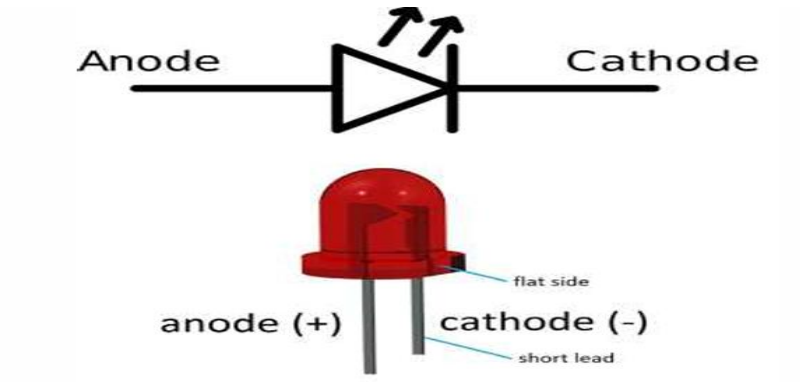
- Determine the net capacitance C of the Capacitor combination when the capacitances are C1=12.0μF, C2=2.0μF, and C3=4.0μF. When a 12V potential difference is maintained across the combination, find the charge and the Voltage across each capacitor.

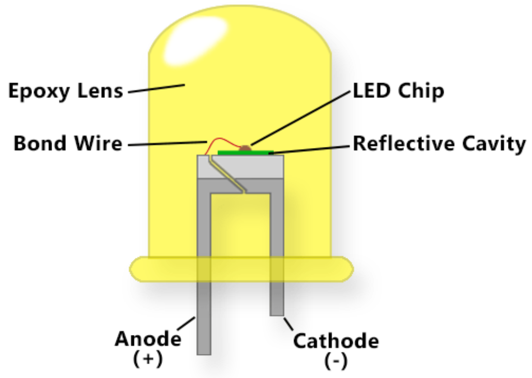


The charging and Discharging Phase of a Capacitor

LIGHT EMITTING DIODE(LED)

- A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it.

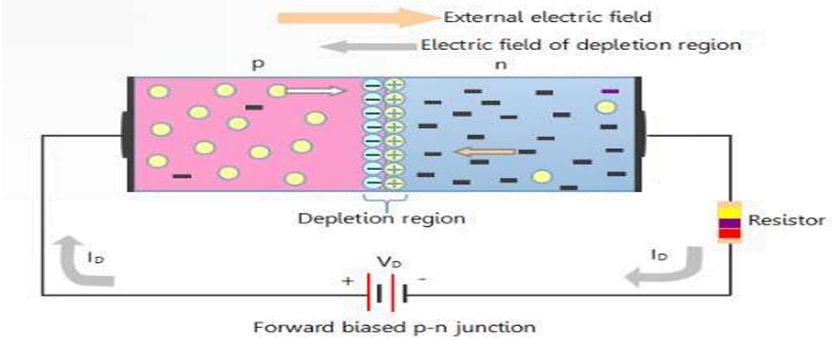




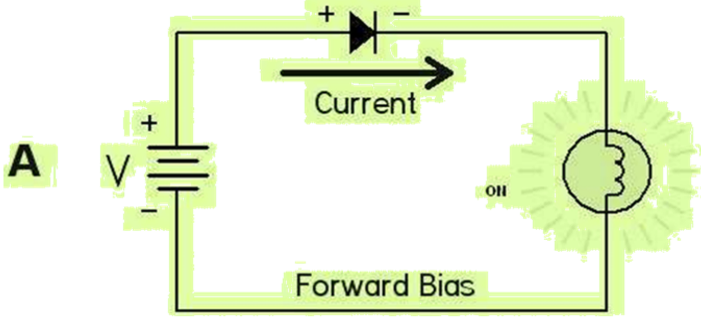
- LEDs are current-dependent devices with the forward voltage and forward current dependent on the semiconductor materials. Typically, the forward voltage is between 1.2 to 3.6 V, and the forward current is between 10 to 30 mA.

How LED works

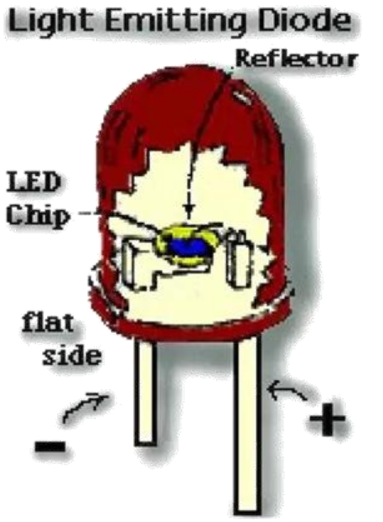
- **Forward Bias: When voltage is applied across a diode in such a way that the diode allows current and the potential barrier reduced, the diode is said to be forward-biased.**

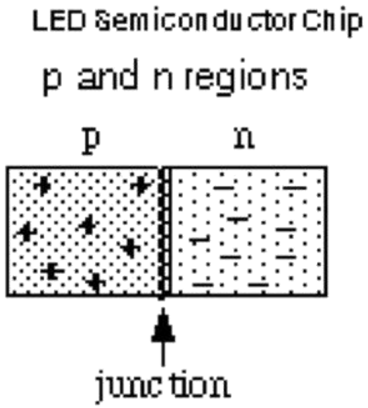


- **Under Forward bias, the depletion region reduce. After go current through the diode.**



How LED works (Cont.)



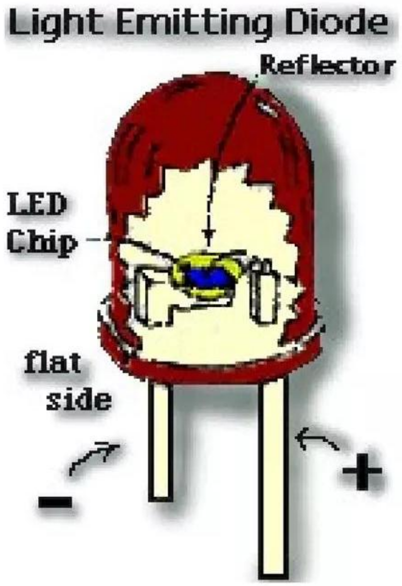


When sufficient voltage is applied to the chip across the leads of the LED, electrons can move easily in only one direction across the *junction* between the *p* and *n* regions.

In the *p region* there are many more positive than negative charges.

When a voltage is applied and the current starts to flow, electrons in the *n region* have sufficient energy to move across the junction into the *p region.*

How LED works (Cont.)



Each time an electron *recombines* with a positive charge, electric potential energy is converted into electromagnetic energy.

For each recombination of a negative and a positive charge, a quantum of electromagnetic energy is emitted in the form of a photon of light with a frequency characteristic of the semi-conductor material (usually a combination of the chemical elements gallium, arsenic and phosphorus)..

Thank you for hearing with patience

Department of Computer Science and Engineering
Daffodil International University

Capacitor

Course Title: Embedded Systems and IoT
Course Code: CSE233

Prepared By:
Jul Jalal Al-Mamur Sayor (JMS)
Lecturer, Dept. of CSE, DIU

Today's Lecture

1. Capacitor's Basic Math
2. Charging and Discharging Phase of Capacitor

Capacitor

- Device that store electric charges.
- C=Q/V
- Capacitance of capacitor is directly proportional to Charge store on that capacitor and inversely proportional to Voltage.

EXAMPLE 1

a. If 82.4×10^{14} electrons are deposited on the negative plate of a capacitor by an applied voltage of 60 V, find the capacitance of the capacitor.
b. If 40 V are applied across a 470 μF capacitor, find the charge on the plates.

Solutions:

a. First find the number of coulombs of charge as follows:

$$82.4 \times 10^{14} \cancel{\text{electrons}}\left(\frac{1\text{ C}}{6.242 \times 10^{18} \cancel{\text{electrons}}}\right) = 1.32\text{ mC}$$

- The charge of one electron (or proton) is 1.602×10^{-19} C.

- Therefore, to find how many electrons make up one coulomb:

$$\text{Number of electrons in 1 C} = \frac{1\text{ C}}{1.602 \times 10^{-19}\text{ C/electron}} \approx 6.242 \times 10^{18}\text{ electrons}$$

Capacitor electric field

and then

$$C = \frac{Q}{V} = \frac{1.32\ \text{mC}}{60\ \text{V}} = \mathbf{22\ \mu F} \qquad \text{(a standard value)}$$

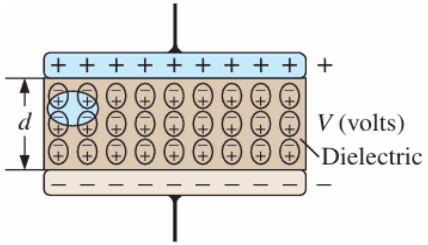
b. Applying Eq. (6) gives

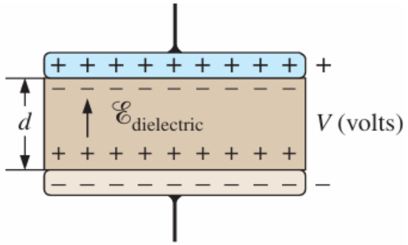
$$Q = CV = (470\ \mu\text{F})(40\ \text{V}) = \mathbf{18.8\ mC}$$

- **The electric field strength between the plates is determined by the voltage across the plates and the distance between the plates.**

$$\boxed{\mathscr{E} = \frac{V}{d}}$$

$\mathscr{E}$ = volts/m (V/m)
V = volts (V)
d = meters (m)





Capacitor permittivity

- **Relative permittivity (dielectric constant) €r of various dielectrics.**

Dielectric	ϵ_r (Average Values)
Vacuum	1.0
Air	1.0006
Teflon®	2.0
Paper, paraffined	2.5
Rubber	3.0
Polystyrene	3.0
Oil	4.0
Mica	5.0
Porcelain	6.0
Bakelite®	7.0
Aluminum oxide	7
Glass	7.5
Tantalum oxide	30
Ceramics	20–7500
Barium-strontium titanite (ceramic)	7500.0

$$\boxed{C = \epsilon_o \epsilon_r \frac{A}{d}}$$

$$\boxed{C = 8.85 \times 10^{-12} \epsilon_r \frac{A}{d}}$$

$$\frac{C = \epsilon \frac{A}{d}}{C_o = \epsilon_o \frac{A}{d}} \Rightarrow \frac{C}{C_o} = \frac{\epsilon}{\epsilon_o} = \epsilon_r$$

$$\boxed{C = \epsilon_r C_o}$$

Capacitor Basic Math

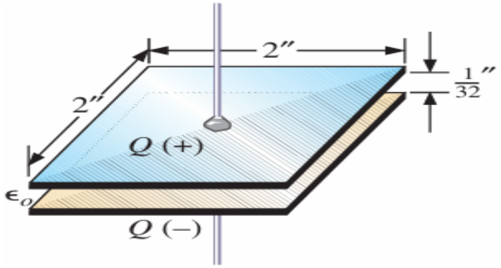


FIG. 10
Air capacitor for Example 3.

EXAMPLE 3 For the capacitor in Fig. 10:

a. Find the capacitance.
b. Find the strength of the electric field between the plates if 48 V are applied across the plates.
c. Find the charge on each plate.

Solutions:

a. First, the area and the distance between the plates must be converted to the SI system as required by Eq. (11):

$$d = \frac{1}{32}\,\cancel{\text{in.}}\left(\frac{1\text{ m}}{39.37\,\cancel{\text{in.}}}\right) = 0.794\text{ mm}$$

and $$A = (2\,\cancel{\text{in.}})(2\,\cancel{\text{in.}})\left(\frac{1\text{ m}}{39.37\,\cancel{\text{in.}}}\right)\left(\frac{1\text{ m}}{39.37\,\cancel{\text{in.}}}\right) = 2.581 \times 10^{-3}\text{ m}^2$$

Eq. (11):

$$C = 8.85\times10^{-12}\,\epsilon_r\frac{A}{d} = 8.85\times10^{-12}(1)\frac{(2.581\times10^{-3}\text{m}^2)}{0.794\text{ mm}} = \mathbf{28.8\text{ pF}}$$

b. The electric field between the plates is determined by Eq. (7):

$$\mathscr{E} = \frac{V}{d} = \frac{48\text{ V}}{0.794\text{ mm}} = \mathbf{60.5\text{ kV/m}}$$

c. The charge on the plates is determined by Eq. (6):

$$Q = CV = (28.8\text{ pF})(48\text{ V}) = \mathbf{1.38\text{ nC}}$$

Capacitor Basic Math

EXAMPLE 4

a. Insert a ceramic dielectric with an ϵ_r of 250 between the plates of the capacitor in Fig. 10. Then determine the new level of capacitance. Compare your results to the solution in Example 3.
b. Find the resulting electric field strength between the plates, and compare your answer to the result in Example 3.
c. Determine the charge on each of the plates, and compare your answer to the result in Example 3.

Solutions:

a. From Eq. (12), the new capacitance level is

$$C = \epsilon_r C_o = (250)(28.8\text{ pF}) = \mathbf{7200\ pF = 7.2\ nF = 0.0072\ \mu F}$$

which is *significantly higher* than the level in Example 3.

b. $$\mathscr{E} = \frac{V}{d} = \frac{48\text{ V}}{0.794\text{ mm}} = \mathbf{60.5\ kV/m}$$

Since the applied voltage and the distance between the plates did not change, *the electric field between the plates remains the same.*

c. $$Q = CV = (7200\text{ pF})(48\text{ V}) = \mathbf{345.6\ nC = 0.35\ \mu C}$$

Capacitor

Types of Capacitors:

1. Fixed Capacitor: A capacitor with a constant value of capacitance that cannot be changed.
Use: Found in almost every electronic circuit (like fans, TVs, chargers).
Analogy: Like a cup with a fixed size, it always holds the same amount of water (energy).
Examples: Ceramic capacitor, Electrolytic capacitor.
2. Variable Capacitor: A capacitor whose capacitance can be adjusted manually or automatically.
Use: Common in radios and tuning circuits, where the signal needs to be fine-tuned.
Analogy: Like a measuring cup with adjustable sides, you can change how much water (energy) it holds.
Examples: Air-gap variable capacitor

FIG. 11
Symbols for the capacitor: (a) fixed; (b) variable.

Capacitor(Cont.)

- **ESR:** There is another quantity of importance when defining the complete capacitive equivalent: the equivalent series resistance (ESR). It is a quantity of such importance to the design of switching and linear power supplies that it holds equal weight with the actual capacitance level.

Capacitor Charging

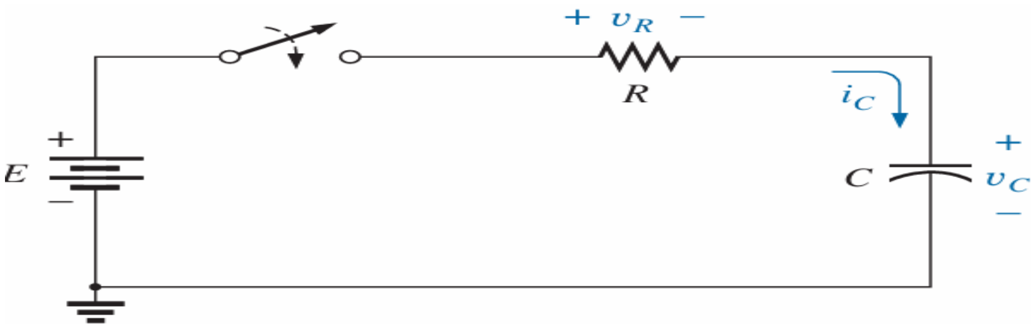


FIG. 26
Basic R-C charging network.

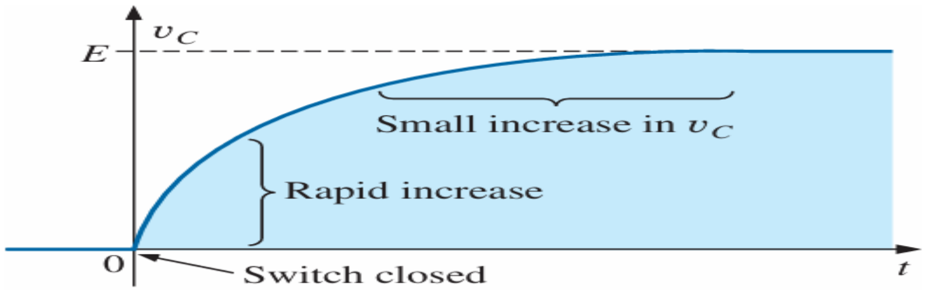


FIG. 27
v_C during the charging phase.

- **Since the voltage across the plates is directly related to the charge on the plates by V=Q/C, a plot of the voltage across the capacitor will have the same shape as a plot of the charge on the plates over time.**

Capacitor Charging & Discharging time constant chart

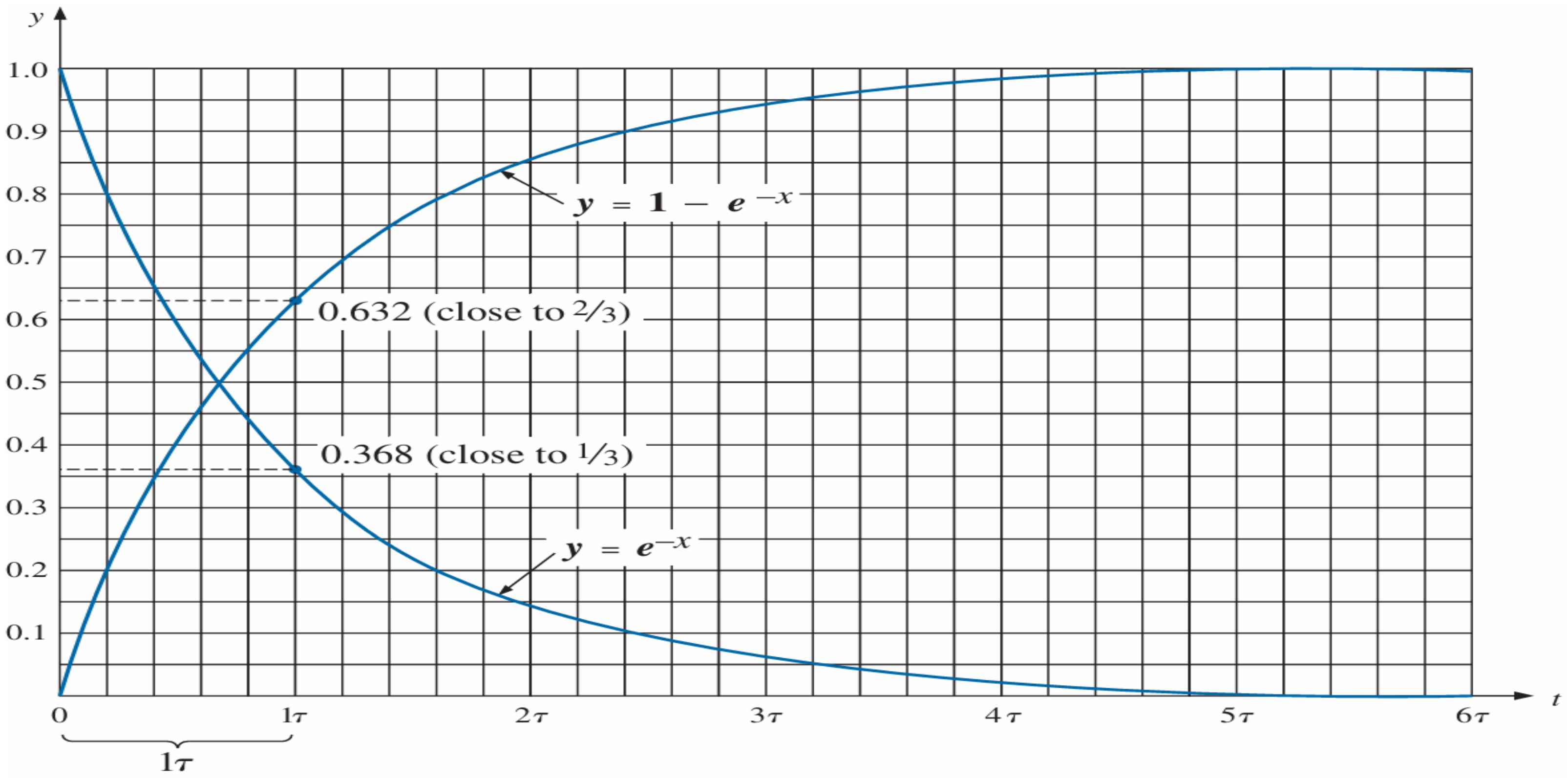


FIG. 28
Universal time constant chart.

Capacitor Charging equation

- **In charging, voltage through capacitor will be,**

$$\boxed{v_C = E(1 - e^{-t/\tau})}_{\text{charging}} \qquad \text{(volts, V)}$$

- **The factor τ, called the time constant of the network, has the units of time, as shown below using some of the basic equations introduced earlier in this text:**

$$\boxed{\tau = RC} \qquad \text{(time, s)}$$

$$\tau = RC = \left(\frac{V}{I}\right)\left(\frac{Q}{V}\right) = \left(\frac{\cancel{V}}{\cancel{Q}/t}\right)\left(\frac{\cancel{Q}}{\cancel{V}}\right) = t \text{ (seconds)}$$

Capacitor voltage graph plot

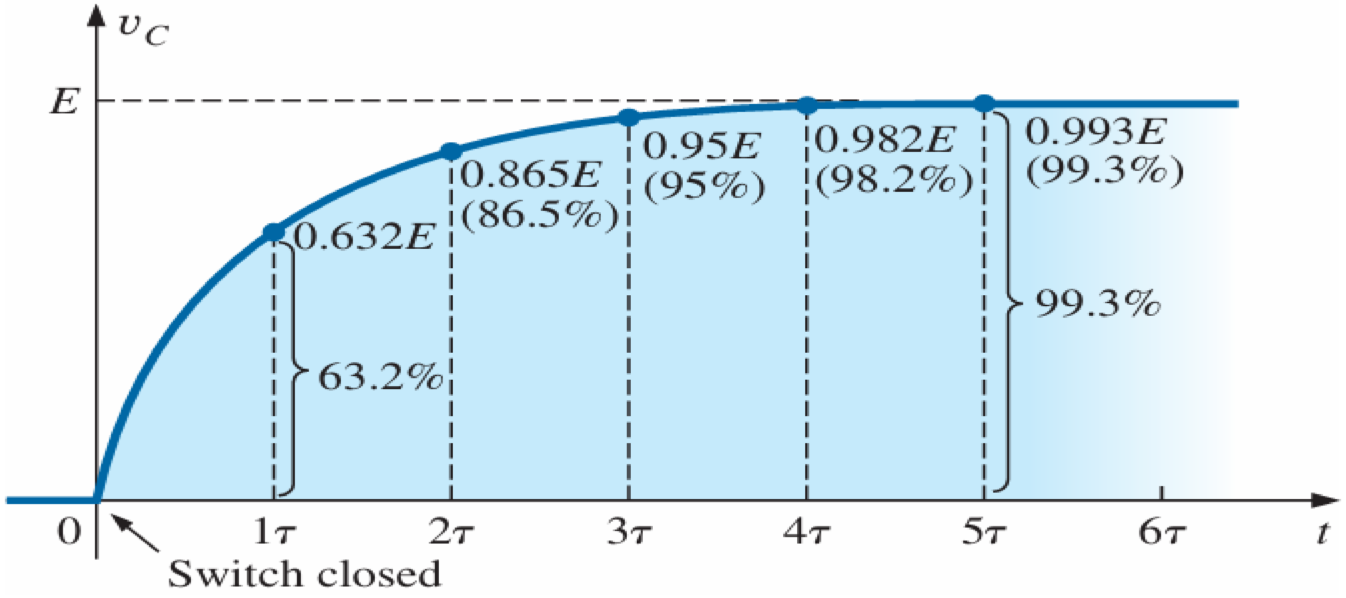


FIG. 29

Plotting the equation $v_C = E(1 - e^{-t/\tau})$ versus time (t).

Capacitor charging voltage calculation

In Eq. (13), if we substitute $t = 0$ s, we find that

$$e^{-t/\tau} = e^{-0/\tau} = e^{-0} = \frac{1}{e^0} = \frac{1}{1} = 1$$

and $$\upsilon_C = E(1 - e^{-t/\tau}) = E(1 - 1) = \mathbf{0\ V}$$

as appearing in the plot in Fig. 29.

It is important to realize at this point that the plot in Fig. 29 is not against simply time but against τ, the time constant of the network. If we want to know the voltage across the plates after one time constant, we simply plug $t = 1\tau$ into Eq. (13). The result is

$$e^{-t/\tau} = e^{-\tau/\tau} = e^{-1} \cong 0.368$$

and $$\upsilon_C = E(1 - e^{-t/\tau}) = E(1 - 0.368) = \mathbf{0.632}\boldsymbol{E}$$

as shown in Fig. 29.

At $t = 2\tau$

$$e^{-t/\tau} = e^{-2\tau/\tau} = e^{-2} \cong 0.135$$

and $$\upsilon_C = E(1 - e^{-t/\tau}) = E(1 - 0.135) \cong \mathbf{0.865}\boldsymbol{E}$$

as shown in Fig. 29.

As the number of time constants increases, the voltage across the capacitor does indeed approach the applied voltage.

At $t = 5\tau$

$$e^{-t/\tau} = e^{-5\tau/\tau} = e^{-5} \cong 0.007$$

and $$\upsilon_C = E(1 - e^{-t/\tau}) = E(1 - 0.007) = \mathbf{0.993}\boldsymbol{E} \cong E$$

In fact, we can conclude from the results just obtained that

the voltage across a capacitor in a dc network is essentially equal to the applied voltage after five time constants of the charging phase have passed.

Capacitor charging current calculation

- **In charging, the equation for the current is,**

$$\boxed{i_C = \frac{E}{R}e^{-t/\tau}}_{\text{charging}} \qquad \text{(amperes, A)}$$

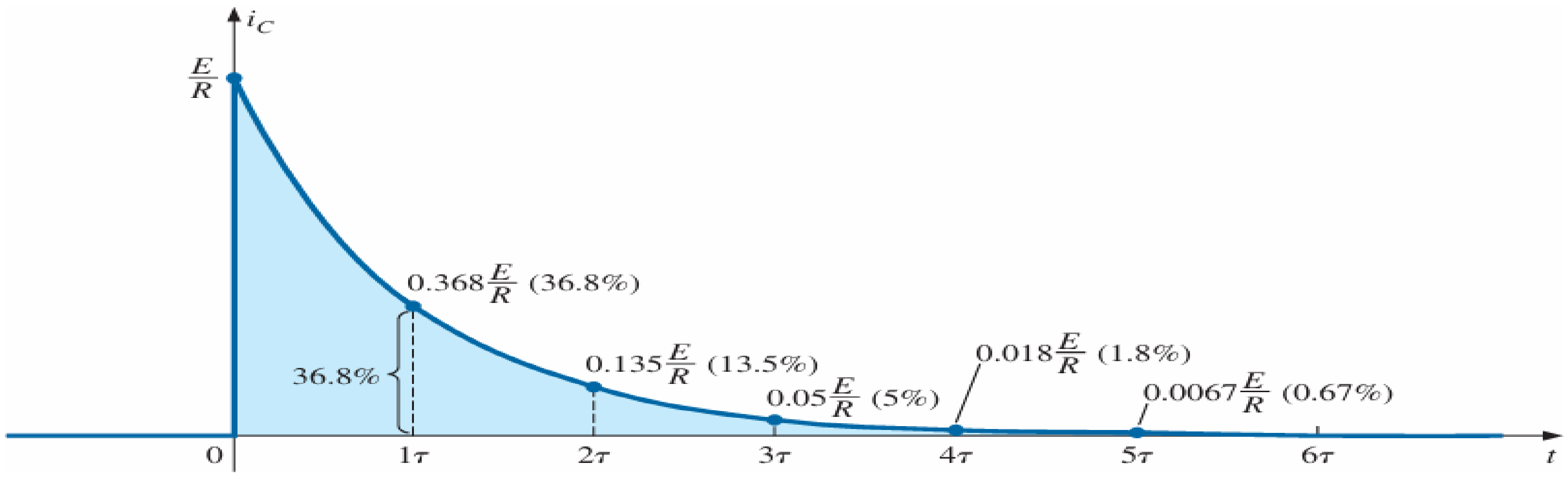


FIG. 30

Plotting the equation $i_C = \frac{E}{R}e^{-t/\tau}$ versus time (t).

Capacitor charging current calculation

At $t = 0$ s

$$e^{-t/\tau} = e^{-0} = 1$$

and
$$i_C = \frac{E}{R}e^{-t/\tau} = \frac{E}{R}(1) = \boldsymbol{\frac{E}{R}}$$

At $t = 1\tau$

$$e^{-t/\tau} = e^{-\tau/\tau} = e^{-1} \cong 0.368$$

and
$$i_C = \frac{E}{R}e^{-t/\tau} = \frac{E}{R}(0.368) = \mathbf{0.368}\,\boldsymbol{\frac{E}{R}}$$

In general, Fig. 30 clearly reveals that

the current of a capacitive dc network is essentially zero amperes after five time constants of the charging phase have passed.

It is also important to recognize that

during the charging phase, the major change in voltage and current occurs during the first time constant.

Capacitor Charging resistor voltage calculation

- Since the resistor and the capacitor in Fig. 26 are in series, the current through the resistor is the same as that associated with the capacitor. Th voltage across the resistor can be determined by using Ohm's law in the following manner:

$$\upsilon_R = i_R R = i_C R$$

$$\upsilon_R = \left(\frac{E}{R} e^{-t/\tau}\right) R$$

$$\boxed{\upsilon_R = E e^{-t/\tau}}_{\text{charging}} \qquad \text{(volts, V)}$$

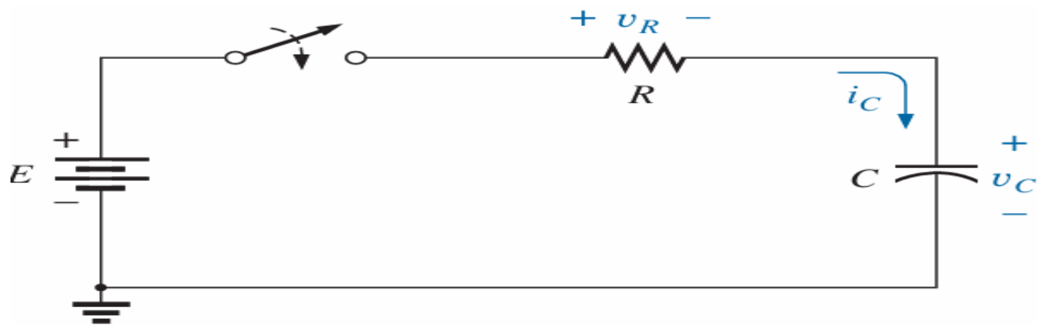


FIG. 26
Basic R-C charging network.

Capacitor Charging resistor voltage graph

- That will be same as Ic

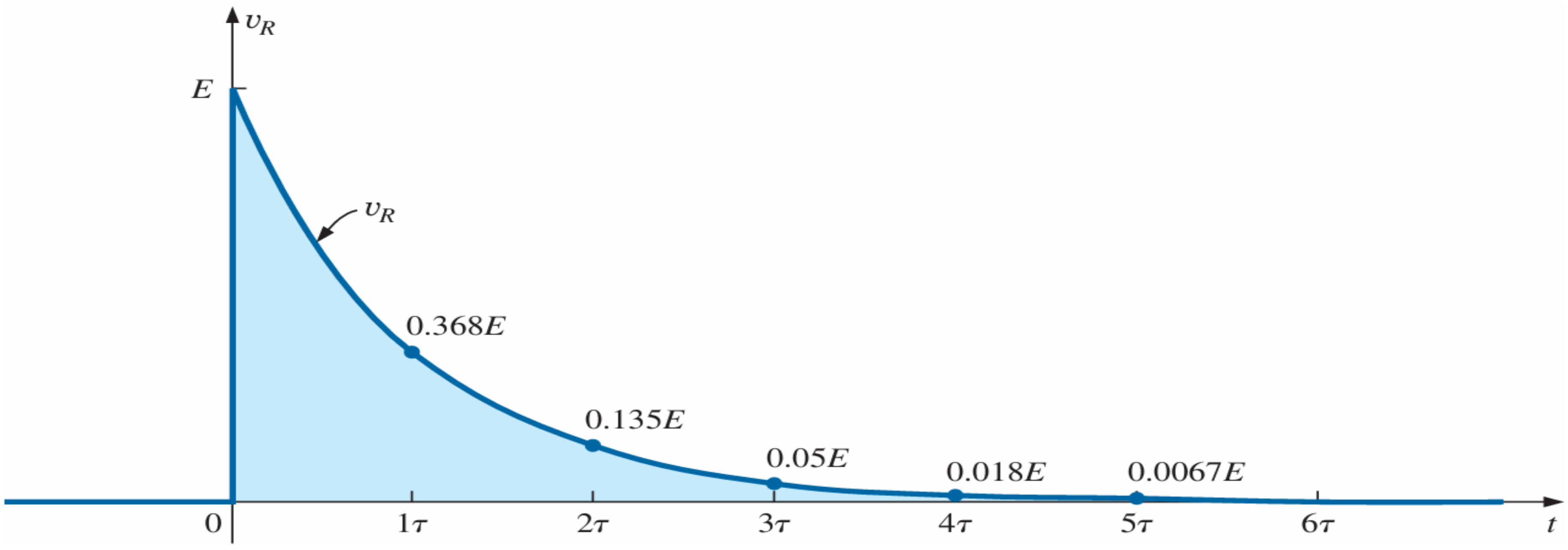


FIG. 33
Plotting the equation $v_R = Ee^{-t/\tau}$ versus time (t).

Capacitor Charging Math Example-6 Page 425-426

EXAMPLE 6 For the circuit in Fig. 35:

a. Find the mathematical expression for the transient behavior of v_C, i_C, and v_R if the switch is closed at $t = 0$ s.
b. Plot the waveform of v_C versus the time constant of the network.
c. Plot the waveform of v_C versus time.
d. Plot the waveforms of i_C and v_R versus the time constant of the network.
e. What is the value of v_C at $t = 20$ ms?
f. On a practical basis, how much time must pass before we can assume that the charging phase has passed?
g. When the charging phase has passed, how much charge is sitting on the plates?
h. If the capacitor has a leakage resistance of 10,000 MΩ, what is the initial leakage current? Once the capacitor is separated from the circuit, how long will it take to totally discharge, assuming a linear (unchanging) discharge rate?

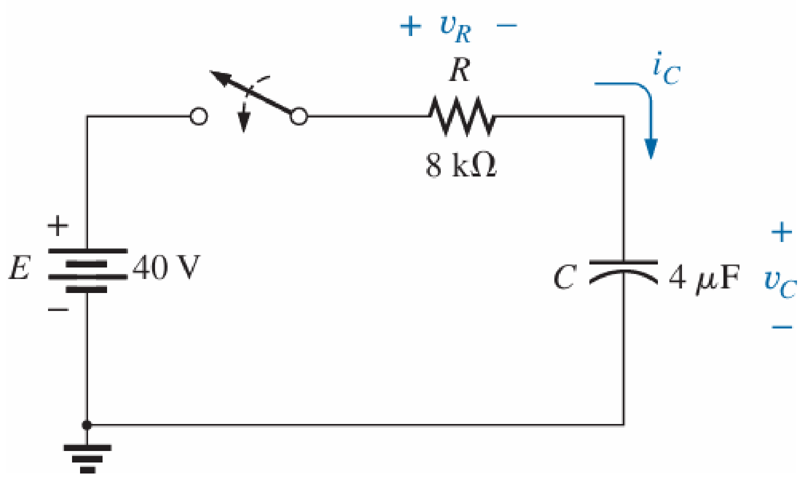


FIG. 35
Transient network for Example 6.

Capacitor Discharging

- For the voltage across the capacitor that is decreasing with time, the mathematical expression is:

$$\boxed{\upsilon_C = Ee^{-t/\tau}} \quad \text{discharging}$$

- For this circuit, the time constant t is defined by the same equation as used for the charging phase. That is,

$$\boxed{\tau = RC} \quad \text{discharging}$$

- Since the current decreases with time, it will have a similar format:

$$\boxed{i_C = \frac{E}{R}e^{-t/\tau}} \quad \text{discharging}$$

- For the voltage R has the same format:

$$\boxed{\upsilon_R = Ee^{-t/\tau}} \quad \text{discharging}$$

Capacitor Discharging Vc, Ic, Vr graph plot

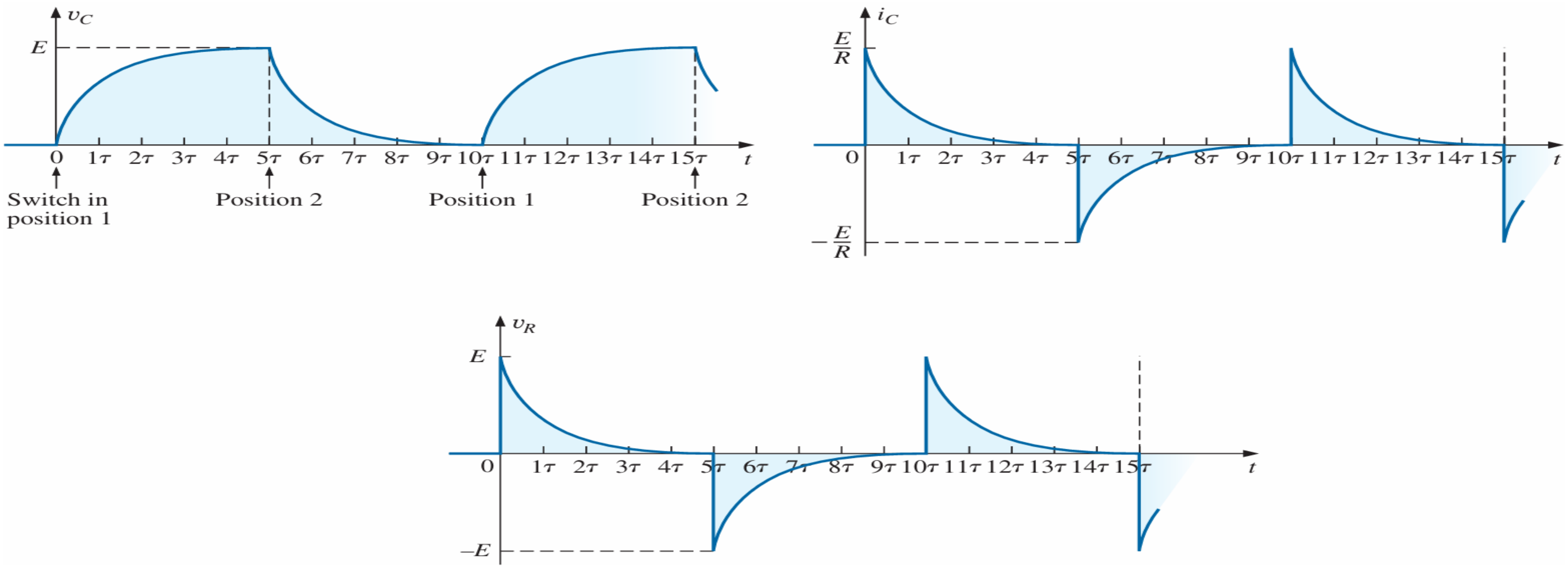


FIG. 40
v_C, i_C, and v_R for 5τ switching between contacts in Fig. 39(a).

Capacitor Discharging Example

EXAMPLE 7 Using the values in Example 6, plot the waveforms for v_C and i_C resulting from switching between contacts 1 and 2 in Fig. 39 every five time constants.

Solution: The time constant is the same for the charging and discharging phases. That is,

$$\tau = RC = (8\text{ k}\Omega)(4\ \mu\text{F}) = 32\text{ ms}$$

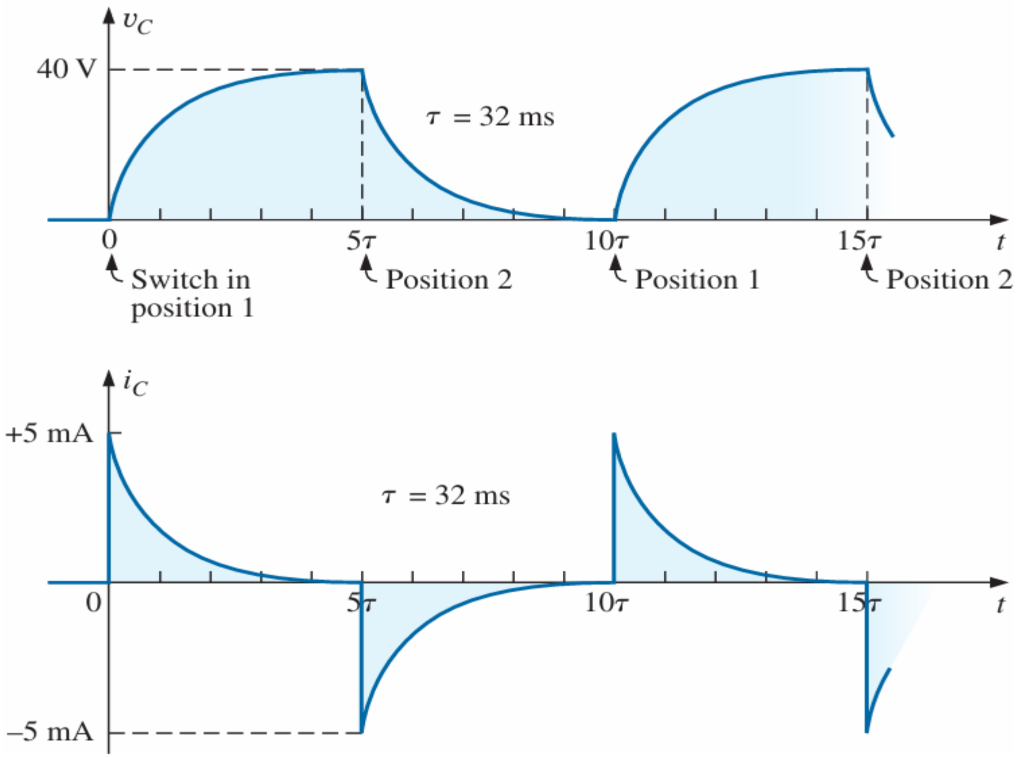
For the discharge phase, the equations are

$$v_C = Ee^{-t/\tau} = \mathbf{40\ V}e^{-t/32\text{ ms}}$$

$$i_C = -\frac{E}{R}e^{-t/\tau} = \frac{40\text{ V}}{8\text{ k}\Omega}e^{-t/32\text{ ms}} = \mathbf{-5\ mA}e^{-t/32\text{ ms}}$$

$$v_R = v_C = \mathbf{40\ V}e^{-t/32\text{ ms}}$$

A continuous plot for the charging and discharging phases appears in Fig. 41.



Example

EXAMPLE 8 For the circuit in Fig. 44:

a. Find the mathematical expressions for the transient behavior of the voltage v_C and the current i_C if the capacitor was initially uncharged and the switch is thrown into position 1 at $t = 0$ s.

b. Find the mathematical expressions for the voltage v_C and the current i_C if the switch is moved to position 2 at $t = 10$ ms. (Assume that the leakage resistance of the capacitor is infinite ohms; that is, there is no leakage current.)

c. Find the mathematical expressions for the voltage v_C and the current i_C if the switch is thrown into position 3 at $t = 20$ ms.

d. Plot the waveforms obtained in parts (a)–(c) on the same time axis using the defined polarities in Fig. 44.

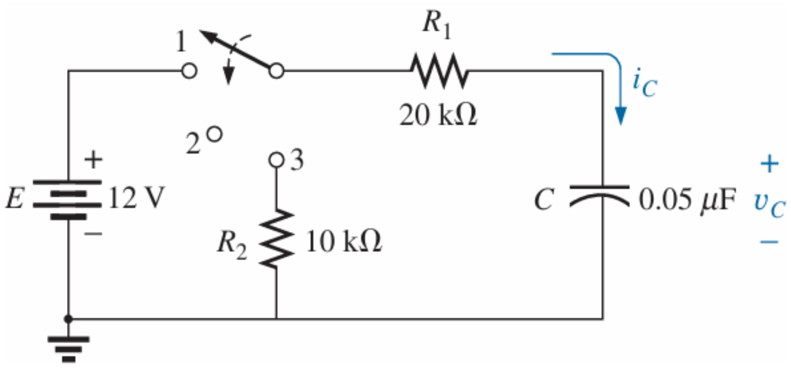


FIG. 44
Network to be analyzed in Example 8.

Solution

Solutions:

a. *Charging phase:*

$$\tau = R_1C = (20\text{ k}\Omega)(0.05\ \mu\text{F}) = 1\text{ ms}$$

$$\upsilon_C = E(1 - e^{-t/\tau}) = \mathbf{12\ V(1 - e^{-t/1\,ms})}$$

$$i_C = \frac{E}{R_1}e^{-t/\tau} = \frac{12\text{ V}}{20\text{ k}\Omega}e^{-t/1\text{ ms}} = \mathbf{0.6\ mA}e^{-t/1\,\mathbf{ms}}$$

b. *Storage phase:* At 10 ms, a period of time equal to 10τ has passed, permitting the assumption that the capacitor is fully charged. Since $R_{\text{leakage}} = \infty\ \Omega$, the capacitor will hold its charge indefinitely. The result is that both υ_C and i_C will remain at a fixed value:

$$\upsilon_C = \mathbf{12\ V}$$

$$i_C = \mathbf{0\ A}$$

c. *Discharge phase* (using 20 ms as the new $t = 0$ s for the equations): The new time constant is

$$\tau' = RC = (R_1 + R_2)C = (20\text{ k}\Omega + 10\text{ k}\Omega)(0.05\ \mu\text{F}) = 1.5\text{ ms}$$

$$\upsilon_C = Ee^{-t/\tau'} = \mathbf{12\ V}e^{-t/1.5\,\mathbf{ms}}$$

$$i_C = -\frac{E}{R}e^{-t/\tau'} = -\frac{E}{R_1 + R_2}e^{-t/\tau'}$$

$$= -\frac{12\text{ V}}{20\text{ k}\Omega + 10\text{ k}\Omega}e^{-t/1.5\text{ ms}} = \mathbf{-0.4\ mA}e^{-t/1.5\,\mathbf{ms}}$$

d. See Fig. 45.

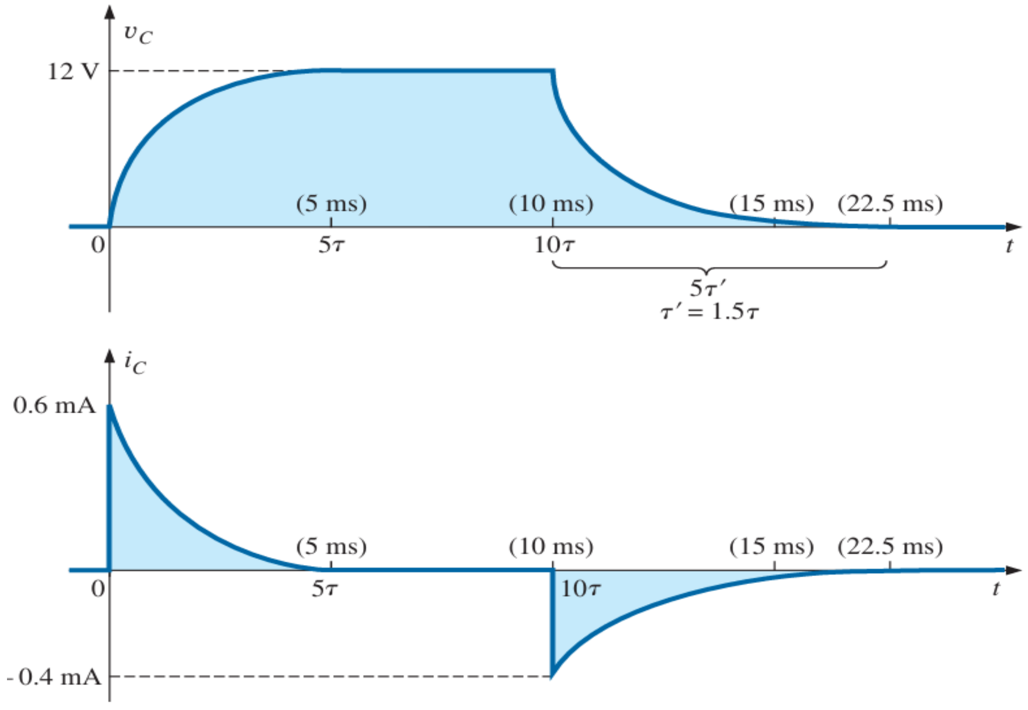


FIG. 45
υ_C and i_C for the network in Fig. 44.

Example

EXAMPLE 15 For the circuit in Fig. 68:

a. Find the total capacitance.
b. Determine the charge on each plate.
c. Find the voltage across each capacitor.

Solutions:

a. $$\frac{1}{C_T} = \frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3}$$
$$= \frac{1}{200 \times 10^{-6}\text{ F}} + \frac{1}{50 \times 10^{-6}\text{ F}} + \frac{1}{10 \times 10^{-6}\text{ F}}$$
$$= 0.005 \times 10^6 + 0.02 \times 10^6 + 0.1 \times 10^6$$
$$= 0.125 \times 10^6$$

and $$C_T = \frac{1}{0.125 \times 10^6} = \mathbf{8\ \mu F}$$

b. $$Q_T = Q_1 = Q_2 = Q_3$$
$$= C_T E = (8 \times 10^{-6}\text{ F})(60\text{ V}) = \mathbf{480\ \mu C}$$

c. $$V_1 = \frac{Q_1}{C_1} = \frac{480 \times 10^{-6}\text{ C}}{200 \times 10^{-6}\text{ F}} = \mathbf{2.4\ V}$$
$$V_2 = \frac{Q_2}{C_2} = \frac{480 \times 10^{-6}\text{ C}}{50 \times 10^{-6}\text{ F}} = \mathbf{9.6\ V}$$
$$V_3 = \frac{Q_3}{C_3} = \frac{480 \times 10^{-6}\text{ C}}{10 \times 10^{-6}\text{ F}} = \mathbf{48.0\ V}$$

and $E = V_1 + V_2 + V_3 = 2.4\text{ V} + 9.6\text{ V} + 48\text{ V} = \mathbf{60\ V}$ (checks)

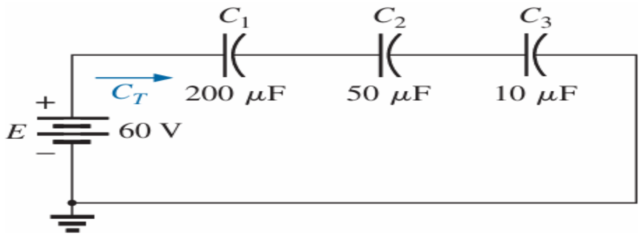


FIG. 68
Example 15.

Example

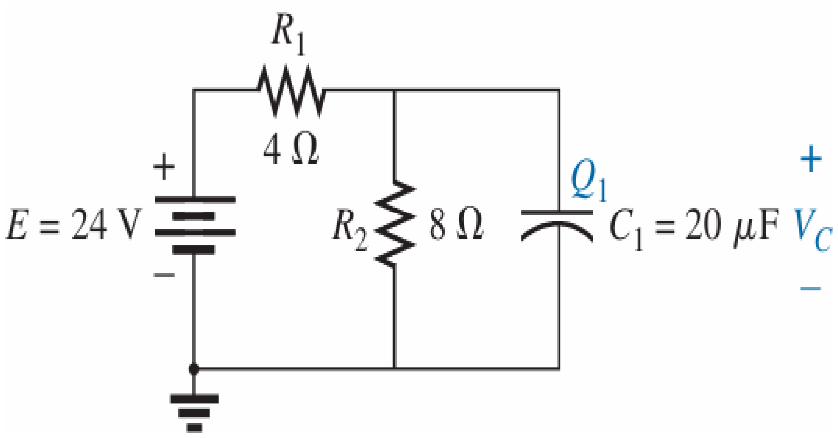


FIG. 72
Example 18.

EXAMPLE 18 Find the voltage across and the charge on capacitor C_1 in Fig. 72 after it has charged up to its final value.

Solution: As previously discussed, the capacitor is effectively an open circuit for dc after charging up to its final value (Fig. 73).

Therefore,

$$V_C = \frac{(8\ \Omega)(24\ \text{V})}{4\ \Omega + 8\ \Omega} = \mathbf{16\ V}$$

$$Q_1 = C_1 V_C = (20 \times 10^{-6}\ \text{F})(16\ \text{V}) = \mathbf{320\ \mu C}$$

Example

EXAMPLE 19 Find the voltage across and the charge on each capacitor of the network in Fig. 74(a) after each has charged up to its final value.

Solution: See Fig. 74(b). We have

$$V_{C_2} = \frac{(7\ \Omega)(72\ \text{V})}{7\ \Omega + 2\ \Omega} = \mathbf{56\ V}$$

$$V_{C_1} = \frac{(2\ \Omega)(72\ \text{V})}{2\ \Omega + 7\ \Omega} = \mathbf{16\ V}$$

$$Q_1 = C_1 V_{C_1} = (2 \times 10^{-6}\ \text{F})(16\ \text{V}) = \mathbf{32\ \mu C}$$

$$Q_2 = C_2 V_{C_2} = (3 \times 10^{-6}\ \text{F})(56\ \text{V}) = \mathbf{168\ \mu C}$$

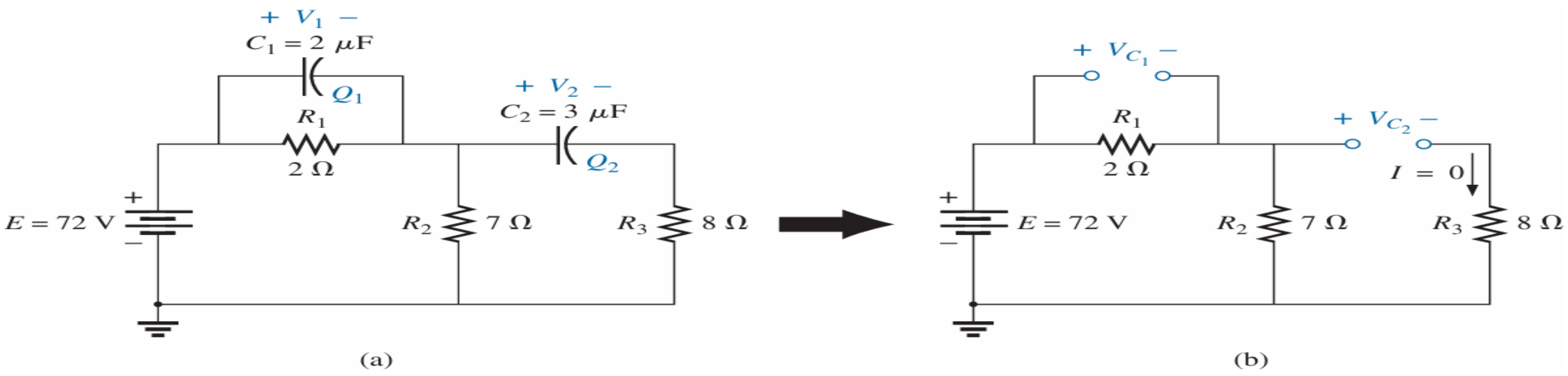


FIG. 74

Important Topics

- What is capacitor, difference from a battery.
- Usage/Application and advantages.
- Types of Capacitor.
- Basic maths of serial, parallel, charge and voltage calculation.
- Charging, discharging phase equation and curves.

"Understanding beats memorizing, learn the 'why' behind the 'what'."

Department of Computer Science and Engineering
Daffodil International University

Transistor & Mosfet

Course Title: Embedded Systems and IoT

Course Code: CSE233

Prepared By:

Bakhtiar Muiz

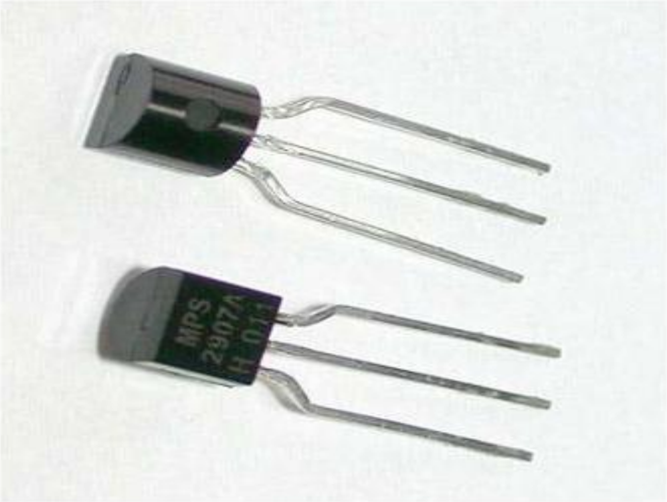
Lecturer, Dept. of CSE, DIU

Transistor

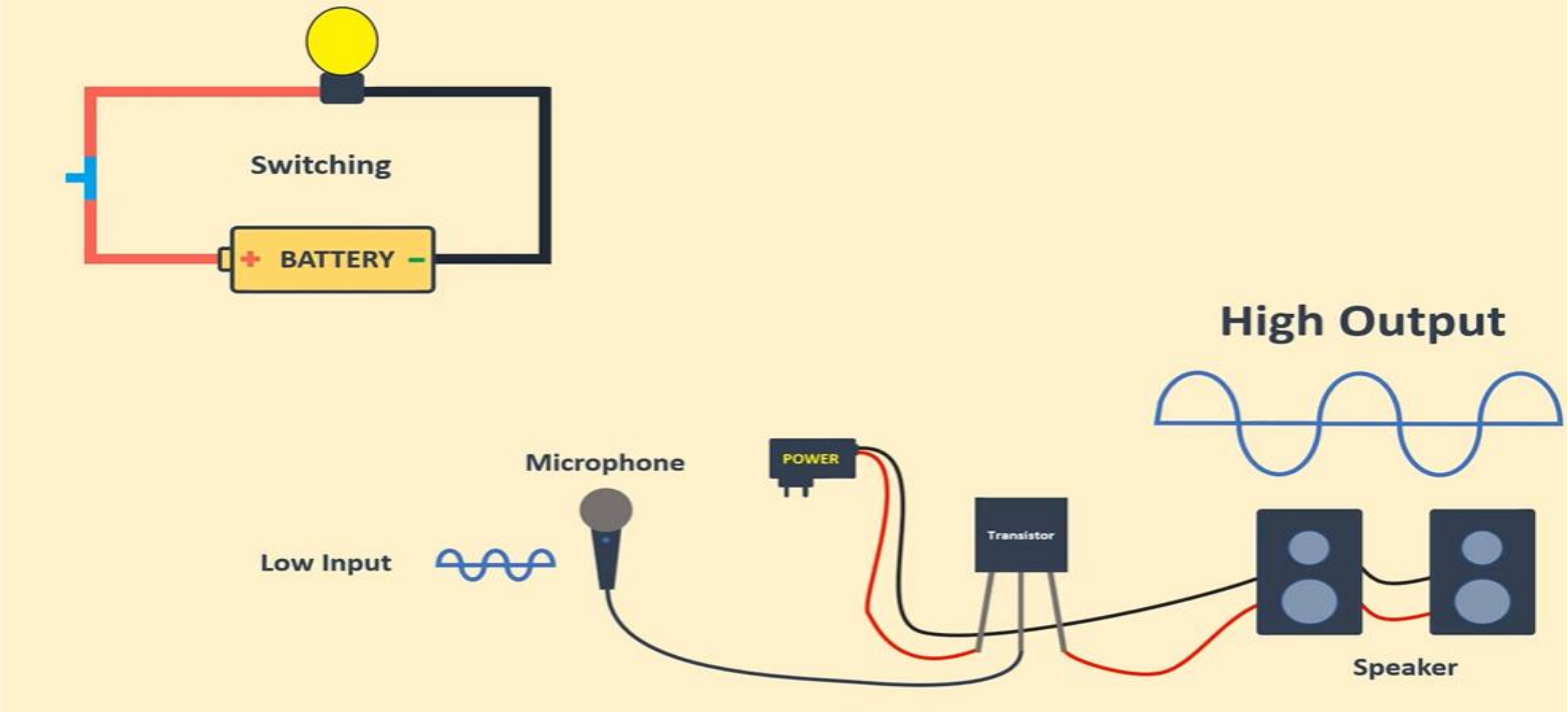
- A transistor is a type of semiconductor device that can be **used to conduct and insulate electric current or voltage.** A transistor basically **acts as a switch and an amplifier**. In simple words, a transistor is a miniature device that is used to control or regulate the flow of electronic signals.

- Semiconductors: ability to change from conductor to insulator
- Can either allow current or prohibit current to flow
- Useful as a switch, but also as an amplifier
- Essential part of many technological advances

A transistor is a semiconductor device used to amplify and switch electronic signals and electrical power.



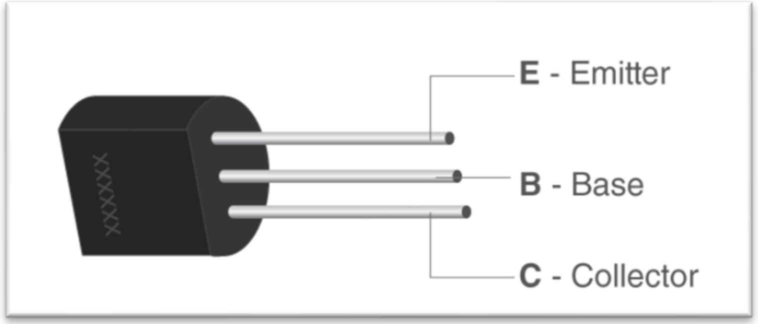
Use of transistor



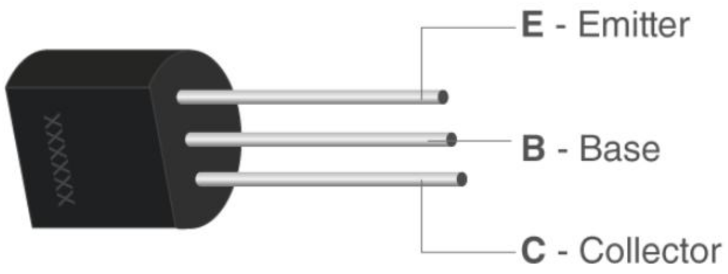
Parts of a Transistor

- A typical transistor is composed of three layers of semiconductor materials or, more specifically, terminals which help to make a connection to an external circuit and carry the current. A voltage or current that is applied to any one pair of the terminals of a transistor controls the current through the other pair of terminals. There are three terminals for a transistor. They are listed below:

- **Base:** This is **used to activate the transistor**.
- **Collector:** It is the **positive lead of the transistor**.
- **Emitter:** It is the **negative lead of the transistor**.



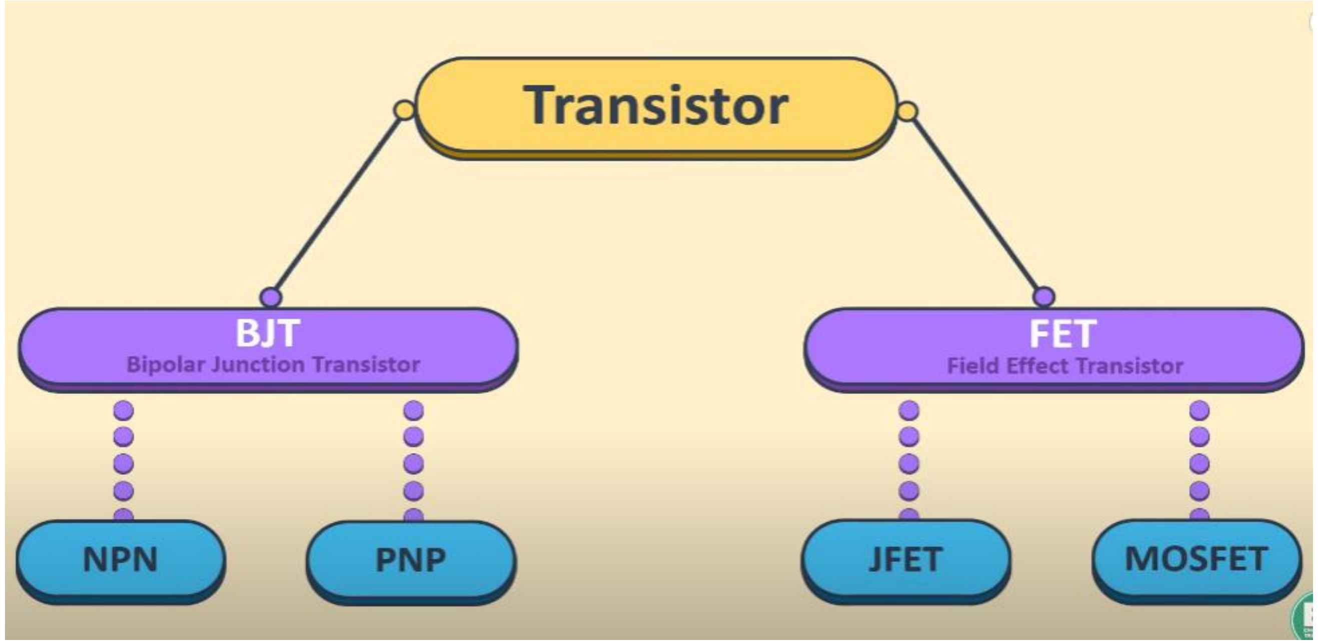
Parts of a Transistor



- Basic working principle of a transistor is based on controlling the flow of current through one channel by varying the intensity of a smaller current that is flowing through a second channel.

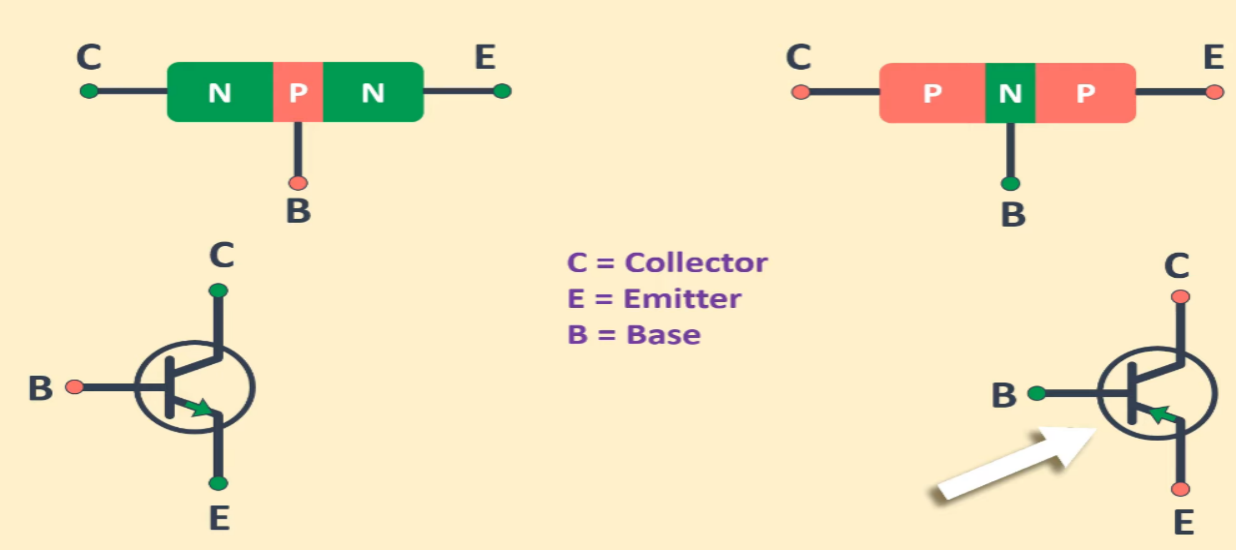
Types of Transistor

- There are mainly two types of transistors, based on how they are used in a circuit.



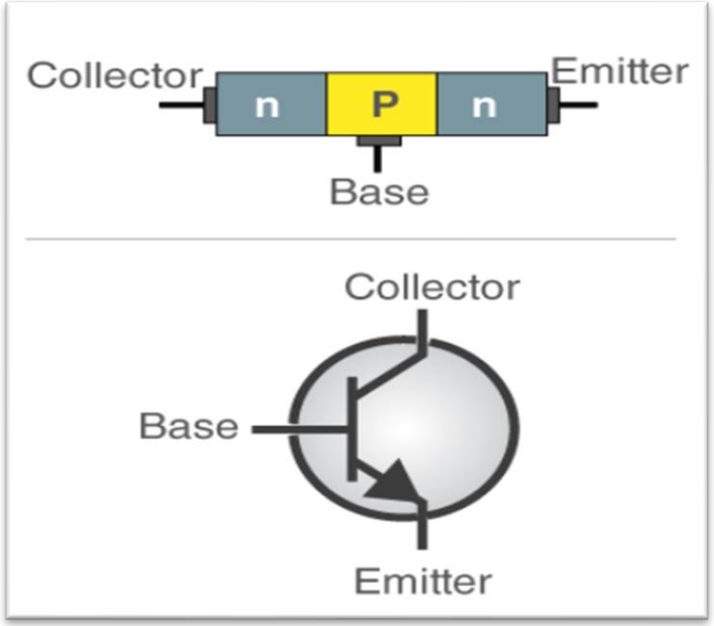
Bipolar Junction Transistor (BJT)

- The three terminals of BJT are the base, emitter and collector. A very small current flowing between the base and emitter can control a larger flow of current between the collector and emitter terminal.



NPN

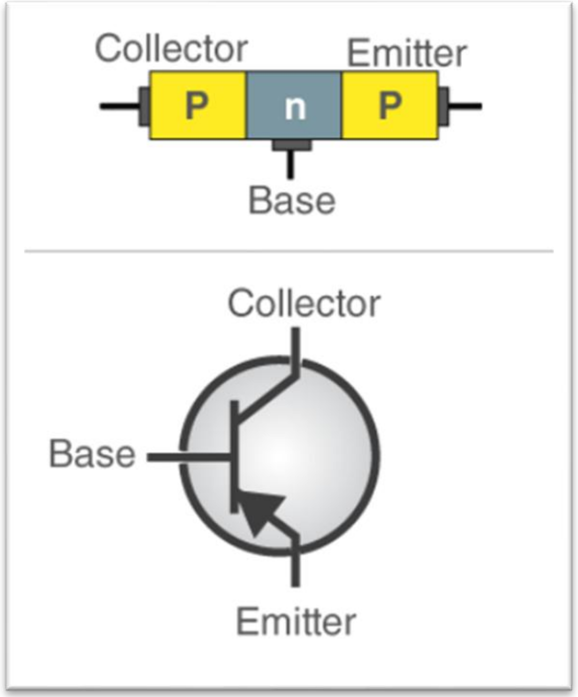
- N-P-N Transistor: In this transistor, we will find one p-type material that is present between two n-type materials.
  1. N-P-N transistor is basically used to amplify weak signals to strong signals.
  2. In an NPN transistor, the electrons move from the emitter to the collector region, resulting in the formation of current in the transistor. This transistor is widely used in the circuit.



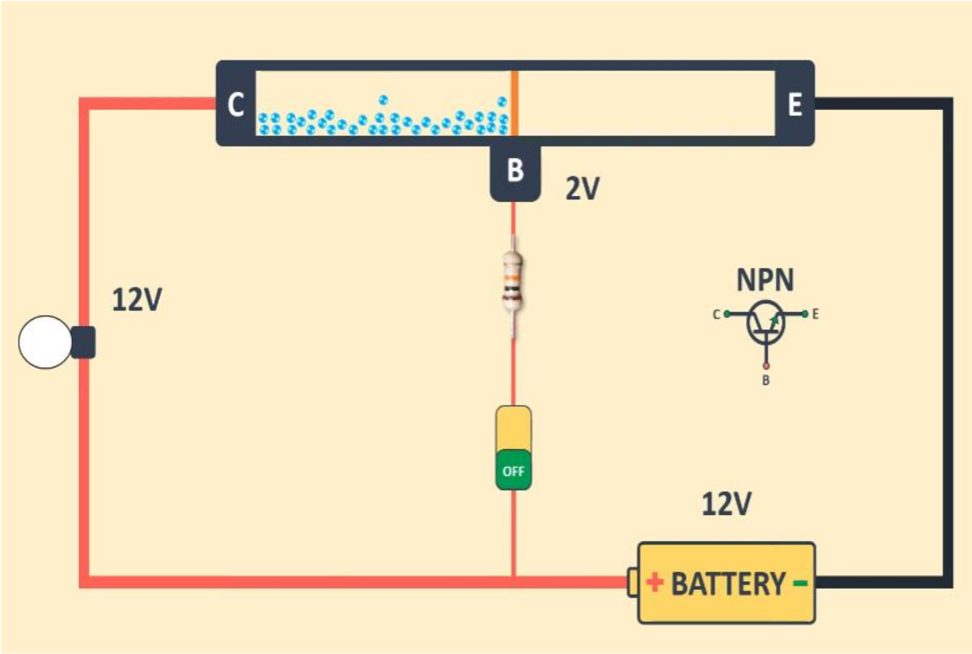
PNP

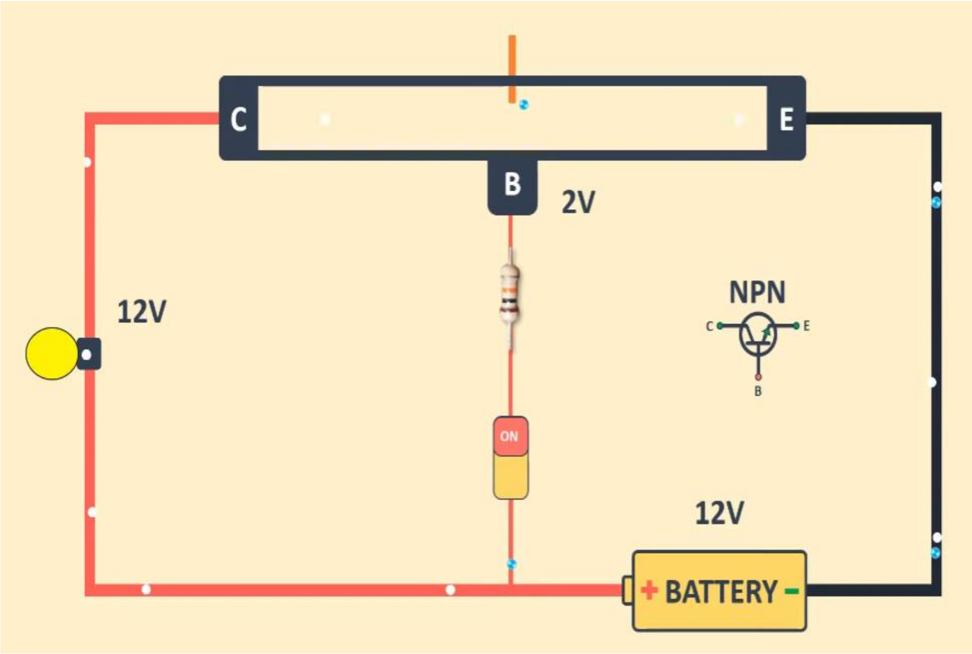
- **P-N-P Transistor: It is a type of BJT where one n-type material is introduced or placed between two p-type materials. In such a configuration, the device will control the flow of current.**

1. **PNP transistor consists of 2 crystal diodes which are connected in series. The right side and left side of the diodes are known as the collector-base diode and emitter-base diode, respectively.**

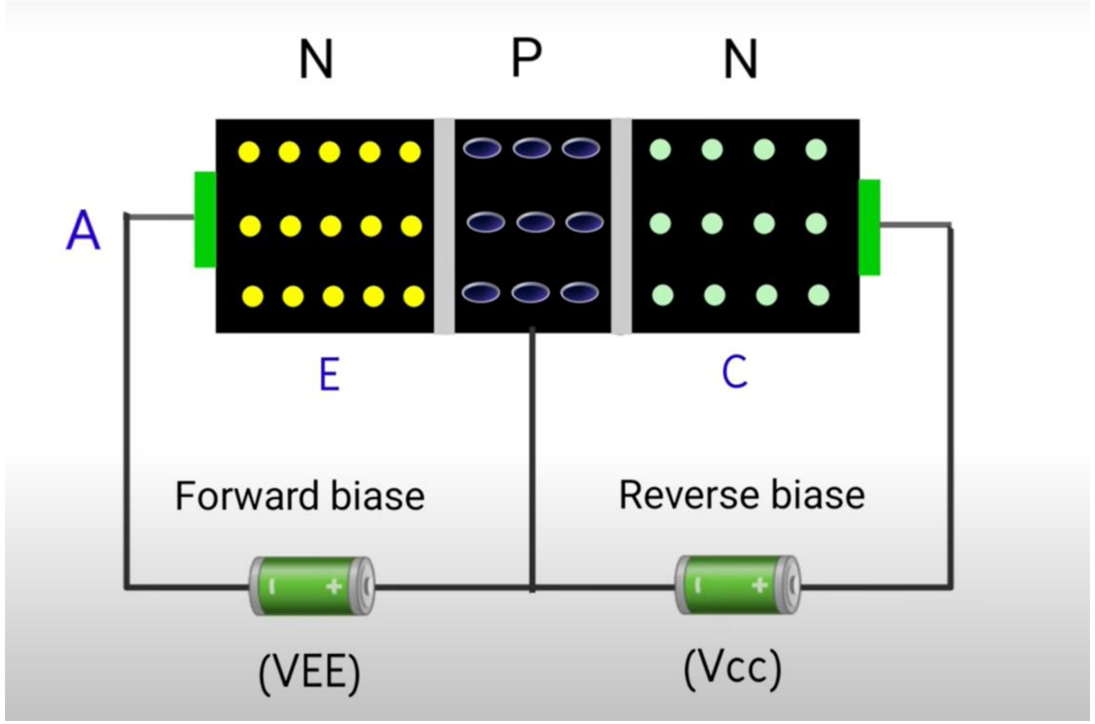


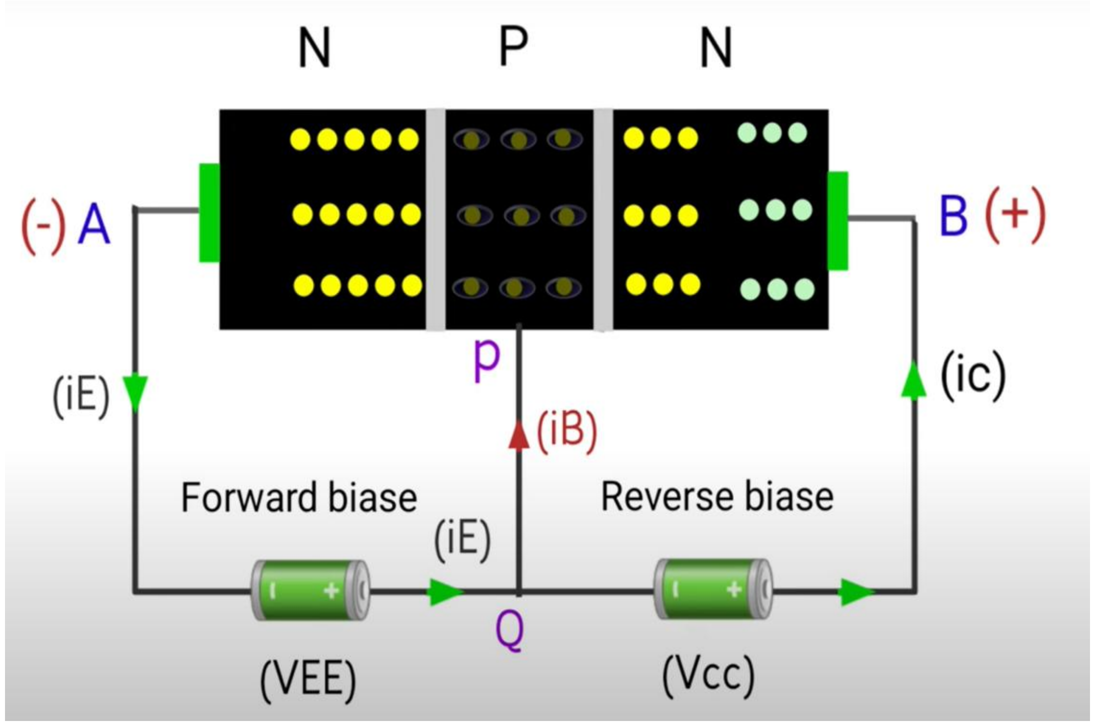
How Do Transistors Work?



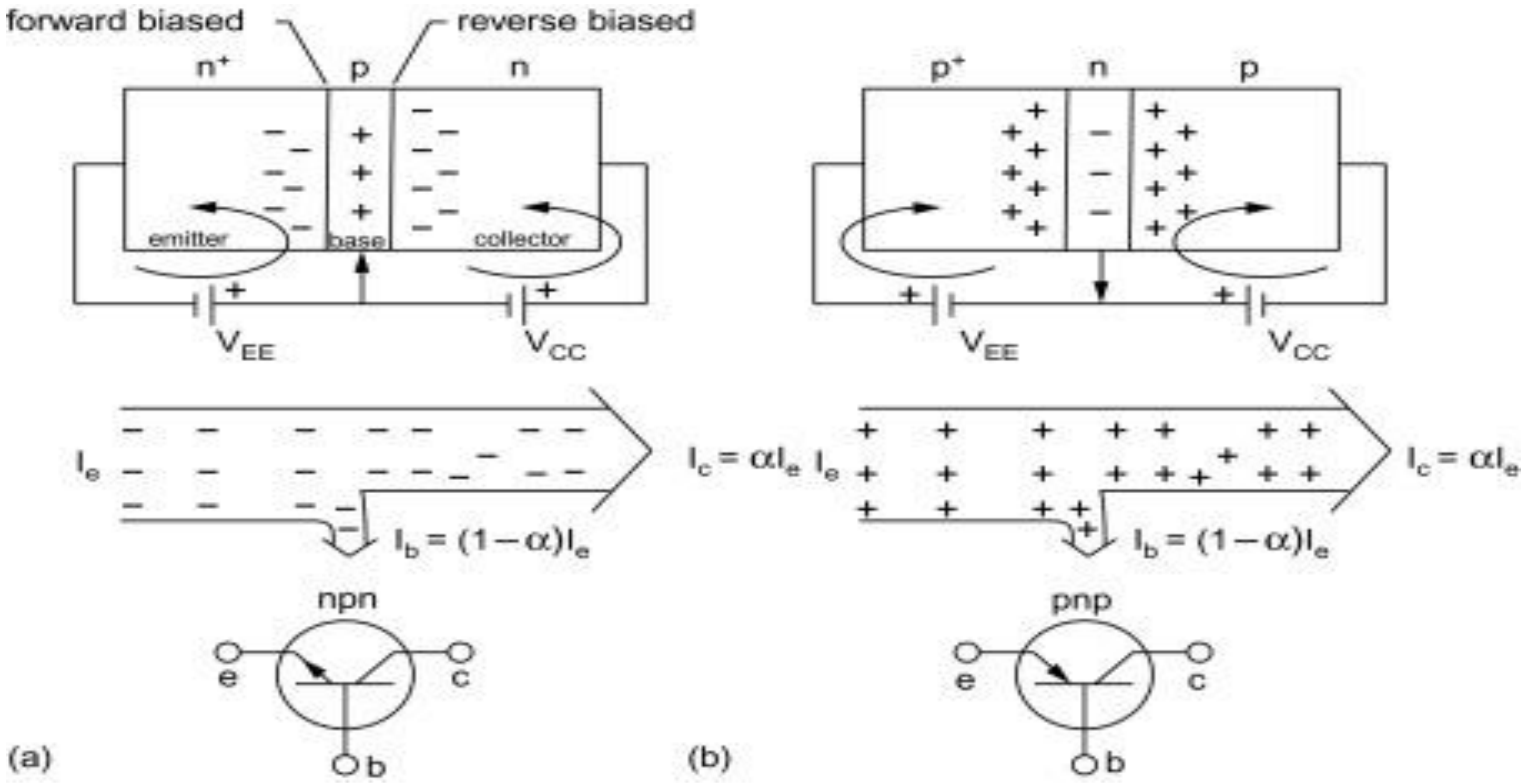


Operation of NPN Transistor



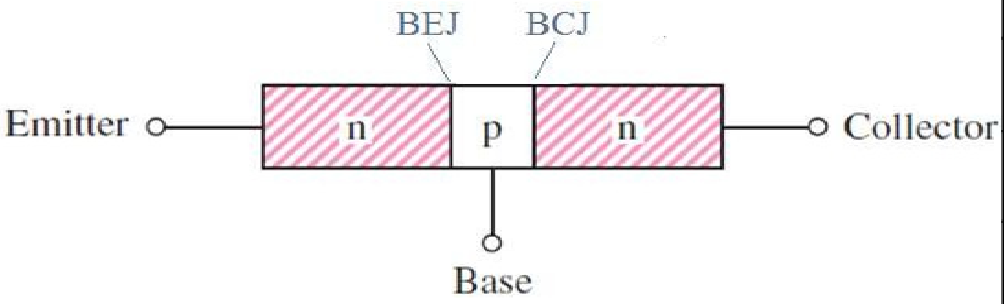


Operation of PNP Transistor



Operation of PNP Transistor

- A single p-n junction has two different types of bias:
- Forward Bias
- Reverse Bias
- Thus, a two pn-jnuction device has four types of bias.



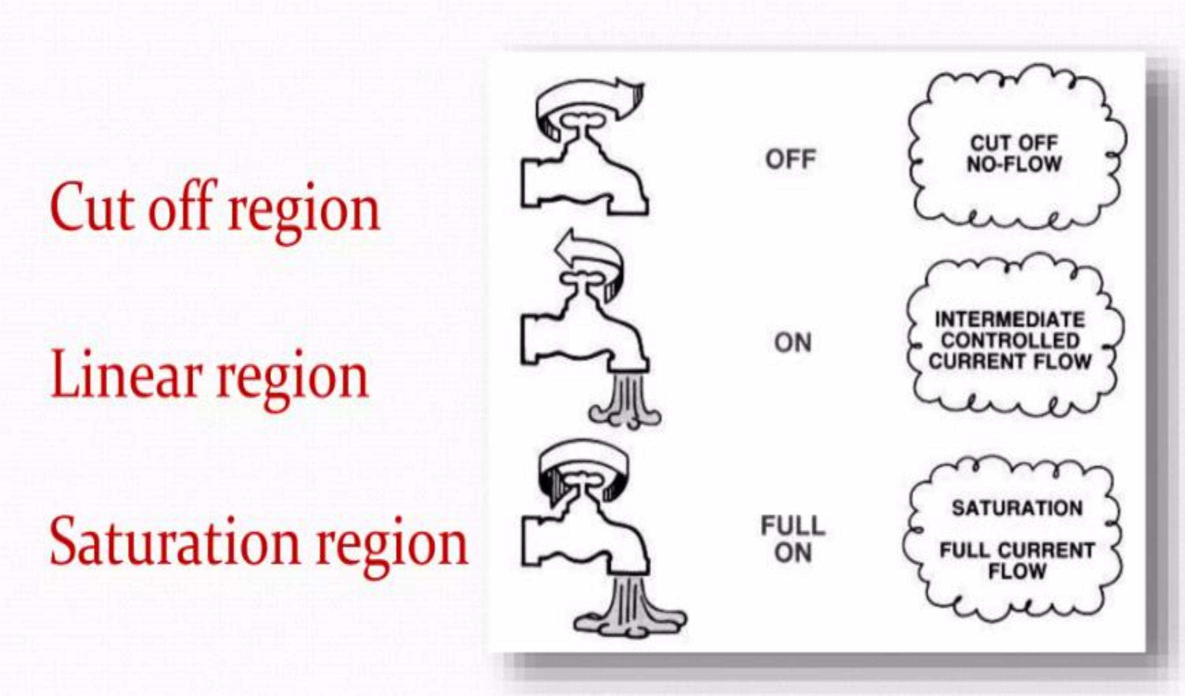
BC junction (BCJ)	BE junction (BEJ)	Mode of operation
Reverse	Reverse	Cut-off
Forward	Reverse	Cut-off
Reverse	Forward	Active
Forward	Forward	Saturation

Limitations of Transistors

- Transistors have a few limitations, and they are as follows:

  i. Transistors lack higher electron mobility.
  ii. Transistors can be easily damaged when electrical and thermal events arise.
  iii. For example, electrostatic discharge in handling.
  iv. Transistors are affected by cosmic rays and radiation.

Operation Regions of BJT



Operation Regions of BJT

OFF Mode (Cutoff Region)

- No base current ($I_b = 0$).
- No current flows between collector and emitter ($I_c = 0$).
- Transistor acts as an open switch.

ON Mode (Saturation Region)

- Base current is high.
- Maximum current flows from collector to emitter.
- Transistor acts as a closed switch.

Amplification Mode (Active Region)

- A small base current controls a large collector-emitter current.
- Transistor acts as an amplifier.

Common configuration

1. Common Emitter Configuration –
has both Current and Voltage Gain.

2. Common Base Configuration –
has Voltage Gain but no Current Gain.

3. Common Collector Configuration –
has Current Gain but no Voltage Gain.

Common configuration

Emitter and collector currents:

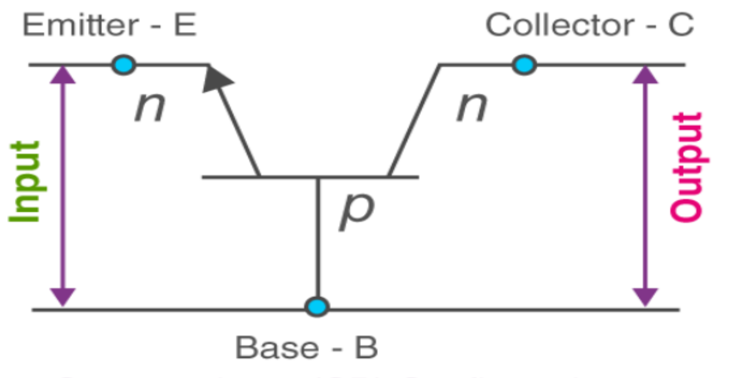
$$I_C \cong I_E$$

Base-emitter voltage:

$$V_{BE} = 0.7\ \text{V} \quad \text{(for Silicon)}$$

Common Base (CB)

- In common base (CB) configuration, the base terminal of the transistor is common between input and output terminals.



Common base (CB) Configuration

Current Amplification Factor α

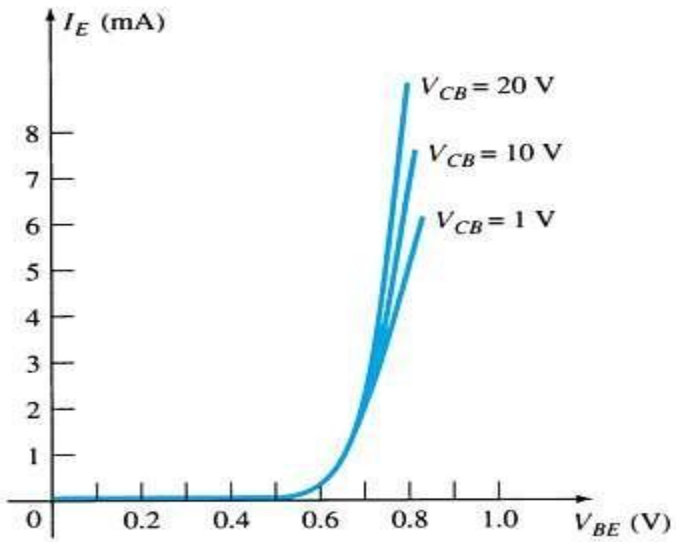
The ratio of change in collector current $\$\Delta I_C\$$ to the change in emitter current $\$\Delta I_E\$$ when collector voltage $\mathbf{V_{CB}}$ is kept constant, is called as **Current amplification factor**. It is denoted by α.

$$\alpha = \frac{\Delta I_C}{\Delta I_E} \; at\ constant\ V_{CB}$$

Common Base (CB)

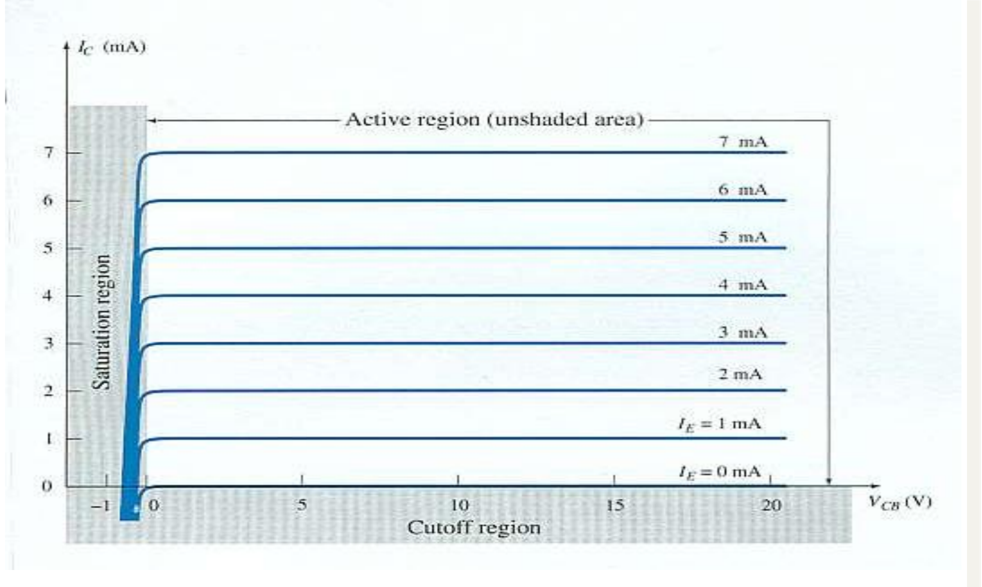
Input Characteristics

This curve shows the relationship between of input current (I_E) to input voltage (V_{BE}) for three output voltage (V_{CB}) levels.



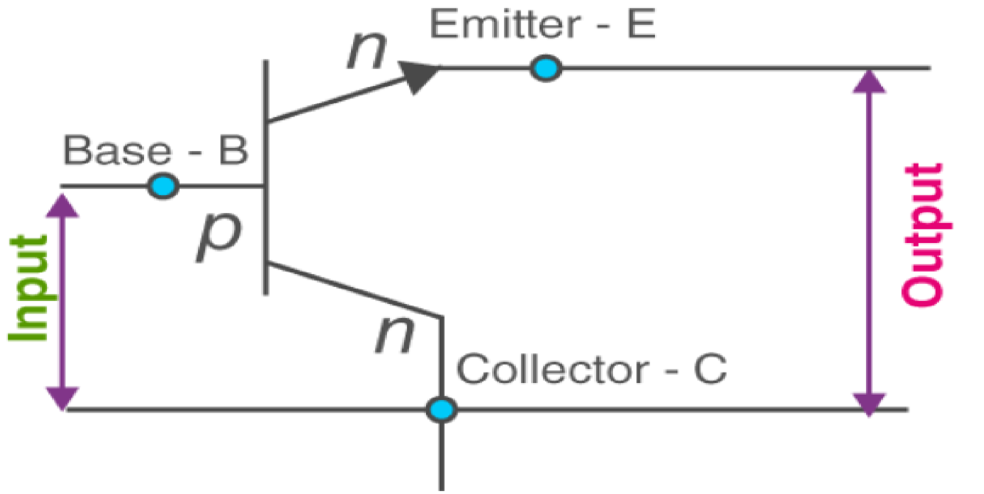
Output Characteristics

This graph demonstrates the output current (I_C) to an output voltage (V_{CB}) for various levels of input current (I_E).



Common Collector (CC)

- In common collector (CC) configuration, the collector terminals are common between the input and output terminals.



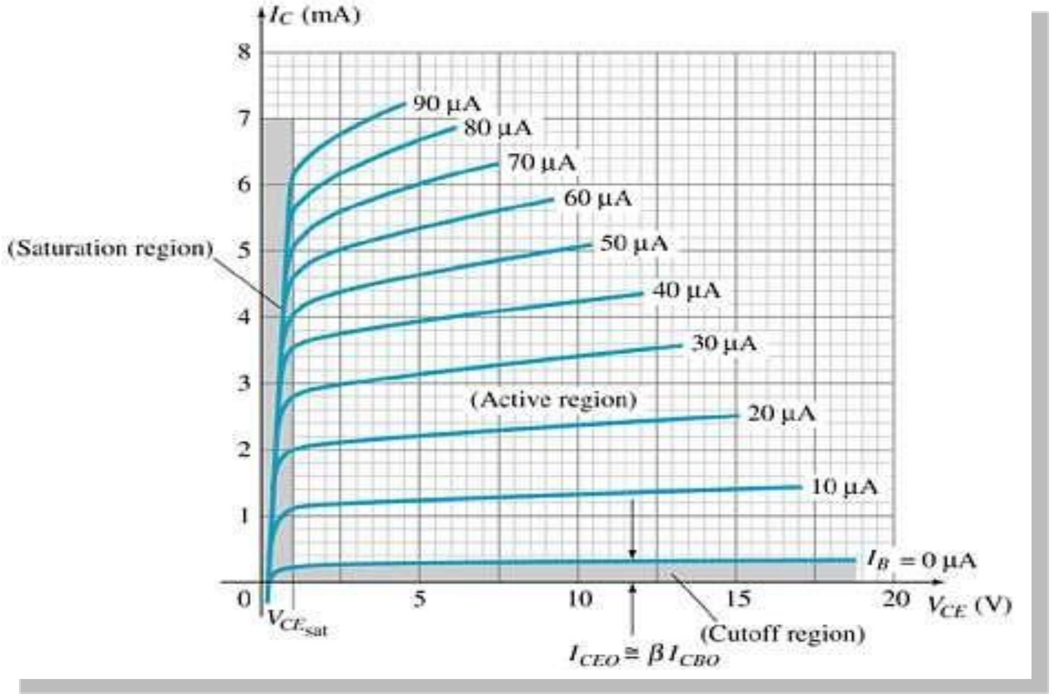
Common Collector (CC) Configuration

The ratio of change in emitter current ΔI_E to the change in base current ΔI_B is known as **Current Amplification factor** in common collector CC configuration. It is denoted by **γ**.

$$\gamma = \frac{\Delta I_E}{\Delta I_B}$$

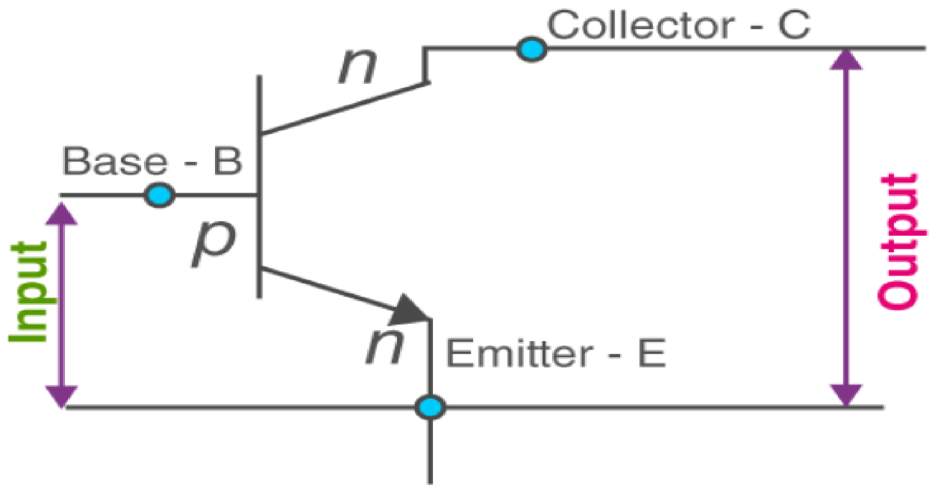
Common Collector (CC)

The characteristics are similar to those of the common-emitter configuration, except the vertical axis is I_E.



Common Emitter (CE)

- In common emitter (CE) configuration, the emitter terminal is common between the input and the output terminals.



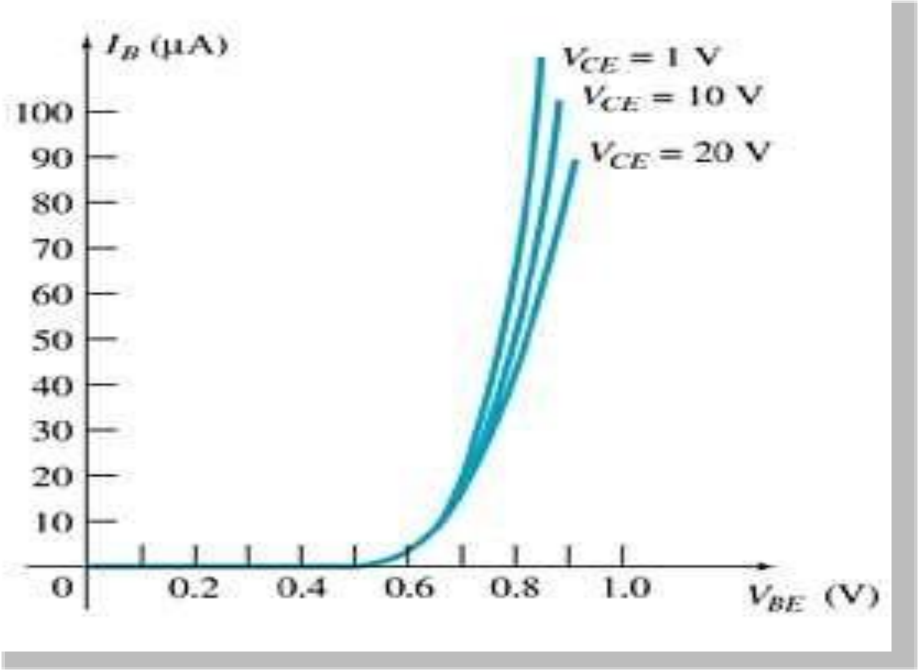
Common Emitter (CE) Configuration

Base Current Amplification factor β

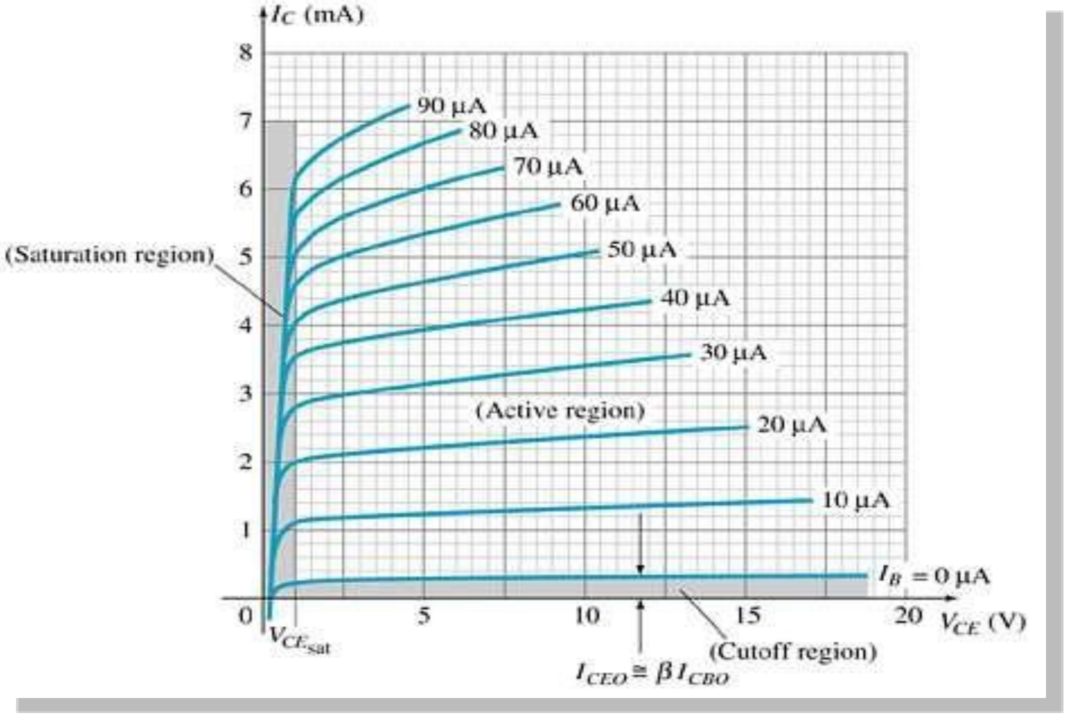
The ratio of change in collector current ΔI_C to the change in base current ΔI_B is known as **Base Current Amplification Factor.** It is denoted by β

$$\beta = \frac{\Delta I_C}{\Delta I_B}$$

Common Emitter (CE)



Base Characteristics



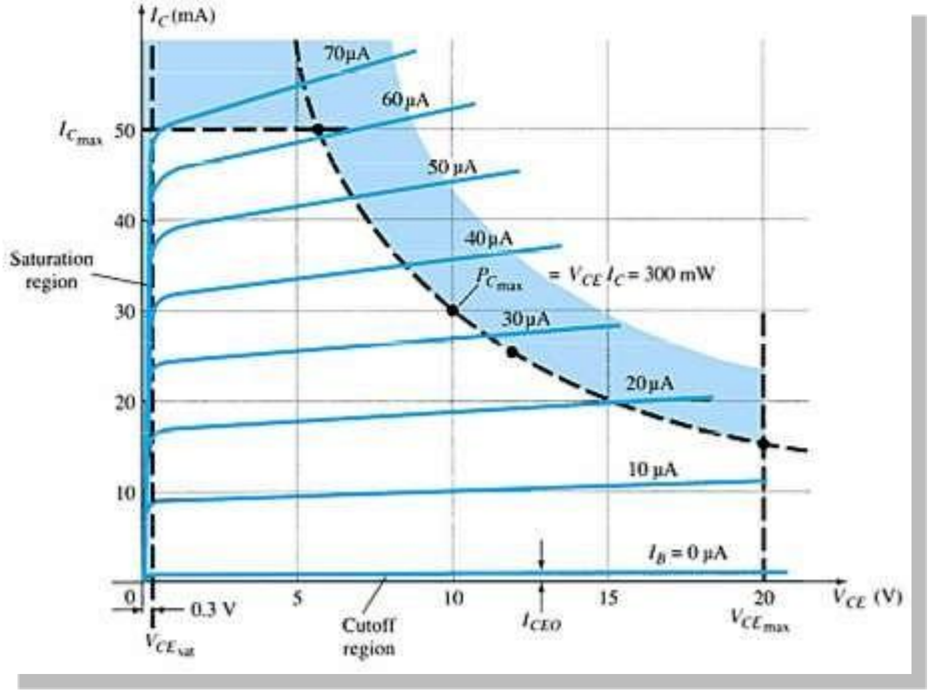
Collector Characteristics

Common configuration

V_{CE} is at maximum and I_C is at minimum ($I_{Cmax}= I_{CEO}$) in the cutoff region.

I_C is at maximum and V_{CE} is at minimum ($V_{CE\ max} = V_{CEsat} = V_{CEO}$) in the saturation region.

The transistor operates in the active region between saturation and cutoff.



Amplification Factor

Relation between β and α

Let us try to derive the relation between base current amplification factor and emitter current amplification factor.

$$\beta = \frac{\Delta I_C}{\Delta I_B}$$

$$\alpha = \frac{\Delta I_C}{\Delta I_E}$$

$$I_E = I_B + I_C$$

$$\Delta I_E = \Delta I_B + \Delta I_C$$

$$\Delta I_B = \Delta I_E - \Delta I_C$$

We can write

$$\beta = \frac{\Delta I_C}{\Delta I_E - \Delta I_C}$$

Dividing by $$

$$\beta = \frac{\frac{\Delta I_C}{\Delta I_E}}{\frac{\Delta I_E}{\Delta I_E} - \frac{\Delta I_C}{\Delta I_E}}$$

$$\alpha = \frac{\Delta I_C}{\Delta I_E}$$

We have

$$\alpha = \frac{\Delta I_C}{\Delta I_E}$$

Therefore,

$$\beta = \frac{\alpha}{1 - \alpha}$$

Amplification Factor

Relation between γ and α

Let us try to draw some relation between γ and α

$$\gamma = \frac{\Delta I_E}{\Delta I_B}$$

$$\alpha = \frac{\Delta I_C}{\Delta I_E}$$

$$I_E = I_B + I_C$$

$$\Delta I_E = \Delta I_B + \Delta I_C$$

$$\Delta I_B = \Delta I_E - \Delta I_C$$

Substituting the value of I_B, we get

$$\gamma = \frac{\Delta I_E}{\Delta I_E - \Delta I_C}$$

Dividing by ΔI_E

$$\gamma = \frac{\frac{\Delta I_E}{\Delta I_E}}{\frac{\Delta I_E}{\Delta I_E} - \frac{\Delta I_C}{\Delta I_E}}$$

$$\frac{1}{1 - \alpha}$$

$$\gamma = \frac{1}{1 - \alpha}$$

Transistor Simple Mathematical Problem

- **In a common base connection, IE = 1mA, IC = 0.95mA. Calculate the value of IB .**
- **In a common base connection, current amplification factor is 0.9. If the emitter current is 1mA, determine the value of base current.**
- **In a common base connection, IC = 0.95 mA and IB = 0.05 mA. Find the value of α.**

- **In a common base connection, the emitter current is 1mA. If the emitter circuit is open, the collector current is 50 µA. Find the total collector current. Given that α = 0.92.**

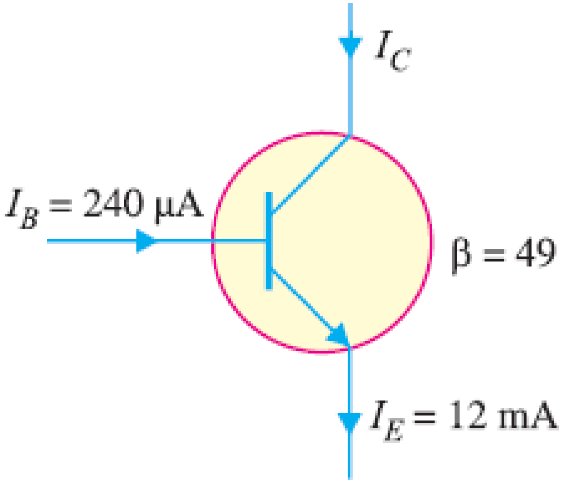
$$\text{Here, } I_E = 1 \text{ mA}, \alpha = 0.92, \quad I_{CBO} = 50 \ \mu\text{A}$$

$$\text{Total collector current, } I_C = \alpha I_E + I_{CBO} = 0.92 \times 1 + 50 \times 10^{-3}$$

$$= 0.92 + 0.05 = \mathbf{0.97 \ mA}$$

Transistor Simple Mathematical Problem

- **Find the α rating of the transistor shown in Fig. Hence determine the value of IC using both α and β rating of the transistor.**



- Solve

$$\alpha = \frac{\beta}{1+\beta} = \frac{49}{1+49} = 0.98$$

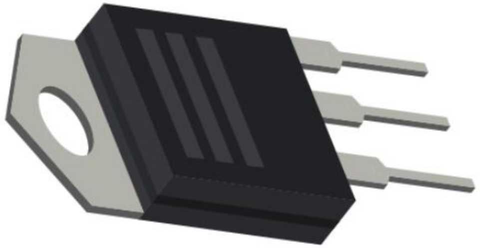
The value of I_C can be found by using either α or β rating as under :

$$I_C = \alpha I_E = 0.98\ (12\ \text{mA}) = 11.76\ \text{mA}$$

Also $$I_C = \beta I_B = 49\ (240\ \mu\text{A}) = 11.76\ \text{mA}$$

What Is a MOSFET?

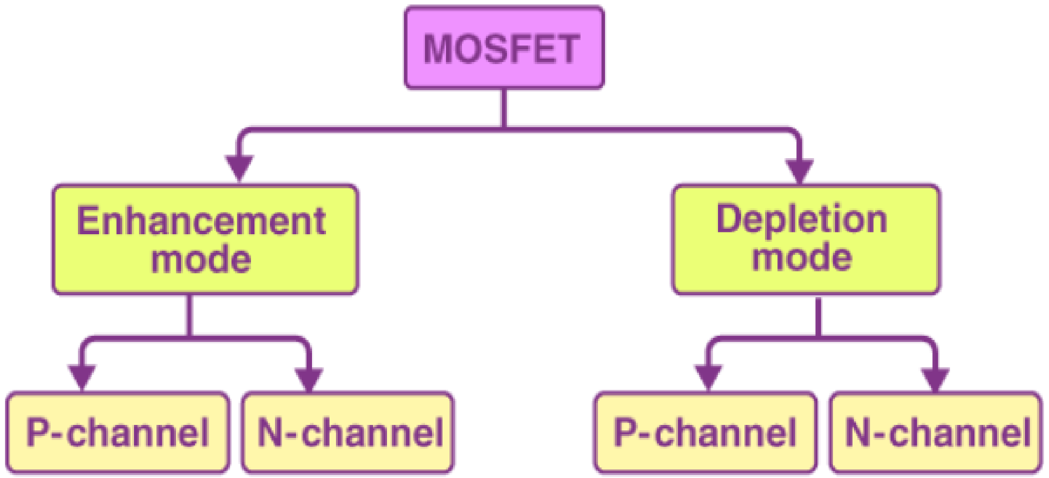
- Metal Oxide Silicon Field Effect Transistors commonly known as MOSFETs are electronic devices used to switch or amplify voltages in circuits. It is a voltage controlled device and is constructed by three terminals. The terminals of MOSFET are named as follows:

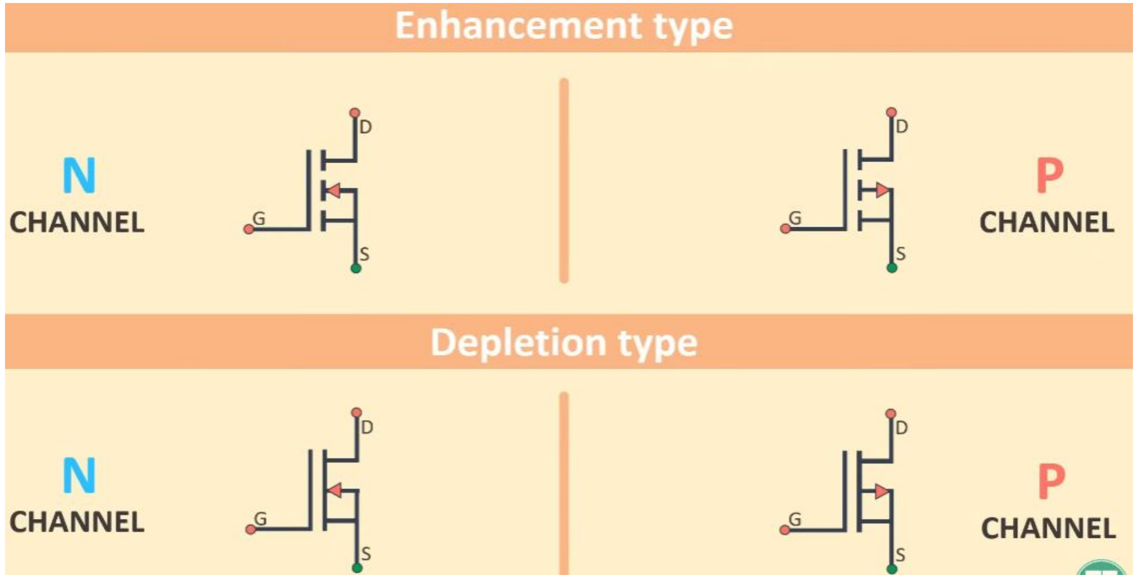


- Source
- Gate
- Drain

MOSFET Types

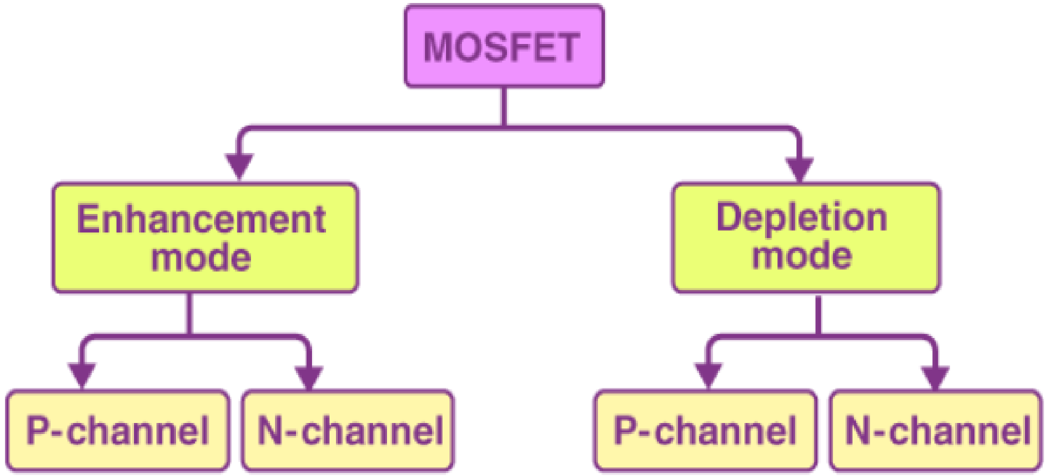
- The classification of MOSFET based on the construction and the material used is given below in the flowchart.

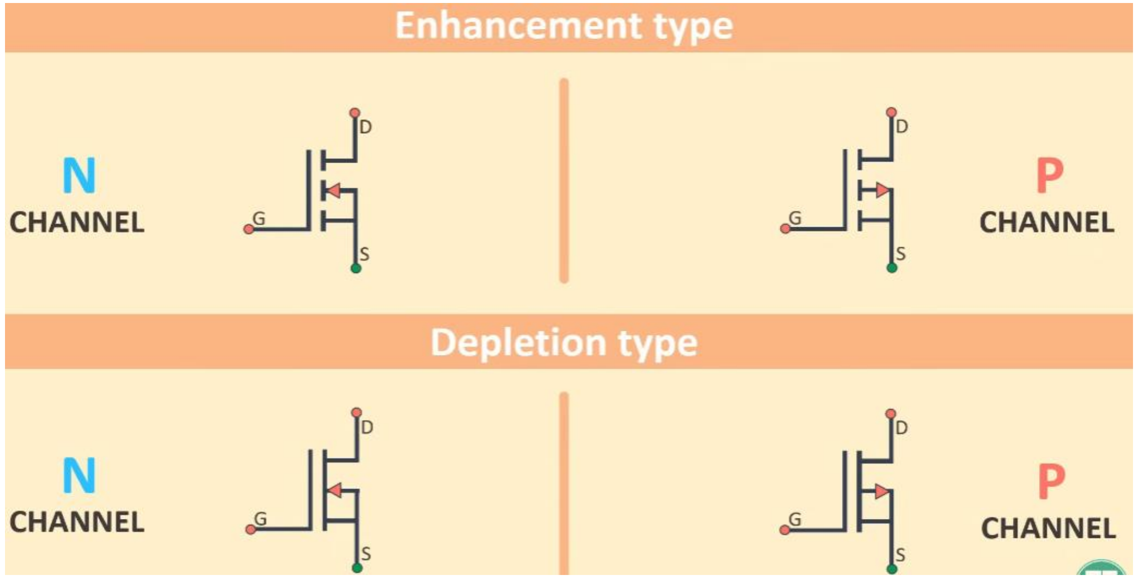




MOSFET Types

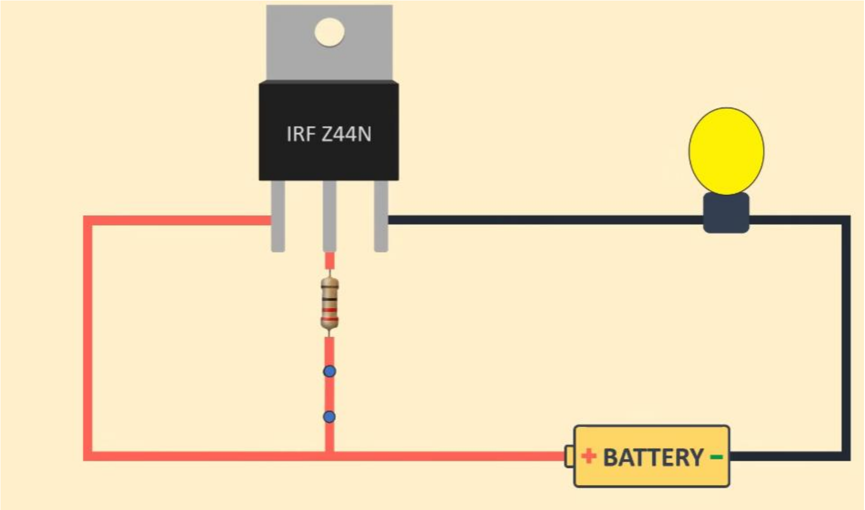
- The classification of MOSFET based on the construction and the material used is given below in the flowchart.

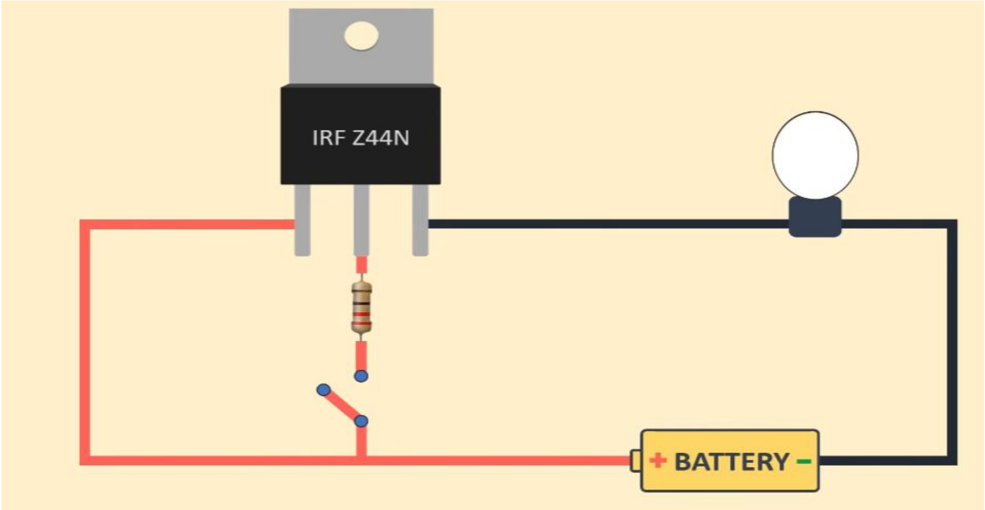




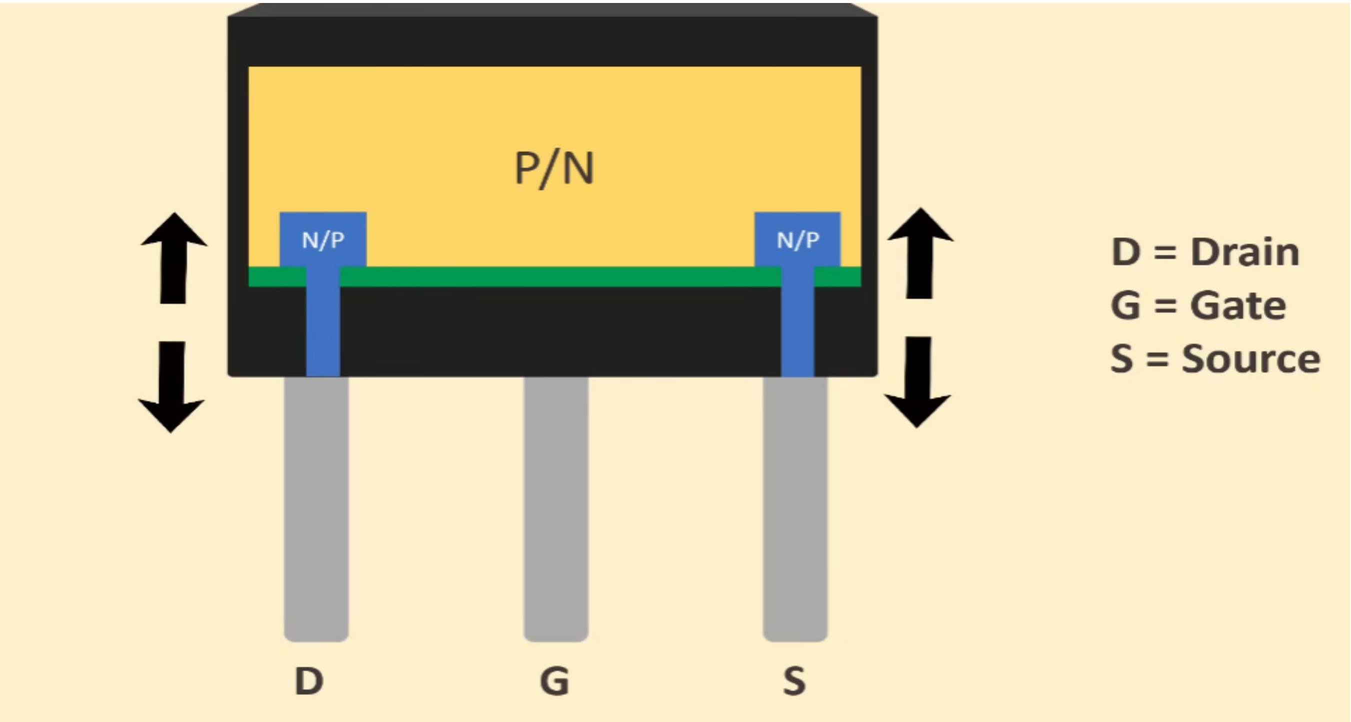
Working Principle of MOSFET

- When voltage is applied to the gate, an electrical field is generated that changes the width of the channel region, where the electrons flow. The wider the channel region, the better conductivity of a device will be.

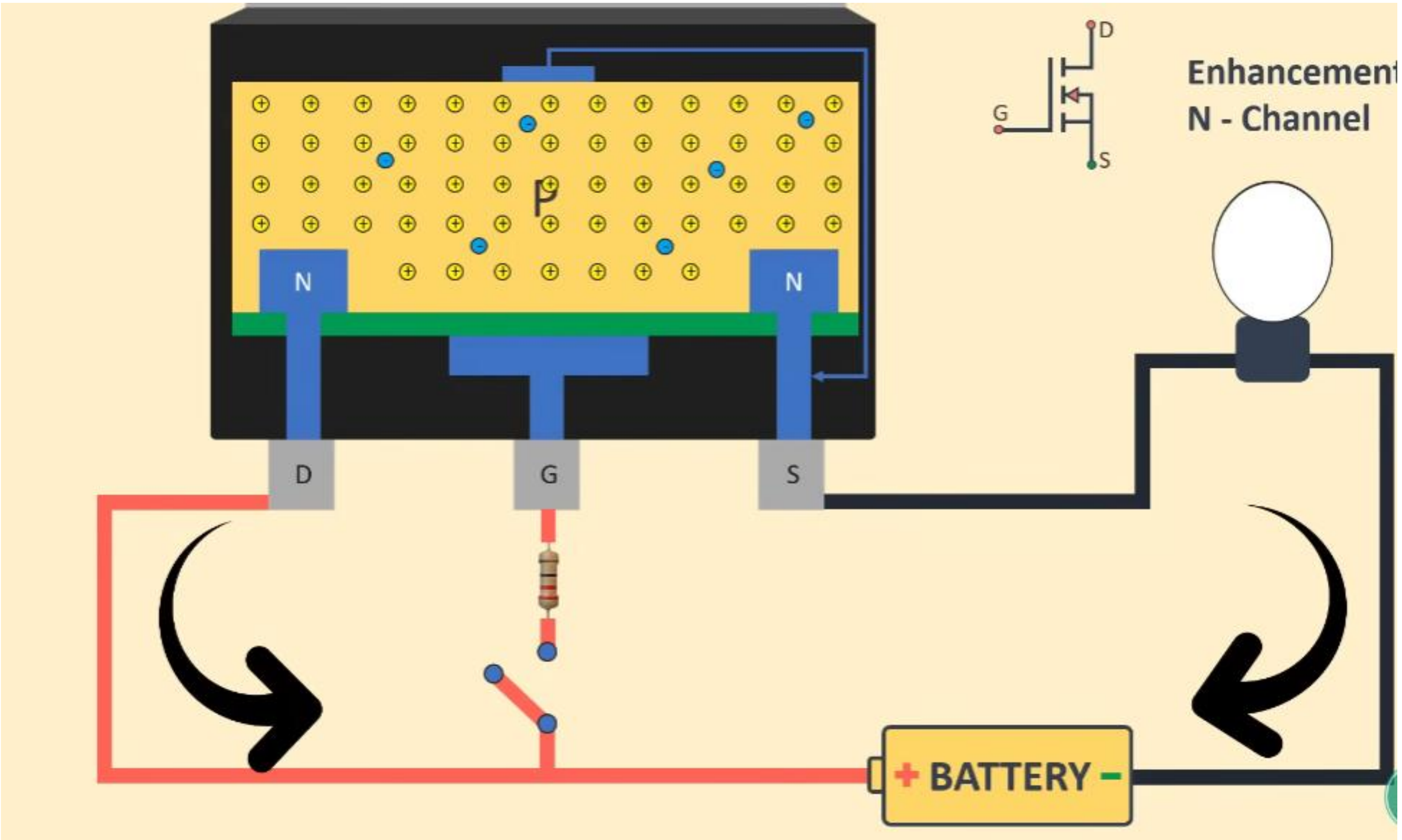




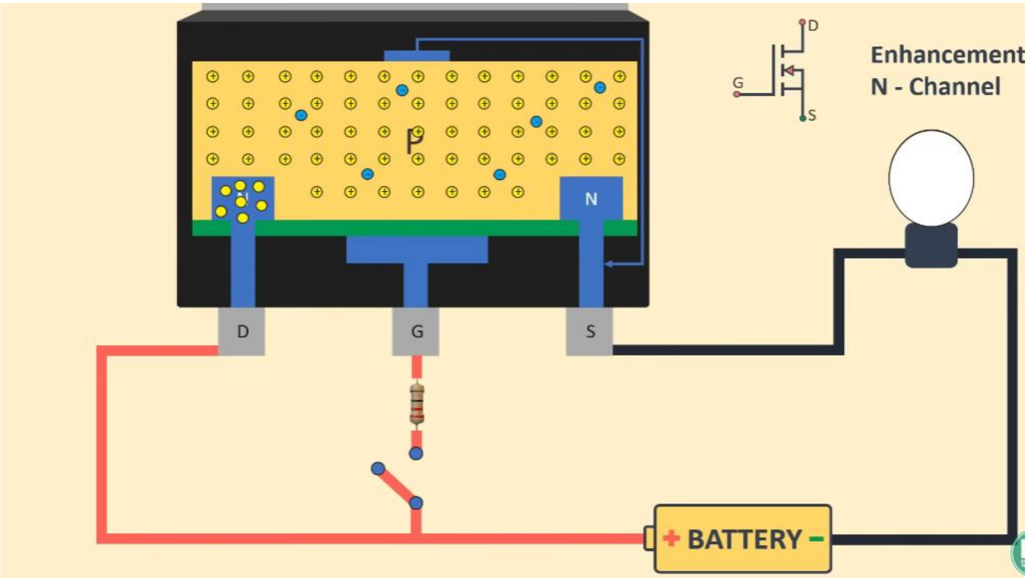
Working Principle of MOSFET

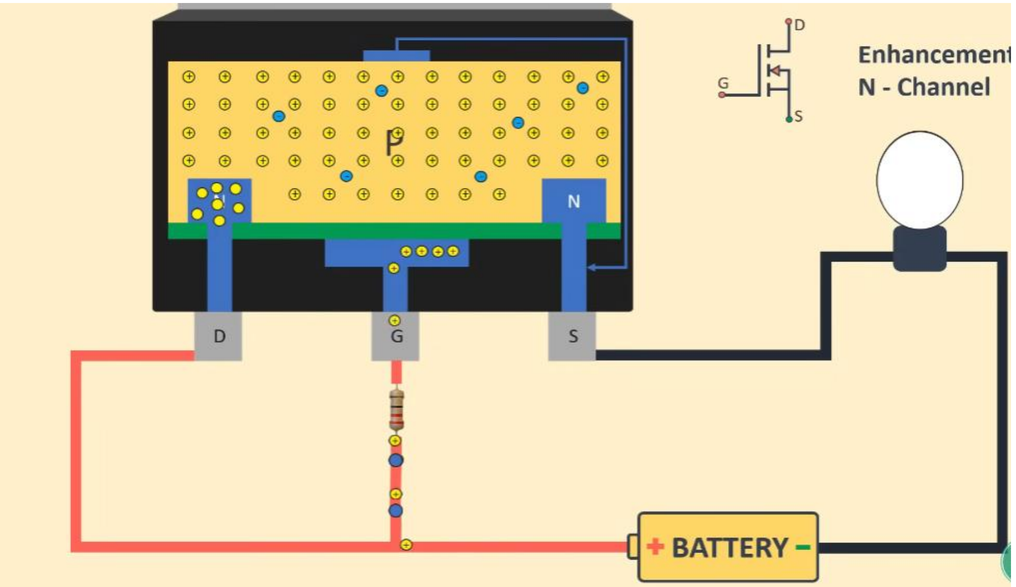


Enhancement N - Channel MOSFET

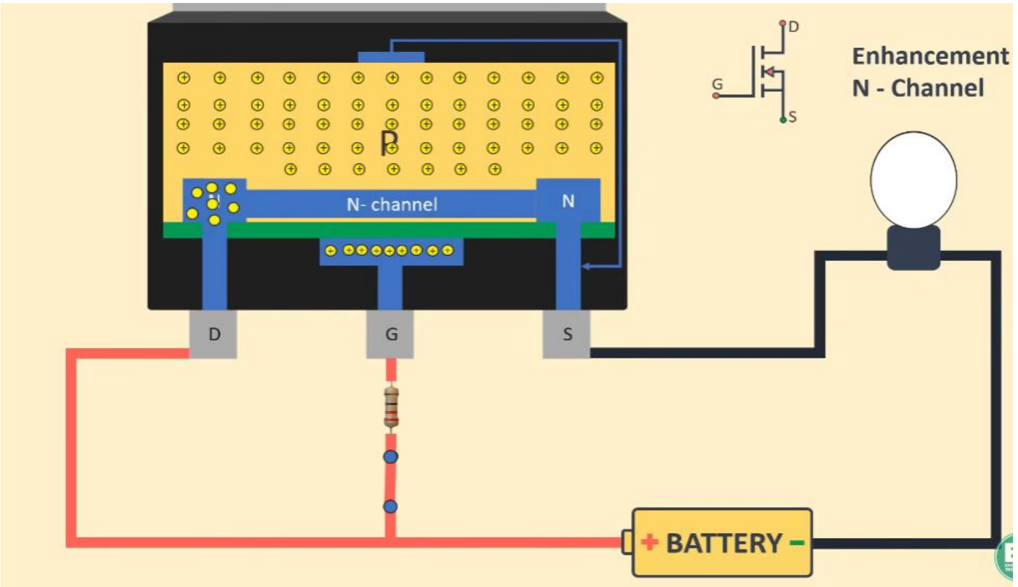


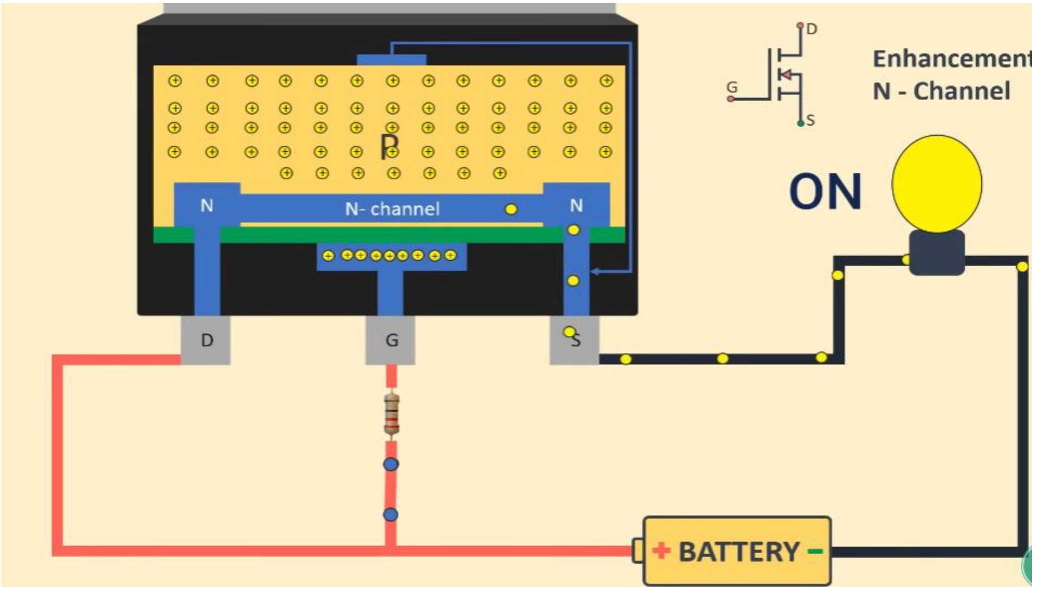
Enhancement N - Channel MOSFET



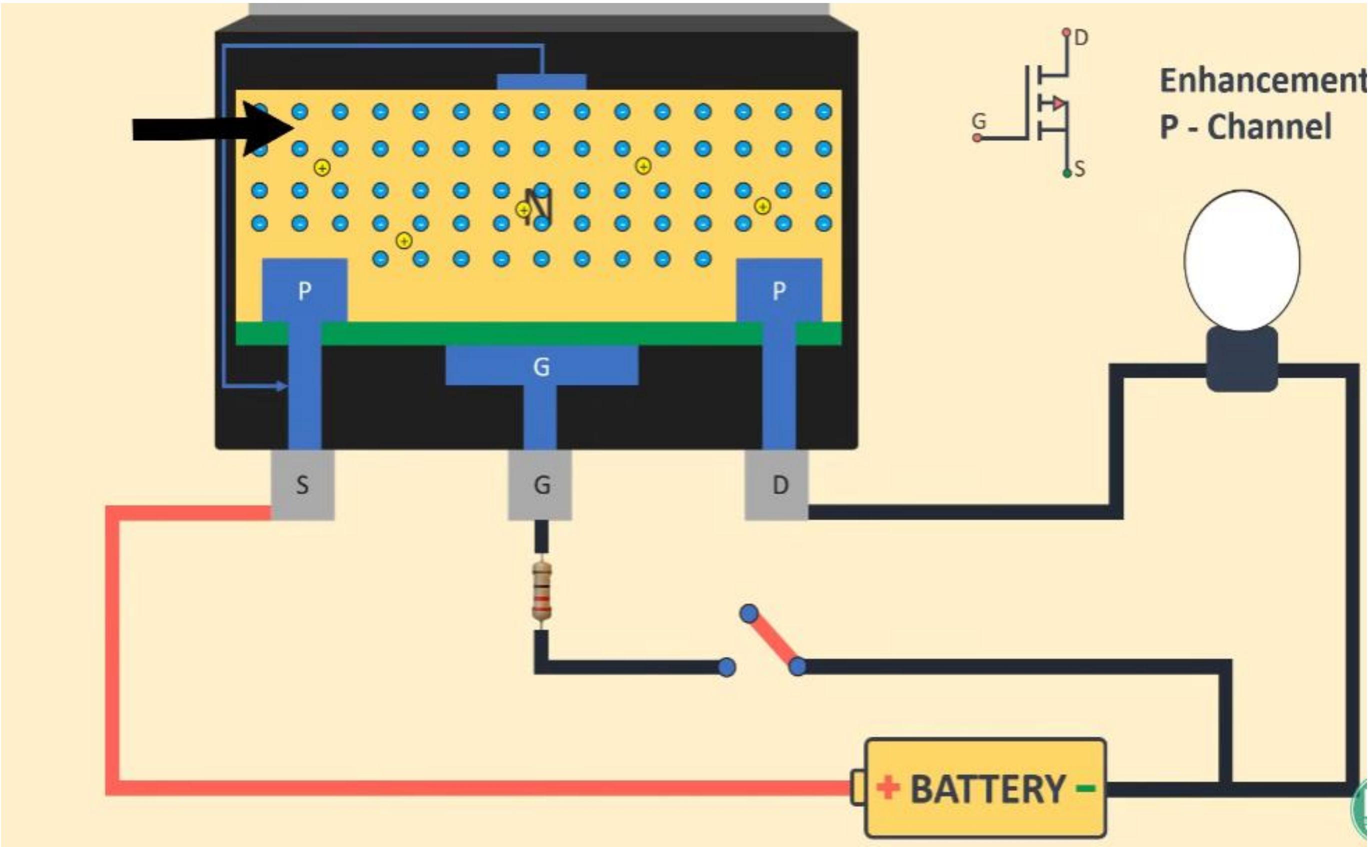


Enhancement N - Channel MOSFET

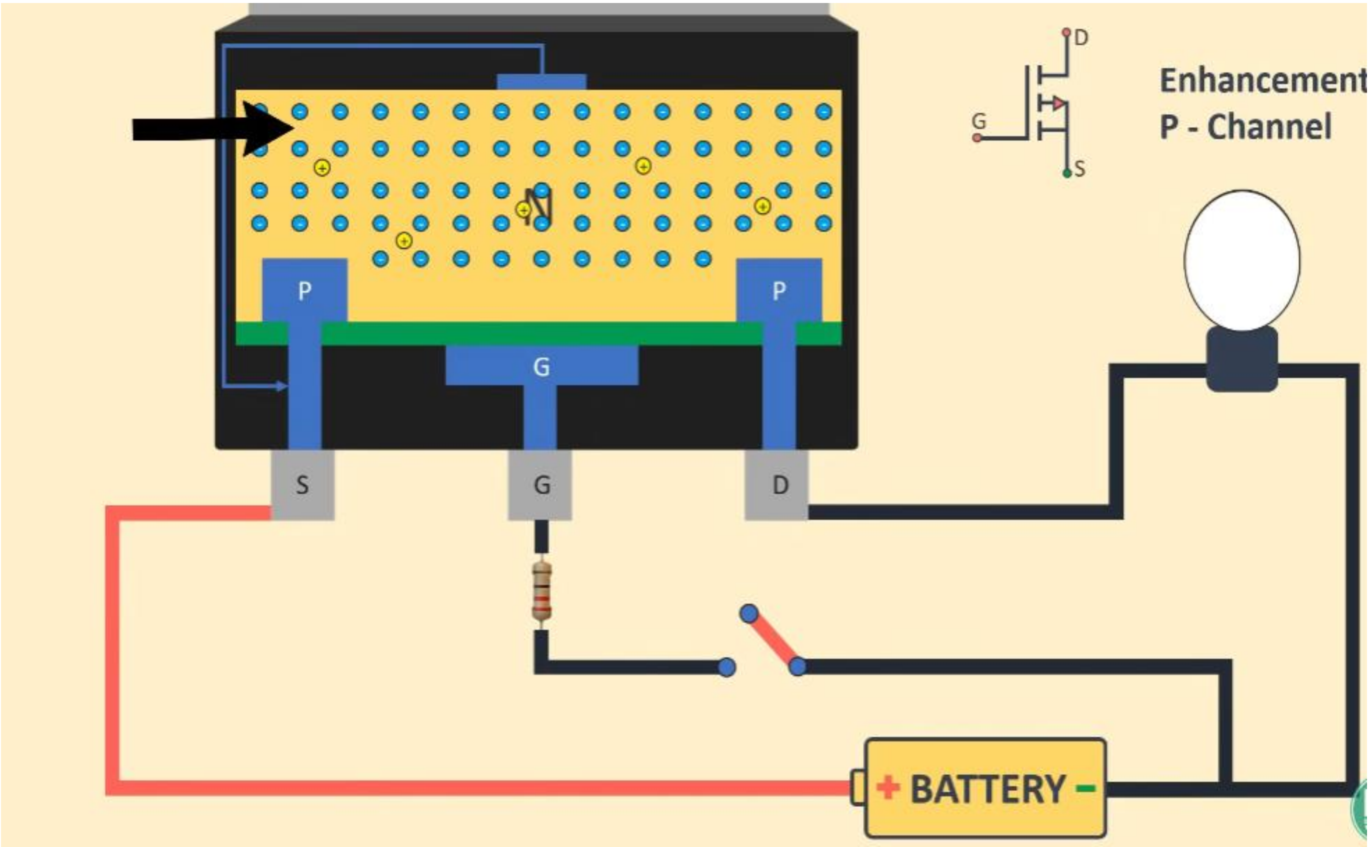




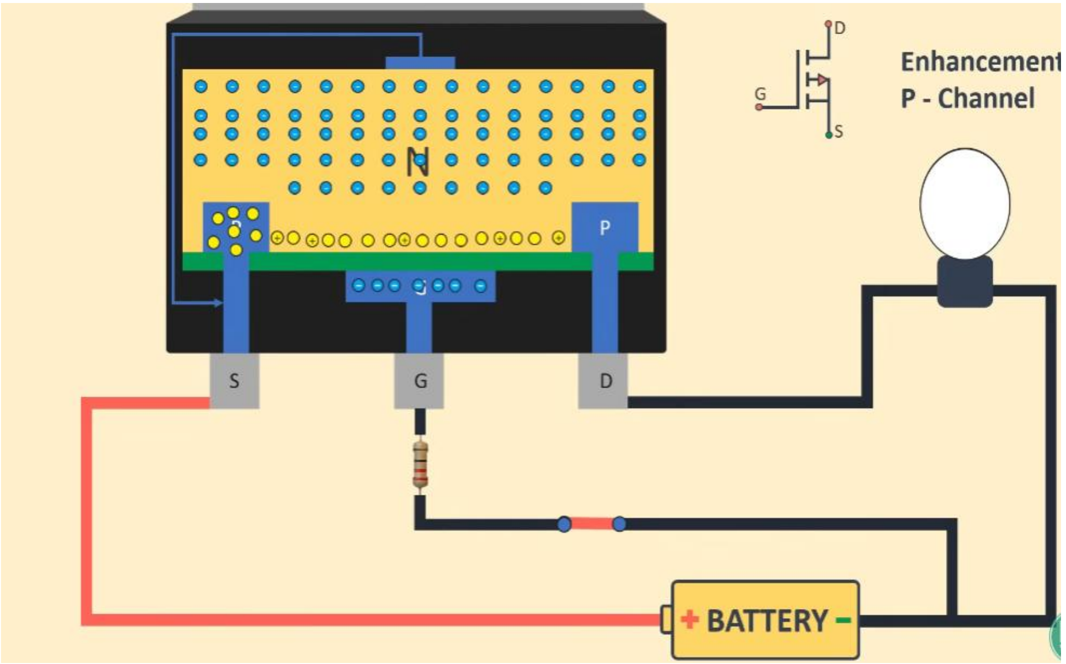
Enhancement P - Channel MOSFET

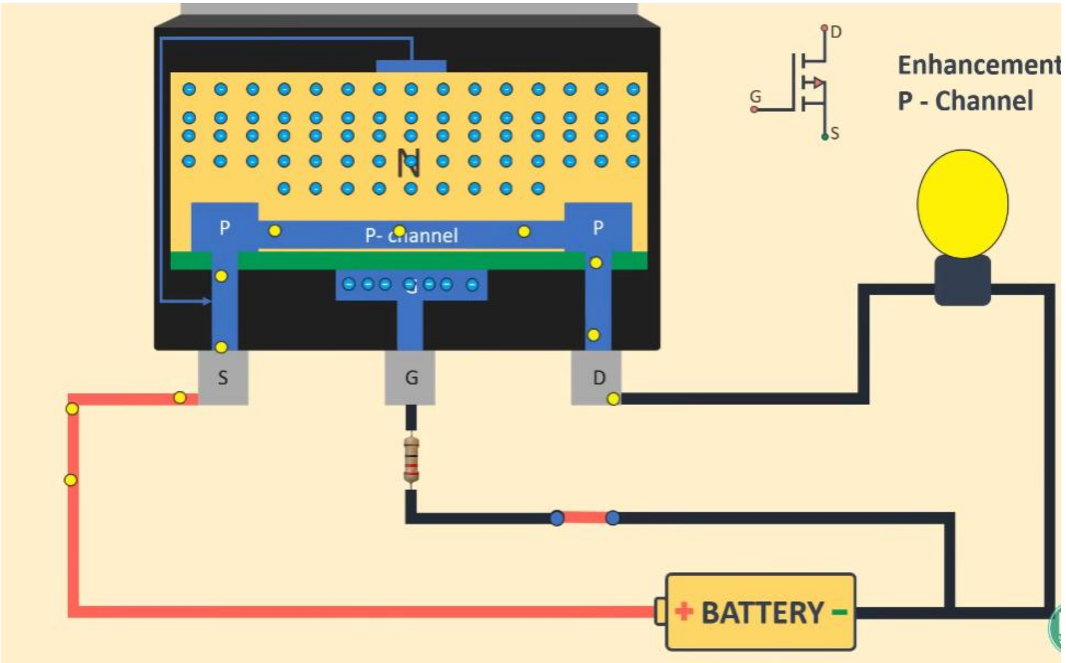


Enhancement P - Channel MOSFET

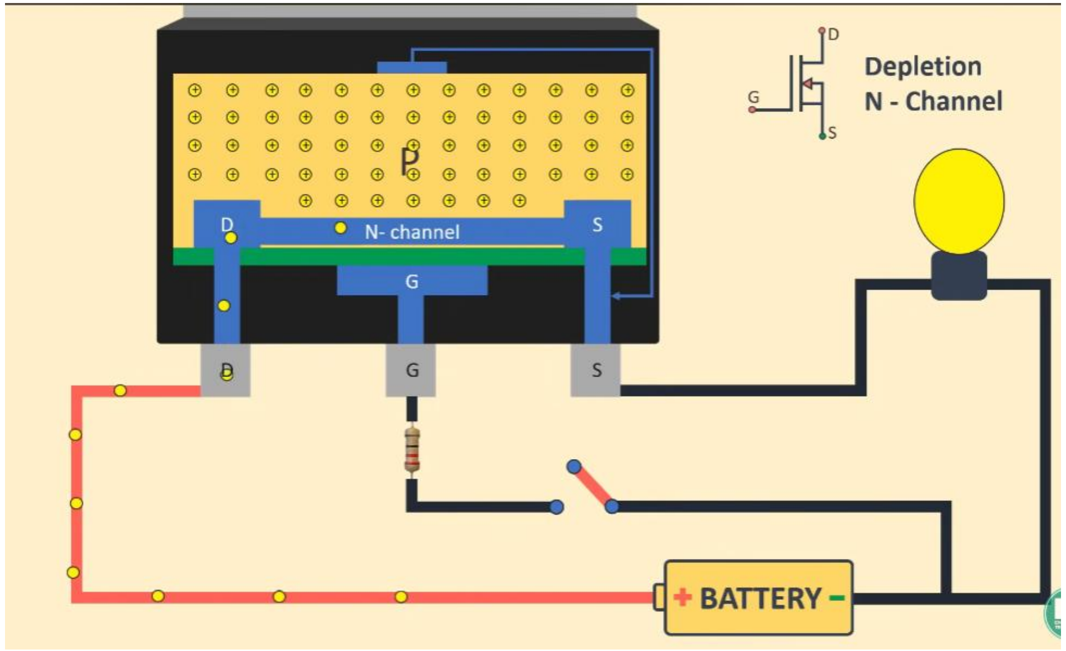


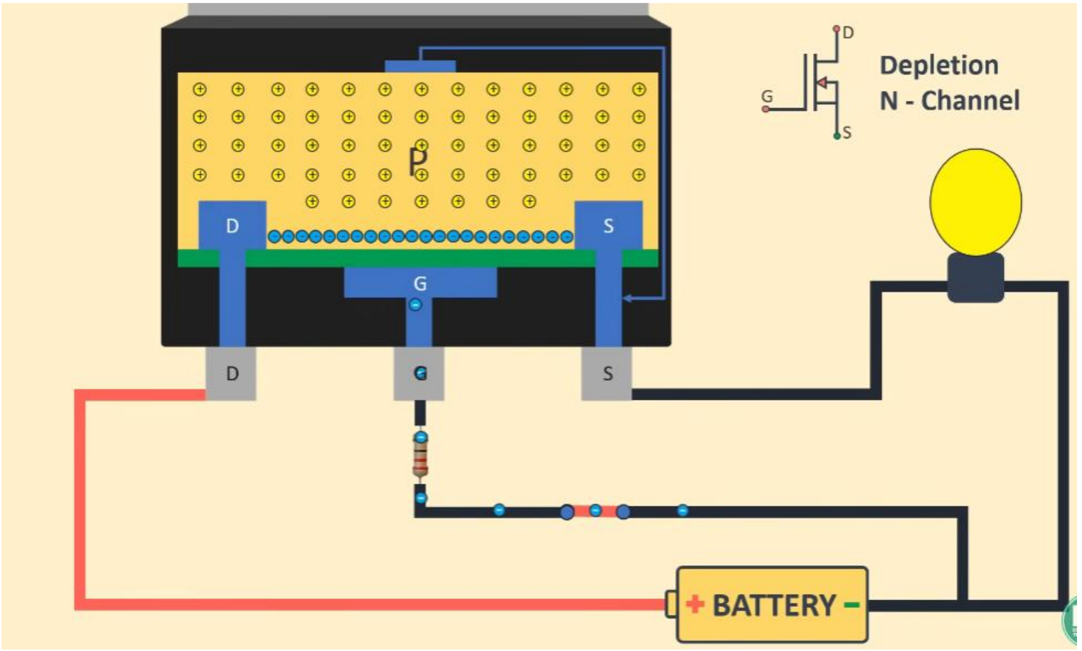
Enhancement P - Channel MOSFET



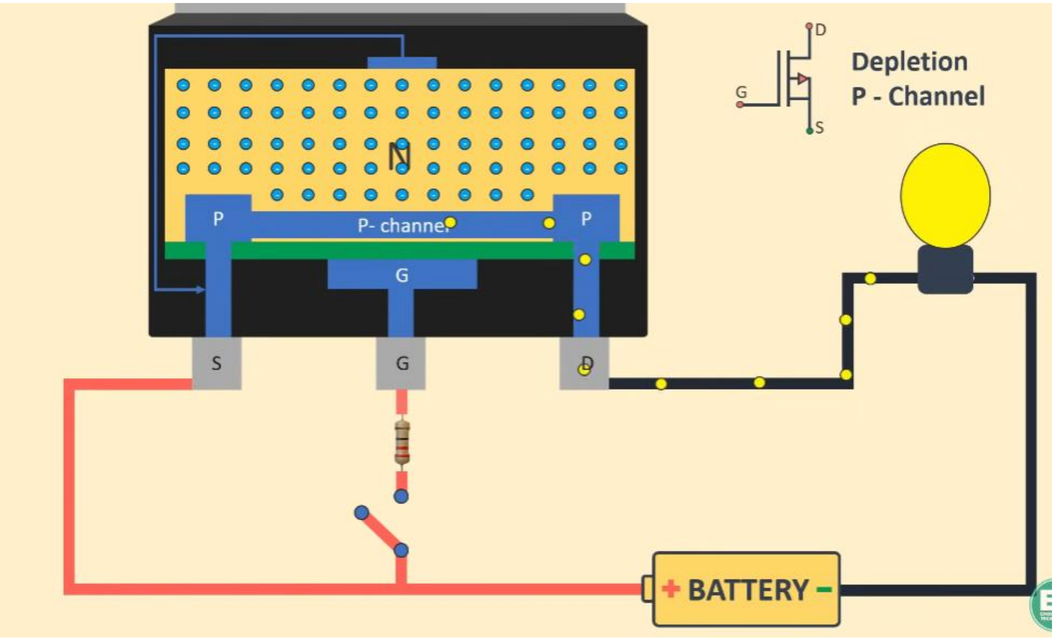


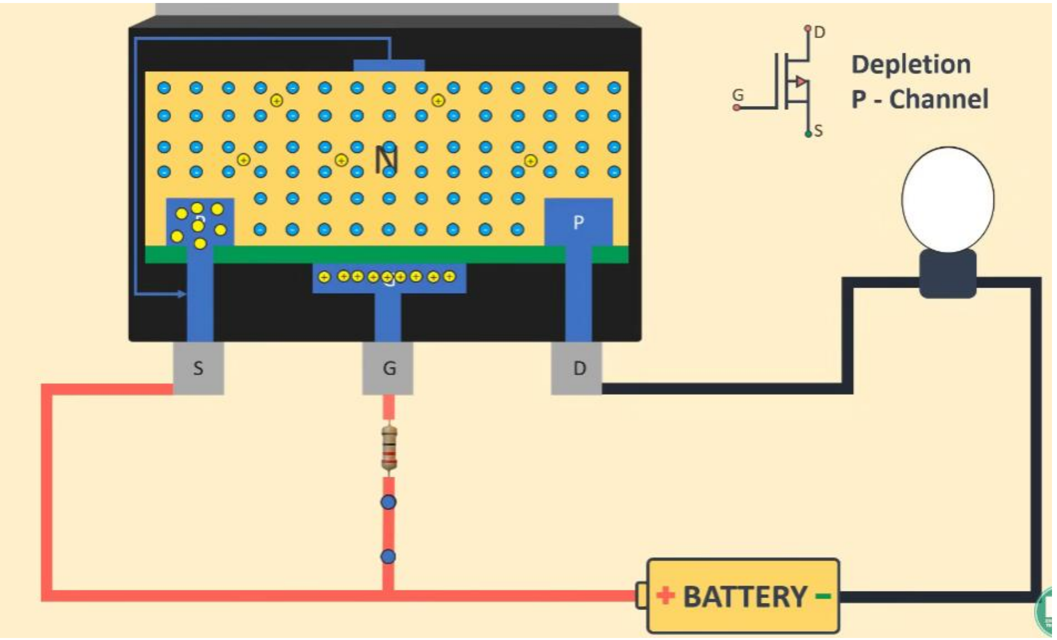
Depletion N - Channel MOSFET



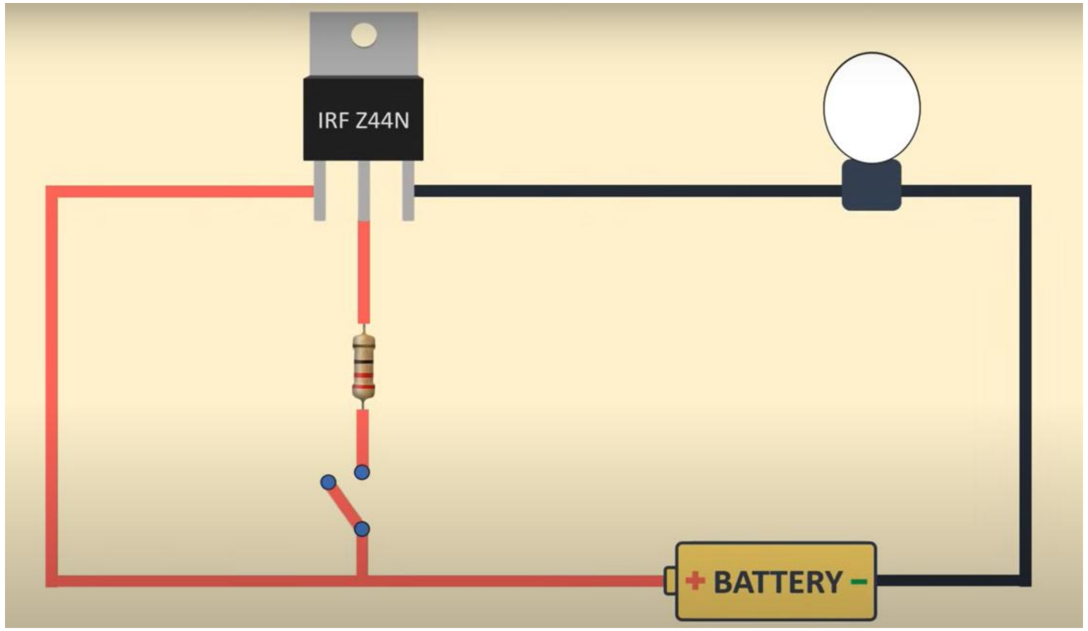


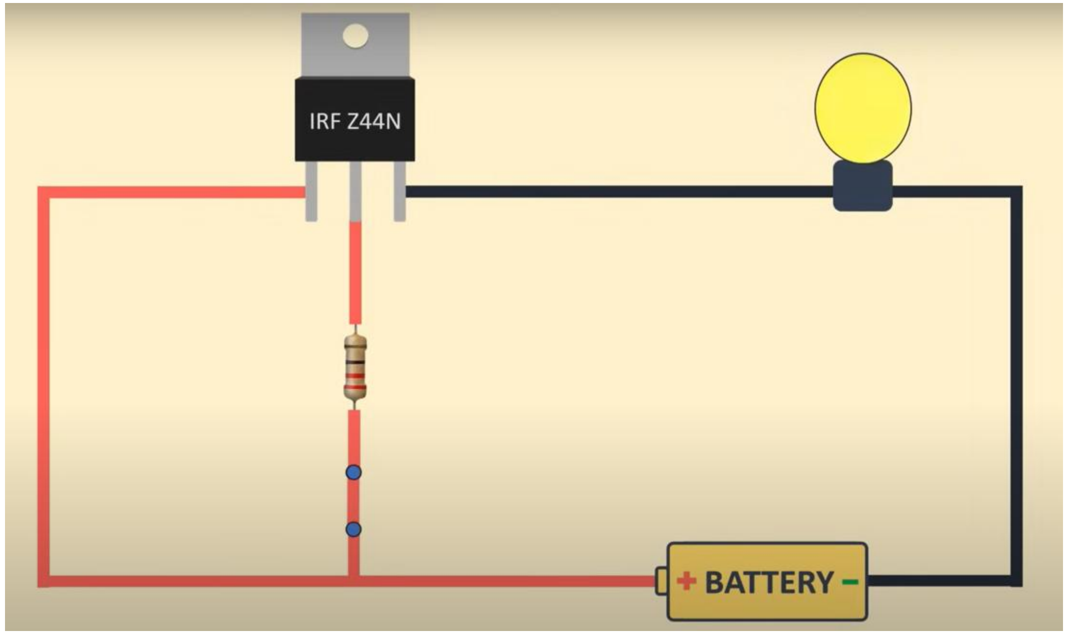
Depletion N - Channel MOSFET



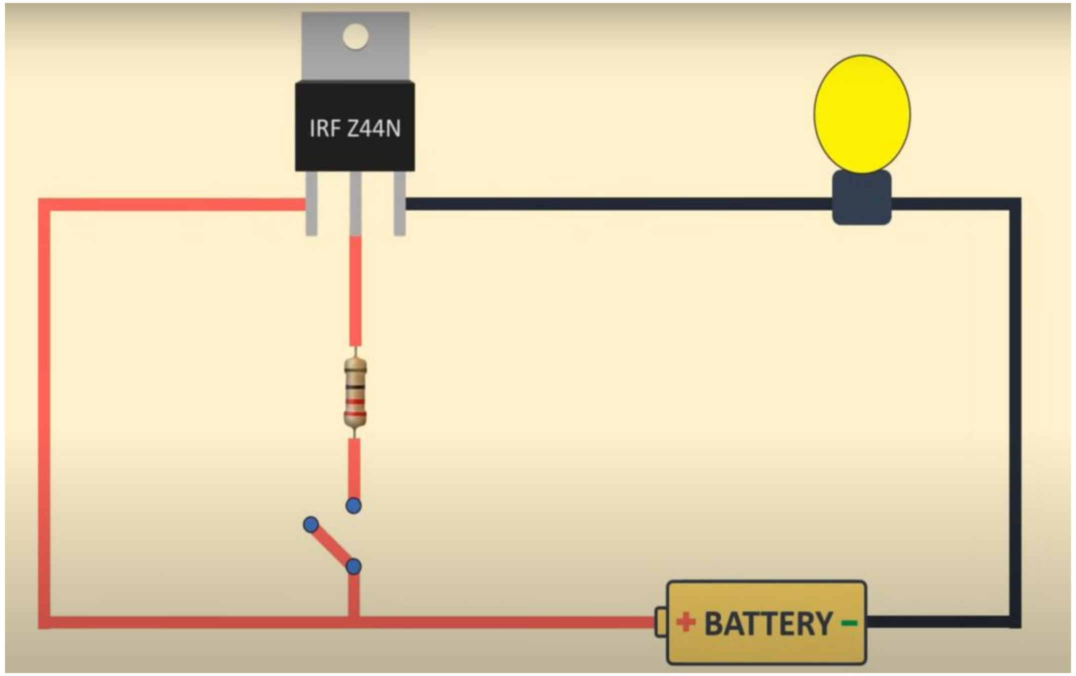


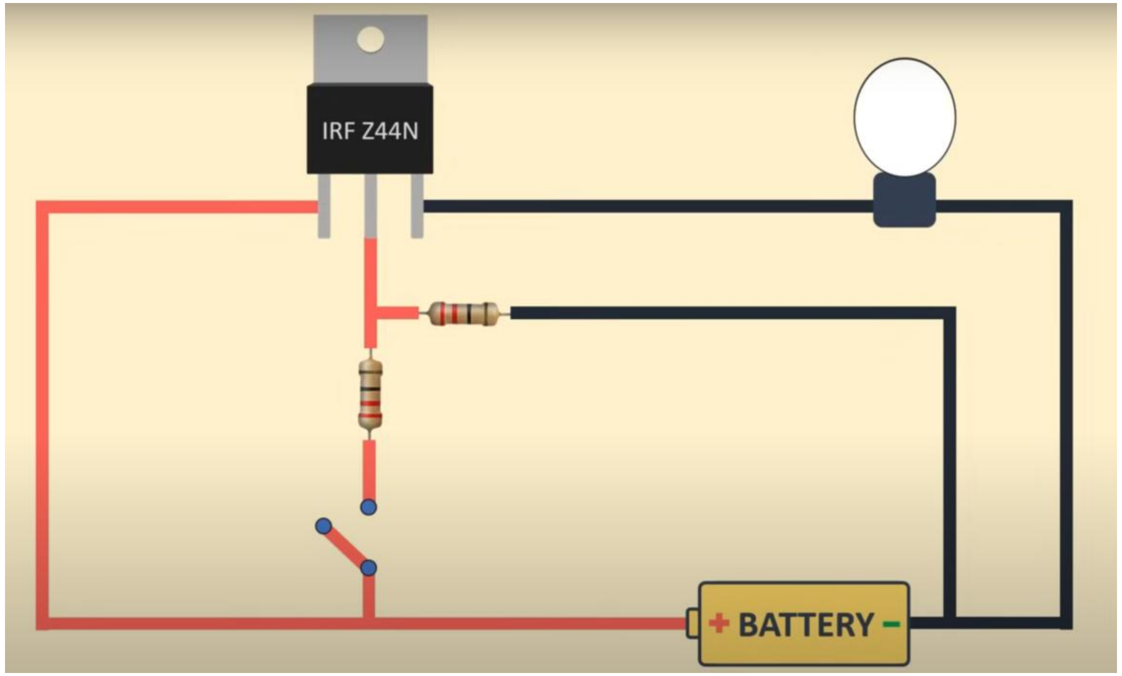
MOSFET Setup





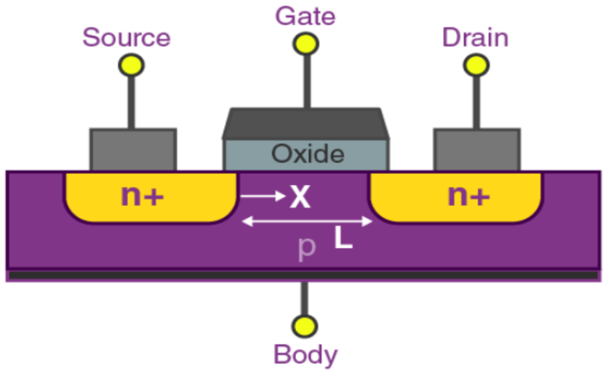
MOSFET Setup





MOSFET Construction

- The p-type semiconductor forms the base of the MOSFET.
- The two types of the base are highly doped with an n-type impurity which is marked as n+ in the diagram.
- From the heavily doped regions of the base, the terminals source and drain originate.
- The layer of the substrate is coated with a layer of silicon dioxide for insulation.
- A thin insulated metallic plate is kept on top of the silicon dioxide and it acts as a capacitor.
- The gate terminal is brought out from the thin metallic plate.
- A DC circuit is then formed by connecting a voltage source between these two n-type regions.



Operating Regions of MOSFET

A MOSFET is seen to exhibit three operating regions. Here, we will discuss those regions.

❑ Cut-Off Region

The cut-off region is a region in which there will be no conduction and as a result, the MOSFET will be OFF. In this condition, MOSFET behaves like an open switch.

❑ Ohmic Region

The ohmic region is a region where the current (I_{DS})increases with an increase in the value of V_{DS}. When MOSFETs are made to operate in this region, they are used as amplifiers.

❑ Saturation Region

In the saturation region, the MOSFETs have their I_{DS} constant in spite of an increase in V_{DS} and occurs once V_{DS} exceeds the value of pinch-off voltage V_P. Under this condition, the device will act like a closed switch through which a saturated value of I_{DS} flows. As a result, this operating region is chosen whenever MOSFETs are required to perform switching operations.

Symbols details

I_D **(Drain Current)** – The current flowing from drain to source in a MOSFET.

I_D(on) **(Drain Current when ON)** – The drain current when the MOSFET is fully turned ON.

V_GS **(Gate-Source Voltage)** – The voltage applied between the gate and source terminals.

V_GS(off) **(Gate-Source Cutoff Voltage)** – The gate-source voltage at which the MOSFET turns OFF (for JFETs and depletion-mode MOSFETs).

V_GS(th) **(Threshold Voltage)** – The minimum gate-source voltage required to turn ON an enhancement-mode MOSFET.

Example

Example 19.32. *Determine the drain-to-source voltage (V_{DS}) in the circuit shown in Fig. 19.51 above if V_{DD} = +18V and R_D = 620Ω. The MOSFET data sheet gives $V_{GS(off)}$ = – 8V and I_{DSS} = 12 mA.*

Solution. Since $I_D = I_{DSS}$ = 12 mA, the V_{DS} is given by;

$$V_{DS} = V_{DD} - I_{DSS} R_D$$

$$= 18\text{V} - (12 \text{ mA})(0.62 \text{ k}\Omega) = \mathbf{10.6V}$$

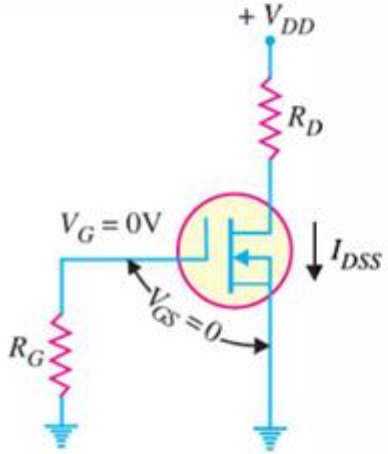


Fig. 19.51

Example

Example 19.33. *The D-MOSFET used in the amplifier of Fig. 19.54 has an I_{DSS} = 12 mA and g_m = 3.2 mS. Determine (i) d.c. drain-to-source voltage V_{DS} and (ii) a.c. output voltage. Given v_{in} = 500 mV.*

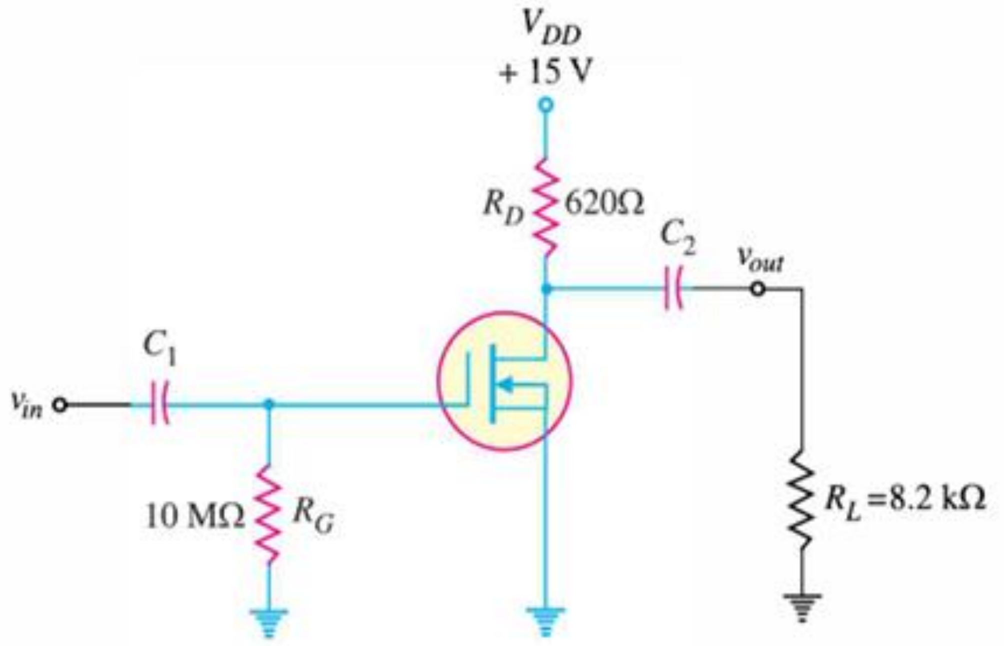


Fig. 19.54

Example

Example 19.33. *The D-MOSFET used in the amplifier of Fig. 19.54 has an I_{DSS} = 12 mA and g_m = 3.2 mS. Determine (i) d.c. drain-to-source voltage V_{DS} and (ii) a.c. output voltage. Given v_{in} = 500 mV.*

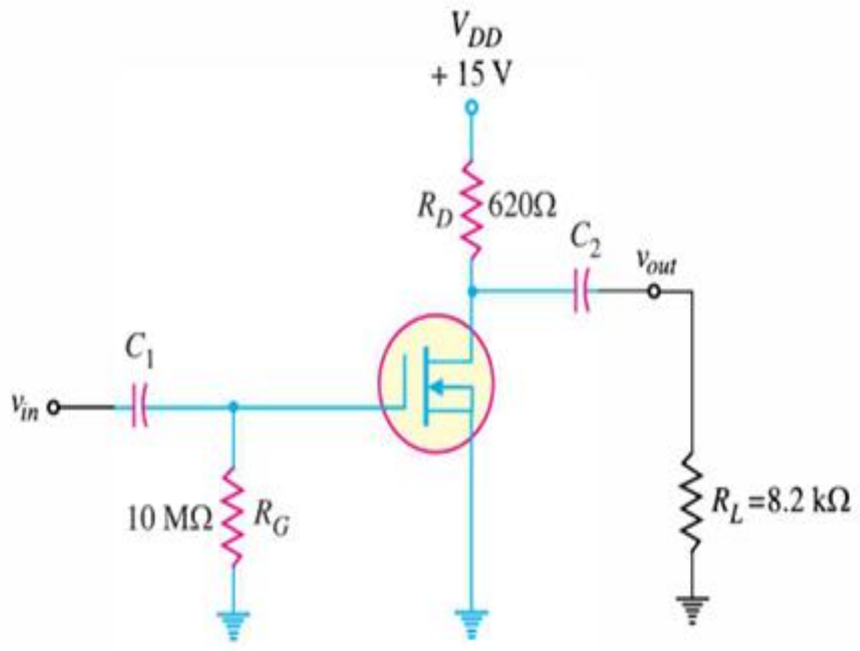


Fig. 19.54

Solution.

(*i*) Since the amplifier is zero biased, $I_D = I_{DSS}$ = 12 mA.

$$\therefore \quad V_{DS} = V_{DD} - I_{DSS} R_D$$
$$= 15\text{V} - (12\text{ mA})(0.62\text{ k}\Omega) = \mathbf{7.56V}$$

(*ii*) Total a.c. drain resistance R_{AC} of the circuit is

$$R_{AC} = R_D \parallel R_L = 620\Omega \parallel 8.2\text{ k}\Omega = 576\Omega$$

$$\therefore \quad v_{out} = A_v \times v_{in} = (g_m R_{AC})(v_{in})$$
$$= (3.2 \times 10^{-3}\, S \times 576\ \Omega)(500\text{ mV}) = \mathbf{922\ mV}$$

Example

Example 19.37. *Determine the values of I_D and V_{DS} for the circuit shown in Fig. 19.62. The data sheet for this particular MOSFET gives $I_{D\,(on)}$ = 10 mA when $V_{GS} = V_{DS}$*

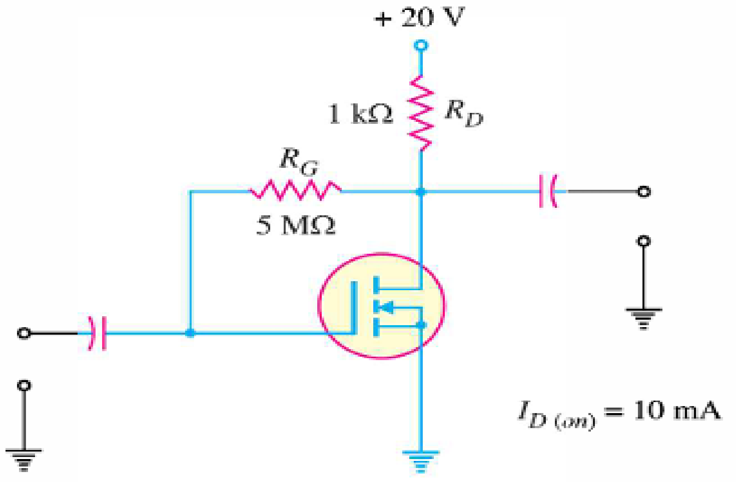


Fig. 19.62

Solution. Since in the drain-feedback circuit $V_{GS} = V_{DS}$,

$$\therefore \quad I_D = I_{D\,(on)} = \mathbf{10\ mA}$$

The value of V_{DS} (and thus V_{GS}) is given by ;

$$V_{DS} = V_{DD} - I_D R_D$$

$$= 20\text{V} - (10\text{ mA})(1\text{ k}\Omega) = 20\text{V} - 10\text{V} = \mathbf{10V}$$

MOSFET vs BJT

MOSFET	BJT
There are two types of MOSFET and they are named: N-type or P-type	BJT is of two types and they are named as: PNP and NPN
MOSFET is a voltage-controlled device	BJT is a current-controlled device
The input resistance of MOSFET is high.	The input resistance of BJT is low.
Used in high current applications	Used in low current applications

MOSFET Applications

- Radiofrequency applications use MOSFET amplifiers extensively.
- MOSFET behaves as a passive circuit element.
- Power MOSFETs can be used to regulate DC motors.
- MOSFETs are used in the design of the chopper circuit.

Advantages & Disadvantages

Advantages of MOSFET

- MOSFETs operate at greater efficiency at lower voltages.
- Absence of gate current results in high input impedance producing high switching speed.

Disadvantages of MOSFET

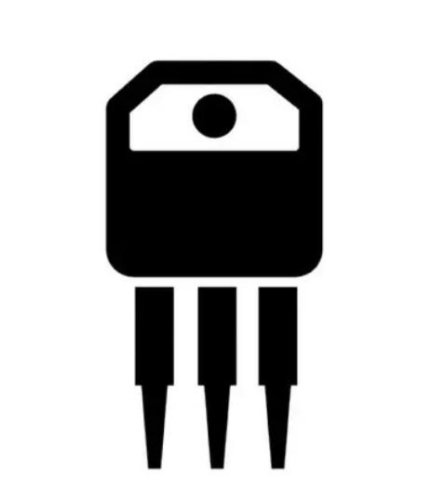
- MOSFETs are vulnerable to damage by electrostatic charges due to the thin oxide layer.
- Overload voltages make MOSFETs unstable.

Thank you for hearing with patience

Course Title: Embedded Systems and IoT

Course Code: CSE233

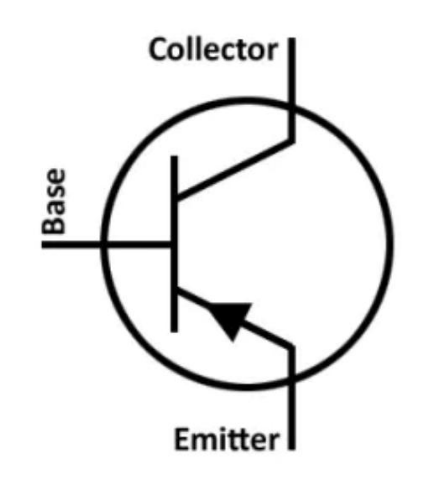
TRANSISTORS



Prepared By:

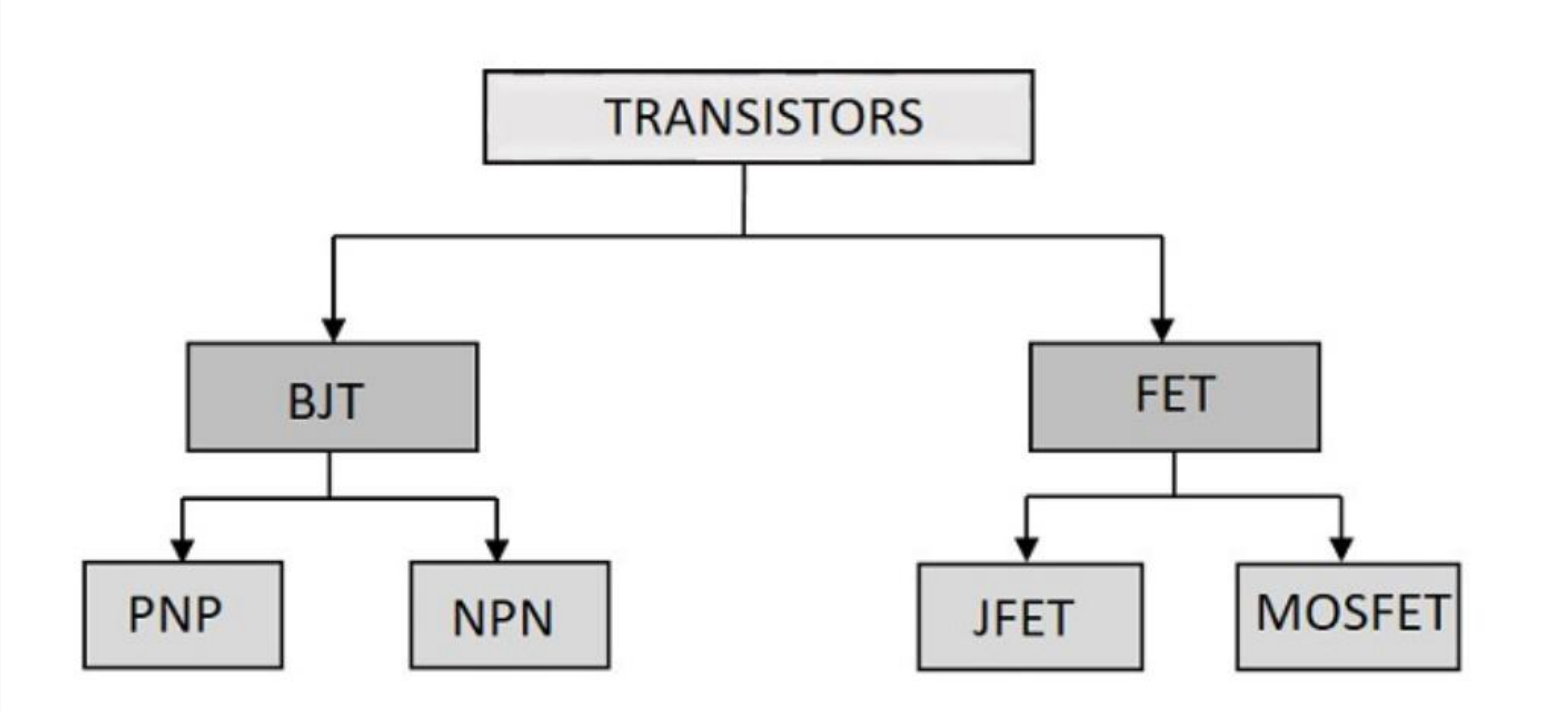
Jul Jalal Al-Mamur Sayor (JMS)

Lecturer, Dept. of CSE, DIU



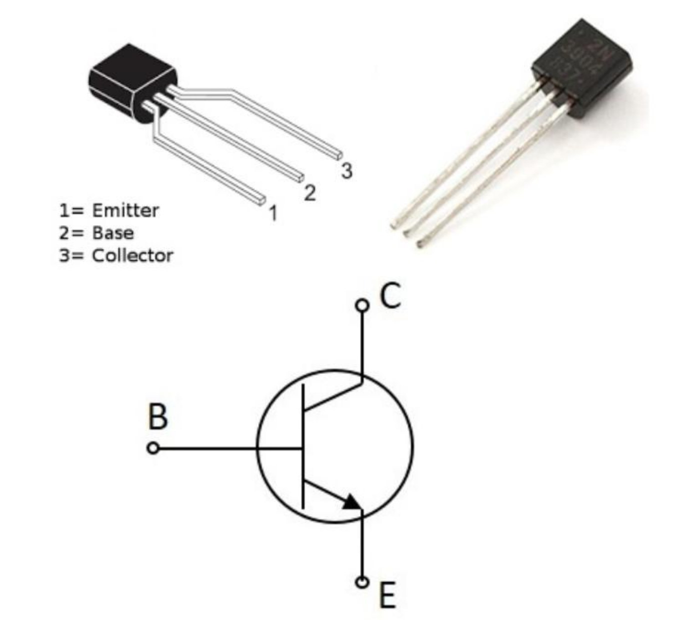
WHAT IS TRANSISTOR

- A transistor is a type of **semiconductor** device that can be used **to conduct** or **insulate** electric current or voltage. A transistor basically acts as **a switch** and/or **an amplifier**. In simple words, a transistor is a device that is used to control or regulate the flow of electronic signals.



TYPES OF TRANSISTORS

- **Bipolar Junction Transistor:** A Bipolar junction transistor, shortly termed as BJT is called so as it has two PN junctions for its function. This BJT is nothing but a normal transistor. Usually NPN transistor is preferred for the sake of convenience. The following image shows how a practical BJT looks like.



A current-controlled device. It uses both electrons and holes (bipolar conduction) to conduct.

Has three terminals:

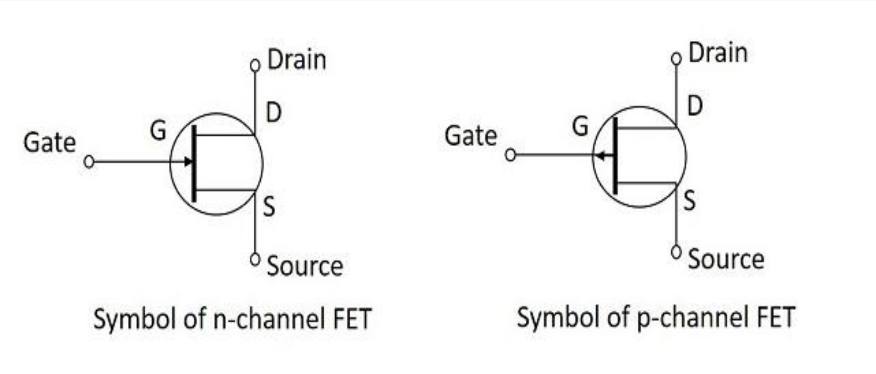
- Base: This is used to activate the transistor.
- Collector: It is the positive lead of the transistor.
- Emitter: It is the negative lead of the transistor.

Two types: NPN and PNP

Current flows from emitter to collector, controlled by base current.

TYPES OF TRANSISTORS

- An FET is a **three-terminal** unipolar semiconductor device. It is a voltage controlled device unlike a bipolar junction transistor. The main advantage of FET is that it has a very high input impedance, which is in the order of Mega Ohms. It has many advantages like low power consumption, low heat dissipation and FETs are highly efficient devices. The following image shows how a practical FET looks like.



A voltage-controlled device. Uses only one type of charge carrier (unipolar): either electrons (N-channel) or holes (P-channel).

Has three terminals: Gate (G), Source (S), Drain (D)

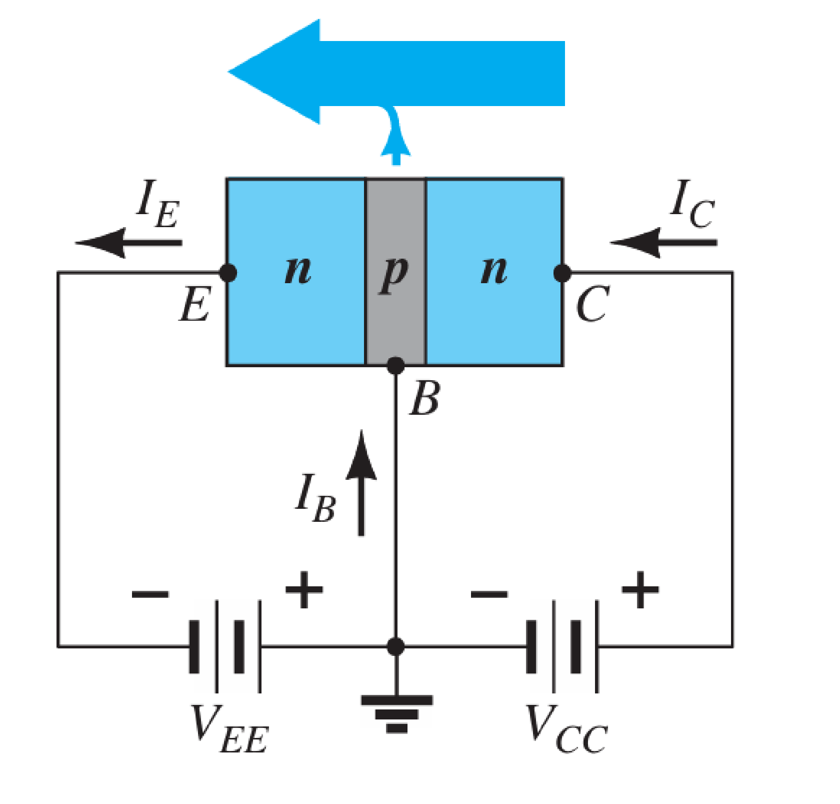
Main types:

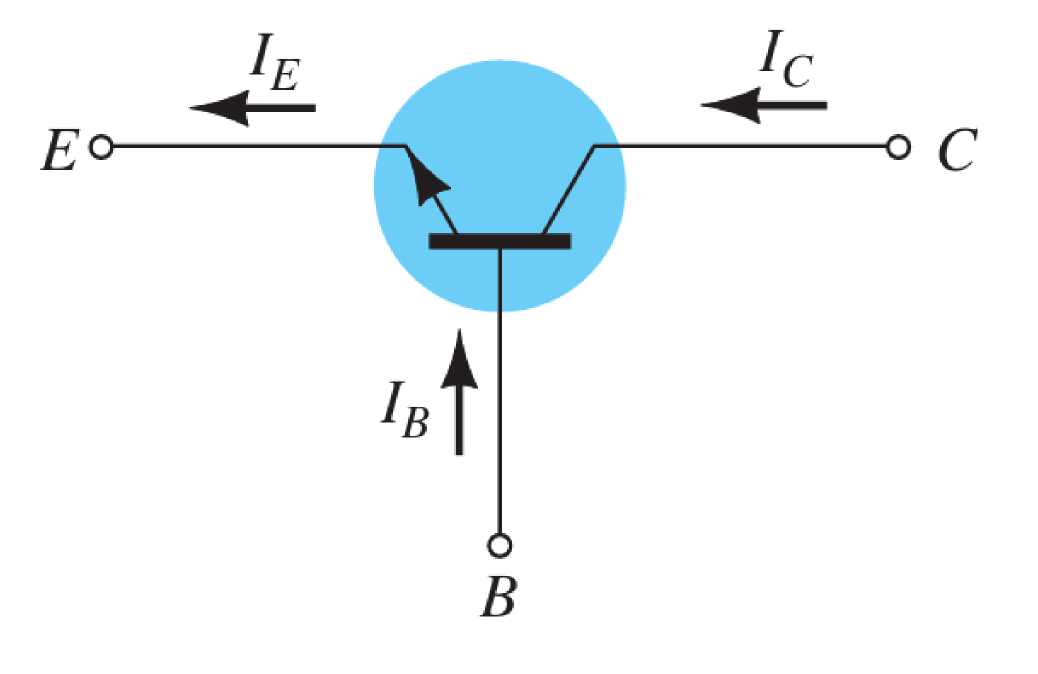
i. JFET (Junction FET)

ii. MOSFET (Metal-Oxide Semiconductor FET)

Current flows from source to drain, controlled by gate voltage

TRANSISTOR OPERATION

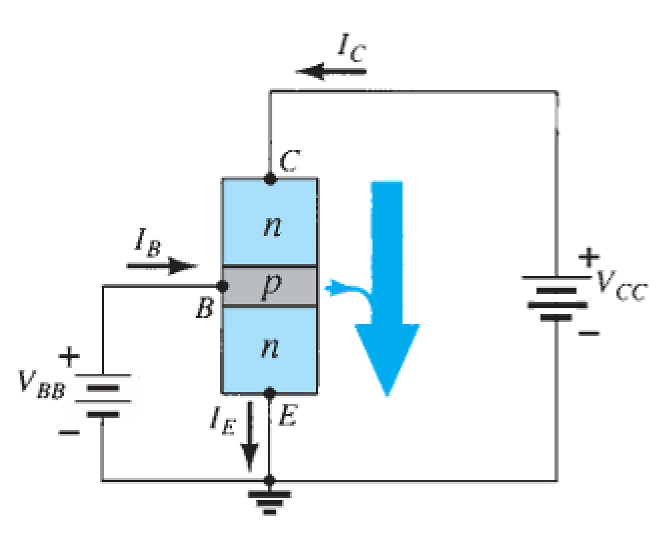




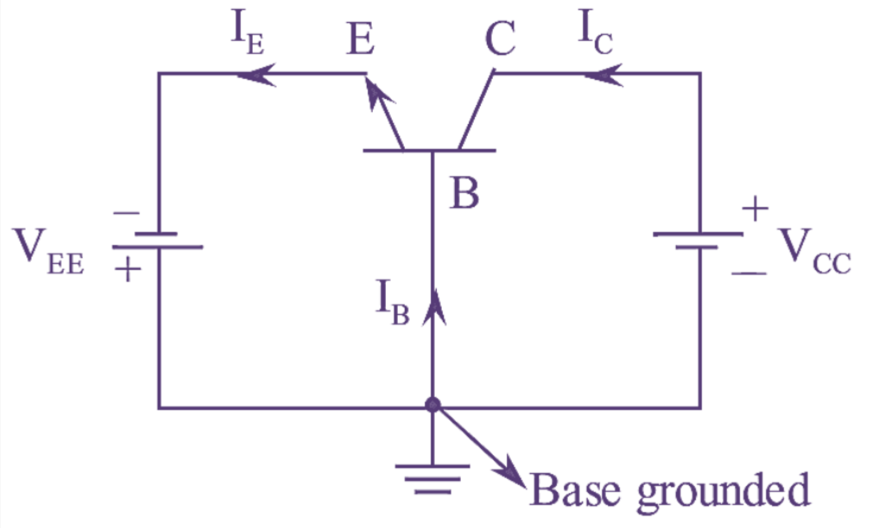
(b)

BASIC TRANSISTOR CONFIGURATIONS

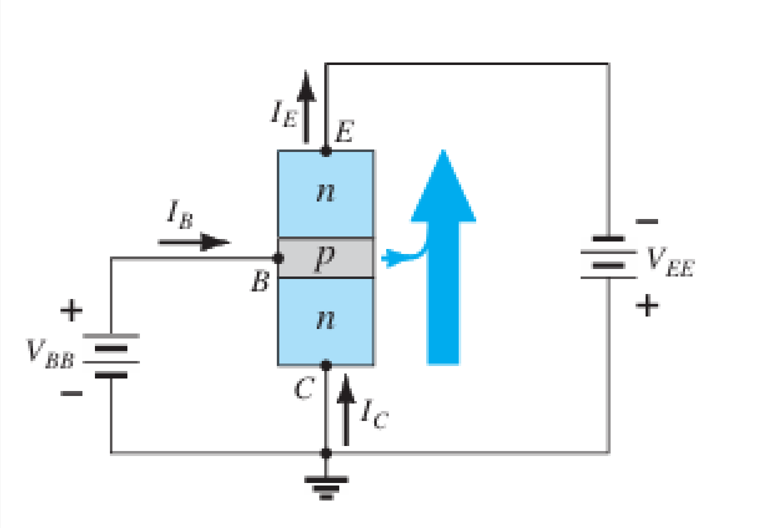
- 1. CB – Common Base Configuration
- 2. CE – Common Emitter Configuration
- 3.



Common-Emitter



Common-Base



Common-Collector

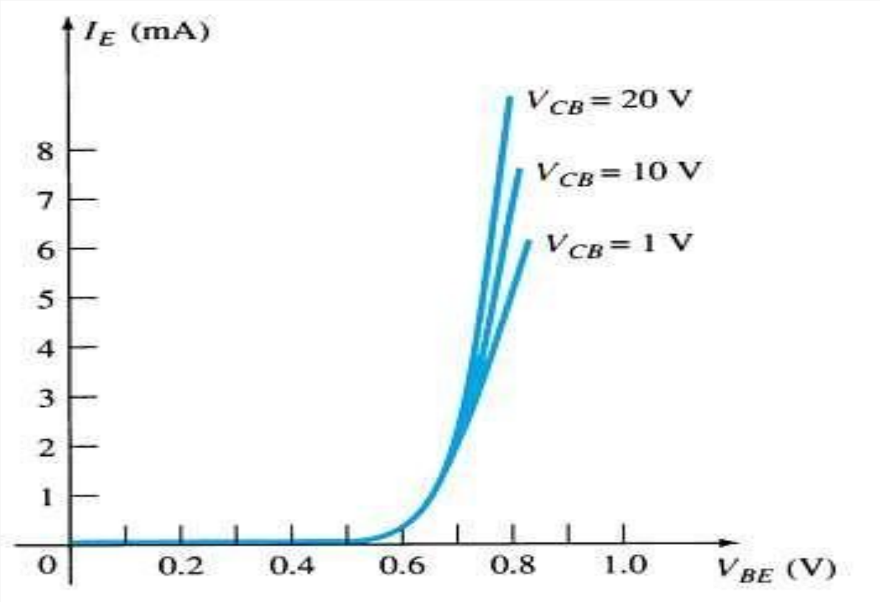
BASIC TRANSISTOR CONFIGURATIONS

Configuration	Common Terminal	Input Between	Output Between
Common Emitter	Emitter	Base–Emitter	Collector–Emitter
Common Base	Base	Emitter–Base	Collector–Base
Common Collector	Collector	Base–Collector	Emitter–Collector

COMMON-BASE

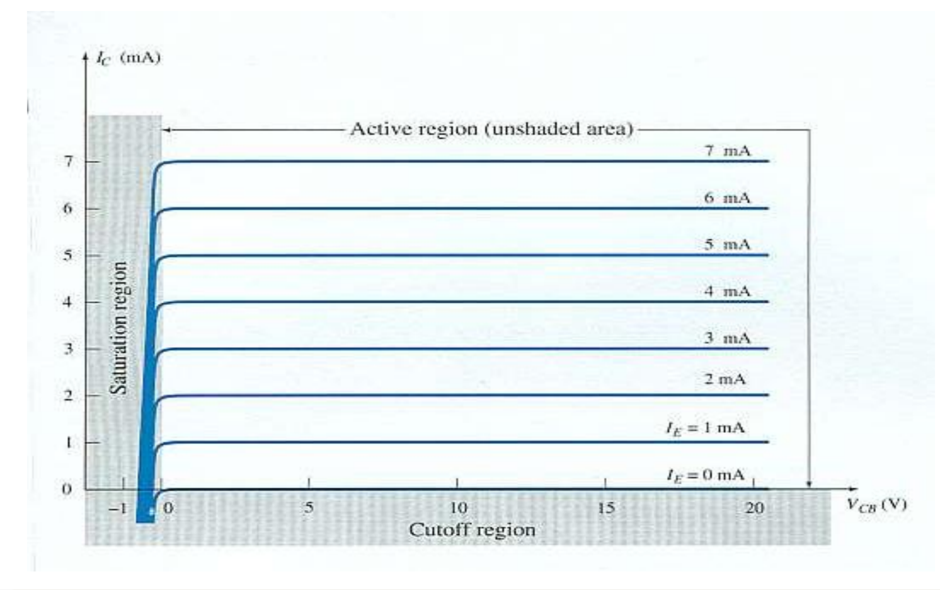
Input Characteristics

This curve shows the relationship between of input current (I_E) to input voltage (V_{BE}) for three output voltage (V_{CB}) levels.



Output Characteristics

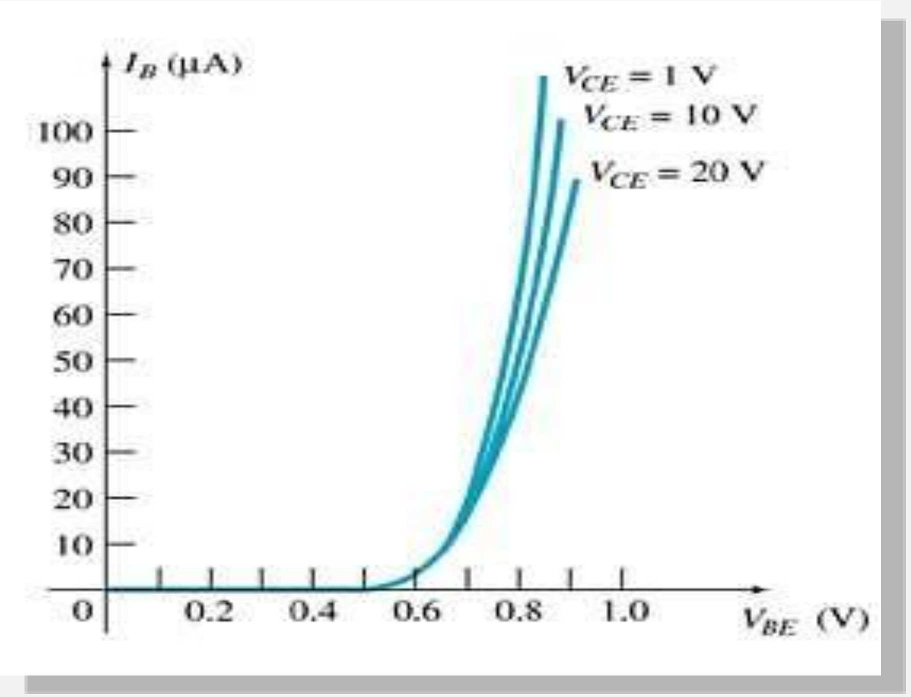
This graph demonstrates the output current (I_C) to an output voltage (V_{CB}) for various levels of input current (I_E).



COMMON-EMITTER

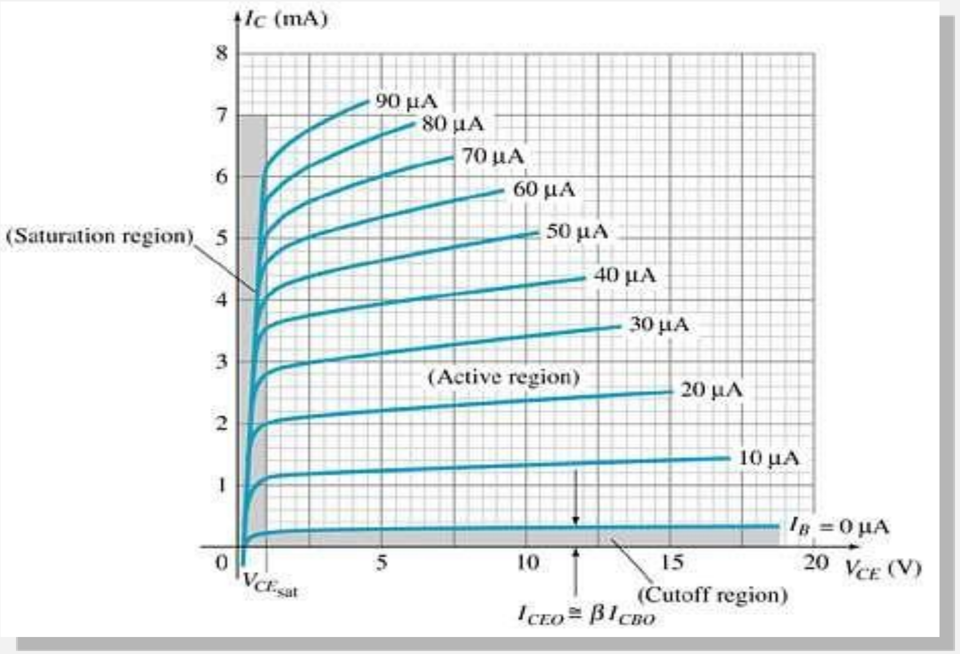
Input Characteristics

- Shows the relationship between input current (IB)
- and input voltage (VBE)
- Measured at constant output voltage (VCE)
- Curve: IB vs. VBE for different fixed VCE values



Output Characteristics

- Shows the relationship between output current (IC) and output voltage (VCE)
- Measured at constant input current (IB)
- Curve: IC vs. VCE for different fixed IB values



Ie = Ib + Ic

BASIC MATHEMATICS

A transistor has the following data:

- $\beta = 49$
- $I_{CBO} = 15\,\mu A$
- $I_B = 40\,\mu A$

Find the following:

1. α
2. I_C using β
3. I_E
4. I_{CEO}
5. I_C including I_{CBO}
6. Verify $I_E = I_C + I_B$

$$\alpha = \frac{\beta}{\beta + 1} = \frac{49}{49 + 1} = \frac{49}{50} = 0.98$$

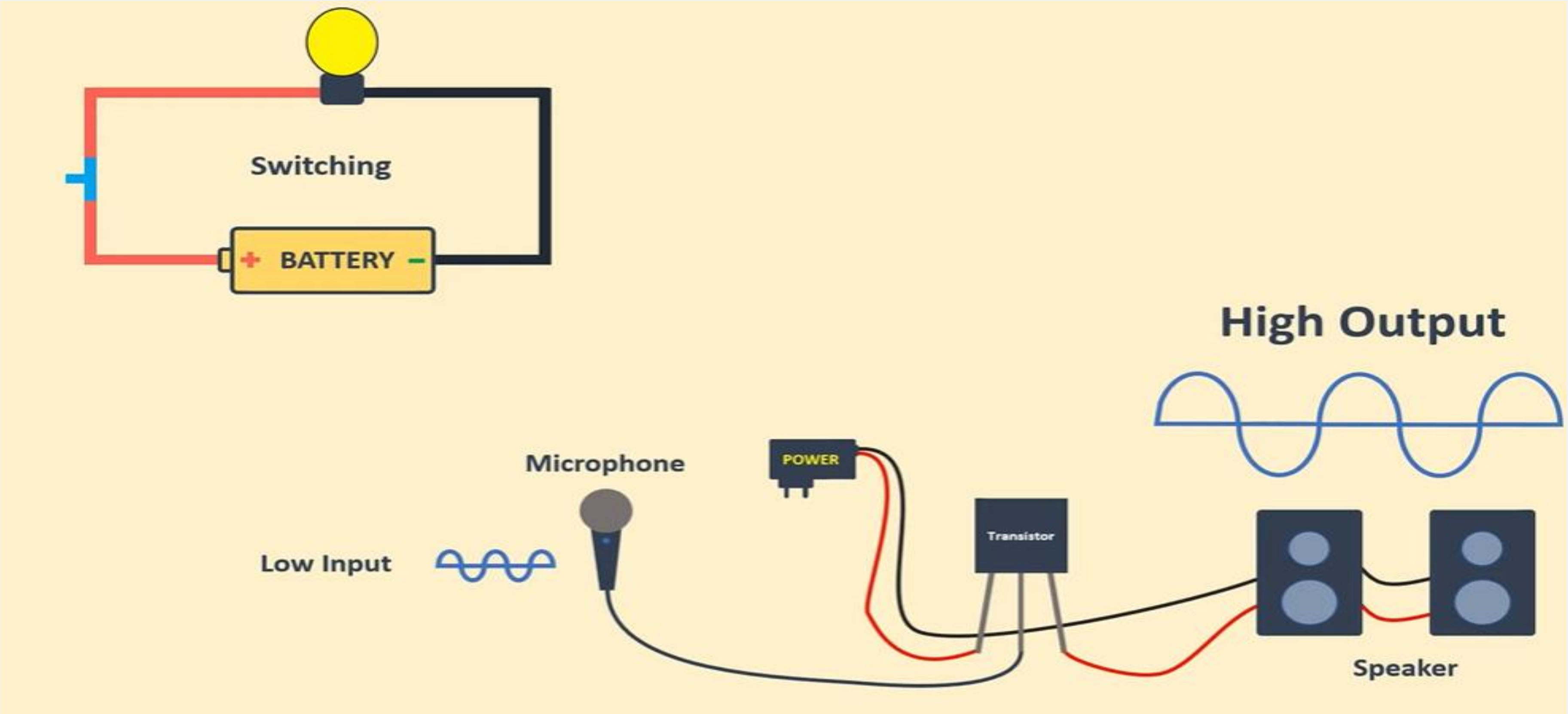
$$I_C = \beta \cdot I_B = 49 \cdot 40\,\mu A = 1960\,\mu A$$

$$I_E = I_C + I_B = 1.96\,mA + 40\,\mu A = 2.00\,mA$$

$$I_{CEO} = (1 + \beta) \cdot I_{CBO} = 50 \cdot 15\,\mu A = 750\,\mu A$$

$$I_C = \beta I_B + I_{CBO} = 1.96\,mA + 15\,\mu A = 1.975\,mA$$

USAGE OF TRANSISTOR



WHAT TO READ BEFORE EXAM?

- Construction*
- Classification
- Input output Curves*-- CB, CE
- Usage
- Advantage disadvantages
- Differences

"LEARNING IS A TREASURE THAT WILL FOLLOW ITS OWNER EVERYWHERE, SO KEEP LEARNING"

THANK YOU

Course Title: Embedded Systems and IoT

Course Code: CSE233

MOSFET

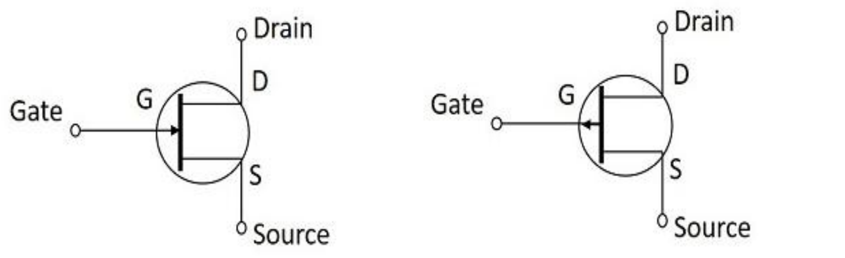
Prepared By:

Jul Jalal Al-Mamur Sayor (JMS)

Lecturer, Dept. of CSE, DIU

TYPES OF TRANSISTORS

- An FET is a **three-terminal** unipolar semiconductor device. It is a voltage controlled device unlike a bipolar junction transistor. The main advantage of FET is that it has a very high input impedance, which is in the order of Mega Ohms. It has many advantages like low power consumption, low heat dissipation and FETs are highly efficient devices. The following image shows how a practical FET looks like.



Symbol of n-channel FET   Symbol of p-channel FET

A voltage-controlled device. Uses only one type of charge carrier (unipolar): either electrons (N-channel) or holes (P-channel).

Has three terminals: Gate (G), Source (S), Drain (D)

Main types:

i. JFET (Junction FET)

ii. MOSFET (Metal-Oxide Semiconductor FET)

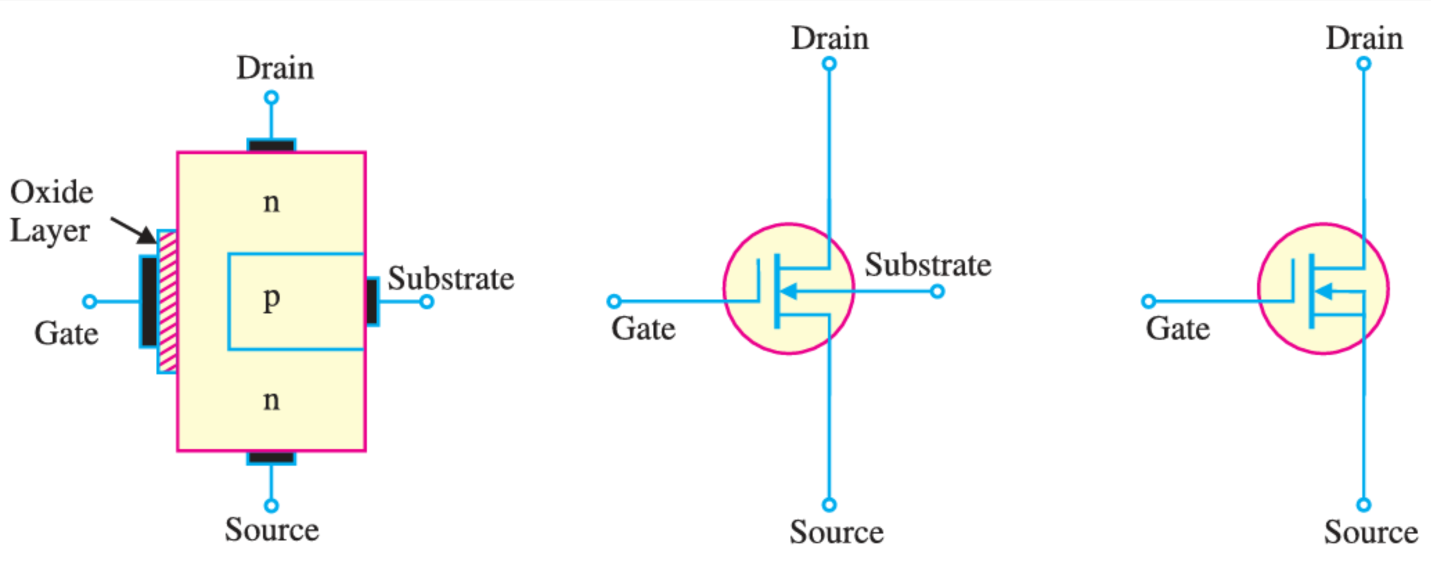
Current flows from source to drain, controlled by gate voltage

ADVANTAGES OF FET OVER BJT

Feature	**BJT (Bipolar Junction Transistor)**	**FET (Field Effect Transistor)**
Control Type	Current-controlled	Voltage-controlled
Input Impedance	Low	High
Switching Speed	Slower	Faster
Power Consumption	Higher (due to base current)	Lower (almost no gate current)
Thermal Stability	Poor (risk of thermal runaway)	Good
Noise	More noise	Less noise
Gain	High current gain	Lower gain
Application	Analog (amplifiers)	Digital, switching, ICs

MOSFET

- A **MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor)** is a type of field-effect transistor that controls the flow of electrical current between two terminals (source and drain) by varying the voltage applied to a third terminal (gate), which is insulated from the channel by a thin oxide layer.



Construction wise we can categories the device

Into four types:

i. P-channel Enhancement /EMOSFET

ii. n-channel Enhancement MOSFET

iii. n-channel Depletion /DMOSFET

iv. p-channel Depletion MOSFET

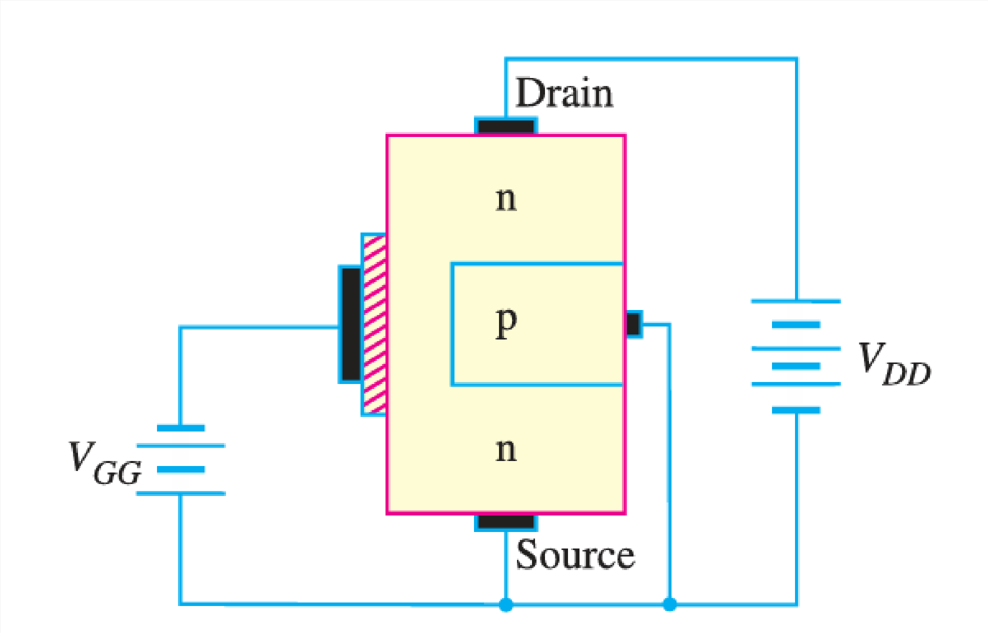
MOSFET

Depletion Mode MOSFET:

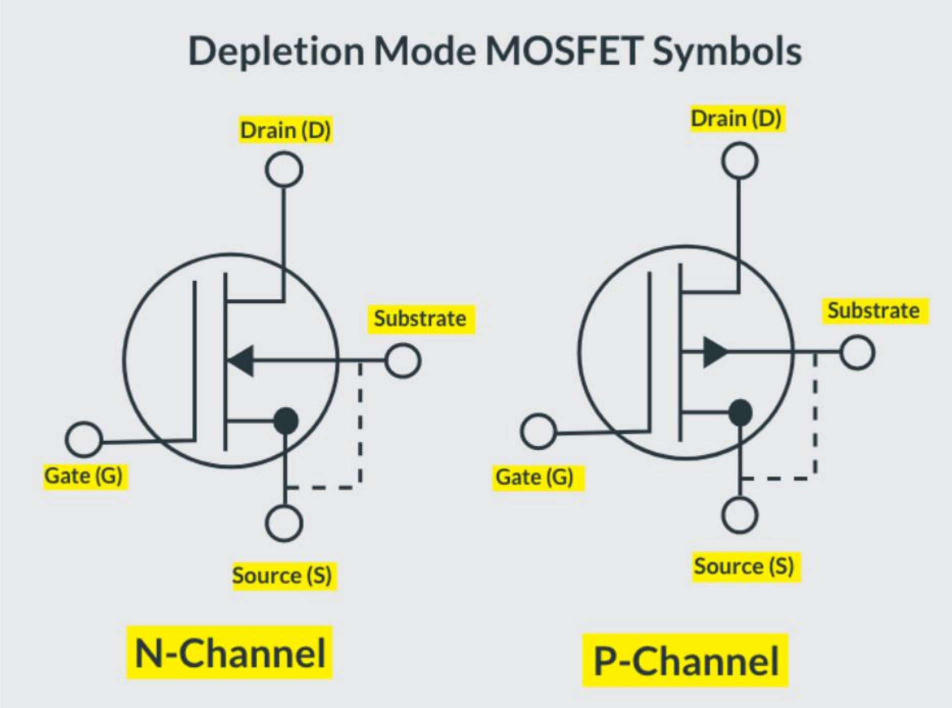
A **depletion mode MOSFET** is a transistor that is normally **ON** at zero gate-to-source voltage because a conductive channel already exists, and applying a gate voltage of opposite polarity reduces or depletes the channel, turning the device OFF.

Enhancement Mode MOSFET:

An **enhancement mode MOSFET** is a transistor that is normally **OFF** at zero gate-to-source voltage and requires a gate voltage to create a conductive channel between the source and drain terminals to turn it ON.



(i)

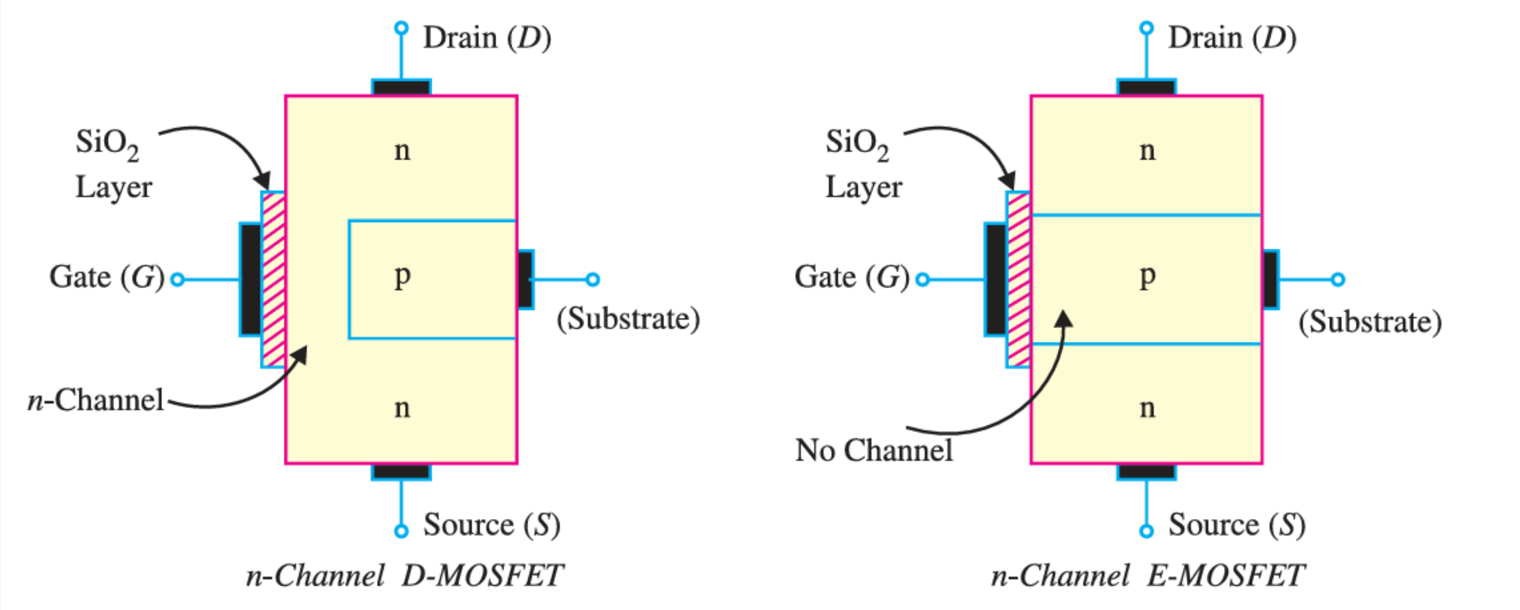


TYPES OF MOSFET

There are two basic types of MOSFETs:

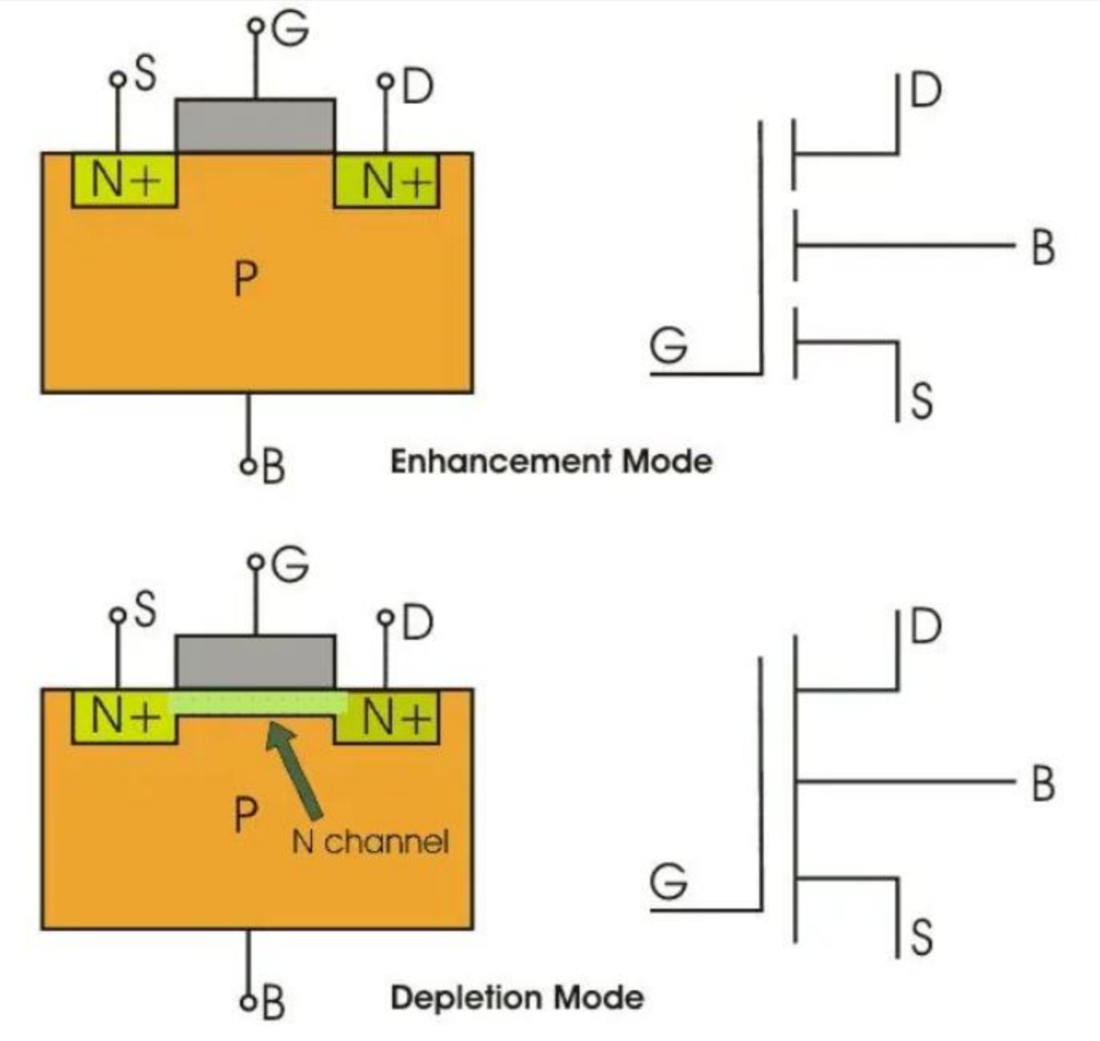
1. Depletion-type MOSFET or D-MOSFET: The D-MOSFET can be operated in both the depletion-mode and the enhancement-mode. For this reason, a D-MOSFET is sometimes called depletion/enhancement MOSFET.

2. Enhancement-type MOSFET or E-MOSFET: The E-MOSFET can be operated only in enhancement-mode.

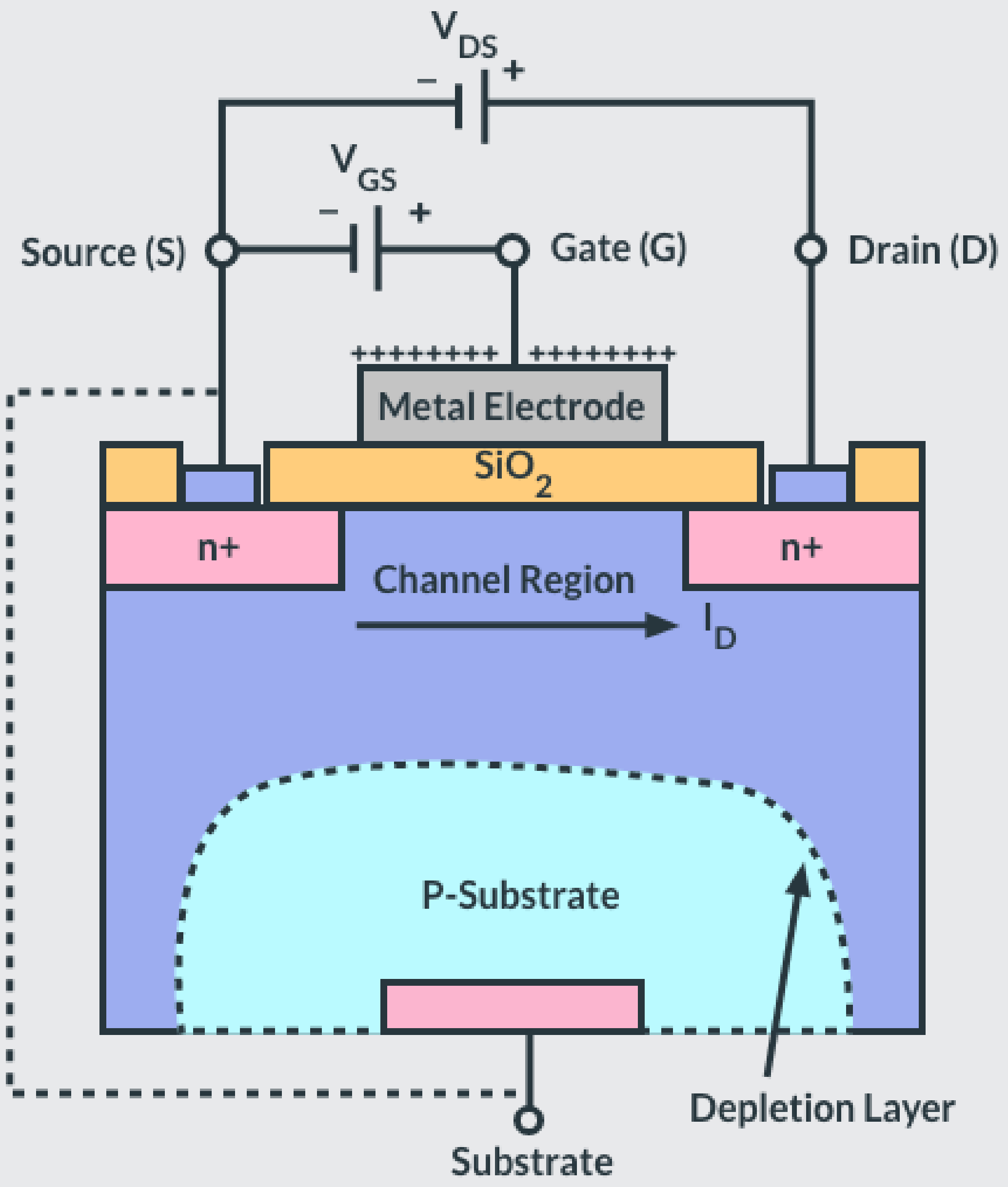


In depletion mode MOSFETs, a conductive channel exists between the source and drain even when no gate voltage is applied. In contrast, enhancement mode MOSFETs have no existing channel between the source and drain at zero gate voltage; a channel is formed only when a suitable gate voltage is applied.

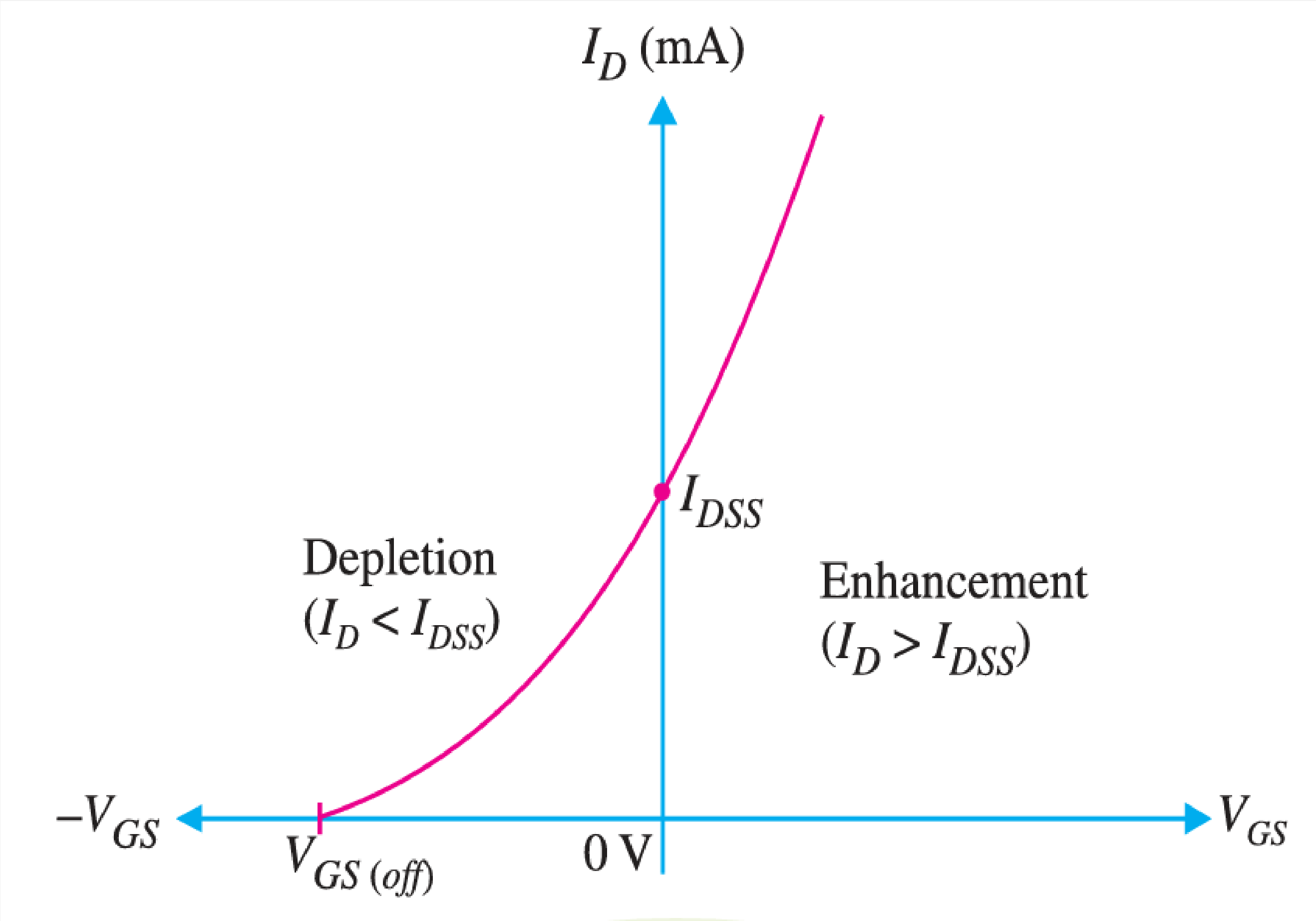
SYMBOL AND CONCEPT

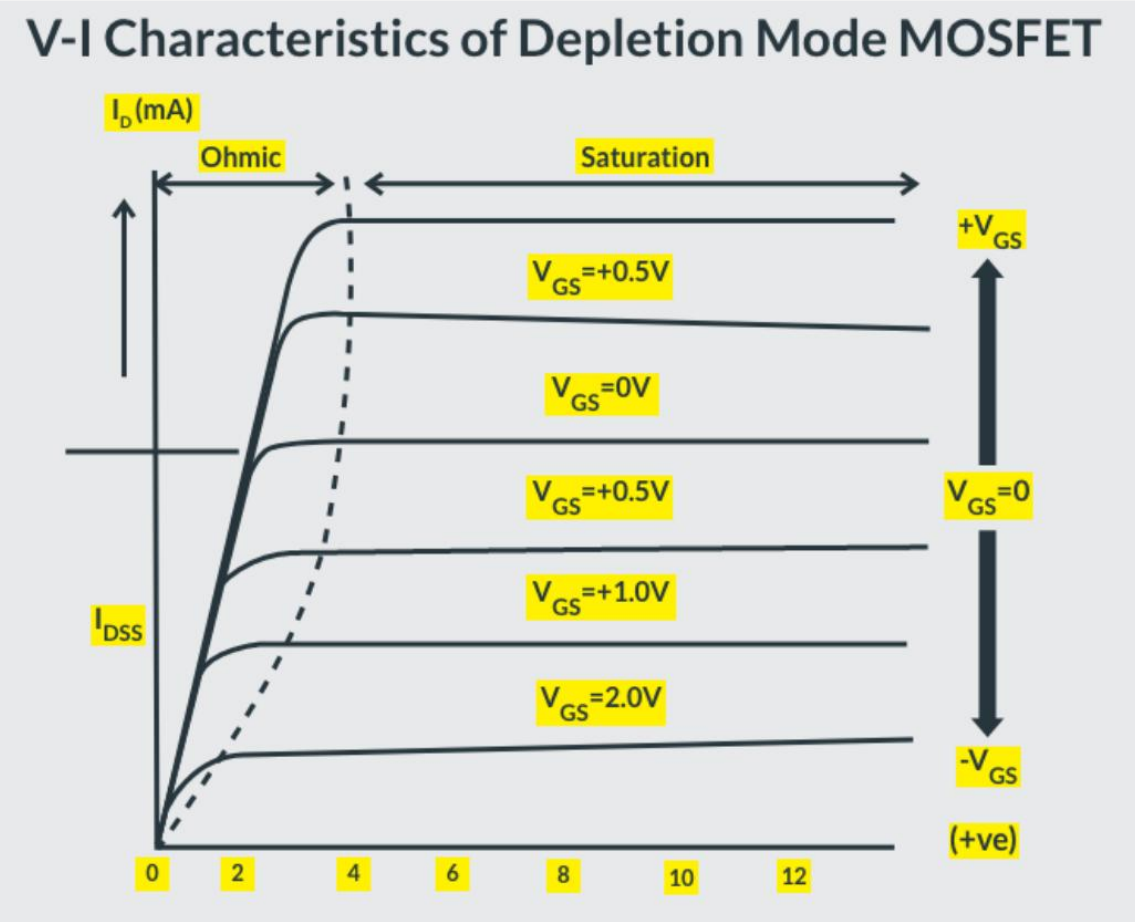


Structure of MOSFET



TRANSFER AND DRAIN CHARACTERISTICS CURVES





WHAT TO READ BEFORE EXAM?

- MOSFETs
- Types of MOSFET, (Construction)
- Symbols
- Operation of D-MOSFET
- D-MOSFET Transfer Characteristics and Drain

“ENGINEERS LIKE TO SOLVE PROBLEMS. IF THERE ARE NO PROBLEMS AVAILABLE, THEY WILL CREATE THEIR OWN ☺”

THANK YOU

Introduction to Microcontrollers

Provided By: Department of Cse
Modified By: Jul Jalal Al-Mamur Sayor (JMS),
Lecturer, Department of CSE

Daffodil International University

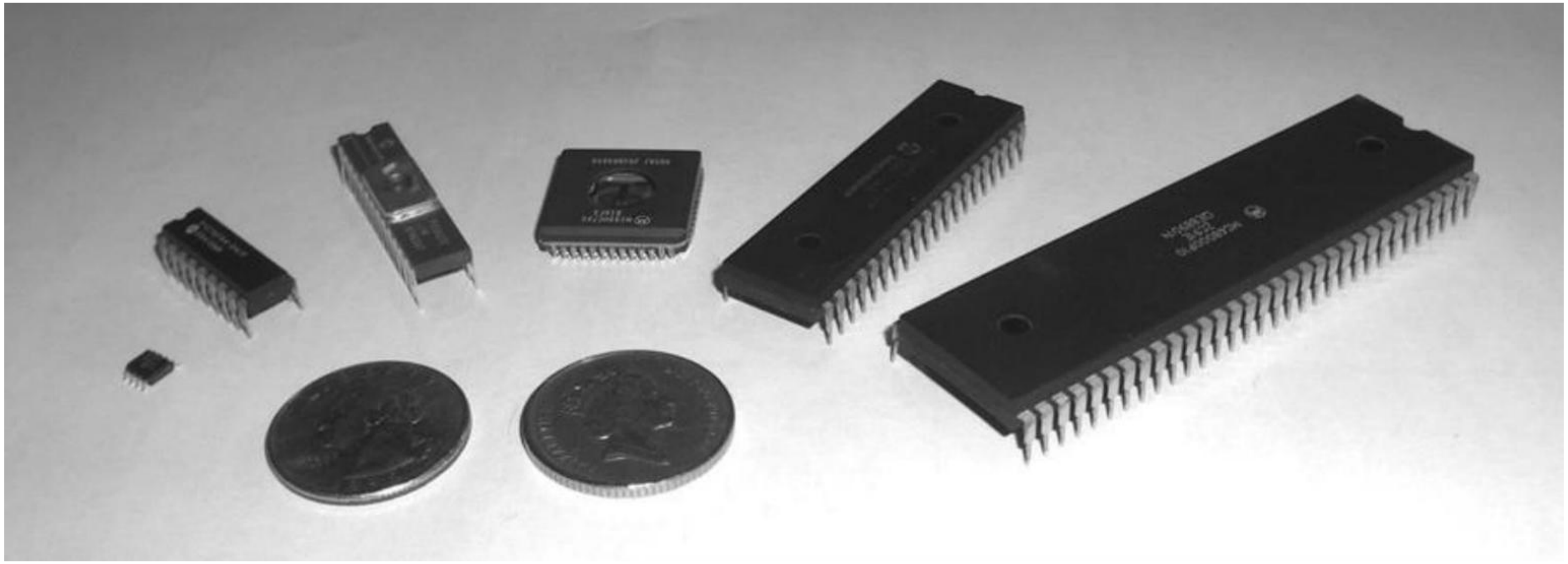
Outline

- **Definition of Microcontroller– IoT, embedded System, M.P**
- **Types of Microcontrollers**
- **CISC and RISC**
- **Microcontroller Architecture**
- **Core Microcontroller Components**
- **Peripheral Microcontroller Components**
- **Internal and External Operation of Microcontrollers**
- **Advantages of Microcontrollers**
- **Application Areas of Microcontrollers**
- **Microcontroller vs. Microprocessor**
- **Choosing a Microcontroller for a Specific Application**

Definition of Microcontroller

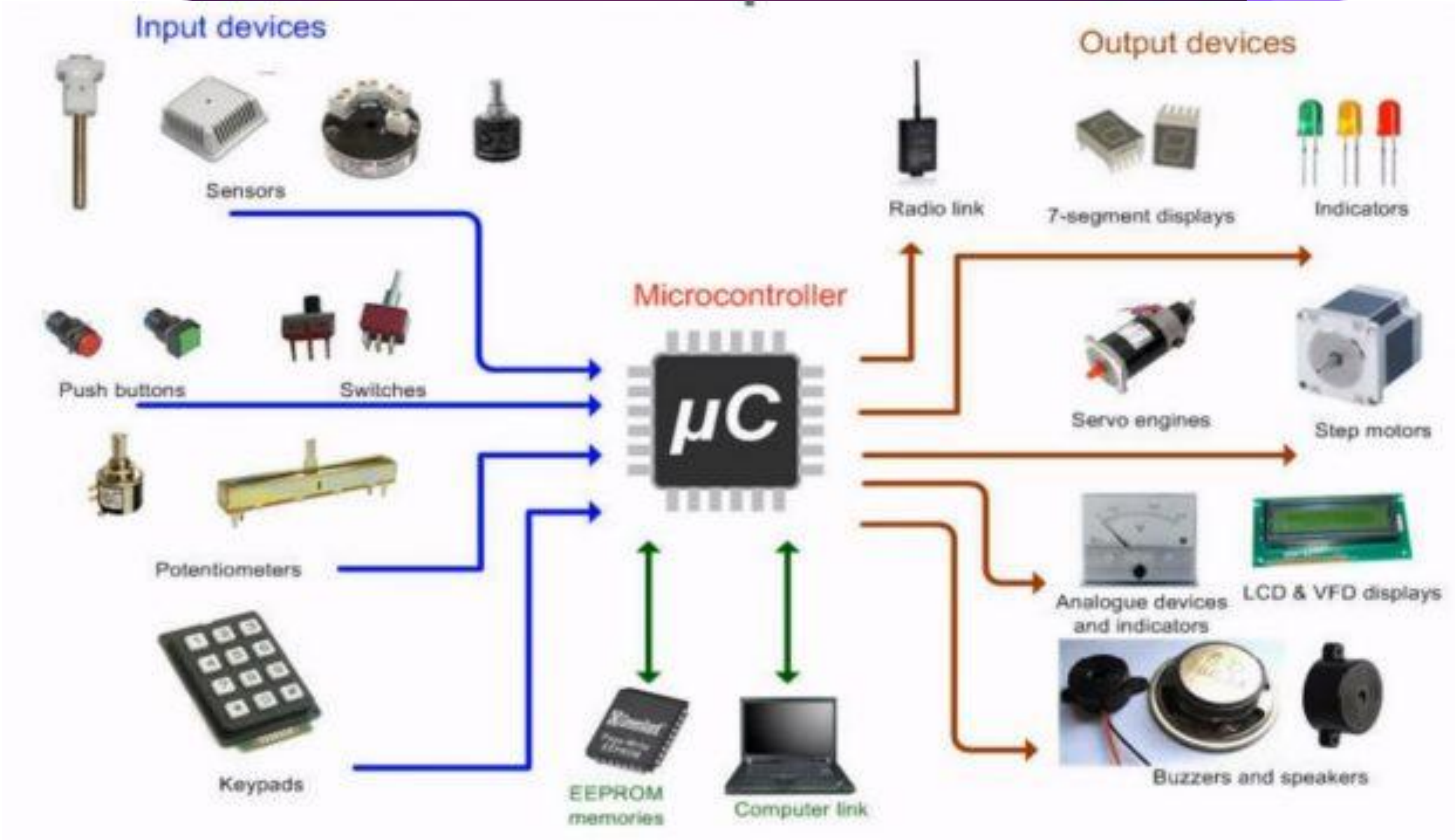
- A microcontroller is a compact integrated circuit (IC) designed to perform specific tasks within an embedded system. It typically includes a central processing unit (CPU), memory (RAM and Flash), and input/output (I/O) peripherals all embedded on a single chip.
- Being programmable is what makes the microcontroller unique.

Definition of Microcontroller (Continued)

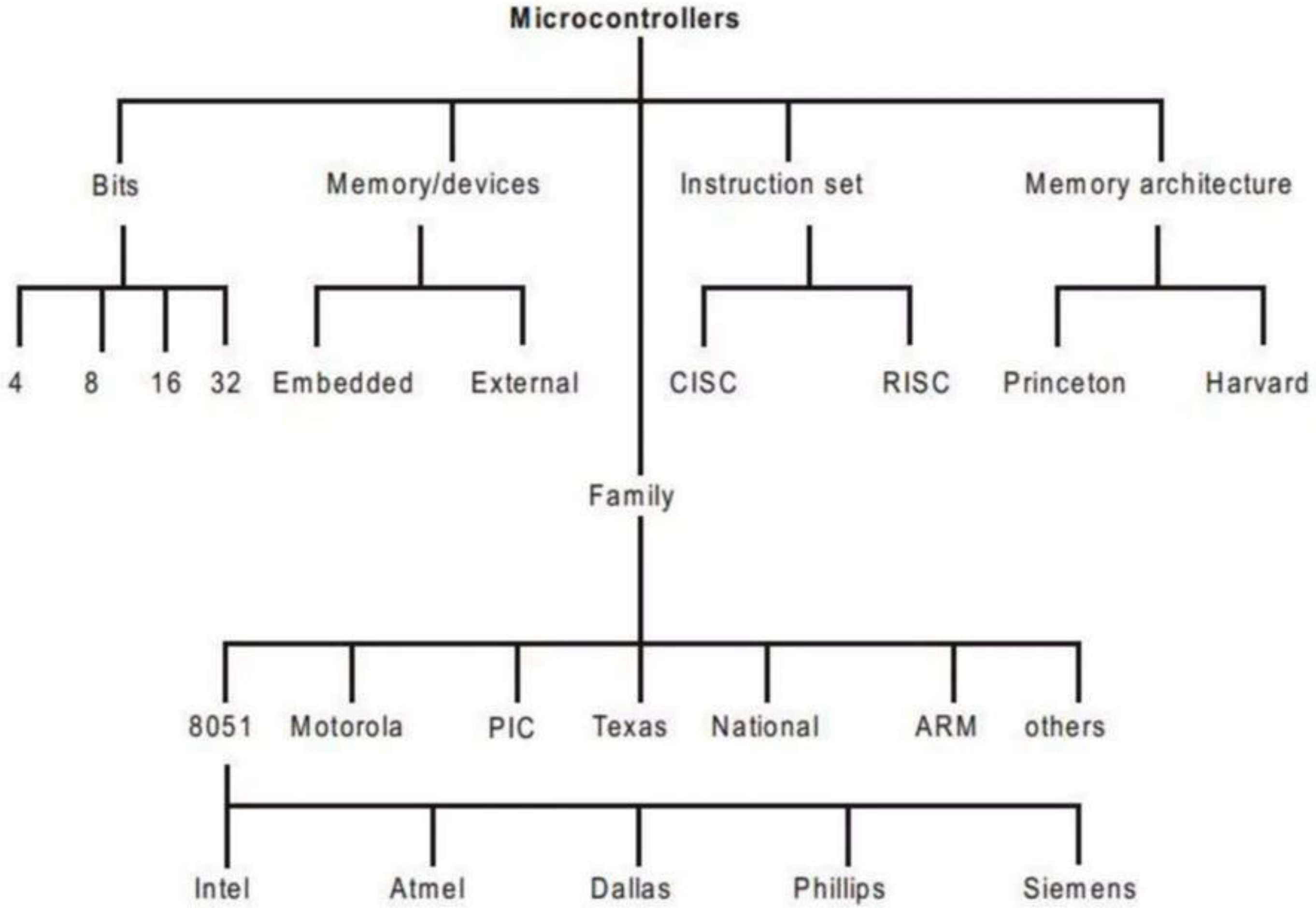


From left to right: PIC 12F508, PIC 16F84A, PIC 16C72, Motorola 68HCO05B16, PIC 16F877, Motorola 68000.

Definition of Microcontroller



Types of Microcontrollers



Types of Microcontroller according to Bit Size

Type	Data Bus Width	Features	Typical Applications	Popular Examples
4-bit MCU	4 bits/ 1 nibble	- Very simple and basic - Extremely low power and cost - Very limited memory and processing power	Simple toys, calculators, LED control, simple embedded controls	Intel 4004 (first microprocessor), PIC 4-bit MCUs
8-bit MCU	8 bits/ 1 byte	- Simple architecture - Low cost & power - Limited memory	Basic control systems, small appliances	Intel 8051, Atmel AVR (Arduino UNO), PIC16F877A
16-bit MCU	16 bits/ 1 word	- Better performance than 8-bit - Moderate memory - Moderate speed	Automotive, mid-level industrial automation	MSP430, PIC24, HCS12
32-bit MCU	32 bits/ 2 words	- High speed & processing power - Larger memory - Complex operations	IoT, robotics, industrial control, smart devices	ARM Cortex-M series (STM32, SAM D21), ESP32

Embedded Microcontrollers

- An embedded system has a microcontroller unit that has all the functional blocks (including program as well as data memory) available on a the same chip.
- Example: 8051 having Program & Data Memory, I/O Ports, Serial Communication, Counters and Timers and Interrupt Control logic on the chip.

External Memory Microcontrollers

- An external system has a microcontroller unit that does not have all the functional blocks available on a chip.
- All or part of the memory units are externally interfaced using an interfacing circuit called the glue circuit.
- Example: Intel 8031 has no program memory on the chip.

CISC and RISC

CISC (Complex Instruction Set Computer) and RISC (Reduced Instruction Set Computer) are two different approaches to designing processor architectures.
CISC focuses on a large, diverse instruction set, aiming to perform complex operations with single instructions, while RISC uses a smaller, simpler instruction set and relies on the compiler to break down complex operations into sequences of simpler ones.

CISC Microcontrollers

- Has an instruction set that supports many addressing modes for the arithmetic and logical instructions, data transfer and memory accesses instructions.
- Many of the instructions are macro like.
- Allows the programmer to use one instruction in place of many simpler instructions.
- Example: Intel 8096 family.

RISC Microcontrollers

- Contains an instruction set that supports fewer addressing modes for the arithmetic and logical instructions and for data transfer instructions.
- Allows simultaneous access of program and data.
- Instruction pipelining increases execution speed.
- Allow each instruction to operate on any register or use any addressing mode.
- Smaller chip and pin count.
- Very low power consumption.

Example : ARM Cortex-M series (e.g., Cortex-M3, M4, M7) — used in smartphones, IoT devices, and embedded systems.

RISC vs CISC Microcontrollers

Feature	RISC Microcontroller	CISC Microcontroller
Instruction Set	Smaller, simpler set of instructions	Larger, more complex set of instructions
Full Form	Reduced Instruction Set Computer	Complex Instruction Set Computer
Execution Speed	Faster, often one cycle per instruction	Can take multiple cycles per instruction
Instruction Format	Fixed format	Variable format
Pipelining	Easier	More challenging
Compiler Complexity	More complex	Less complex
Power Consumption	Generally lower	Generally higher
Code Size	Can be longer	Can be shorter
Development Cost	Generally lower	Potentially higher

RISC vs CISC Microcontrollers

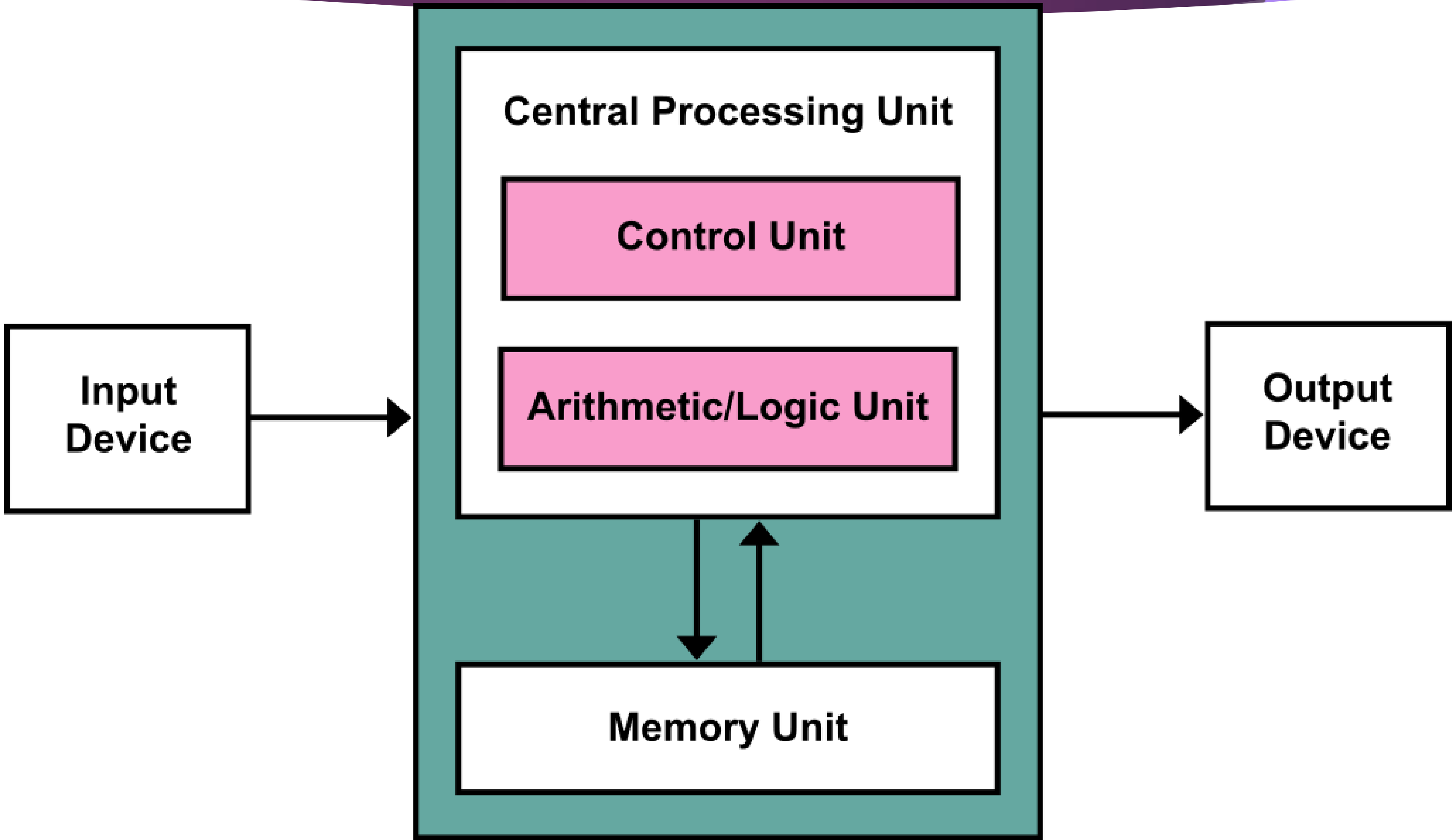
```
LOAD R1, 1000     ; Load value from address 1000 to R1
LOAD R2, 1004     ; Load value from address 1004 to R2
ADD R3, R1, R2    ; Add R1 and R2, store in R3
STORE R3, 1008    ; Store result in address 1008
```

```
ADD [1000], [1004], [1008]
```

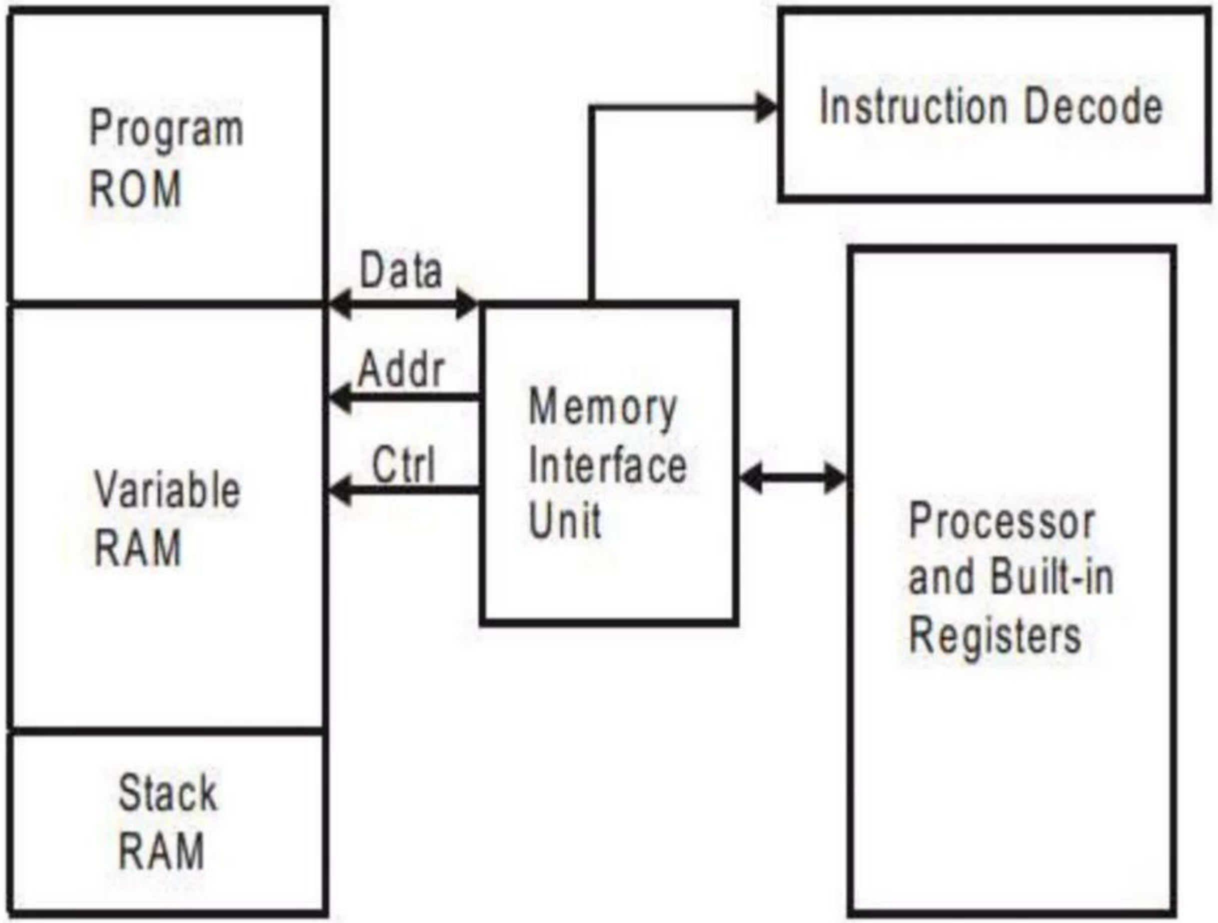
Von-Neuman Architecture

- Single data bus that is used to fetch both instructions and data.
- Program instructions and data are stored in a common main memory.
- When such a controller addresses main memory, it first fetches an instruction, and then it fetches the data to support the instruction.
- Simplifies the microcontroller design because only one memory is accessed.
- The weakness is that two separate fetches can slow up the controller's operation.
- Example: Motorola 68HC11.

Von-Neuman Architecture Block Diagram

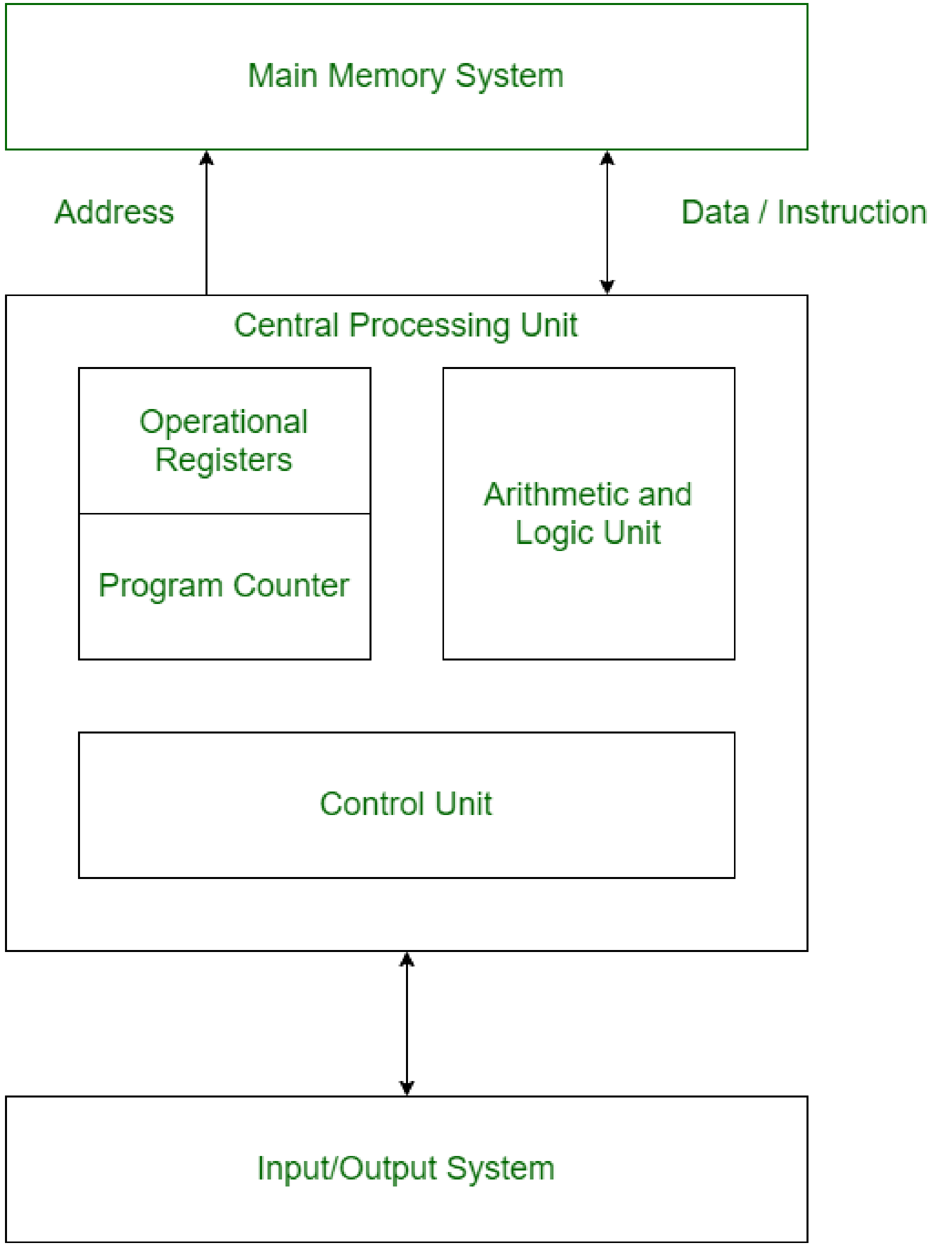


Von-Neuman Architecture Block Diagram

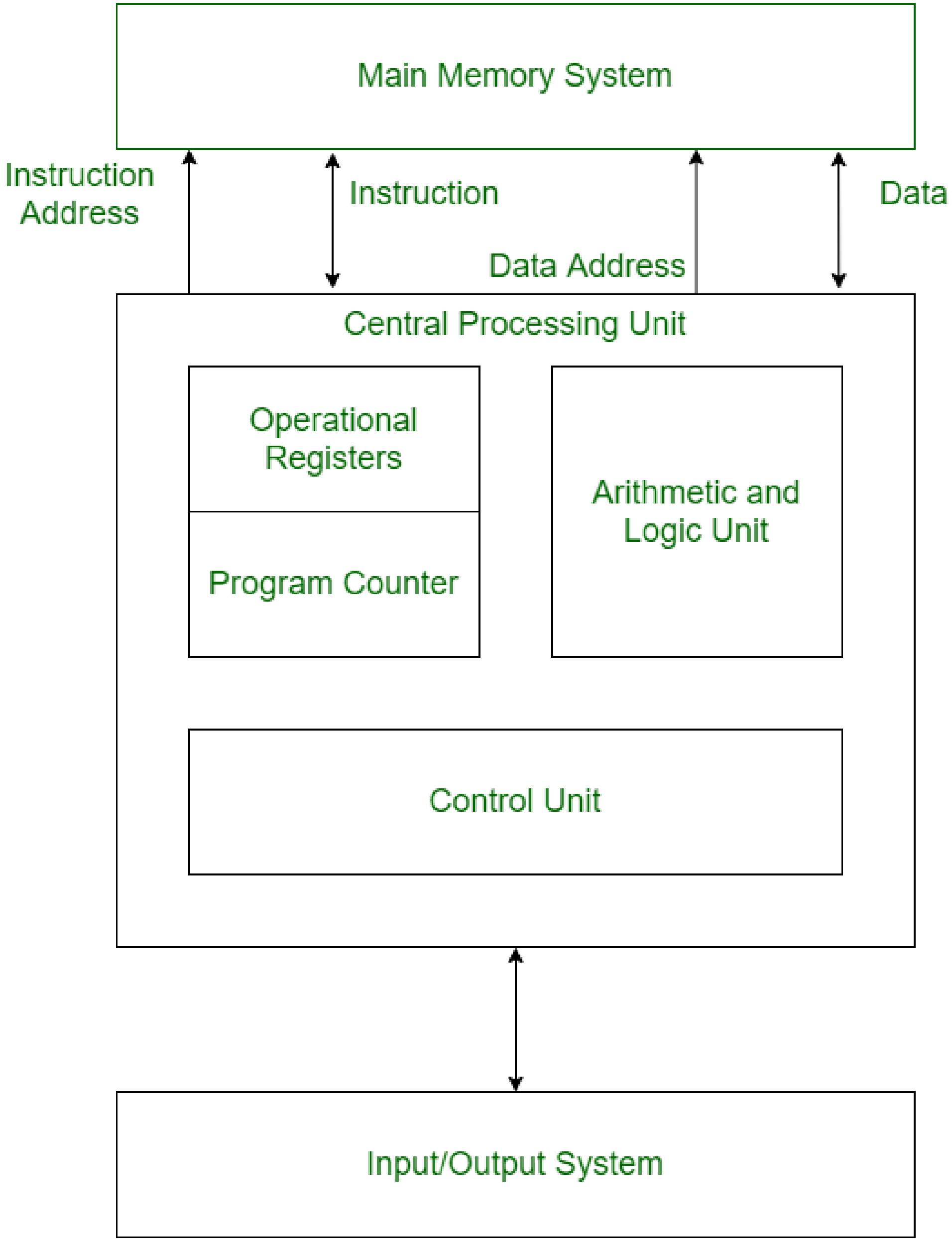


Harvard Architecture

- Separate data bus and an instruction bus.
- Execution occur in parallel.
- Much faster execution than Von-Neuman architecture.
- Design complexity.
- Example: Intel MCS-51 family and PIC microcontrollers.



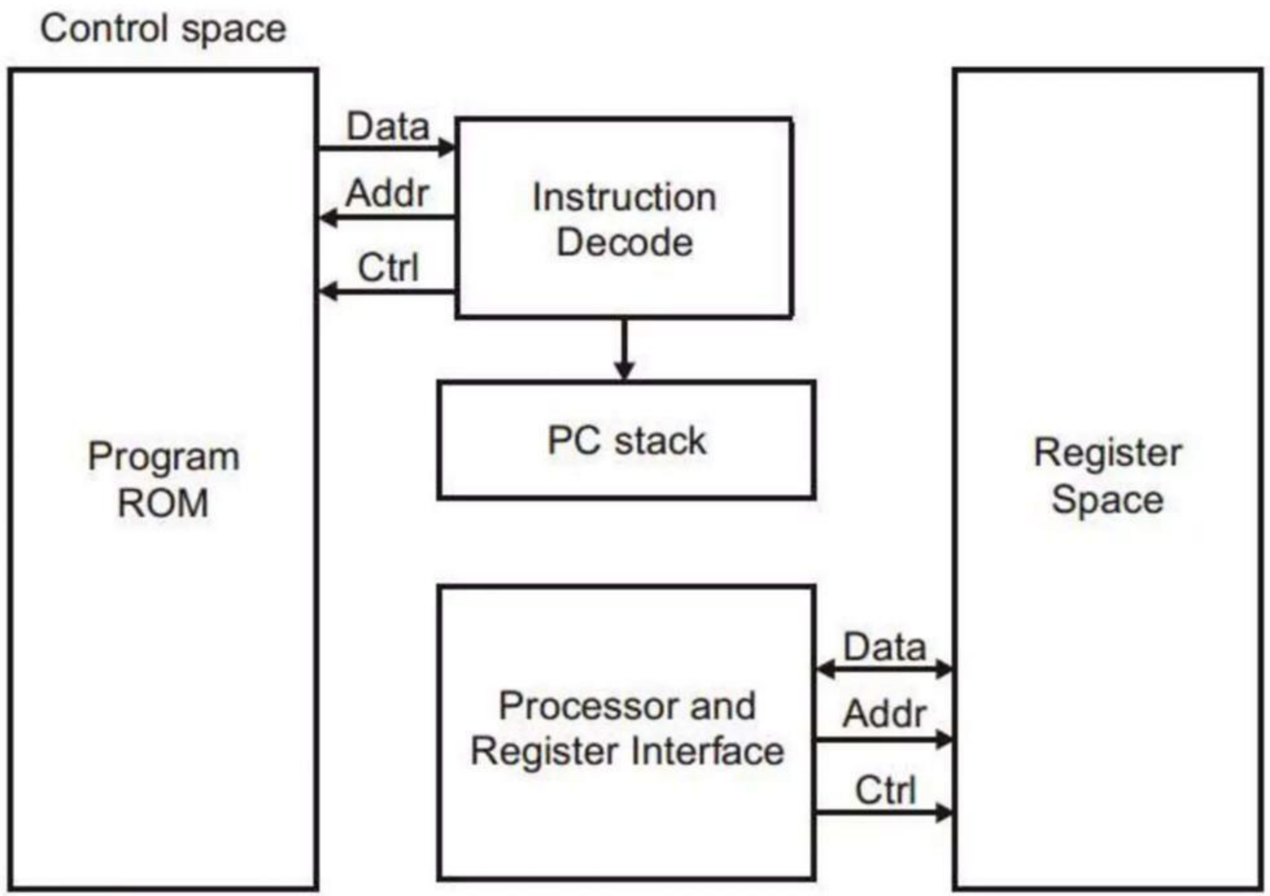
Von Neumann Architecture



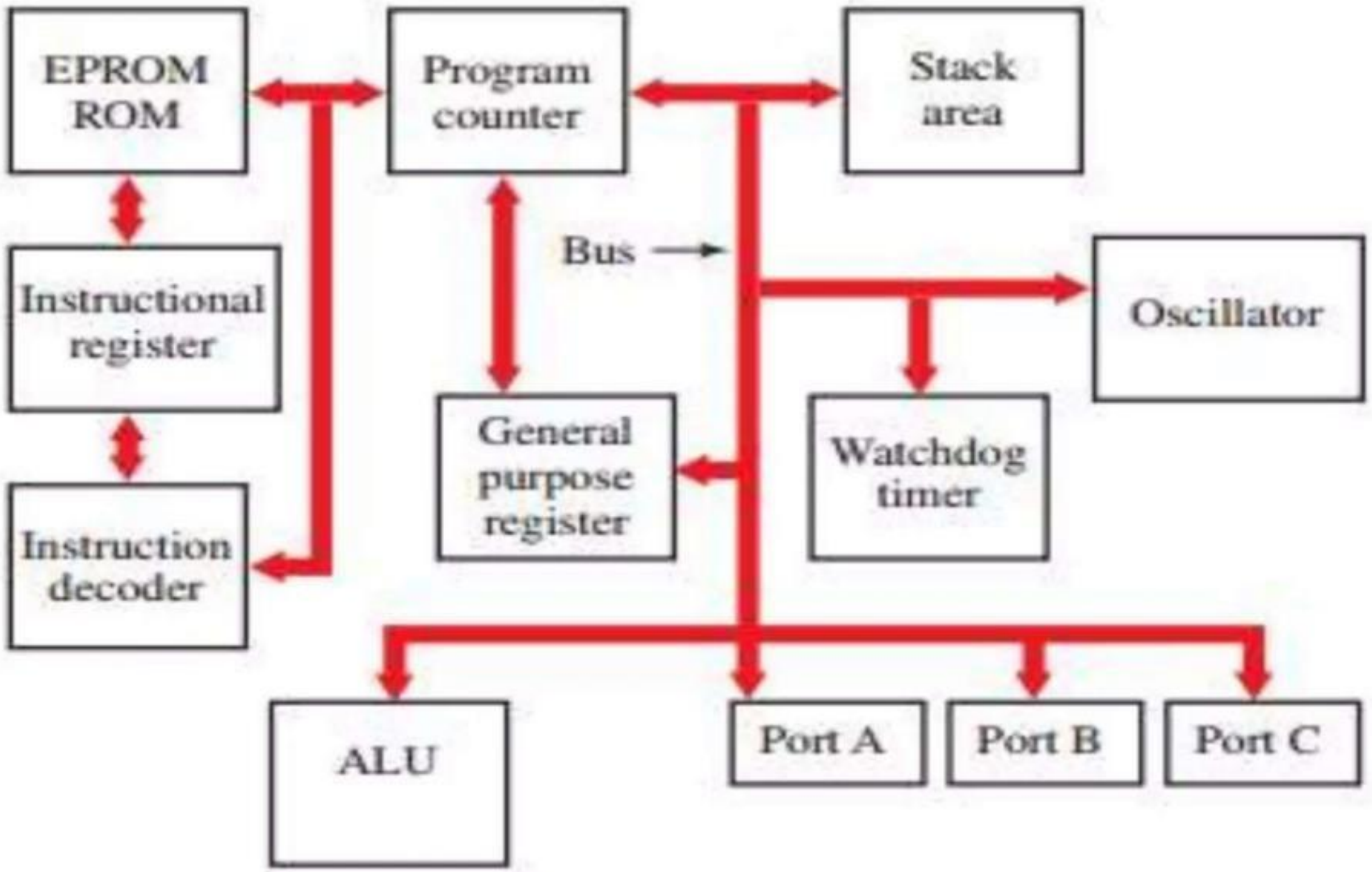
Harvard Architecture

Aspect	Von Neumann Architecture	Harvard Architecture
Memory Structure	Uses a **single memory** for both program (code) and data.	Uses **separate memories** for program (code) and data.
Bus System	Single bus for instruction and data fetch.	Separate buses for instruction and data fetch.
Speed	Slower due to shared bus – can't access code and data at the same time.	Faster – allows simultaneous access to code and data.
Instruction Fetch & Data Access	Occurs **sequentially**, not in parallel.	Occurs **in parallel**, improving throughput.
Complexity	Simpler and cost-effective architecture.	Slightly more complex and costly.
Common Usage	Older or low-cost microcontrollers (e.g., some 8051 variants).	Most modern microcontrollers (e.g., AVR, PIC, ARM Cortex-M).
Example MCUs	Intel 8051, early PIC MCUs.	Atmel AVR (Arduino), PIC16F877A, ARM Cortex-M series. Intel , ryzen

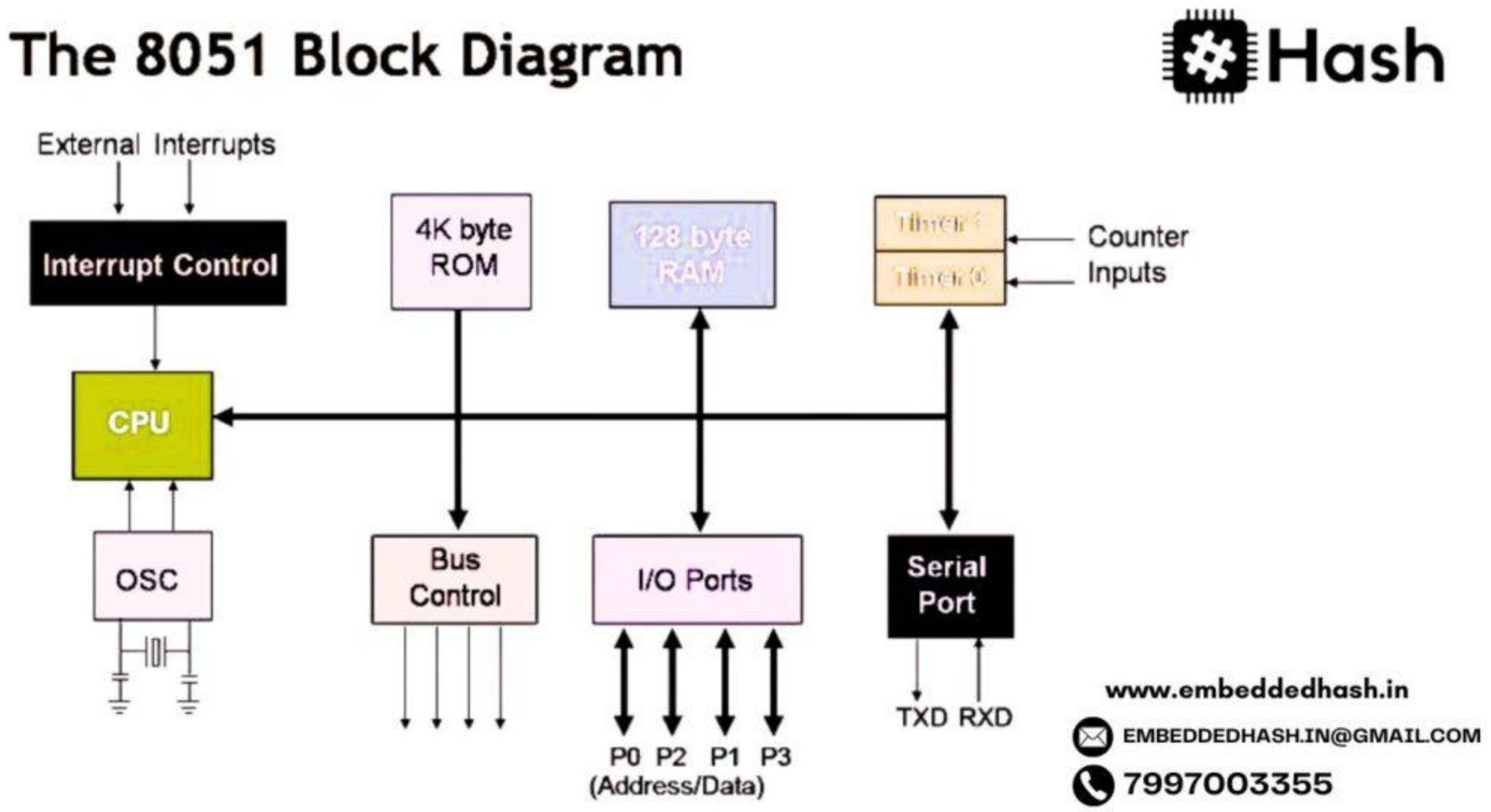
Harvard Architecture Block Diagram



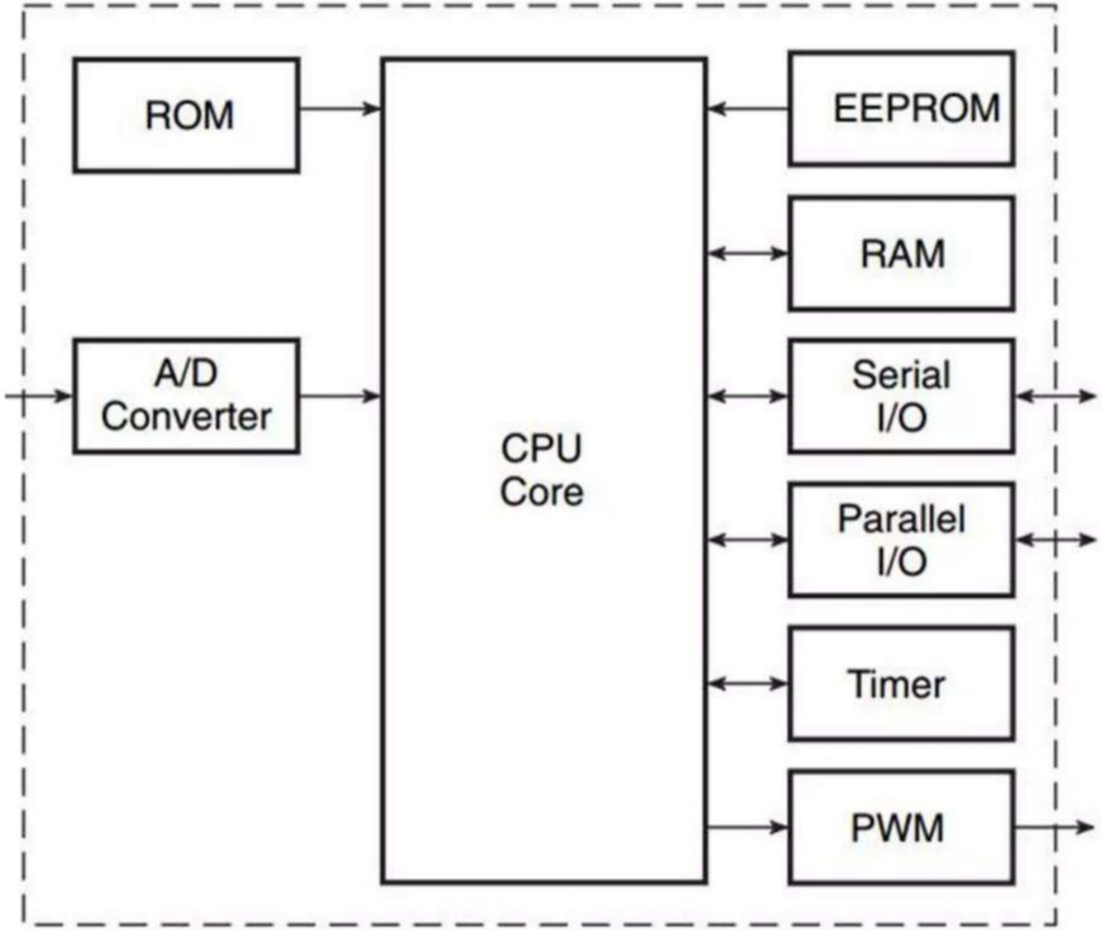
Microcontroller CPU Architecture Block Diagram



Microcontroller CPU Architecture Block Diagram



Generalized Microcontroller Architecture Block Diagram



Core components of Microcontroller

Component	Function/Work
CPU (Central Processing Unit)	Executes instructions, performs arithmetic and logic operations, controls other components.
Program Memory (Flash ROM)	Stores the program code permanently (non-volatile memory).
Data Memory (RAM)	Temporarily stores data and variables during program execution (volatile memory).
Input/Output Ports (GPIO)	Interface pins used to connect and communicate with external devices like sensors, LEDs, switches.
Timers/Counters	Generate time delays, measure time intervals, or count external events.
Watchdog Timer	Monitors program execution; resets the MCU if the program hangs or crashes to prevent system failure.
Oscillator/Clock Generator	Provides clock signals to synchronize all MCU operations; controls execution speed.
Interrupt Controller	Manages interrupt requests from peripherals, allowing immediate attention to critical tasks.
ADC (Analog-to-Digital Converter)	Converts analog signals (e.g., sensor outputs) into digital values the MCU can process.
DAC (Digital-to-Analog Converter)	Converts digital data into analog signals for controlling analog devices (optional).
Serial Communication Interfaces	Protocols like UART, SPI, I2C for communication with other devices or MCUs.
Power Supply & Reset Circuit	Provides stable power and resets the MCU during startup or fault conditions.
Stack Pointer	Keeps track of the last stack location used while processor is busy manipulating checking ports, or checking interrupts.

Peripheral Microcontroller Components

<u>The analog-to-digital converter:</u>

- Provides an interface between the microcontroller and the sensors that produce analogue electrical equivalents of the actual physical parameters to be controlled.

<u>The digital-to-analog converter:</u>

- Provides an interface between the microcontroller and the actuators that provide the control function.

Peripheral Microcontroller Components

<u>I/O ports:</u>

- Provide an interface between the microcontroller and the peripheral I/O devices such as the keyboard, display, etc.

<u>Counters/timers:</u>

- Are used to keep time and/or measure the time interval between events, count the number of events and generate baud rate for the serial ports.

Microcontroller Internal Operation

- The microcontroller consists of thousands of digital circuits that are combined into areas to provide specific functions.
- The CPU components of a microcontroller are used to save data and programs, perform math and logic functions, and generate timing signals.
- The different areas are connected by a bus system. The bus system contains tiny

Parallel circuits that carry the digital pulse patterns from section to section.

Microcontroller Internal Operation

- The ROM stores the program required for the microcontroller to function and controls how the chip components operate and how data and instructions flow through the chip.
- RAM stores programs and data temporarily.
- Ports and registers are special memory locations dedicated to a specific function

  such as a hardware location or a place to manipulate data.

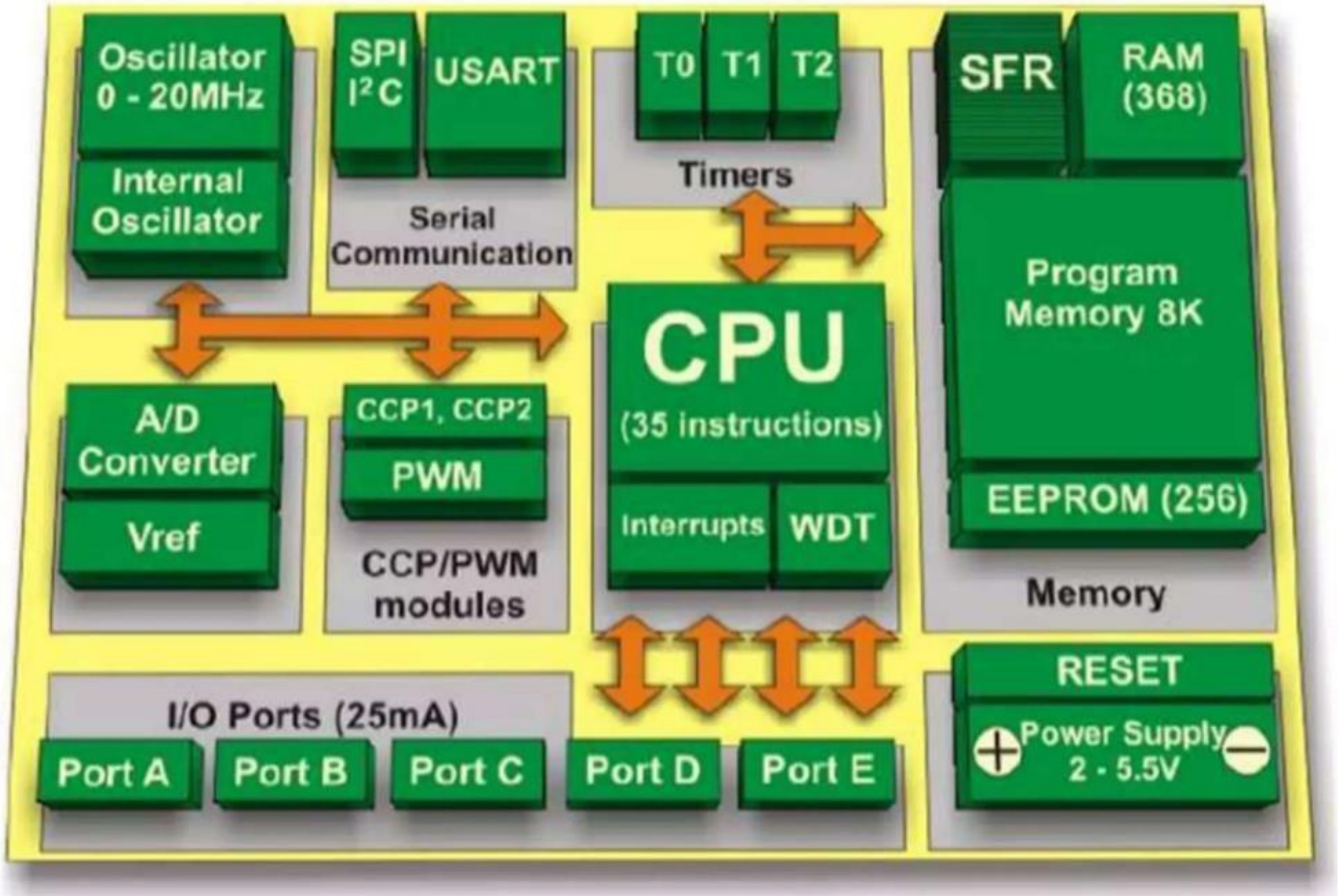
Microcontroller External Operation

- When a microcontroller is mounted on a circuit board with other components, it functions as a single unit, and is referred to as a module or a microcontroller development board.
- A microcontroller module typically consists of a microcontroller, a power source, an interface for connecting to a programming device, I/O ports, and additional memory.
- A power source powers the microcontroller and any accompanying components located on the printed circuit board.

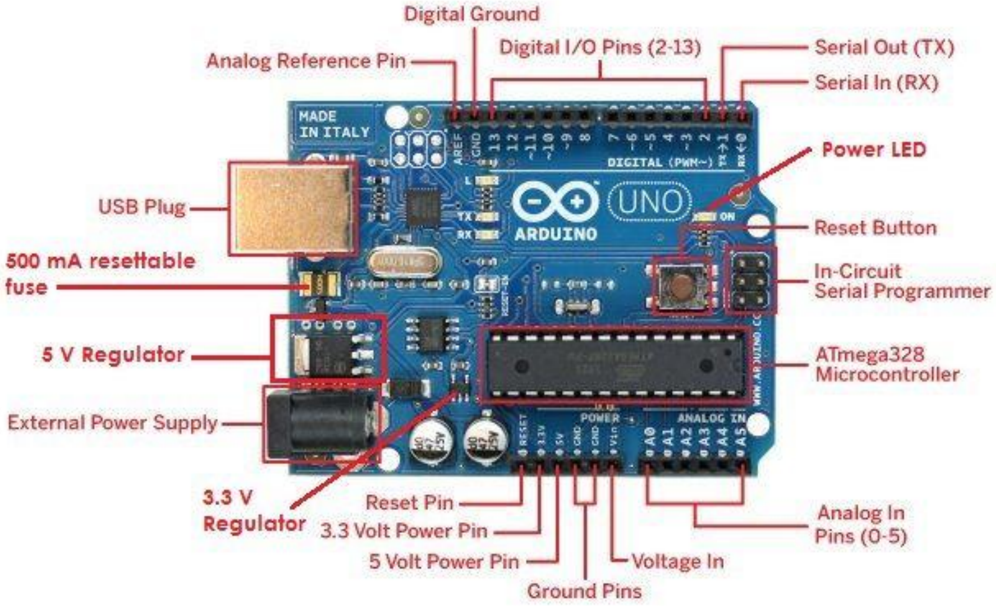
Microcontroller External Operation

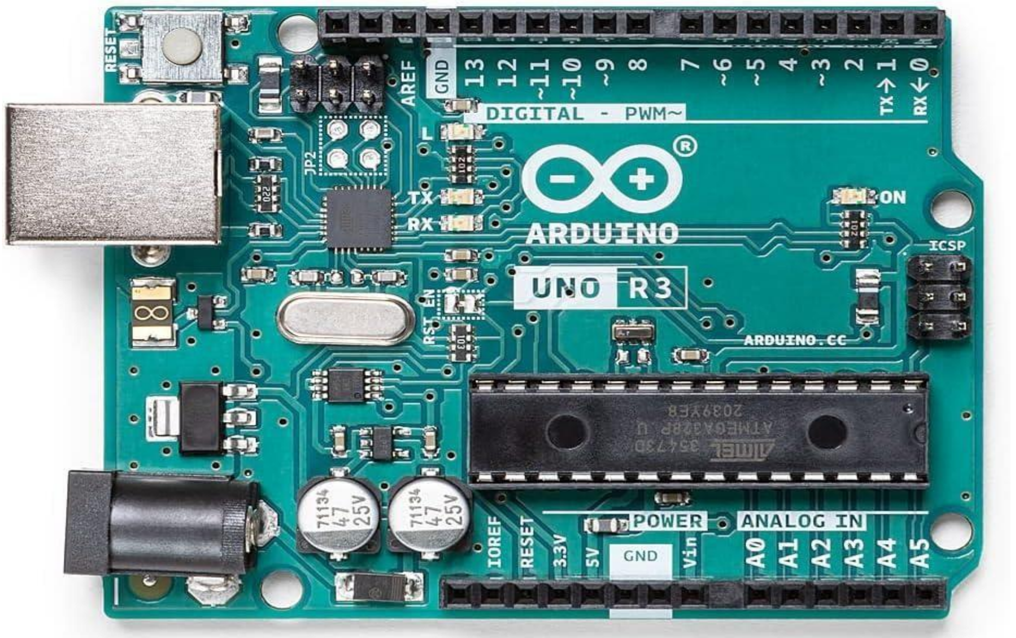
- An interface communicates with the microcontroller.
- A set of input/output (I/O) ports send and receive signals from the devices the microcontroller is designed to control. I/O ports when programmed as an output pin, each pin can output digital signals. When programmed as an input pin, each pin can receive digital signals.
- Digital-to-analog and analog-to-digital converters change the digital pulses into analog signals.

Microcontroller External Operation Block Diagram



Arduino Uno – An Atmel ATMega328P Development Board





Advantages of Microcontroller

Advantage	Description
1. Compact Size	Combines CPU, memory, and peripherals in a single chip, reducing overall size of the system.
2. Low Cost	Cost-effective solution for control-based applications, especially in mass production.
3. Low Power Consumption	Designed for energy-efficient operation, ideal for battery-powered and IoT devices.
4. Easy to Interface	Comes with built-in I/O ports, timers, ADC, UART, SPI, I2C, etc., enabling easy integration.
5. Real-Time Control	Well-suited for real-time tasks due to fast interrupt handling and deterministic behavior.
6. Reliable and Stable	Performs specific tasks consistently with high reliability, especially in embedded systems.
7. Faster Development	Supported by development kits, IDEs, libraries, and online communities, speeding up prototyping.
8. Portable	Small, lightweight, and can be used in handheld and mobile systems.
9. Versatile Applications	Used in diverse fields: automotive, medical, home automation, robotics, and consumer electronics.
10. On-chip Memory	Internal RAM and Flash eliminate the need for external memory in most simple applications.

Comparison Between Microcontroller and Microprocessor

	Microprocessor	Microcontroller
Applications	General computing (i.e. Laptops, tablets)	Appliances, specialized devices
Speed	Very fast	Relatively slow
External Parts	Many	Few
Cost	High	Low
Energy Use	Medium to high	Very low to low
Vendors	intel AMD ARM	ATMEL ST Texas Instruments Microchip

Comparison Between Microcontroller and Microprocessor

Feature	Microcontroller (MCU)	Microprocessor (MPU)
Definition	A compact integrated circuit with CPU, memory, and peripherals on a single chip.	A general-purpose CPU used for computing, requires external components.
Integration	Includes CPU, RAM, ROM, timers, I/O ports, ADC, etc., all on one chip.	Only contains the CPU; memory and I/O must be connected externally.
Power Consumption	Low (suitable for battery-powered devices)	Higher (requires more power for operation)
Speed	Moderate (typically slower, optimized for control applications)	High (optimized for performance-intensive tasks)
Cost	Low	Higher
Size	Small and compact	Larger, especially with external peripherals
Usage	Embedded systems (e.g., washing machines, IoT devices, car systems)	Personal computers, laptops, servers
Multitasking	Limited	Designed for multitasking and complex OS
Real-Time Applications	Ideal for real-time control systems	Not always suitable for strict real-time requirements
Example Devices	Arduino (AVR), STM32, PIC, 8051, ESP32, ESP8266	Intel Core i7, AMD Ryzen, ARM Cortex-A series, Snapdragon

Application Areas of Microcontrollers

Application Area	Examples
Consumer Electronics	TV remotes, washing machines, microwave ovens, air conditioners, cameras
Automotive Systems	Engine control units (ECU), airbag systems, ABS, cruise control, infotainment
Industrial Automation	Motor control, programmable logic controllers (PLC), robotics, CNC machines
Medical Devices	Heart rate monitors, glucose meters, infusion pumps, diagnostic equipment
Home Automation	Smart lighting, security systems, thermostats, smart locks
IoT (Internet of Things)	Smart sensors, wearable devices, environmental monitoring, smart meters
Telecommunication	Mobile phones, signal boosters, routers, modems
Office Equipment	Printers, scanners, copiers, fax machines
Aerospace & Defense	Drones, missile guidance systems, avionics, navigation units
Agriculture	Automatic irrigation, soil moisture sensors, greenhouse control systems
Toys & Entertainment	Electronic toys, drones, interactive games, music systems
Computing Peripherals	Keyboards, mice, USB devices, external hard drives
Renewable Energy	Solar charge controllers, battery management systems

Criteria of Choosing a Microcontroller-- Context dependent

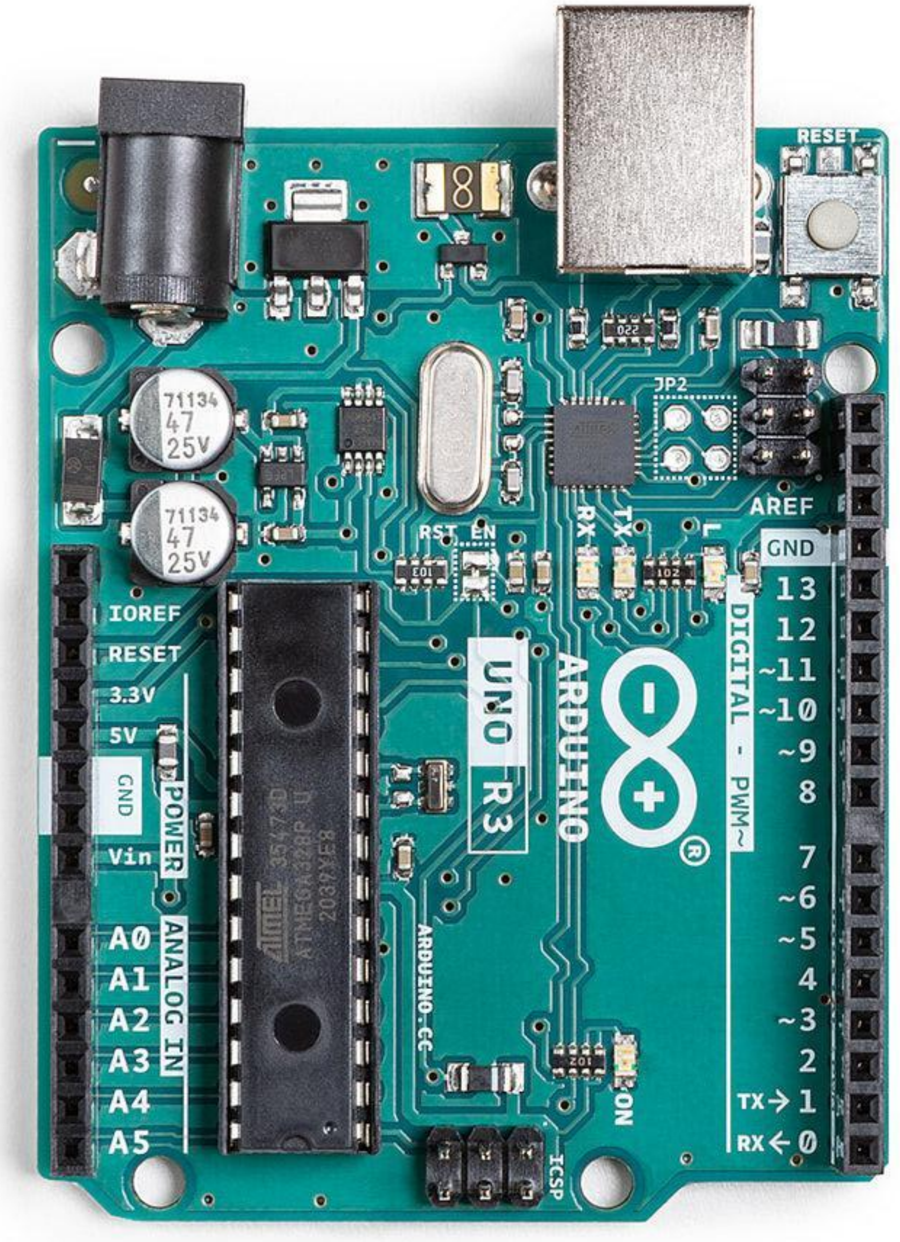
Criteria	Description
1. Application Requirements	Determine whether the MCU is for control, sensing, communication, or processing tasks.
2. Processing Power (CPU)	Consider clock speed (MHz) and architecture (8-bit, 16-bit, 32-bit) depending on computational needs.
3. Memory (RAM & Flash)	Evaluate how much **RAM** (for temporary data) and **Flash** (for program storage) is required.
4. Number of I/O Ports	Ensure enough **GPIO pins** are available for sensors, switches, LEDs, or external devices.
5. Peripheral Support	Look for built-in **ADC, DAC, PWM, Timers, UART, SPI, I2C** etc., depending on interface needs.
6. Power Consumption	Choose low-power MCUs for battery-powered or energy-sensitive applications (e.g., IoT devices).
7. Cost	Balance performance and features with budget constraints, especially in large-scale production.
8. Package Type & Size	Consider physical size and mounting type (DIP, QFP, BGA, etc.) based on PCB design limitations.
9. Availability & Support	Ensure the MCU is widely available, well-documented, and has good community/library support.
10. Development Tools	Check if compatible and user-friendly development tools (IDE, debugger, compiler) are available.
11. Real-Time Requirements	For real-time tasks, choose MCUs with deterministic interrupt response and real-time OS support.
12. Environmental Tolerance	For industrial or outdoor use, select MCUs that can operate in extreme temperatures or conditions.
13. Security Features	Consider built-in encryption, secure boot, or tamper detection for sensitive applications.

“Great design is not just about making things work, it's about making them work efficiently, reliably, and beautifully at the smallest scale.”

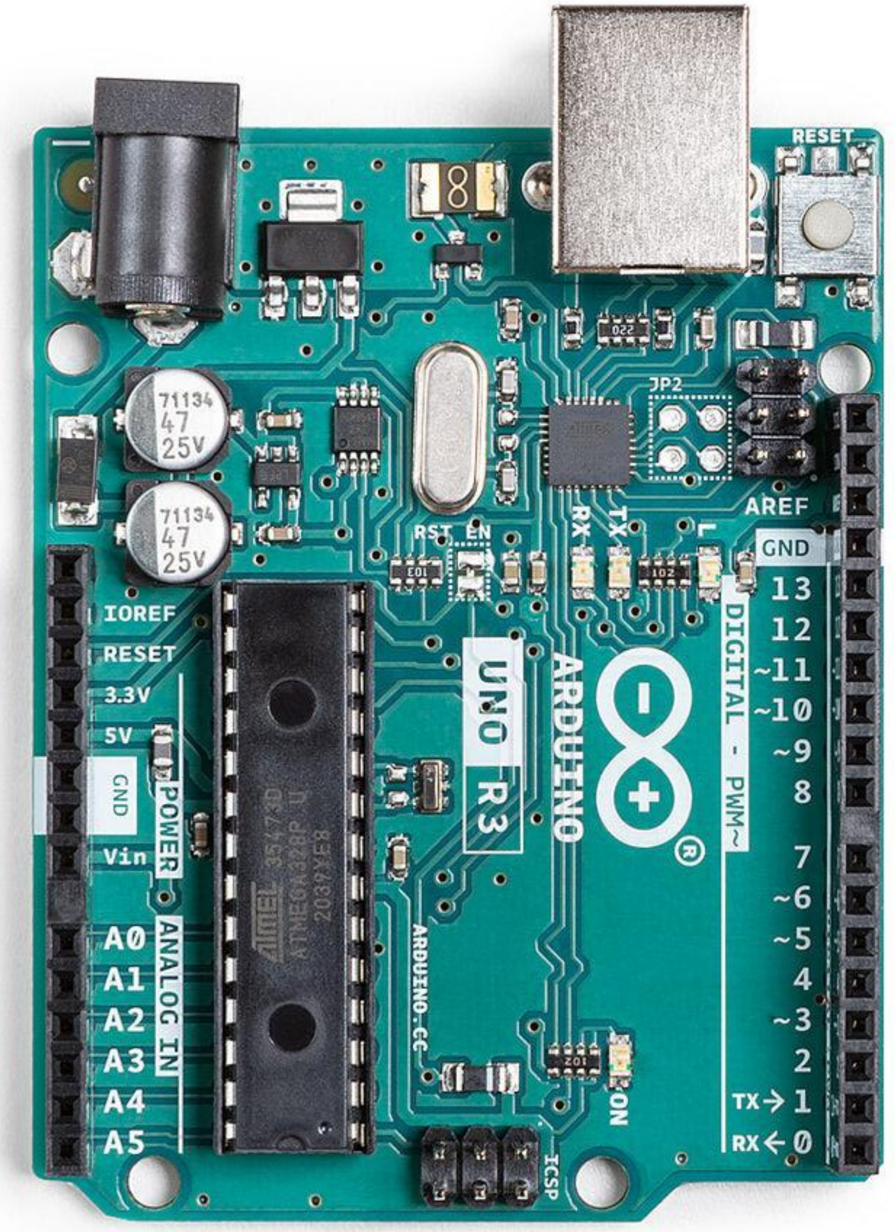
--THANK YOU--

Getting Started With Arduino

Common Functions of Arduino



Function	What it does
pinMode(pin,mode)	Sets a pin as an input or output
digitalWrite(pin, value)	Sets a digital output pin to HIGH or LOW
digitalRead(pin)	Reads a digital input pin as HIGH or LOW
analogWrite(pin, value)	Sets an analog output pin to a value 0-1023
analogRead(pin)	Reads an analog output pin as a value 0-1023
delay(milliseconds)	Pauses the program for a certain amount of time
Serial.begin(value)	Begins the Serial Monitor with a baud rate of *value*
Serial.print(value)	Prints the value (variable) to the Serial Monitor.



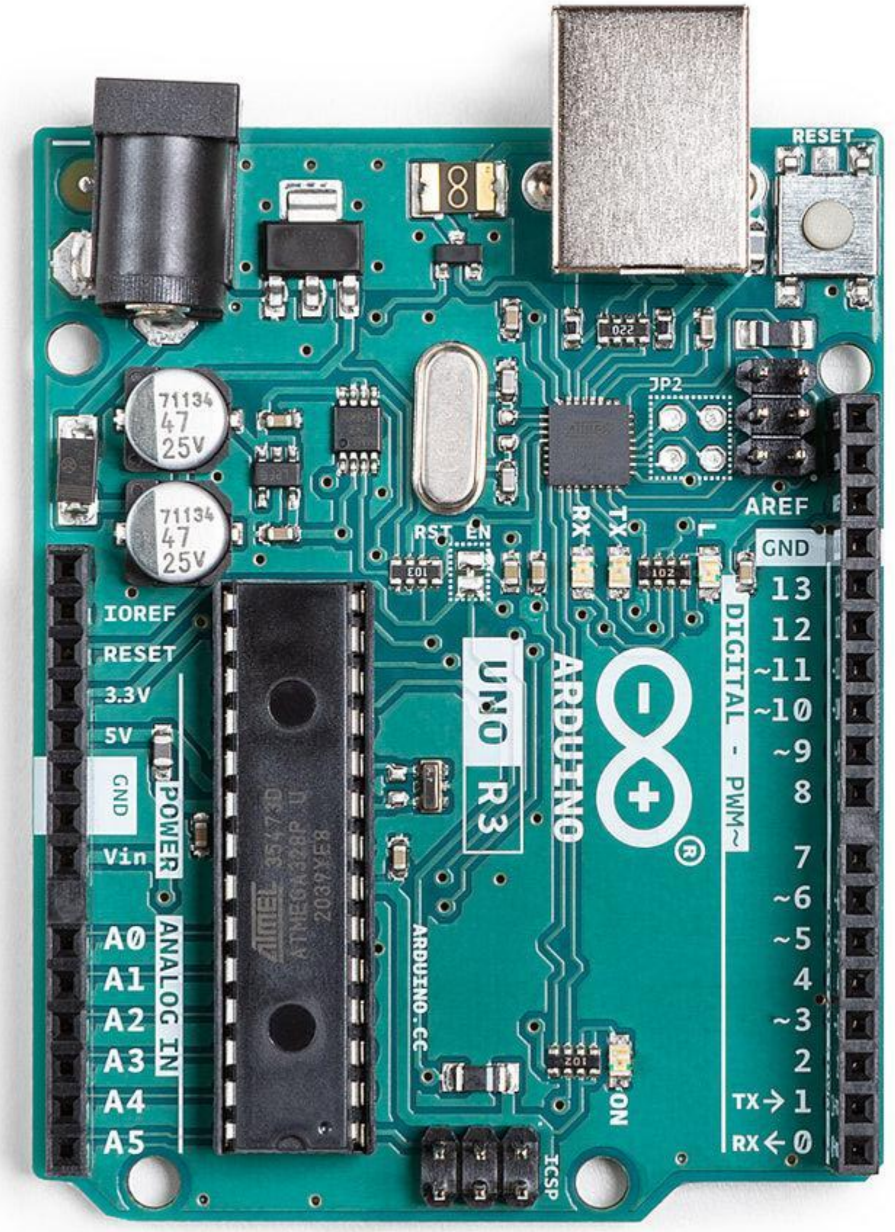
Pin Diagram of Arduino

Digital Pins (0-13)

- **Pins 0 to 13**: These are digital input/output pins that can be used as either inputs or outputs using functions such as pinMode(), digitalWrite(), and digitalRead() etc.

- Each pin operates at 5V and can provide or receive a maximum of 40 mA of current.

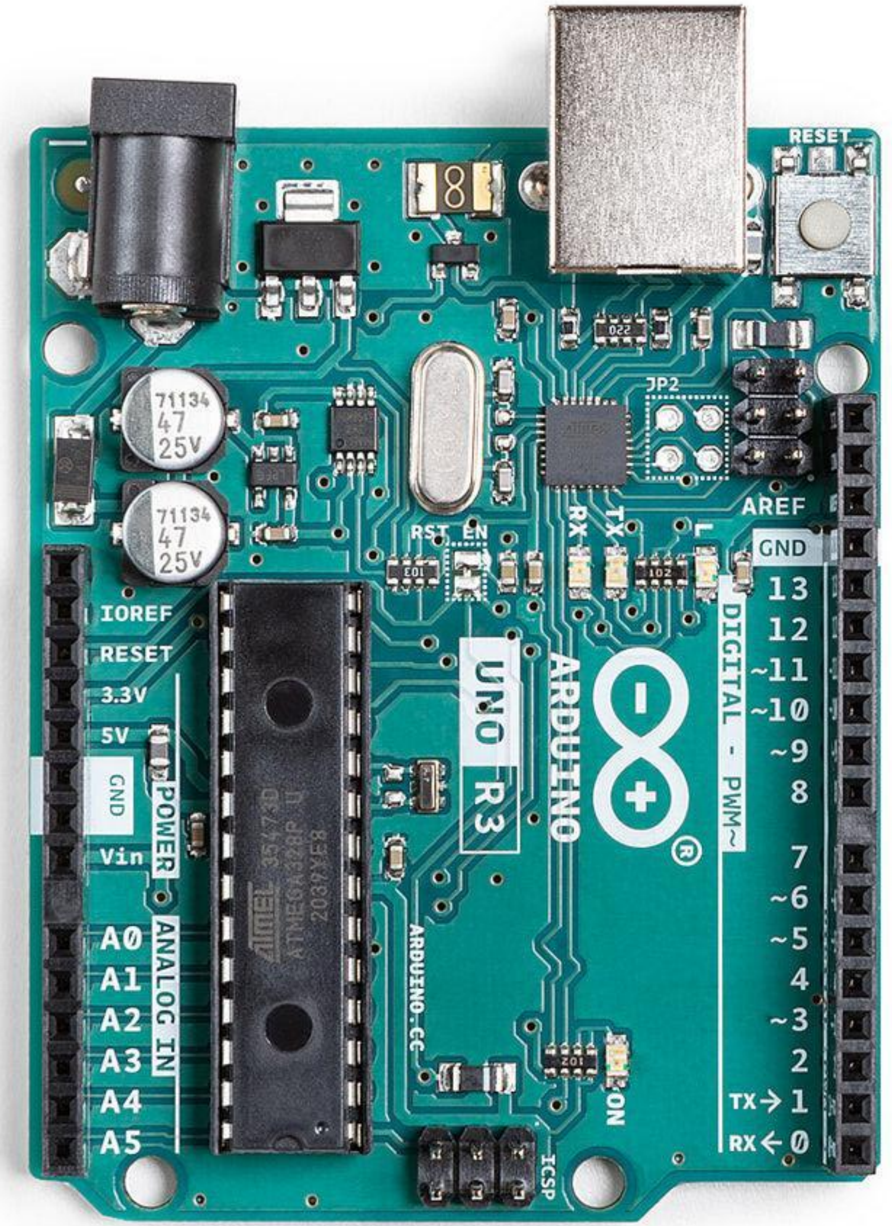
Pin Diagram of Arduino



- **Pin 0 (RX) and Pin 1 (TX)**: Used for serial communication. RX is used for receiving data, and TX is used for transmitting data.

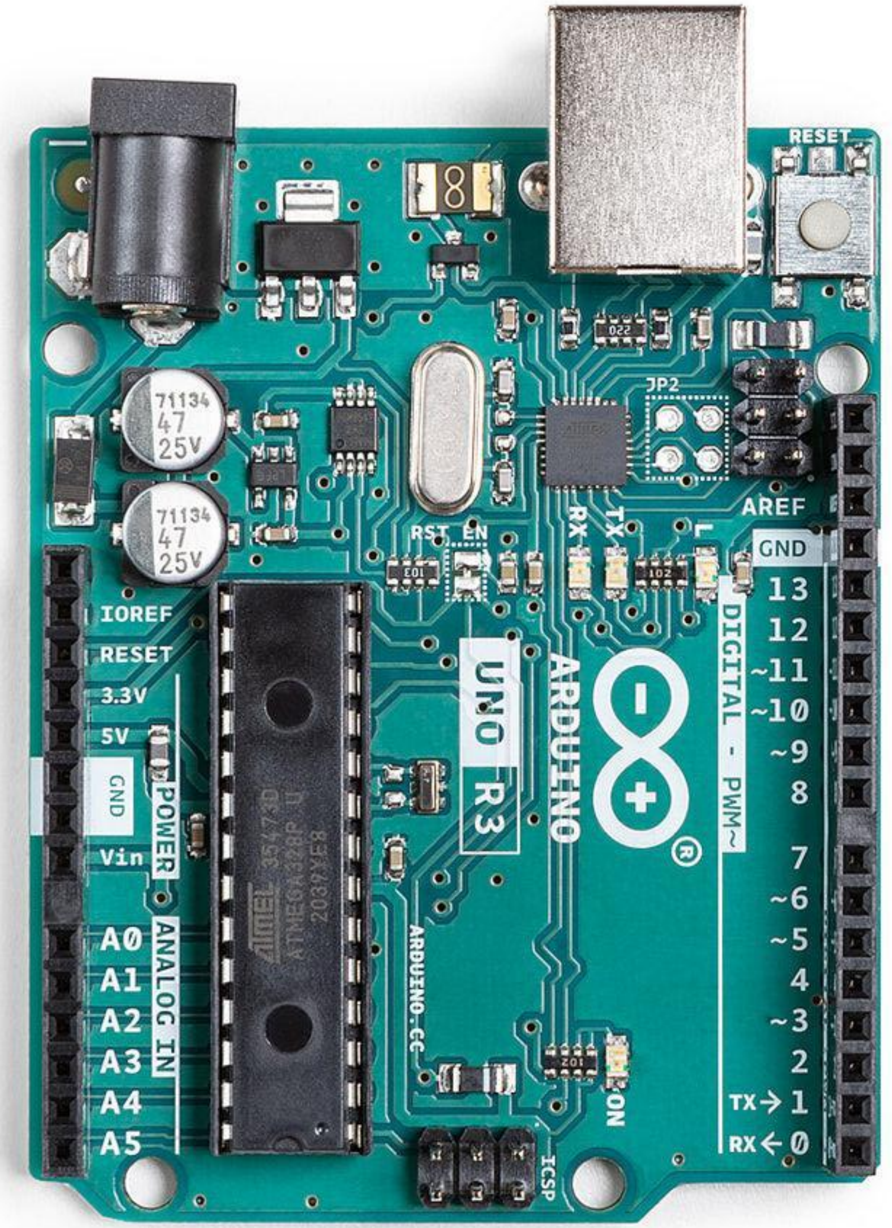
- These are connected to the USB-to-TTL serial chip, so avoid using them if you're using the USB port for programming.

Pin Diagram of Arduino



- **PWM (Pulse Width Modulation) Pins (3, 5, 6, 9, 10, 11)**
- These digital pins can be used as PWM outputs using the analogWrite() function. PWM is used to simulate analog output, commonly used for dimming LEDs or controlling motor speed.

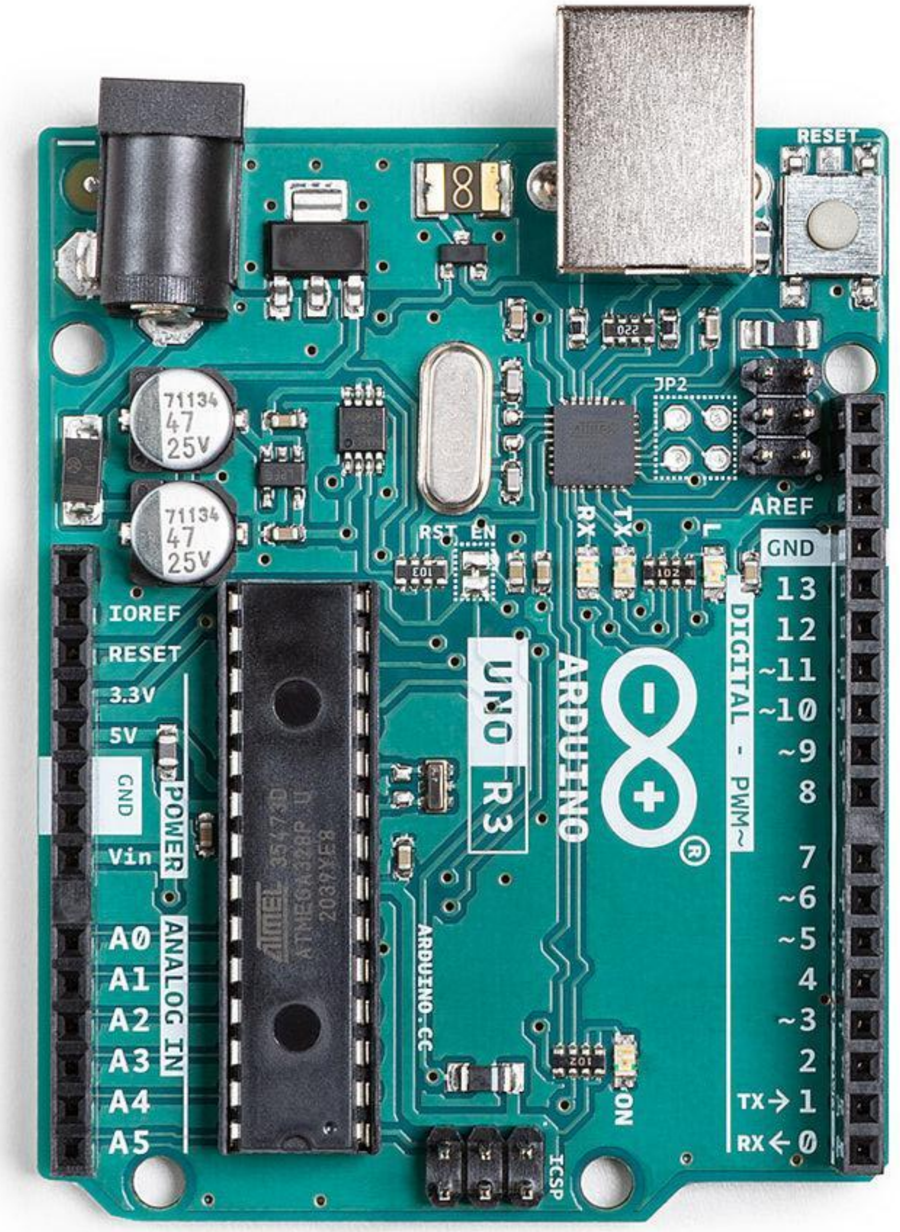
Pin Diagram of Arduino



Analog Pins (A0-A5)

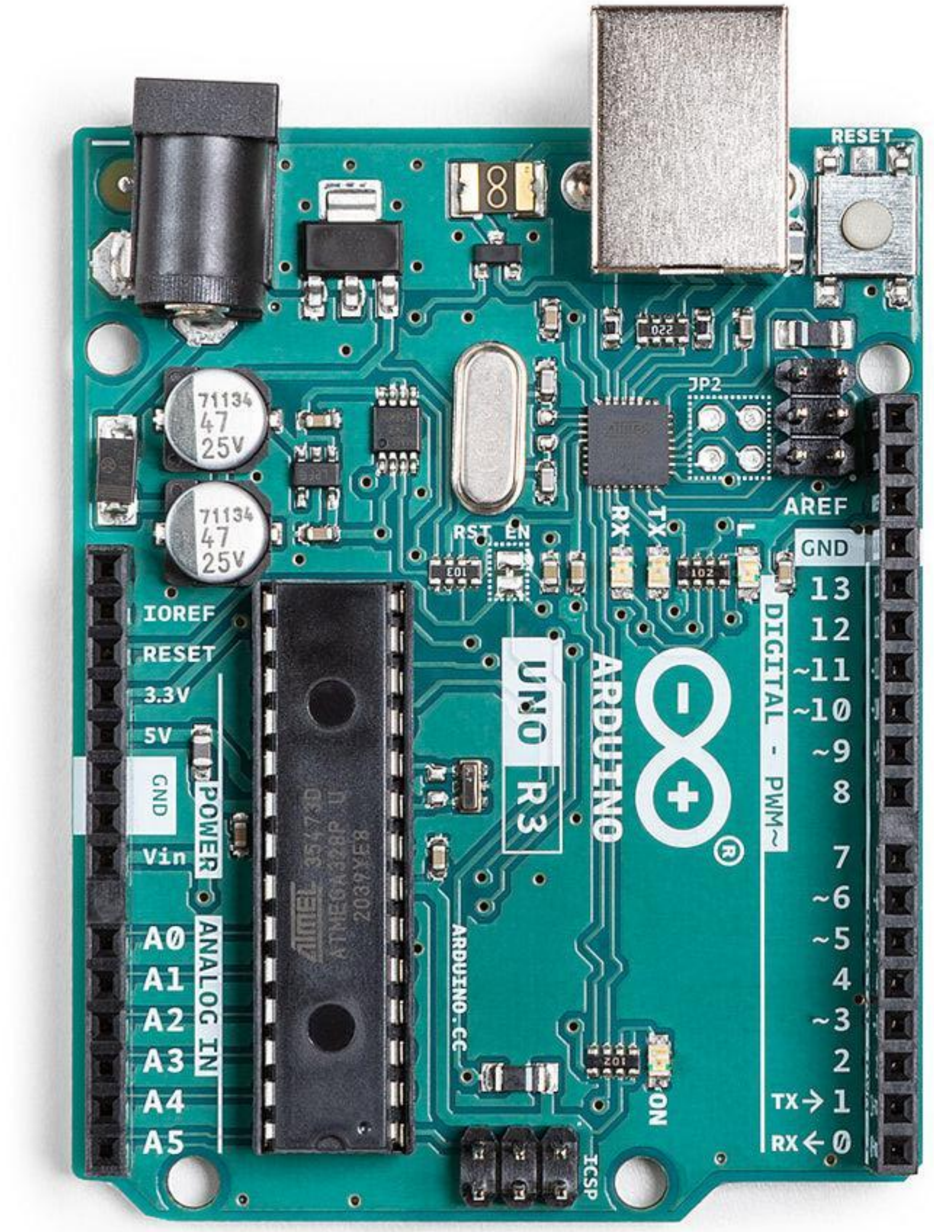
Pins A0 to A5: These are analog input pins, capable of reading a voltage between 0 and 5V using the analogRead() function. They have a 10-bit resolution, returning values from 0 to 1023.

Pin Diagram of Arduino



Power Pins

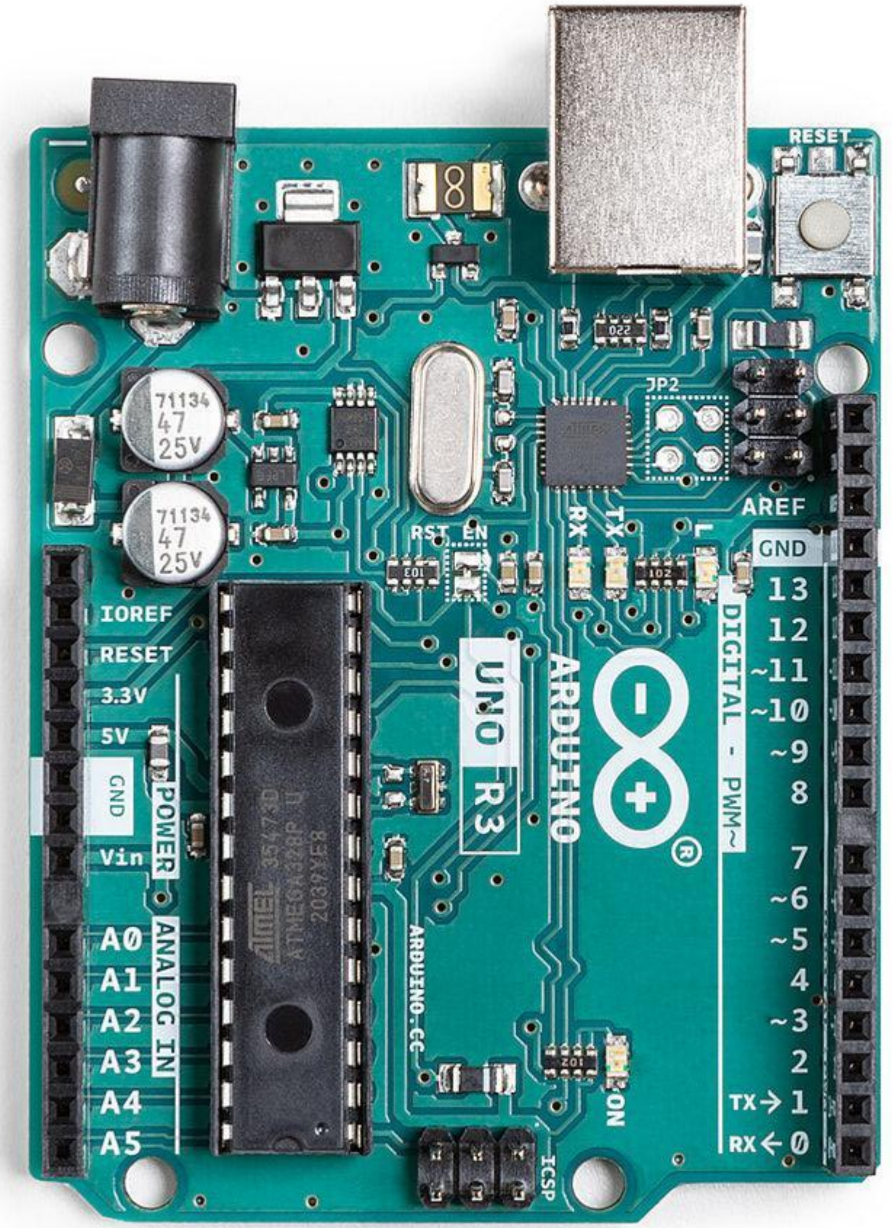
Vin: Input voltage to the Arduino board when using an external power source (6-20V). You can supply voltage through this pin or the power jack.



Pin Diagram of Arduino

5V: The regulated power supply used to power the microcontroller and other components on the board. Can come from USB, Vin (through the regulator), or an external power supply.

3.3V: A 3.3V supply generated by the onboard regulator. Maximum current draw is 50 mA.

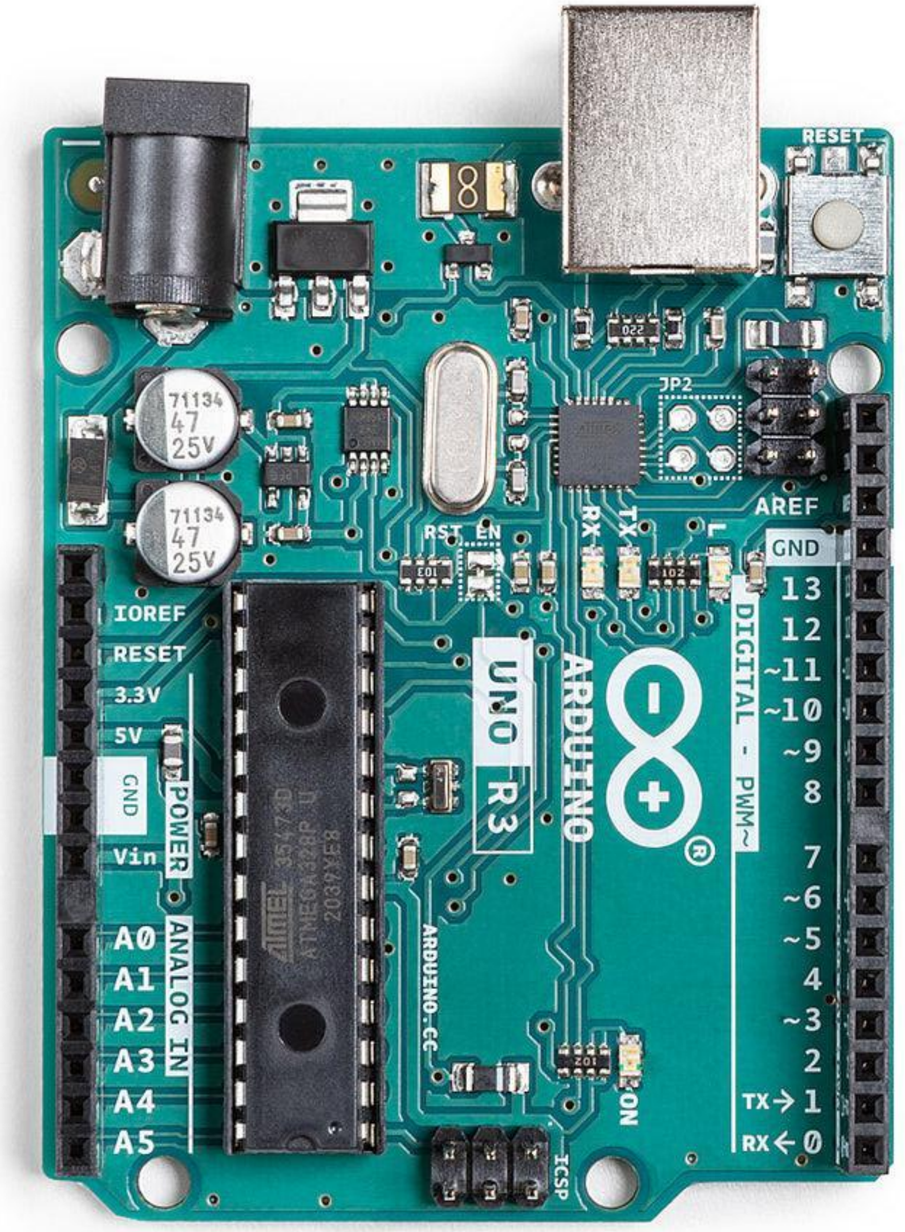


Pin Diagram of Arduino

GND: Ground pins.

RESET: Resets the microcontroller. Can be used to restart the program from the beginning.

Pin Diagram of Arduino

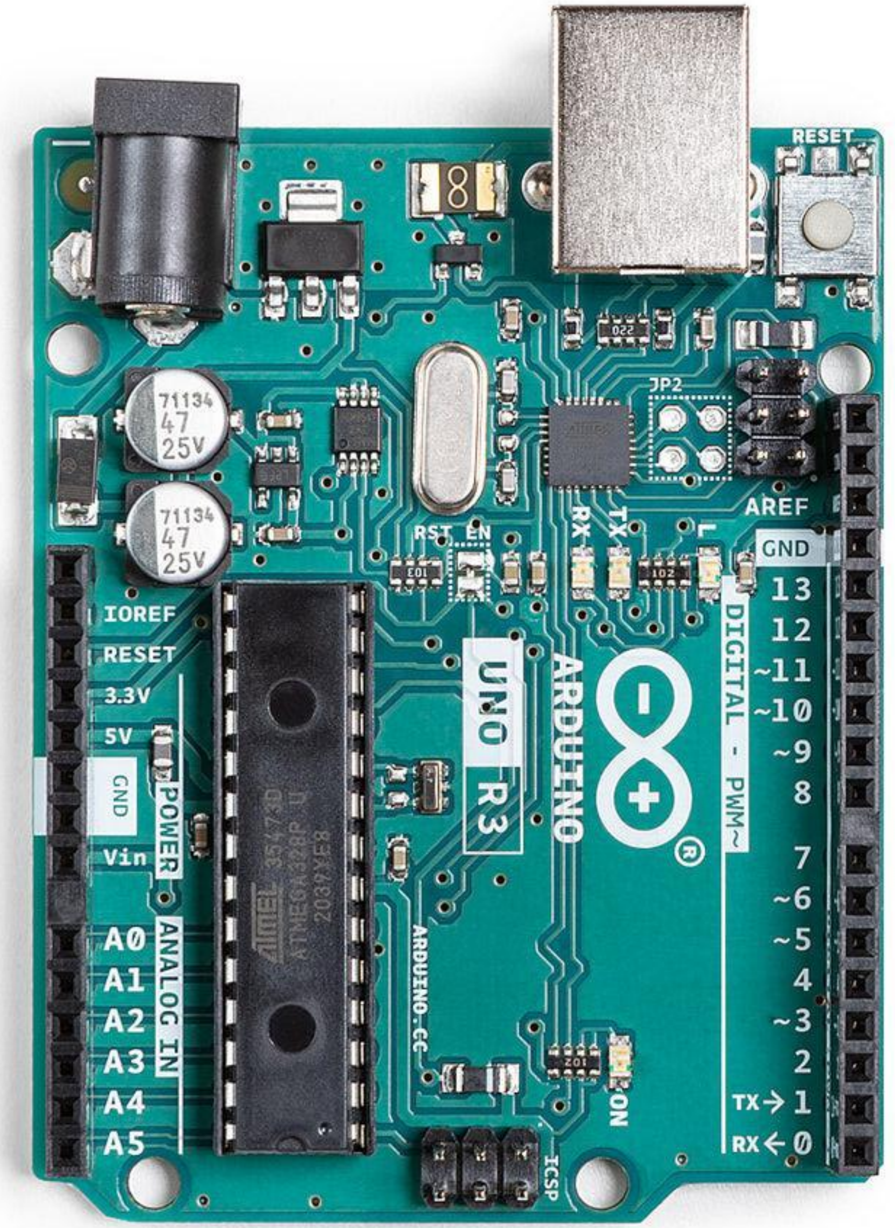


Other Pins

AREF (Analog Reference): Used to set an external reference voltage (0-5V) as the upper limit for analog inputs.

IOREF: Provides the voltage reference at which the microcontroller is operating (usually 5V on Arduino Uno). Allows the shields to adapt to the board's voltage.

Pin Diagram of Arduino



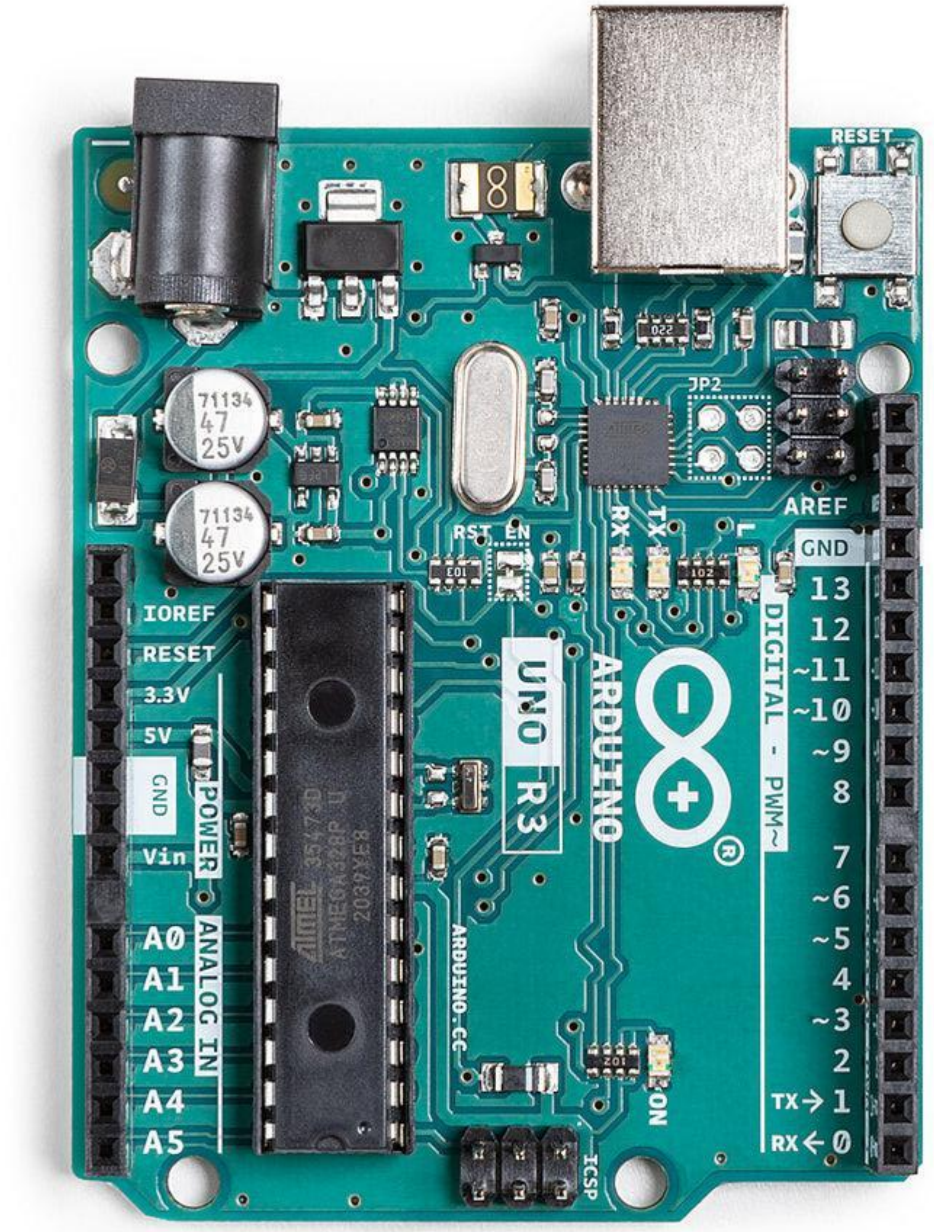
SPI Pins (10, 11, 12, 13)

Pin 10 (SS - Slave Select): Used to select the SPI slave device.

Pin 11 (MOSI - Master Out Slave In): Sends data from the Arduino to the connected SPI device.

Pin 12 (MISO - Master In Slave Out): Receives data from the connected SPI device.

Pin 13 (SCK - Serial Clock): Provides the clock signal used to synchronize data transmission.



Pin Diagram of Arduino

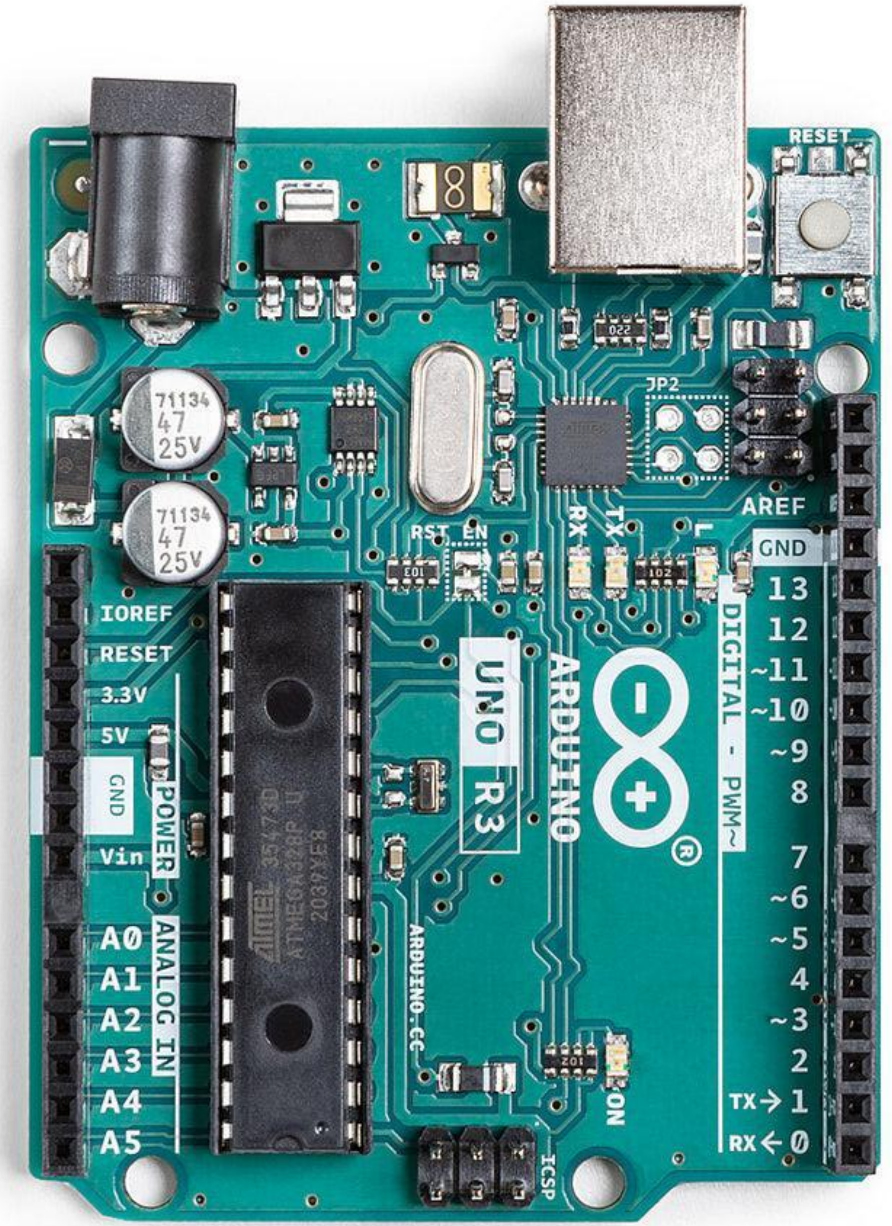
I2C Pins (A4, A5)

SDA (A4): Serial Data Line used for I2C communication.

SCL (A5): Serial Clock Line used for I2C communication.

LED Indicators

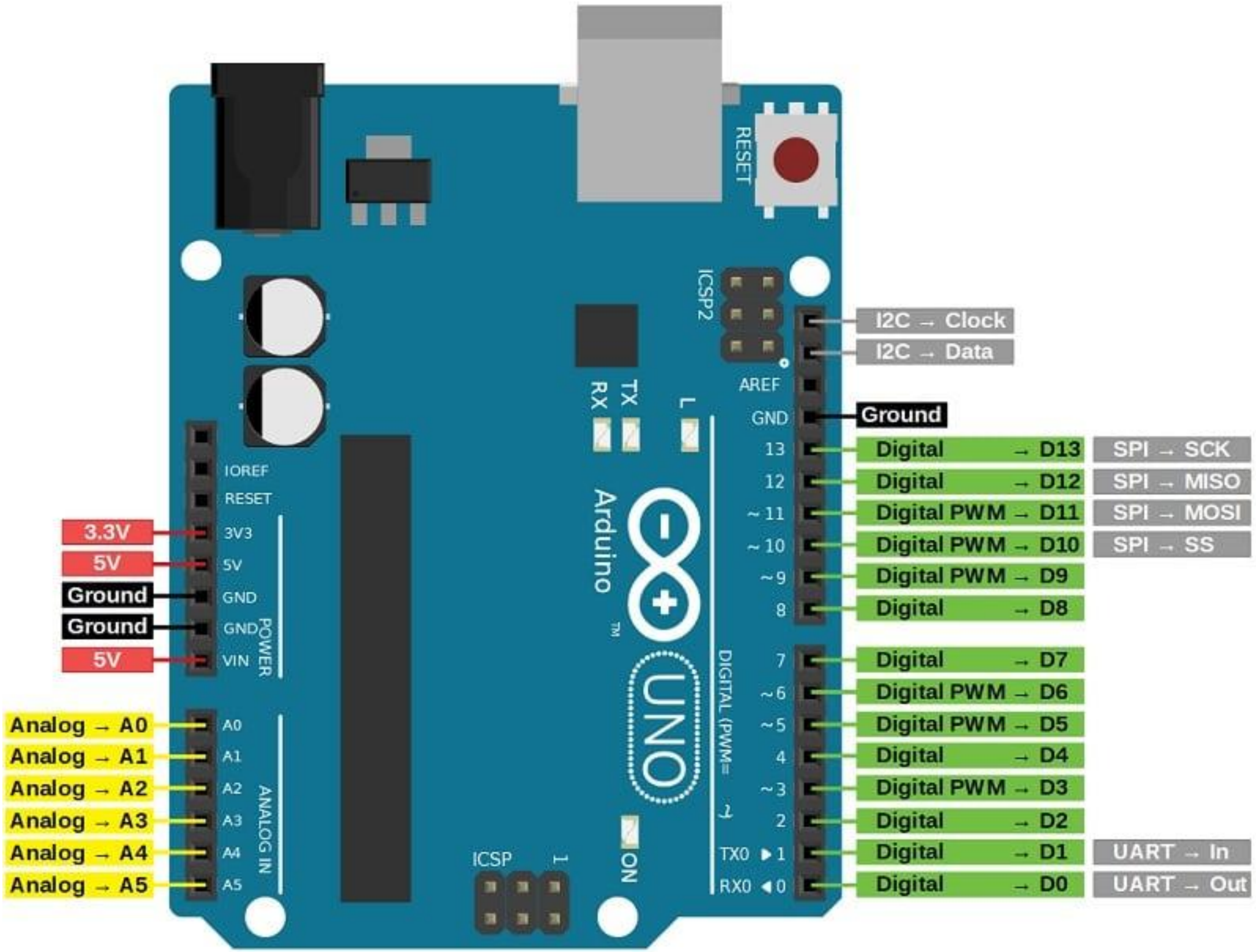
Pin 13 (Built-in LED): Onboard LED connected to digital pin 13. This LED is handy for testing simple sketches without connecting external LEDs.



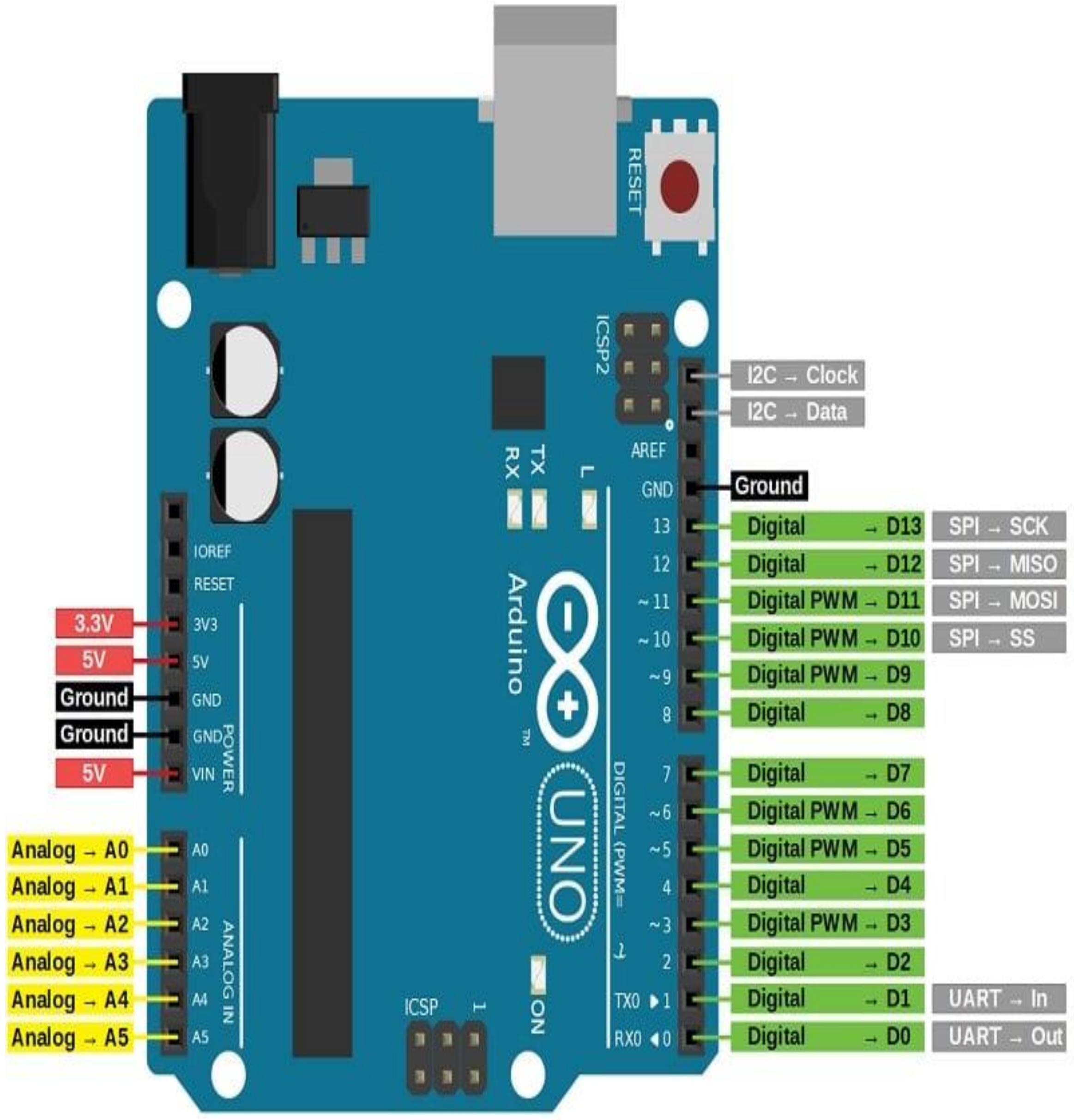
Pin Diagram of Arduino

Communication Pins

TX/RX LEDs: Indicate serial data transmission (TX) and reception (RX). These LEDs blink when data is being sent or received via the USB-to-serial chip and USB connection.

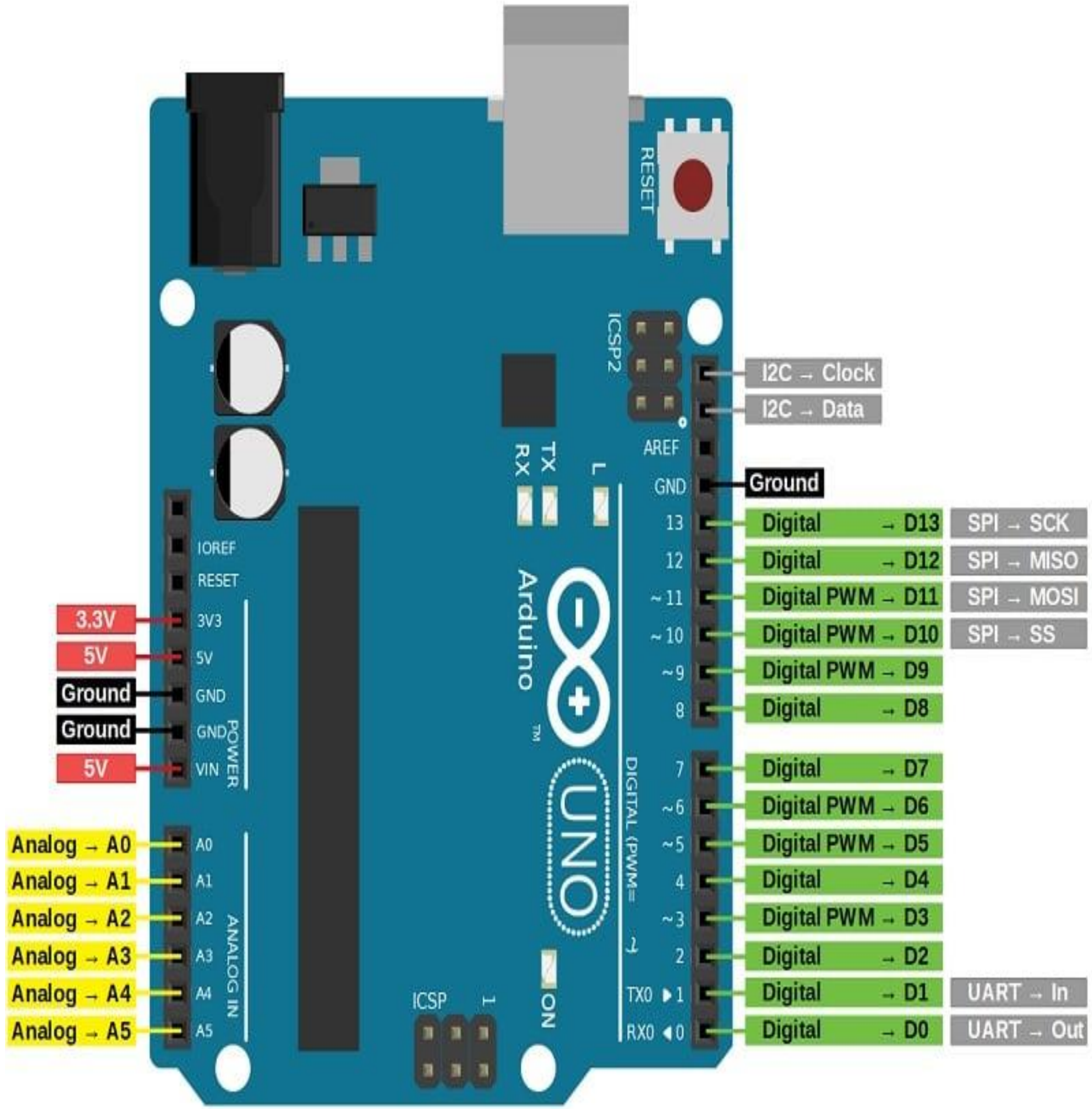


RESET
ICSP2
I2C → Clock
I2C → Data
AREF
GND
Ground
TX
RX
L
13
Digital → D13
SPI → SCK
12
Digital → D12
SPI → MISO
~11
Digital PWM → D11
SPI → MOSI
~10
Digital PWM → D10
SPI → SS
~9
Digital PWM → D9
8
Digital → D8
7
Digital → D7
~6
Digital PWM → D6
~5
Digital PWM → D5
4
Digital → D4
~3
Digital PWM → D3
2
Digital → D2
TX0 ▶ 1
Digital → D1
UART → In
RX0 ◀ 0
Digital → D0
UART → Out
DIGITAL (PWM= ~)
Arduino
™
UNO
ON
ICSP
1
IOREF
RESET
3.3V
3V3
5V
5V
Ground
GND
Ground
GND
5V
VIN
POWER
Analog → A0
A0
Analog → A1
A1
Analog → A2
A2
Analog → A3
A3
Analog → A4
A4
Analog → A5
A5
ANALOG IN



SPI (Serial Peripheral Interface) Devices

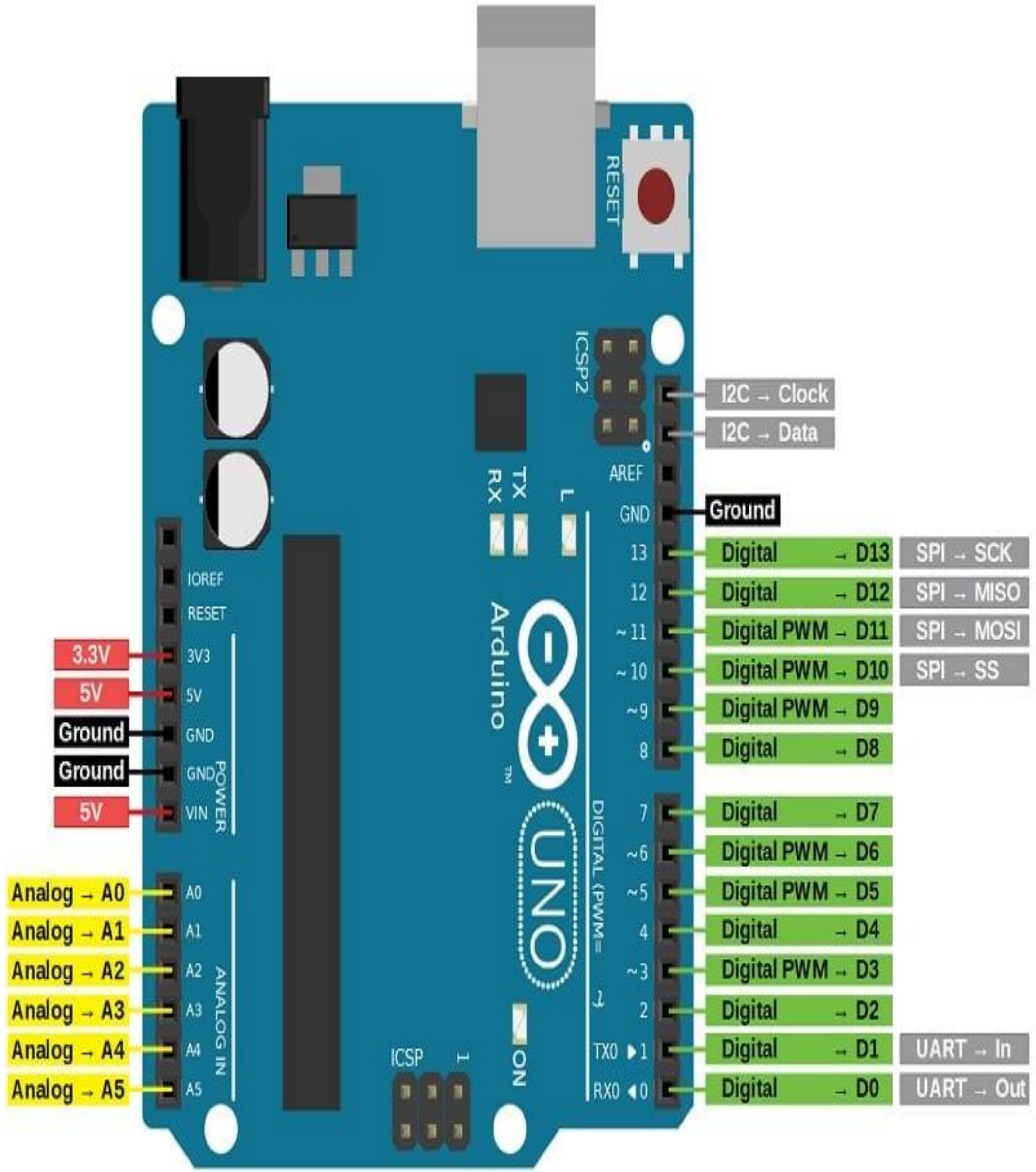
SPI (Serial Peripheral Interface) is a synchronous serial communication protocol used primarily to communicate between a master device (such as a microcontroller like the Arduino) and one or more peripheral devices (like sensors, displays, or memory chips). It's widely used because of its speed and simplicity, especially when high-speed data transfer is required.



Key Components of SPI Communication:

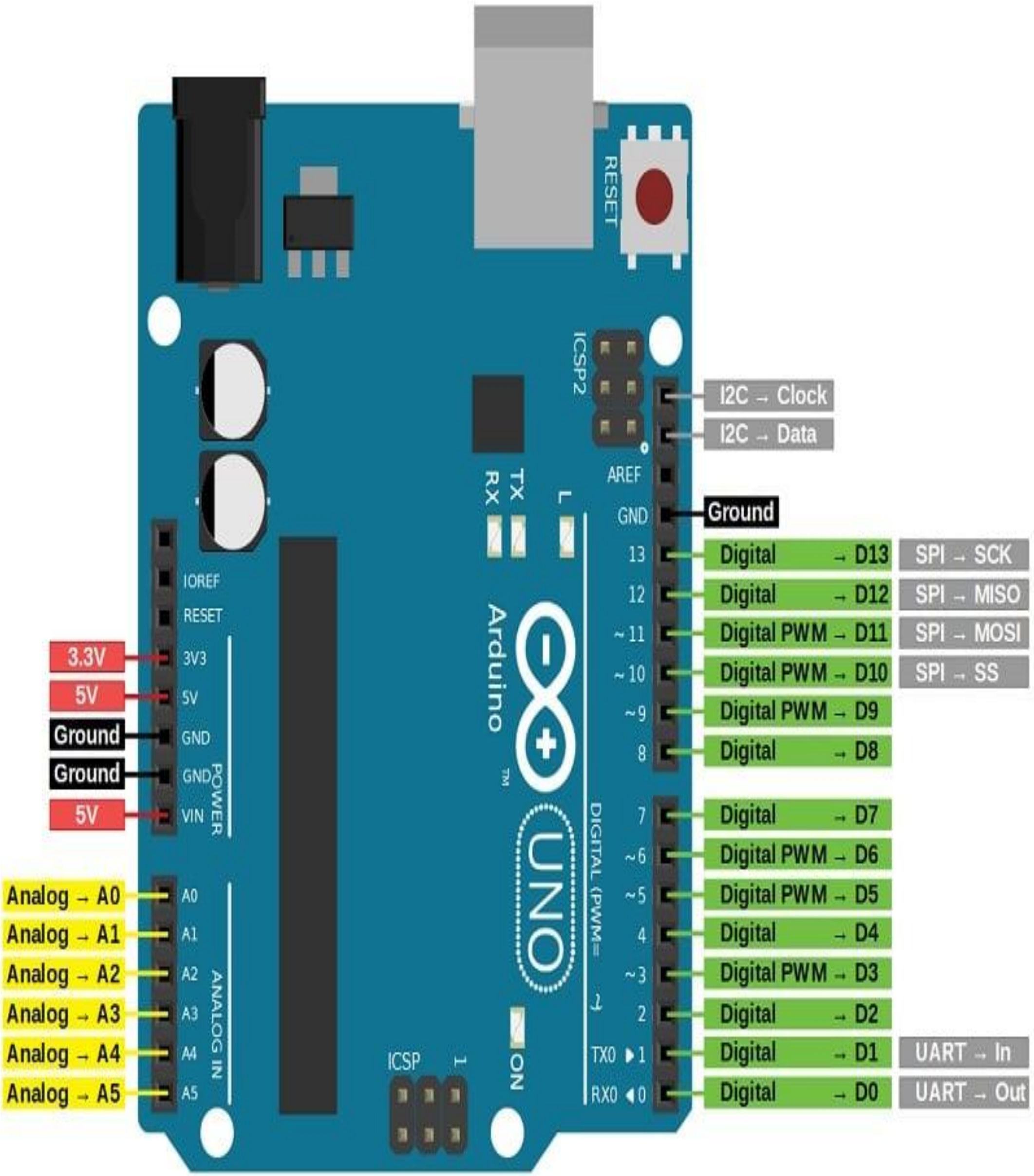
1. Master and Slave Devices:

- **Master**: The device that controls the communication, typically a microcontroller (e.g., Arduino).
- **Slave(s)**: The devices that respond to the master's commands, such as sensors, displays, SD cards, etc.



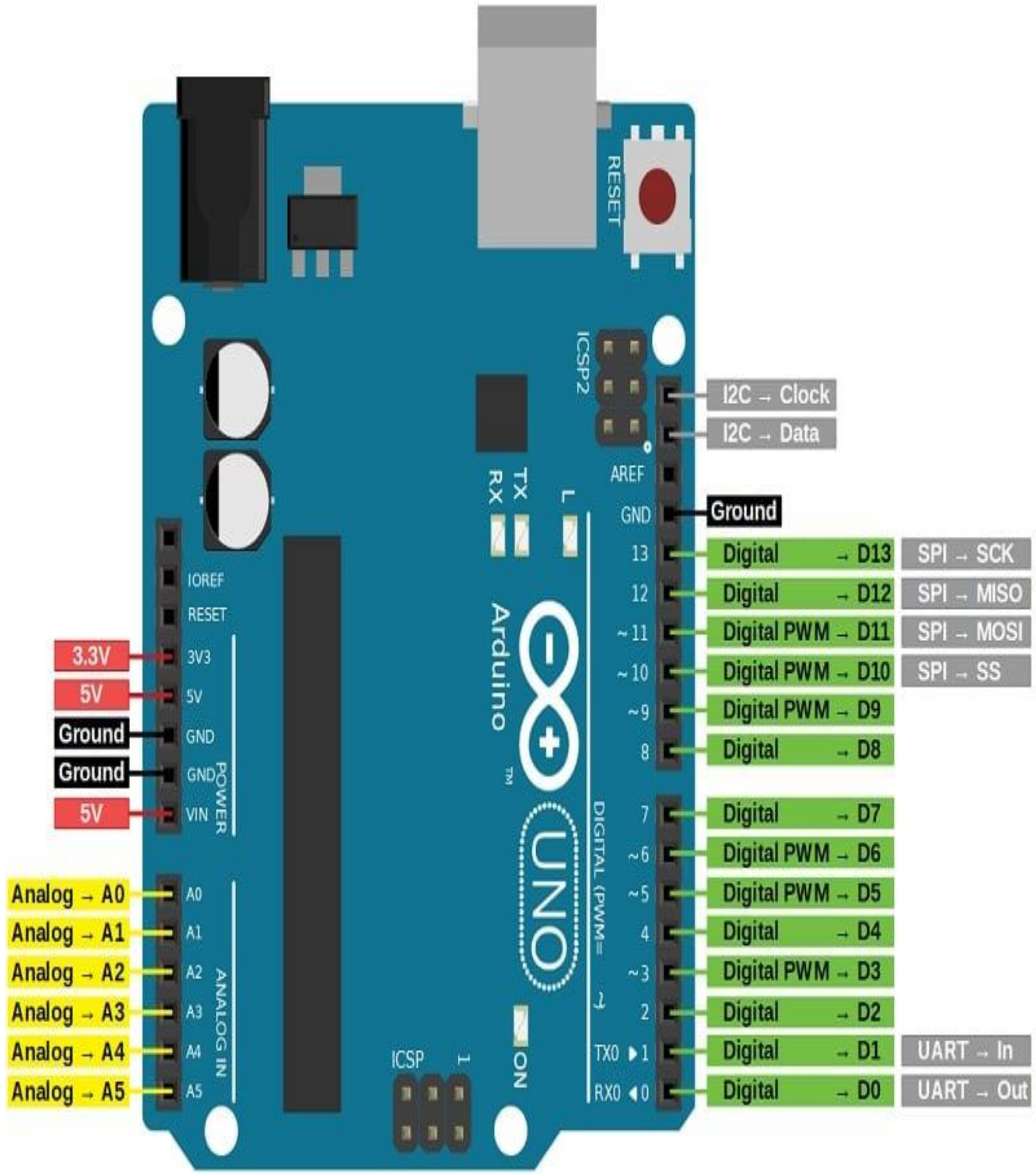
SPI Signals:

- **MOSI (Master Out Slave In)**: The line through which data is sent from the master to the slave.
- **MISO (Master In Slave Out)**: The line through which data is sent from the slave back to the master.
- **SCK (Serial Clock)**: The clock signal generated by the master to synchronize data transfer.
- **SS (Slave Select)**: A line used by the master to select which slave device to communicate with. When SS is low, the slave device is active; when high, it is inactive.

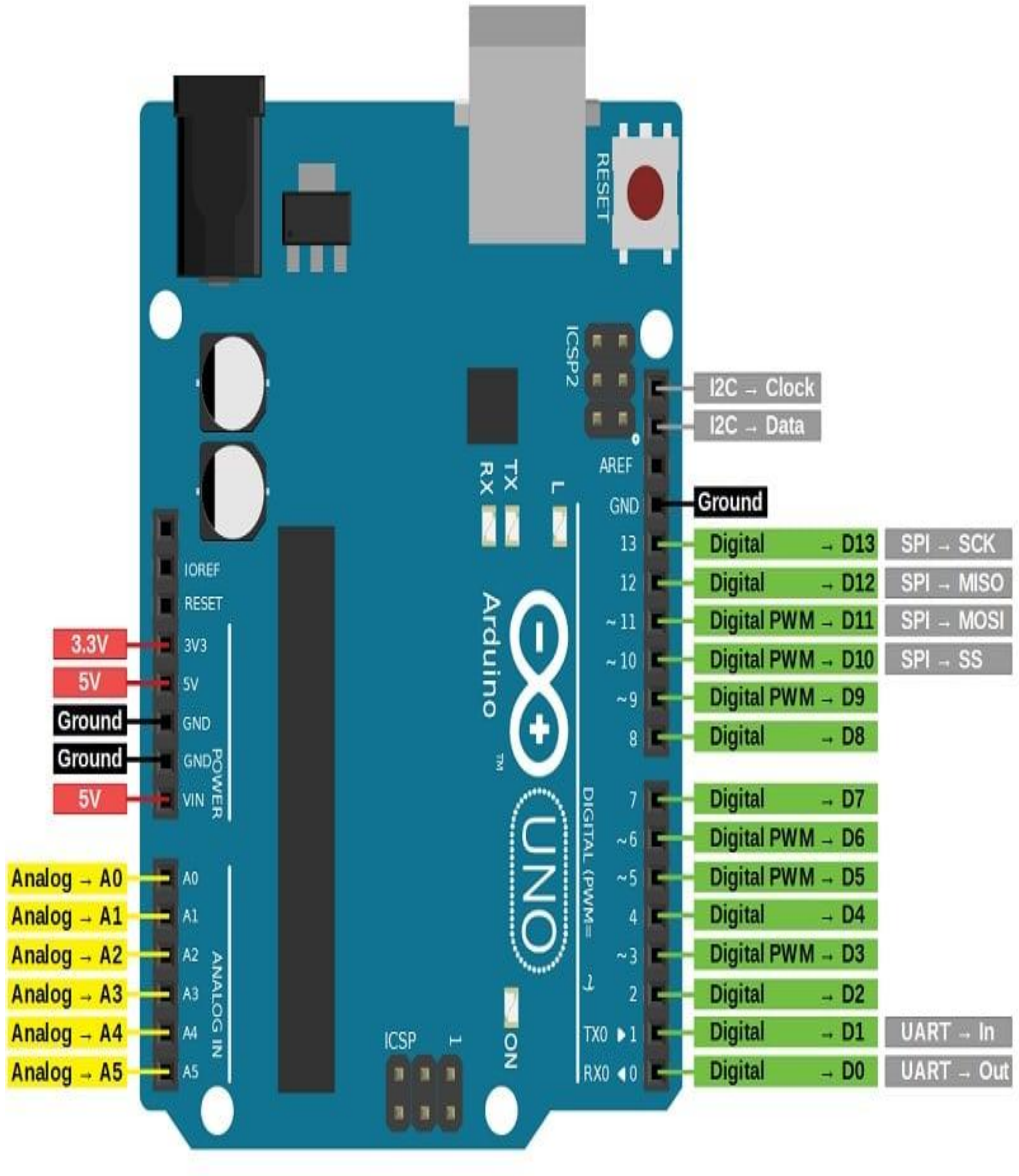


How SPI Works:

1. **Data Transfer**: SPI is a full-duplex communication protocol, meaning data can be sent and received simultaneously.
2. **Clock Control**: The master generates the clock signal on the SCK line, which dictates the timing of data transmission.

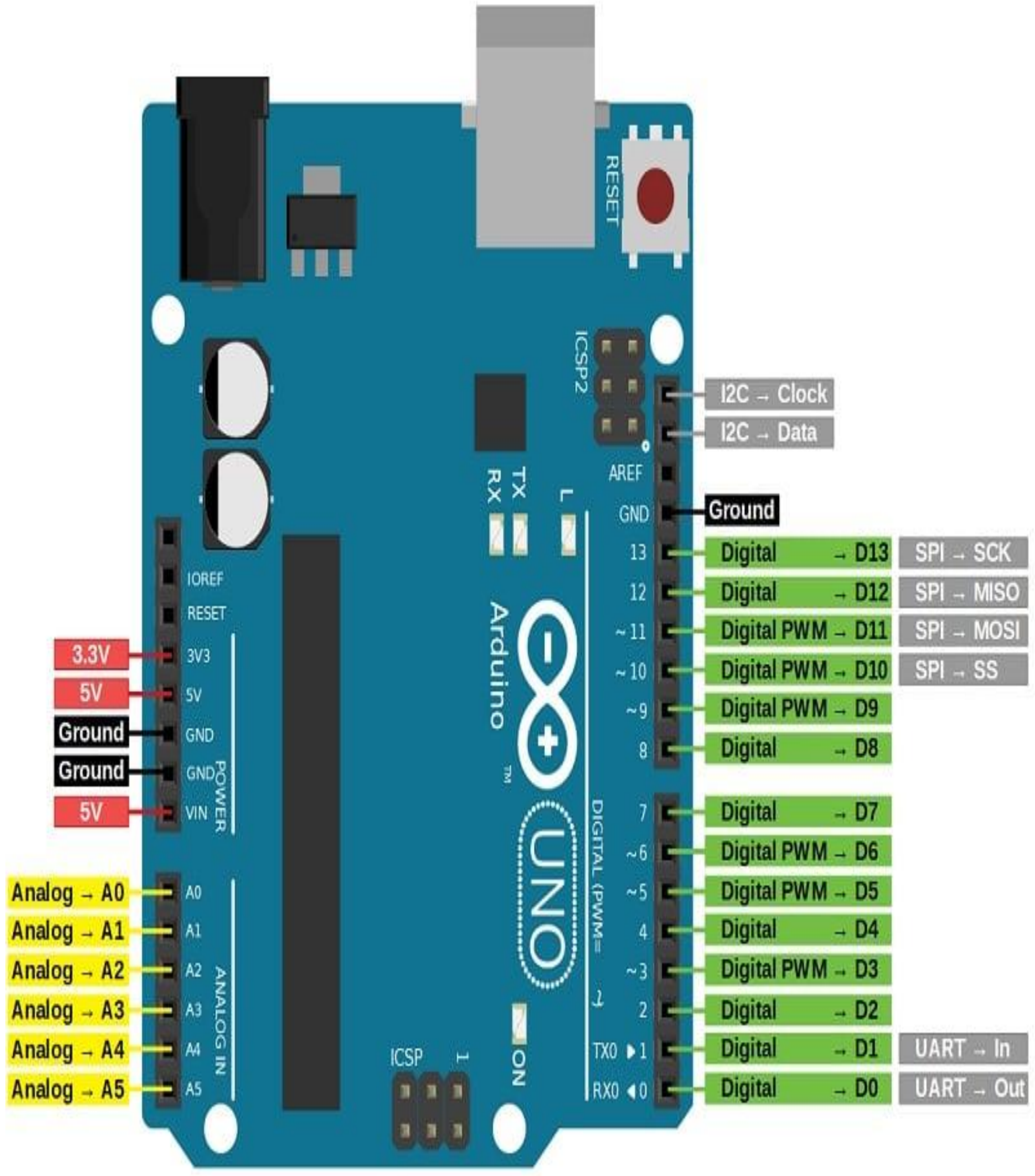


3. Slave Selection: Only the slave with its SS line pulled low communicates with the master. Multiple slaves can be connected to the same SPI bus, and the master controls which one to communicate with by toggling the corresponding SS line.



Advantages of SPI:

- **High Speed**: SPI supports high data rates compared to other protocols like I2C.
- **Simplicity**: The protocol is straightforward, with minimal software overhead.
- **Full Duplex**: Data transmission in both directions simultaneously increases efficiency.

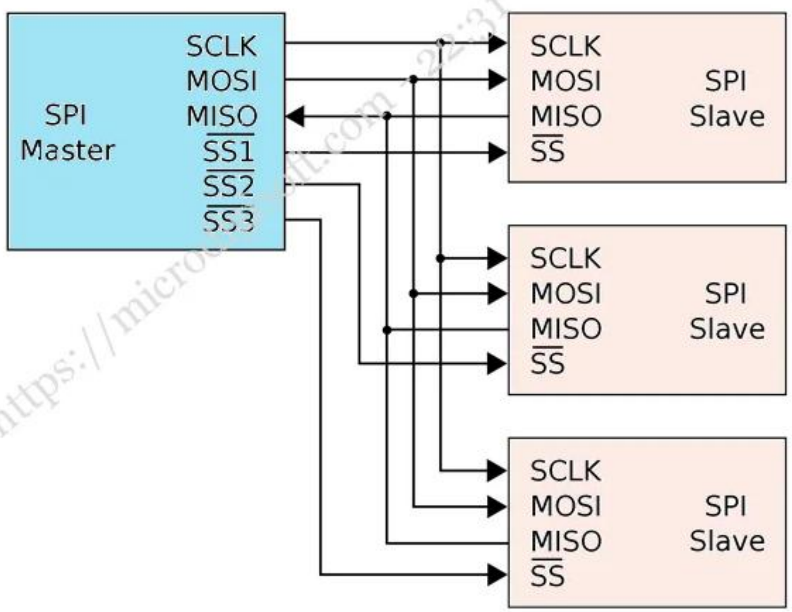


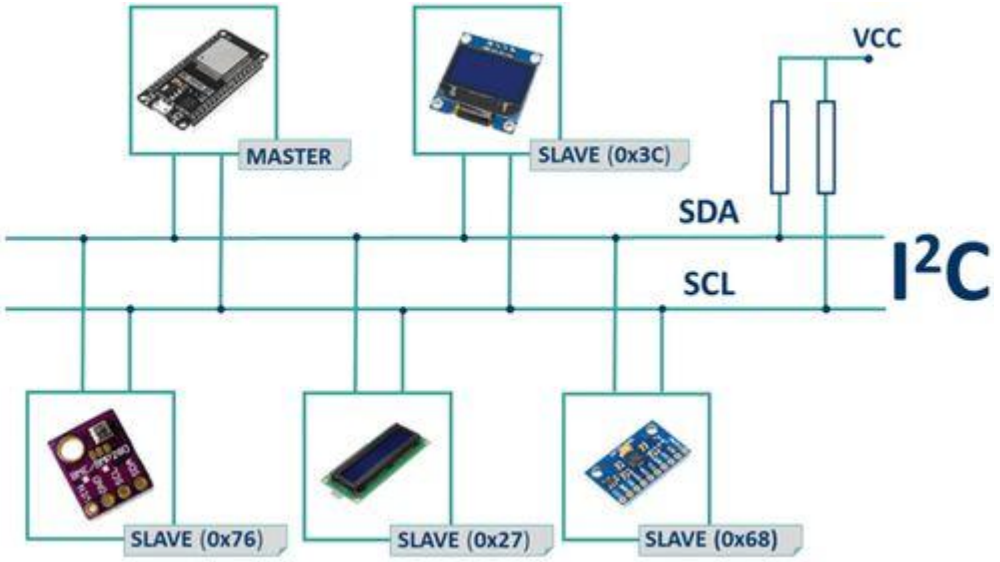
Common SPI Devices:

- **Sensors**: Gyroscopes, accelerometers, temperature sensors.
- **Memory Devices**: EEPROM, Flash memory.
- **Displays**: OLEDs, LCDs.
- **SD Cards**: Used for data storage in many embedded applications.

Why SPI & I2C Communication Are Necessary in Arduino:

1. To Connect External Devices:
Arduino has limited I/O pins. SPI and I2C let it communicate with sensors, displays, memory chips, etc., using fewer pins.
2. To Support Multiple Devices: I2C allows multiple devices on two wires, and SPI allows multiple devices using one shared bus with separate select lines.
3. To Exchange Data Faster & More Efficiently: SPI (fast, up to 10+ Mbps) and I2C (simple, address-based) are ideal for high-speed or multi-device communication—better than using digitalRead/digitalWrite manually.
4. To Interface with Modern Modules: Most modern modules (e.g., SD cards, OLEDs, IMUs) use SPI/I2C. Arduino needs these protocols to talk to them.
In short: Less wiring, Multiple devices, Faster data, Modern module support.



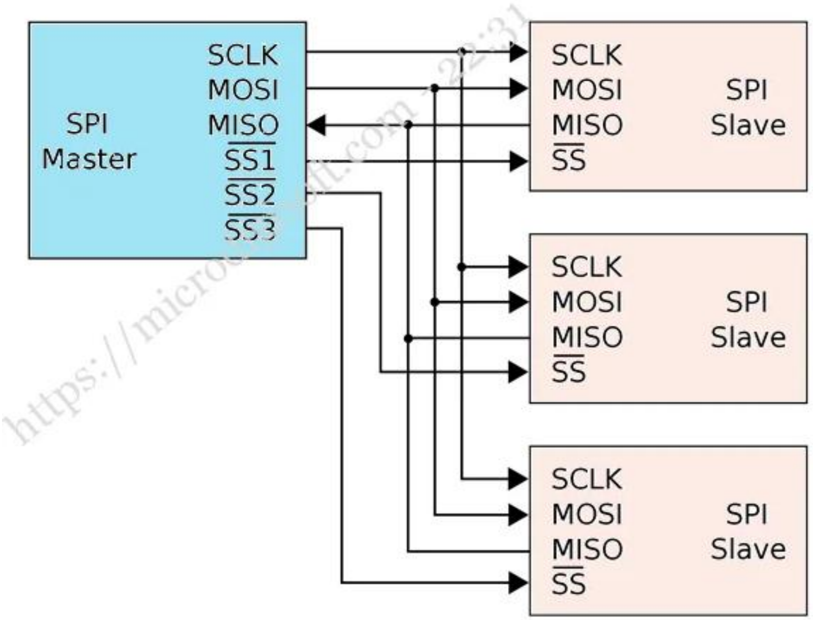


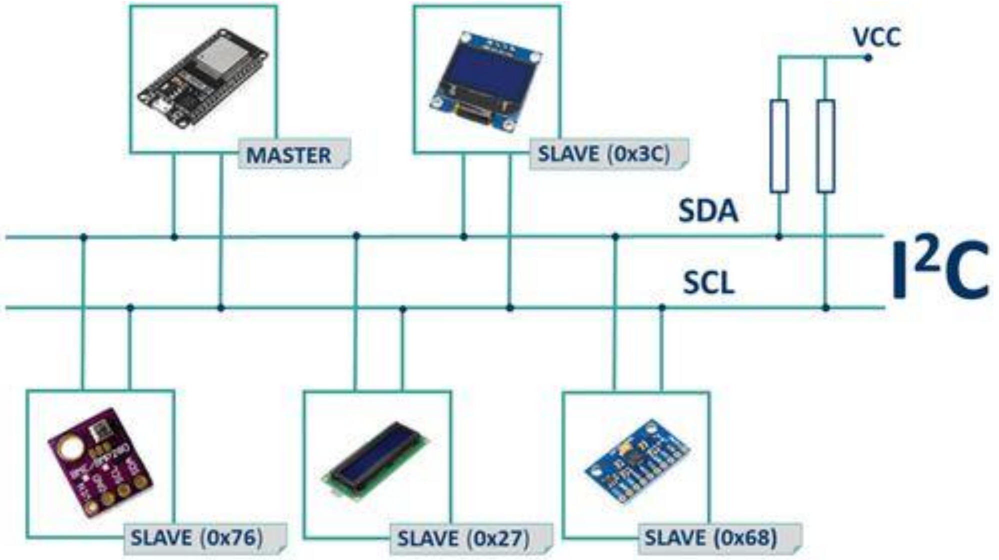
Why SPI & I2C Communication Are Necessary in Arduino:

- SPI is like a manager (Master) giving commands to one worker (Slave) over a walkie-talkie, while listening to their response at the same time.
- I2C is like a teacher asking “Roll number 12, what's your homework?” only student 12 replies. Others stay silent.

Why Use I2C/SPI Instead of Just Analog/Digital Pins?

- Analog/Digital pins can only do basic HIGH/LOW or voltage readings — not enough to transfer structured data (like complex sensor values, text, images).
- I2C/SPI allow fast, reliable, multi-byte communication with smart devices like sensors, displays, SD cards, etc.

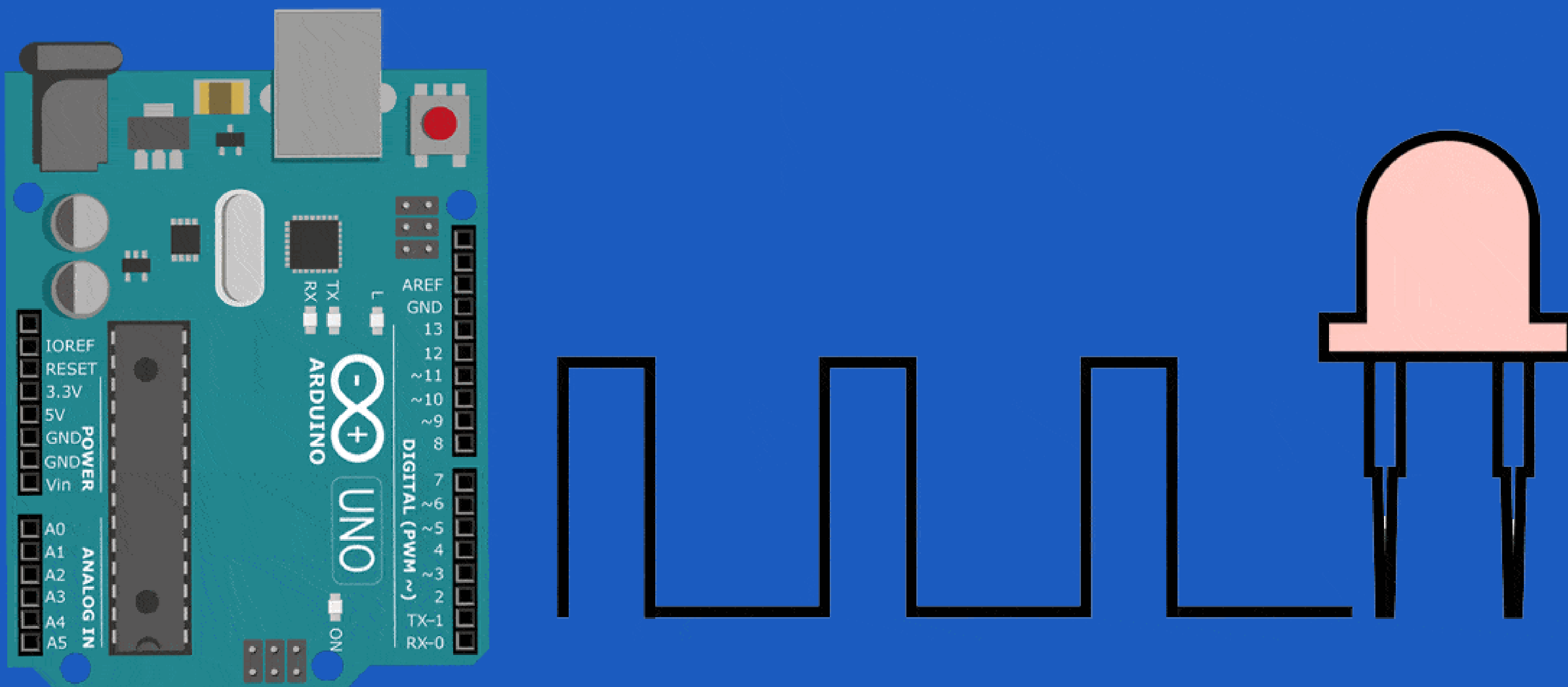




CSE 233: Embedded Systems

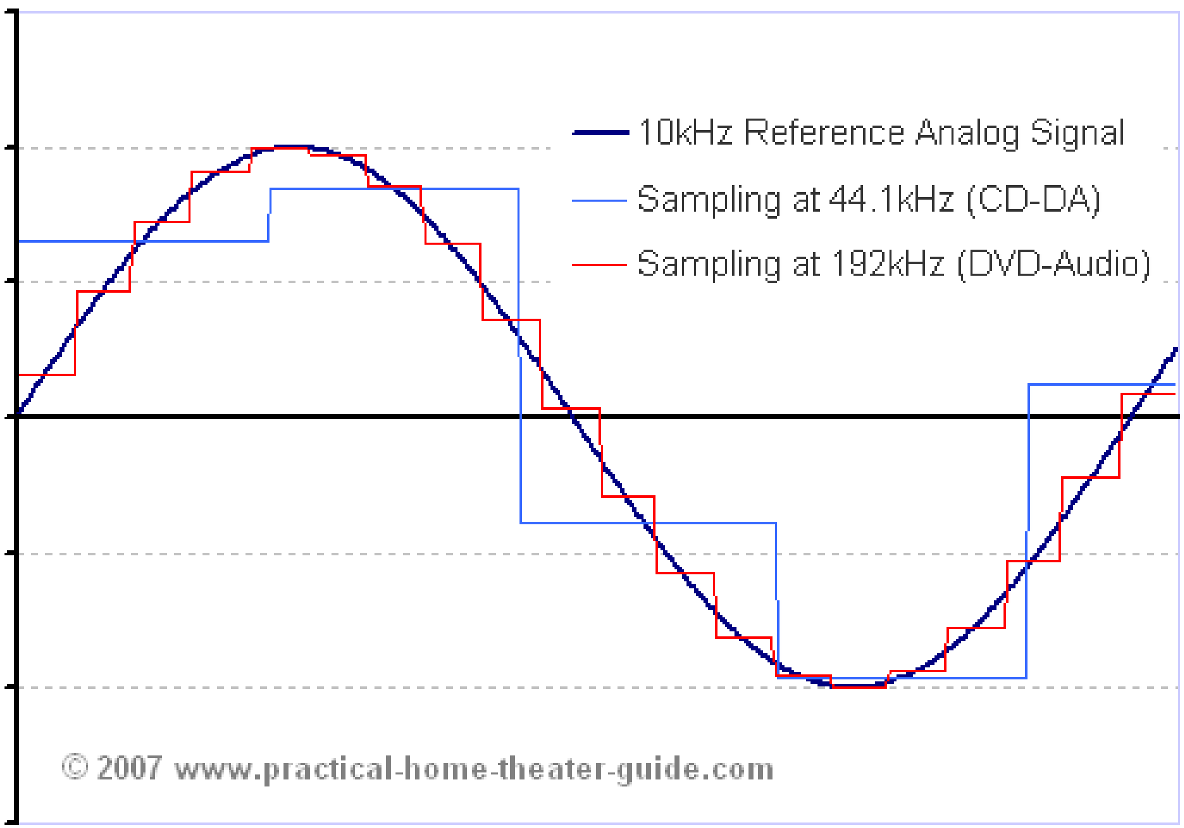
PWM signal (Analog IO)

Arduino PWM Guide

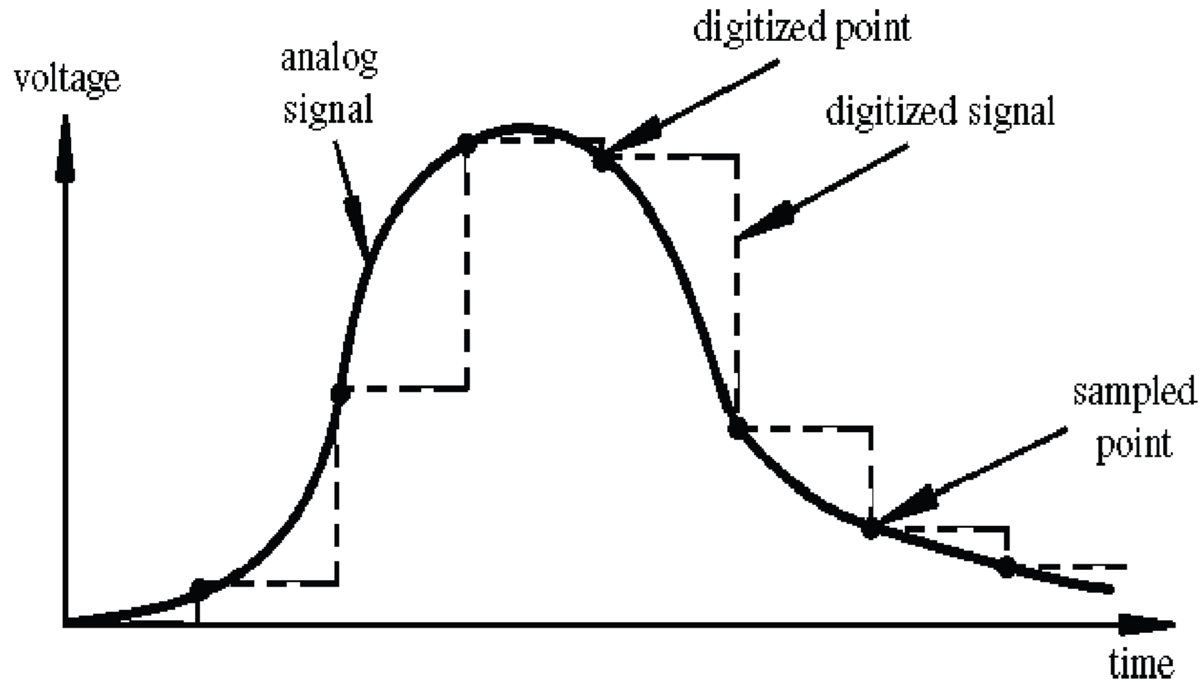


Analog Input

- Think about our voice
- Voice is nothing but analog signal
- Then it is digitized and store or transmit



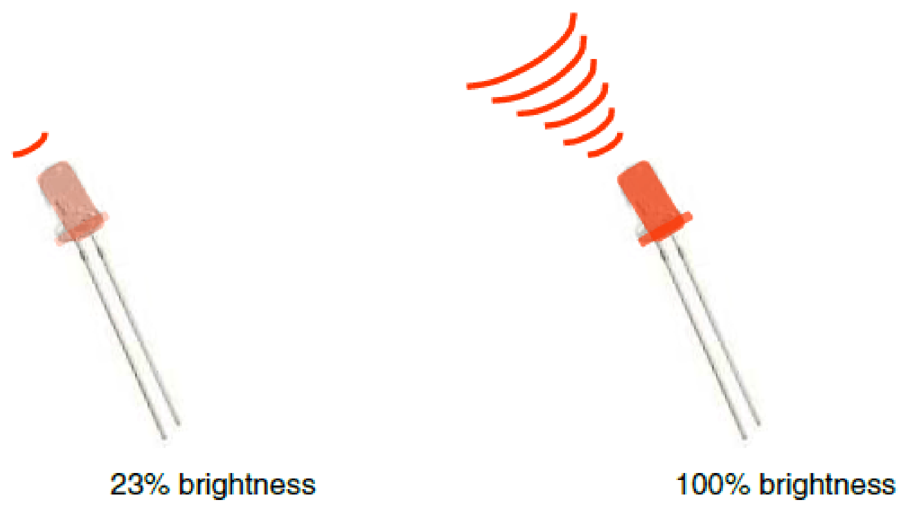
Arduino Analog Input



- *Resolution*: the number of different voltage levels (i.e., *states*) used to digitize an input signal
- Resolution values range from 256 states (8 bits) to 4,294,967,296 states (32 bits)
- The Arduino ADC uses 1024 states (10 bits) but processors uses 8bits in memory and registers
- Smallest measurable voltage change is 5V/1024 or 4.8 mV
- Maximum sample rate is 10,000 times a second

Analog Output

- Can a digital device produce analog output?

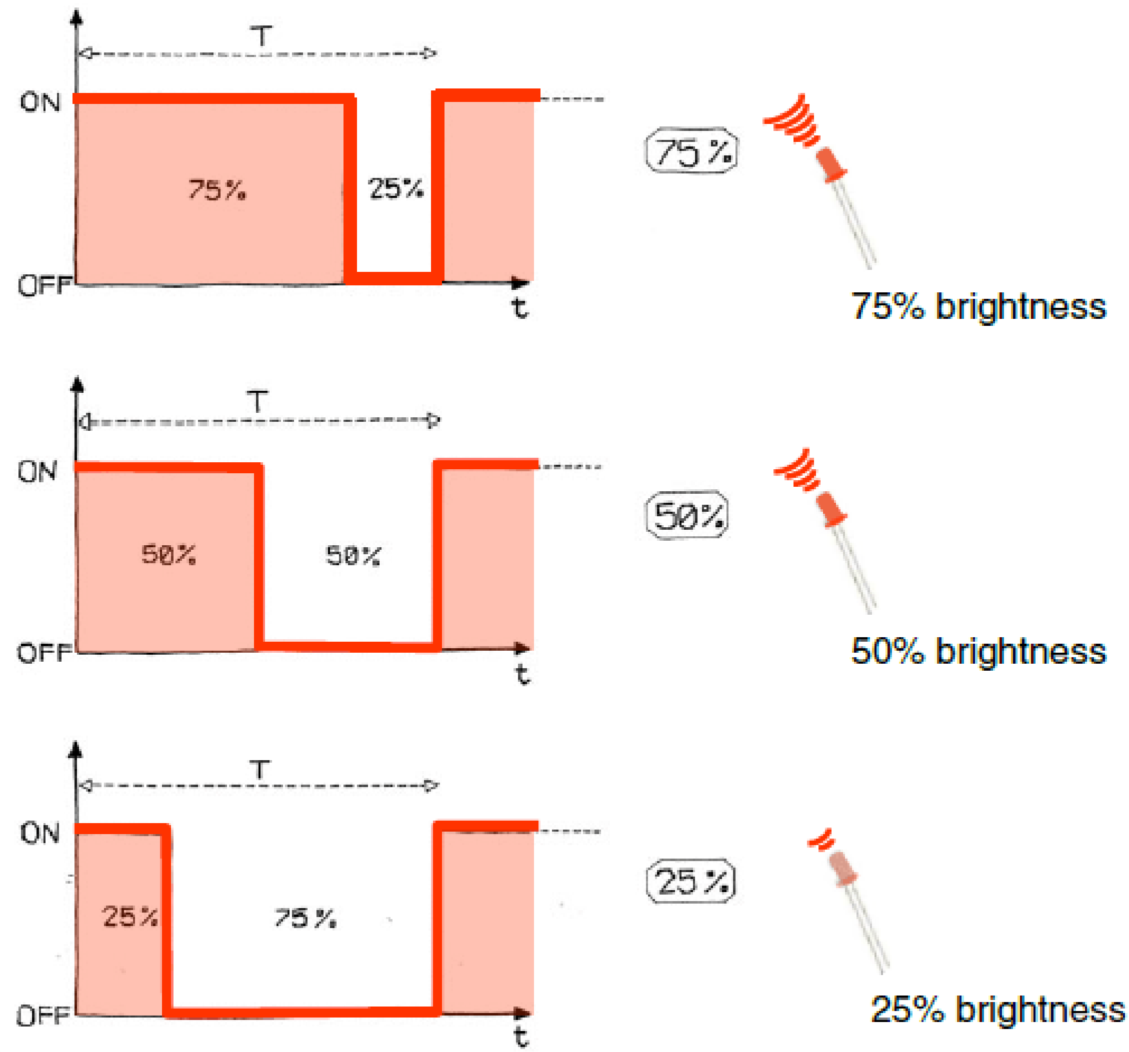


- Analog output can be simulated using pulse width modulation (PWM)

Pulse Width Modulation

- Can't use digital pins to directly supply say 2.5V, but can pulse the output on and off really fast to produce the same effect

- The on-off pulsing happens so quickly, the connected output device "sees" the result as a reduction in the voltage



PWM Duty Cycle

output voltage = (on_time / cycle_time) * 5V

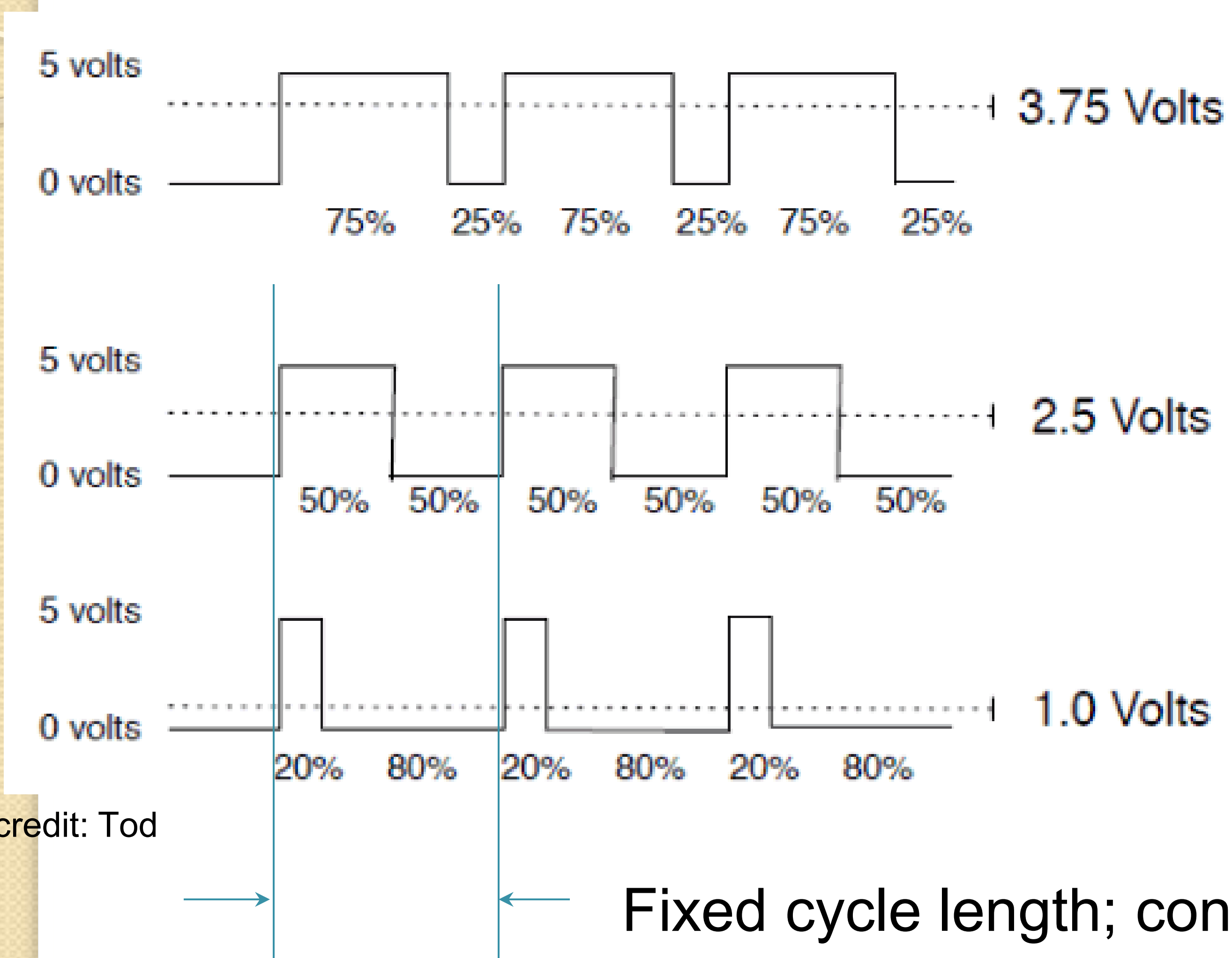
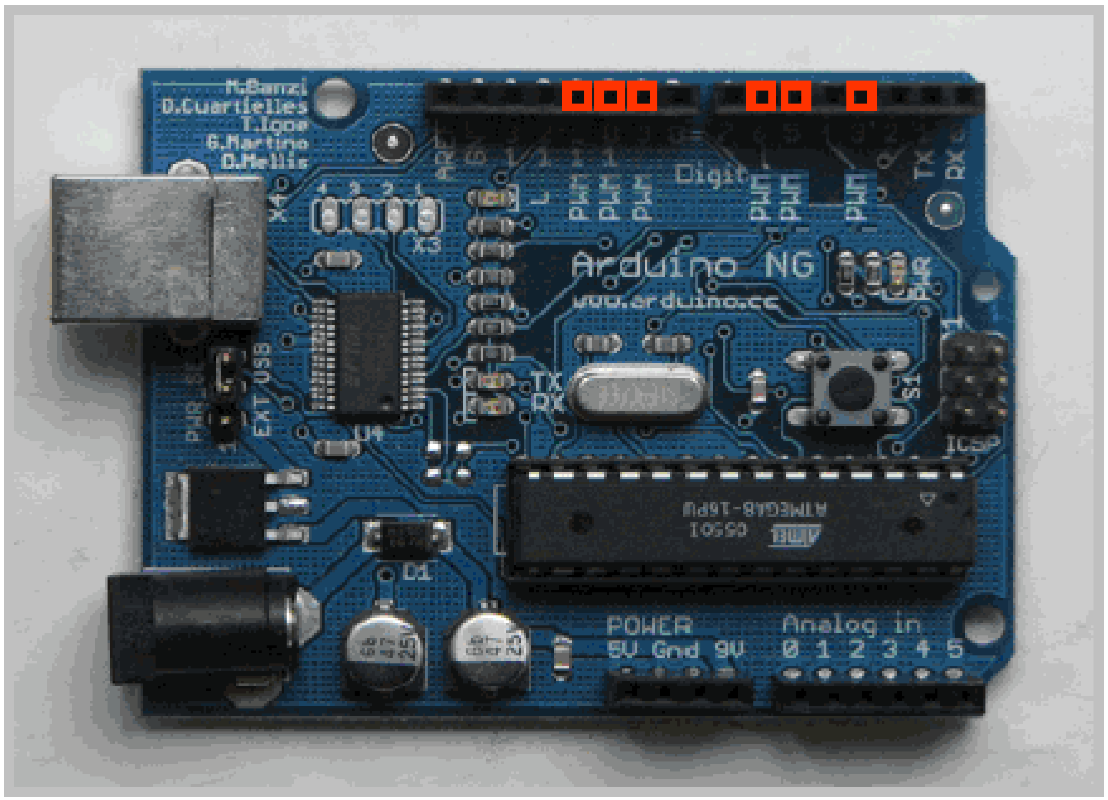


Image credit: Tod Kurt

Fixed cycle length; constant number of cycles/sec

PWM Pins

Your Arduino board has built in PWM circuits, on pins 3, 5, 6, 9, 10, and 11



- **Command:**
  analogWrite(pin,value)
- value is duty cycle: between 0 and 255
- **Examples:**
  analogWrite(9, 127)
  for a 50% duty cycle

  analogWrite(11, 63)
  for a 25% duty cycle

Course Title: Embedded Systems and IoT

Course Code: CSE233

INTRODUCTION TO ARDUINO SENSORS

Prepared By:

Jul Jalal Al-Mamur Sayor (JMS)

Lecturer, Dept. of CSE, DIU

ARDUINO SENSORS

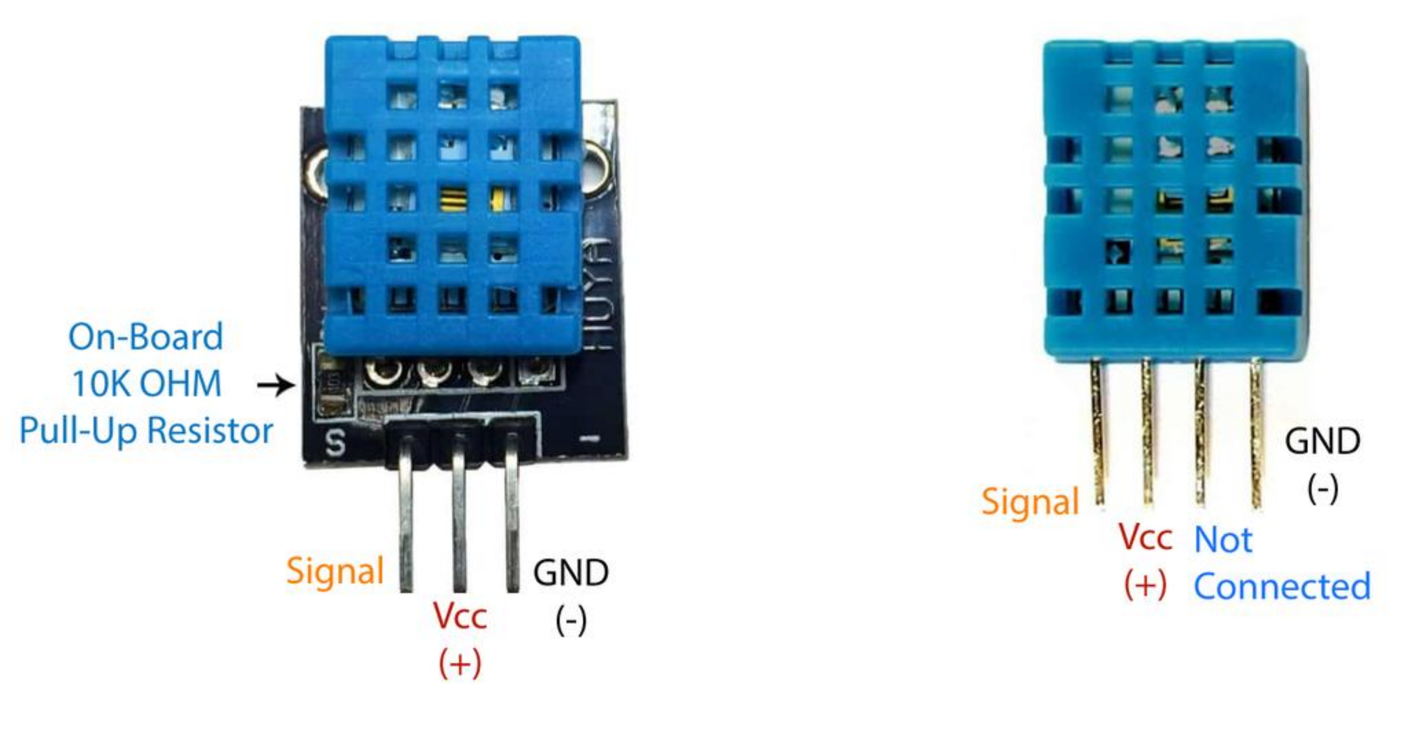
- A sensor is a device that detects and measures physical properties from the environment (like temperature, humidity, or motion) and converts them into data that can be processed by Embedded Systems or Internet of Things (IoT) devices.
- In Embedded Systems and IoT, sensors play a key role by providing real-time data, enabling devices to monitor and respond to their surroundings autonomously.
- This data can be used locally or transmitted over networks, making sensors essential for automation, monitoring, and smart applications.

TYPES OF ARDUINO SENSORS

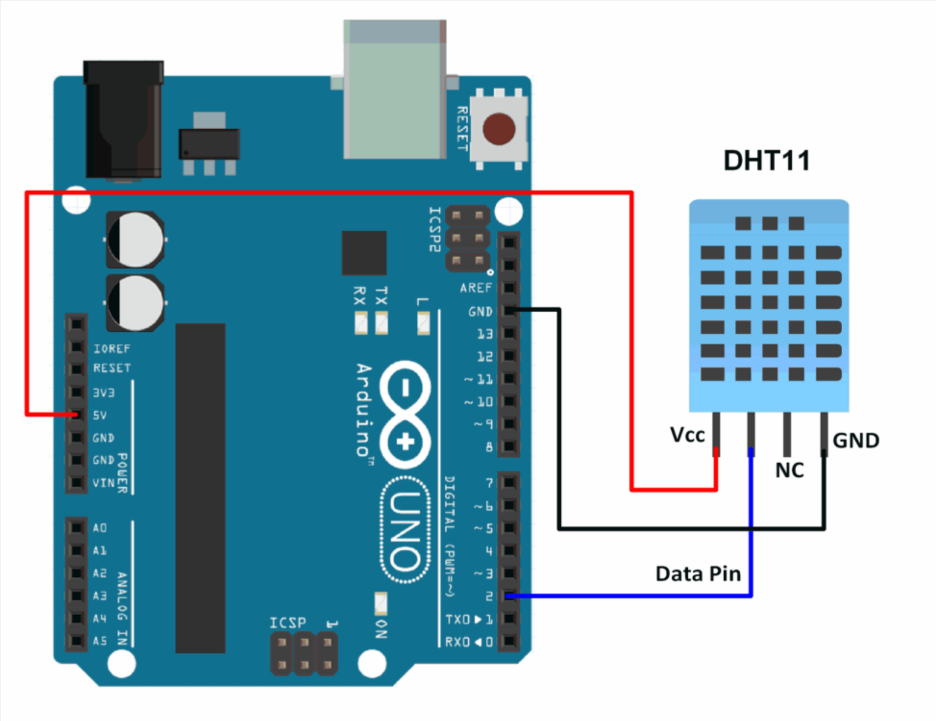
Sensor Type	Examples	Purpose / Application
Light Sensors	Photoresistor (LDR), BH1750, TSL2561	Detect light intensity for automatic lighting, screen brightness, or day/night detection.
Motion Sensors	PIR sensor, ADXL345 (Accelerometer), MPU6050	Detect movement or orientation in security systems, fitness devices, and controllers.
Proximity & Distance Sensors	HC-SR04 (Ultrasonic), IR sensor, LiDAR	Measure distance or detect nearby objects in robots, automation, and vehicles.
Gas & Air Quality Sensors	MQ-2 (Smoke), MQ-135 (Air Quality)	Detect gases like CO, CH_4, pollutants in air monitoring, alarms, and industrial safety.
Pressure Sensors	BMP180, BMP280, MPX5700	Measure pressure for weather data, altitude sensing, and industrial systems.
Sound Sensors	Microphones, sound level meters	Detect noise levels or voice commands for audio analysis and voice-activated devices.
Touch Sensors	TTP223, Force-sensitive resistor	Detect physical touch or pressure in smart devices and interactive displays.
Vibration Sensors	Piezoelectric sensors, vibration switches	Monitor vibrations for diagnostics, industrial health, or security alerts.
Flow Sensors	YF-S201 (Water Flow), gas flow sensors	Measure liquid/gas flow rate for irrigation, fluid control, and smart meters.
Infrared Sensors	IR Sensors, Flame sensors	Detect heat, motion, or fire sources in automation and safety systems.
Magnetic Sensors	Hall Effect sensors, magnetometers	Detect magnetic fields in motor control, positioning, and navigation systems.
Image & Vision Sensors	Cameras, CCD/CMOS sensors	Capture and process images for surveillance, robotics, and facial recognition.
Chemical Sensors	pH sensors, ion-selective electrodes	Detect chemical content for agriculture, medical, and lab testing.

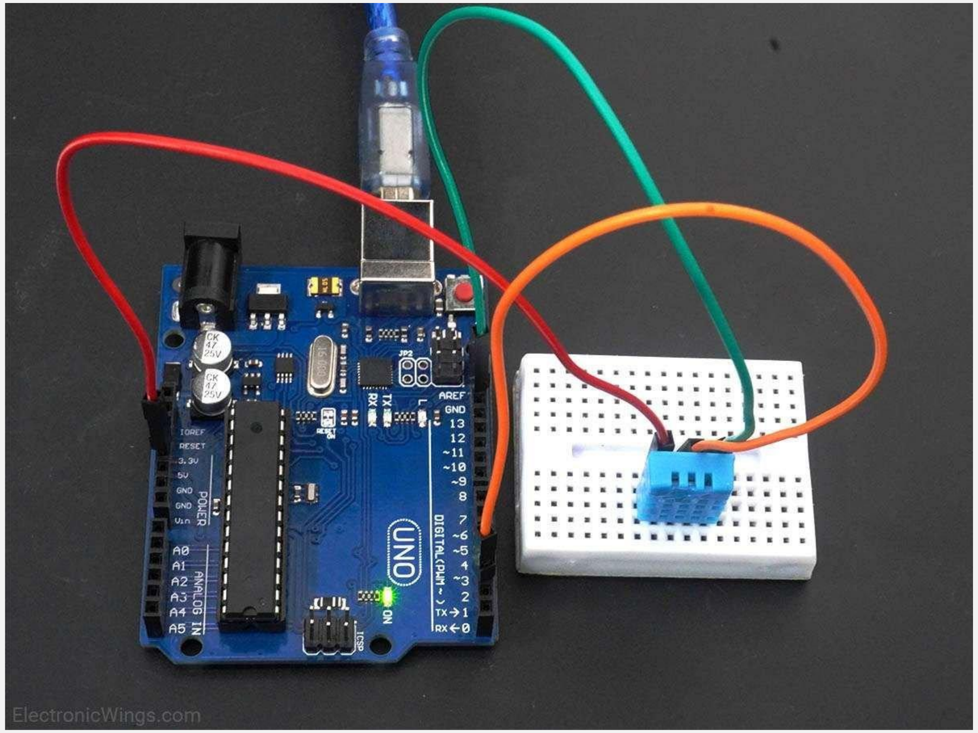
INTRODUCTION TO DHT11

- The **DHT11** is a basic, low-cost digital sensor used to measure **temperature** and **humidity**. It features a **calibrated digital signal output**, making it easy to connect with microcontrollers like **Arduino**. The sensor contains a **thermistor** and a **capacitive humidity sensor,** sending data in a structured digital format through a **single data pin**, ensuring reliable communication.



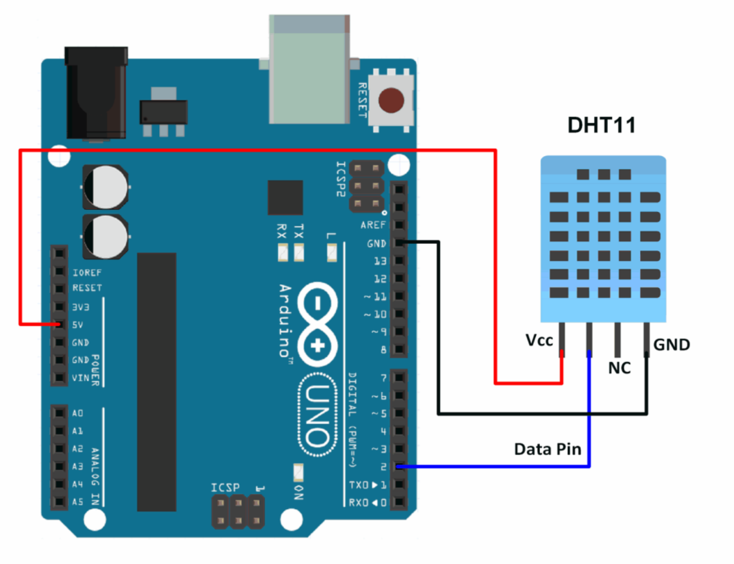
INTERFACING DHT11 WITH ARDUINO





HOW DHT11 WORKS WITH ARDUINO

- Sensing:
  Inside the DHT11, a thermistor measures temperature and a capacitive sensor measures humidity — both produce analog signals.
- Conversion:
  The DHT11 has a built-in ADC (Analog-to-Digital Converter) that converts these analog readings into digital values (like 25°C, 60% RH).
- Communication:
  The sensor sends the data as a digital signal through a single digital I/O pin (e.g., pin 2 on Arduino) using a special timing-based protocol.



SKETCH/CODE FOR DHT11

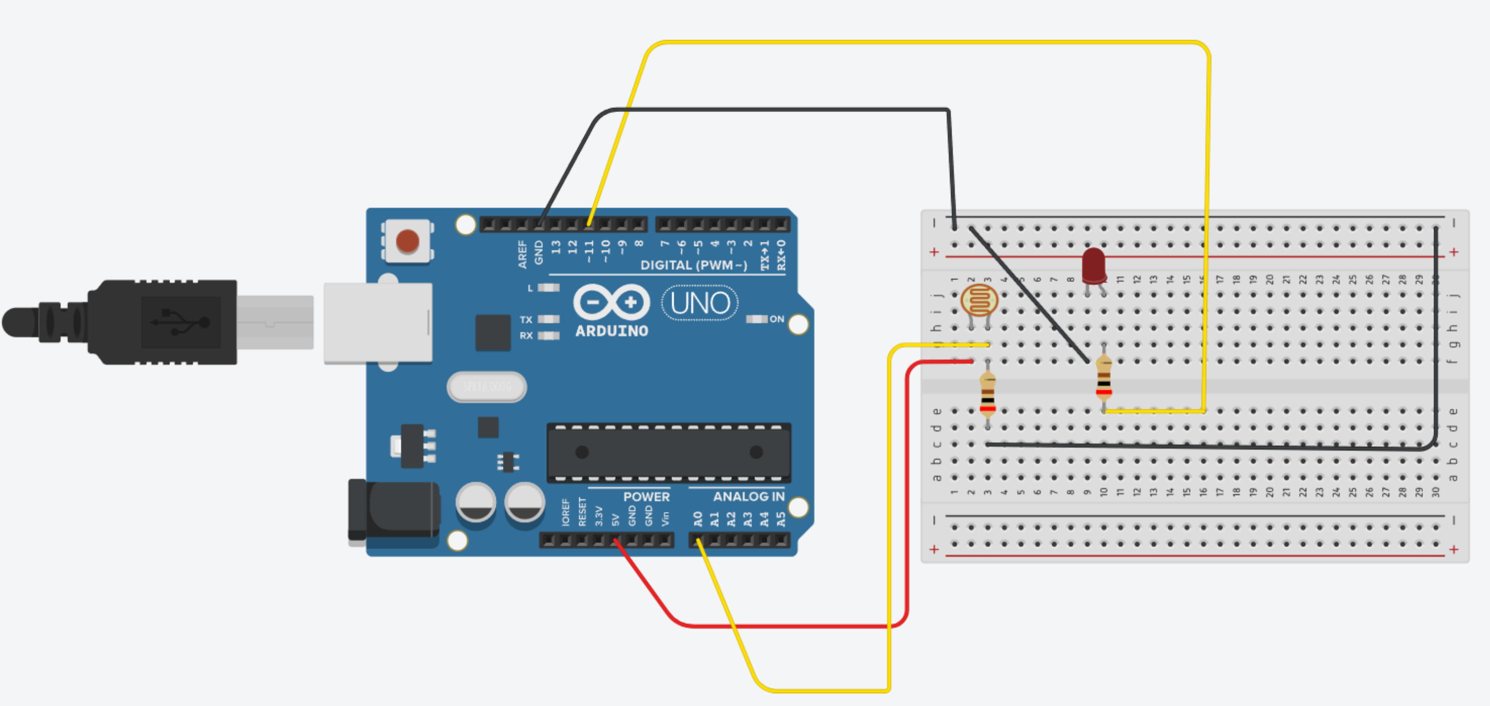
```
#include "DHT.h"  // Include the DHT sensor library

DHT dht;

void setup() {
  Serial.begin(9600);  // Start serial communication
  dht.setup(2);        // Set up DHT sensor on digital pin 2
}

void loop() {
  delay(2000);
  float humidity = dht.getHumidity();        // Read humidity
  float temperature = dht.getTemperature();  // Read temperature in Celsius

  // Print only temperature and humidity
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print(" %  Temperature: ");
  Serial.print(temperature);
  Serial.println(" °C");
}
```

SKETCH/CODE FOR DHT11

```
#include "DHT.h"  // Include the DHT sensor library to handle communication and data conversion

DHT dht;  // Create a DHT object to interact with the sensor

void setup()
{
  Serial.begin(9600);  // Start serial communication at 9600 baud rate
  Serial.println();
  Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");  // Print header

  dht.setup(2);  // Set up DHT sensor on digital pin 2 (data pin)
}

void loop()
{
  // Wait for the sensor's minimum sampling interval before reading again
  delay(dht.getMinimumSamplingPeriod());
  // Read humidity from the DHT sensor
  float humidity = dht.getHumidity();
  // Read temperature in Celsius
  float temperature = dht.getTemperature();
  // Print communication status (e.g., OK, checksum error, etc.)
  Serial.print(dht.getStatusString());
  Serial.print("\t");
  // Print humidity with 1 decimal place
  Serial.print(humidity, 1);
  Serial.print("\t\t");
  // Print temperature in Celsius with 1 decimal place
  Serial.print(temperature, 1);
  Serial.print("\t\t");
  // Convert Celsius to Fahrenheit and print with 1 decimal place
  Serial.println(dht.toFahrenheit(temperature), 1);
}
```

INTRODUCTION TO LDR

- **LDR** stands for **Light Dependent Resistor**, also known as a **photoresistor**.
- An **LDR is a type of resistor whose resistance decreases when the intensity of light increases.**
- In darkness, the LDR has high resistance (mega-ohms), so almost no current flows.
- In bright light, the resistance becomes very low, allowing more current to pass.



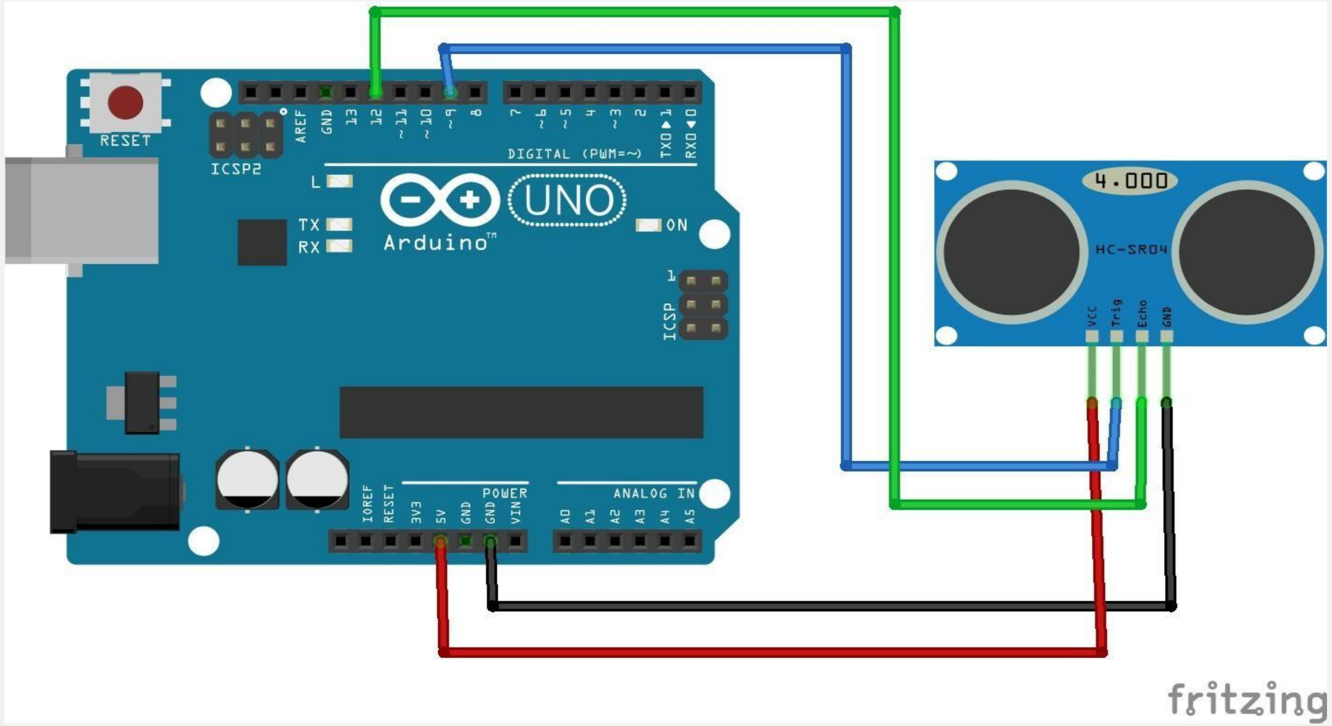

```
// Interfacing Arduino with LDR and LED

int value=0;
void setup()
{
  Serial.begin(9600);
  pinMode(11, OUTPUT); // LED, Actuator
  pinMode(A0, INPUT); // sensor
}
void loop()

{
  value= analogRead(A0);
  if(value<10)
  {
    digitalWrite(11, HIGH);
    Serial.println("Light ON"); // Night
    Serial.println(value);
  }
  else
  {
    digitalWrite(11, LOW);
    Serial.println("Light OFF"); // DAY
    Serial.println(value);
  }
}
```

INTRODUCTION TO ULTRASONIC SENSOR

- An **ultrasonic sensor** (like the **HC-SR04**) is used to **measure distance** using **sound waves**. It works by sending out an **ultrasonic pulse** and measuring how long it takes to **bounce back** after hitting an object.

1. The **trigger pin** sends a short ultrasonic pulse (usually 40kHz).
2. The **echo pin** receives the reflected signal.
3. The time between sending and receiving is used to calculate **distance** using this formula:

$$\text{Distance (cm)} = \frac{\text{Time } (\mu s) \times 0.0343}{2}$$



Common Uses:

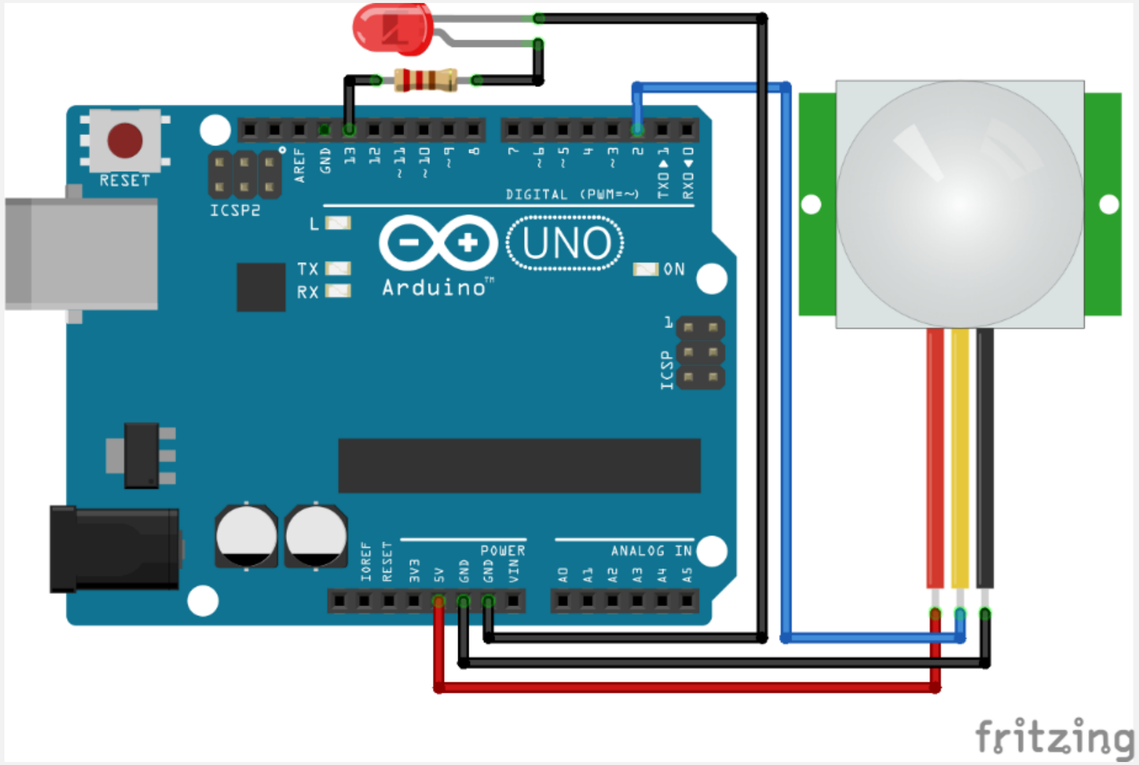
- Obstacle detection in robots
- Parking sensors
- Water level monitoring
- Smart garbage bins

SKETCH/CODE FOR ULTRASONIC SENSOR

```
#define trigPin 9
#define echoPin 10
void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop() {
  long duration;
  float distanceCm;
  // Send a 10 microsecond pulse to trigger
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Read echo time
  duration = pulseIn(echoPin, HIGH);
  // Calculate distance in cm
  distanceCm = duration * 0.0343 / 2;
  // Print the result
  Serial.print("Distance: ");
  Serial.print(distanceCm);
  Serial.println(" cm");
  delay(500);
}
```

INTRODUCTION TO PIR SENSOR

- **PIR** stands for **Passive Infrared Sensor**. It detects **motion** by sensing changes in infrared (IR) radiation, which is emitted by **warm bodies** like humans and animals.
- **How It Works:**
- When a human or animal moves within the sensor's range, the IR radiation changes.
- The PIR sensor detects this change and **sends a HIGH signal** (3.3V or 5V) to indicate motion.



Common Uses:

- Automatic lights
- Home security systems
- Intruder detection
- Smart alarms

SKETCH/CODE FOR PIR SENSOR

```
int pirPin = 2;       // PIR sensor output connected to digital pin 2
int ledPin = 13;      // Built-in LED pin
void setup() {
  pinMode(pirPin, INPUT);     // Set PIR pin as input
  pinMode(ledPin, OUTPUT);    // Set LED pin as output
  Serial.begin(9600);
}
void loop() {
  int motion = digitalRead(pirPin);  // Read PIR sensor output

  if (motion == HIGH) {
    Serial.println("Motion Detected!");
    digitalWrite(ledPin, HIGH);      // Turn on LED
  } else {
    Serial.println("No Motion");
    digitalWrite(ledPin, LOW);       // Turn off LED
  }
  delay(500);  // Small delay to stabilize readings
}
```

WHAT TO READ BEFORE EXAM?

- Workings
- Circuit
- Code
- Usage

"Don't make yourself down for anyone. Be the awkward, funny, intelligent, beautiful little weirdo that you are, don't hold back"

THANK YOU

Course Title: Embedded Systems and IoT

Course Code: CSE233

INTRODUCTION TO ARDUINO ACTUATORS

Prepared By:

Jul Jalal Al-Mamur Sayor (JMS)

Lecturer, Dept. of CSE, DIU

ARDUINO ACTUATORS

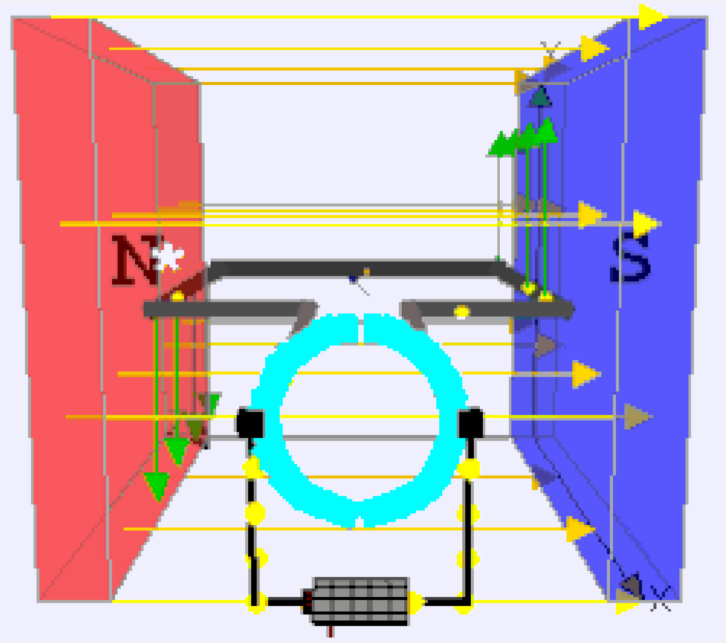
- An **actuator** is a device that **converts electrical signals into physical action**. It performs a **real-world task** like movement, rotation, sound, or light based on a control signal, usually from a microcontroller like an **Arduino**, **Raspberry Pi**, or an **IoT system**.
- **Actuators are output devices that "act" based on Arduino's decisions**, enabling interaction with the physical world.
- **Input → Arduino → Output**
  - ➢ **Input**: Sensors (e.g., motion, temperature, distance)
  - ➢ **Arduino**: Processes the input and makes decisions
  - ➢ **Actuator**: Executes the physical output (e.g., move, turn on, vibrate)

TYPES OF ARDUINO ACTUATORS

Actuator	Description / Working	Example Applications	Control Method
LEDs	Emit light when current flows through them.	Indicators, signal lights, status displays	Digital pin (HIGH/LOW)
DC Motors	Convert electrical energy into rotational motion.	Robotic cars, fans, propellers	H-bridge motor driver, transistor switch
Servo Motors	Allow precise control of angular position (0°–180°).	Robotic arms, camera pan-tilt systems	PWM (Servo library)
Stepper Motors	Rotate in small steps for high-precision movement.	CNC machines, 3D printers	Stepper driver with step/direction pins
Solenoids	Electromagnet creates linear motion (push/pull).	Locks, pinball machines, valves	Transistor or relay circuit
Relays	Electromechanical switch for controlling high-voltage loads using Arduino.	AC lights, pumps, fans	Digital signal via transistor trigger
Linear Actuators	Convert rotary motion into linear movement.	Window openers, lifting arms	Relay or motor driver
Piezo Buzzers / Vibrators	Generate sound or vibration through electrical signal.	Alarm systems, mobile haptics	Digital or PWM pin
Heating Elements	Generate heat instead of motion; used in thermal control.	Soldering iron, 3D printer beds	Controlled via MOSFET or relay

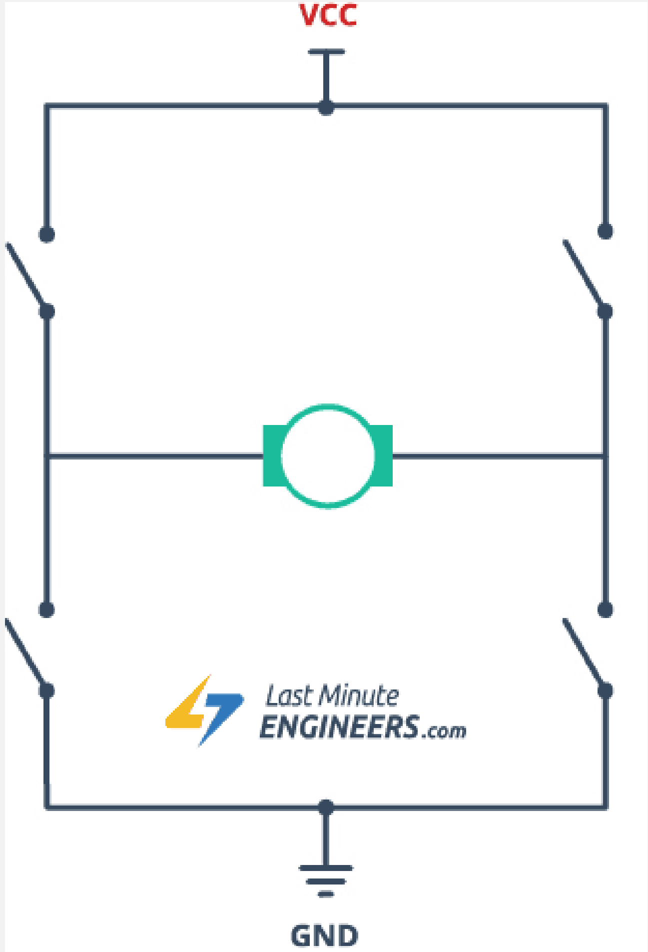
INTRODUCTION TO DC MOTORS

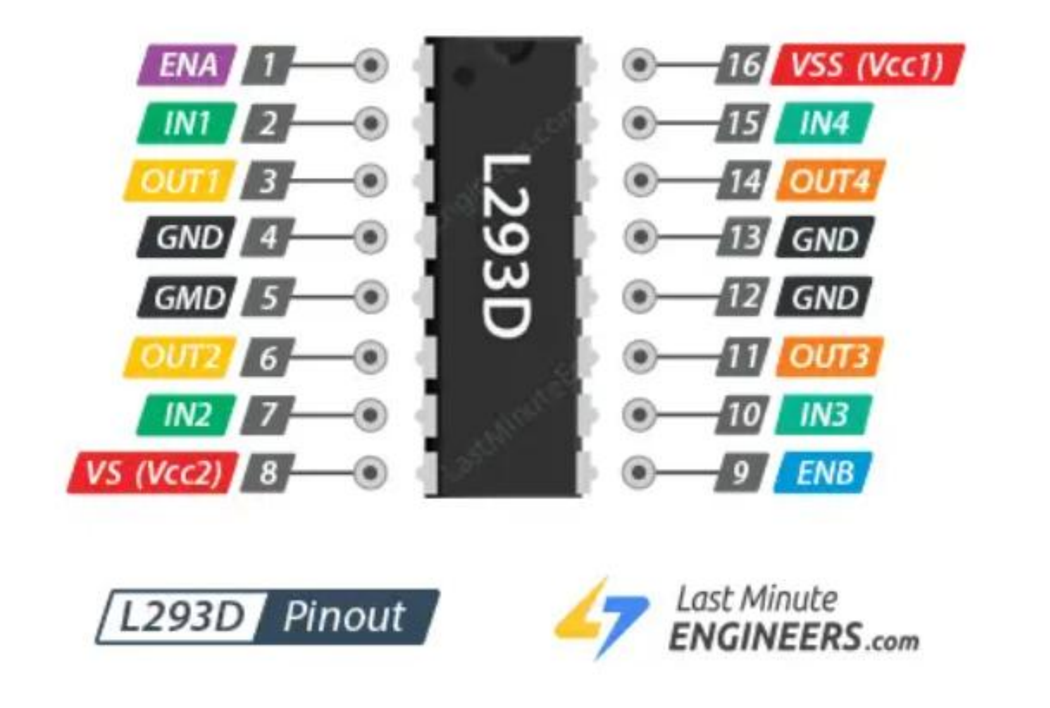
- A **DC (Direct Current) motor** is an actuator that converts **electrical energy into rotational mechanical motion**. It spins when voltage is applied across its terminals, and the **direction** and **speed** of rotation can be controlled using electronic circuits.
- **Why Use DC Motors?**
  - Simple and fast rotation
  - Commonly used in **robotics**, **fans**, **wheels**, **pumps**, etc.
  - Can rotate in **both directions** (forward & reverse)



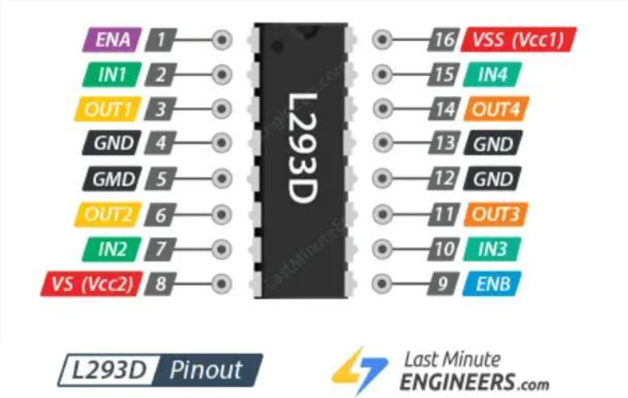
APPLICATION AND OPERATIONS OF L293D MOTOR CONTROLLER

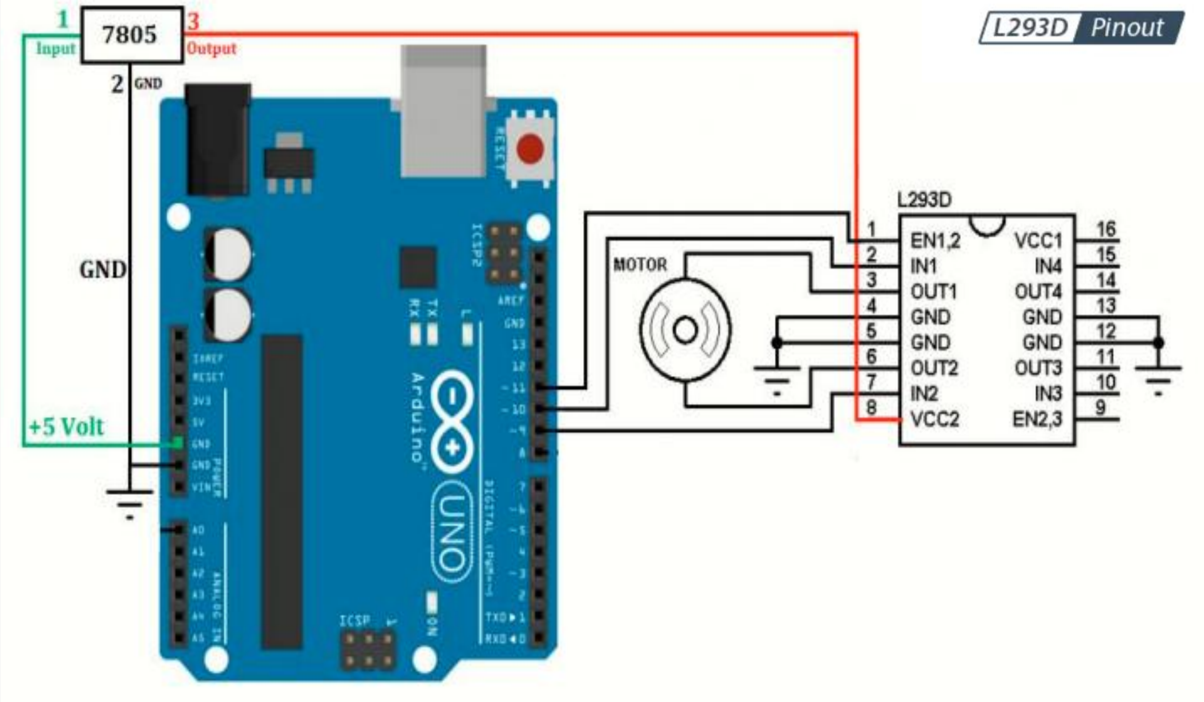
- The L293D is a dual-channel H-Bridge motor driver capable of driving a pair of DC motors or a single stepper motor. This means it can drive up to two motors individually which makes it ideal for building a two-wheeled robotic platform.

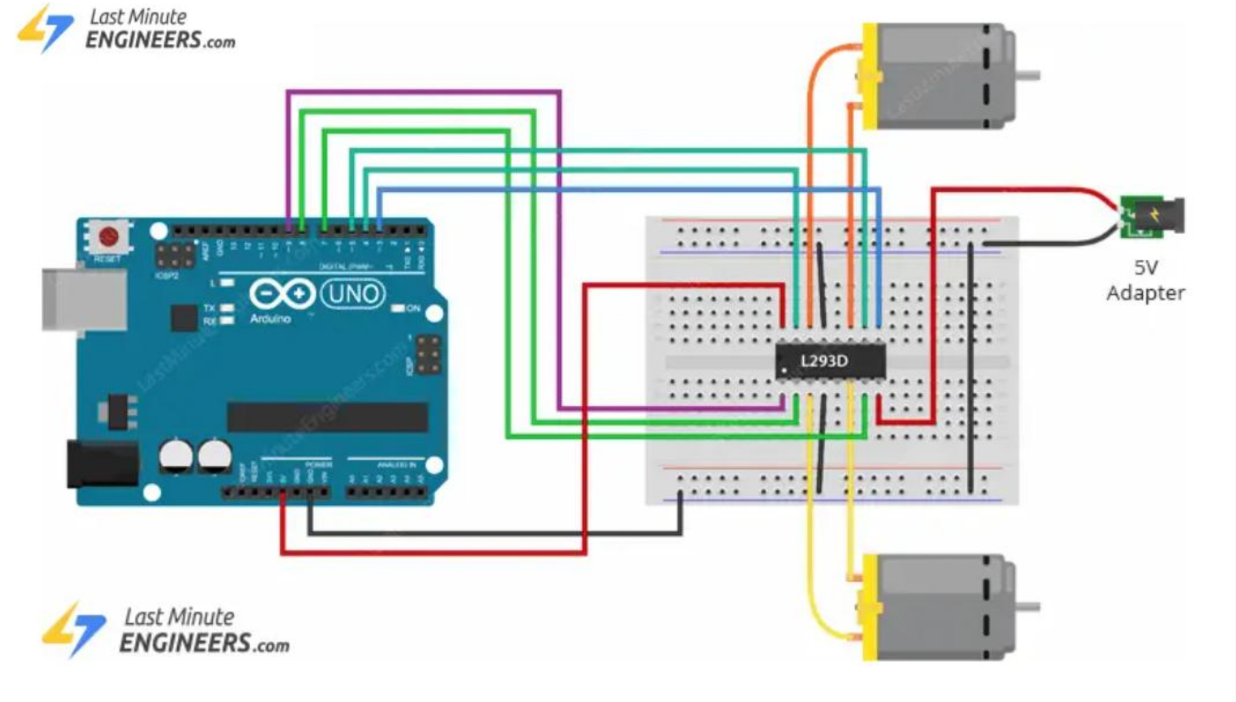




APPLICATION AND OPERATIONS OF L293D MOTOR CONTROLLER

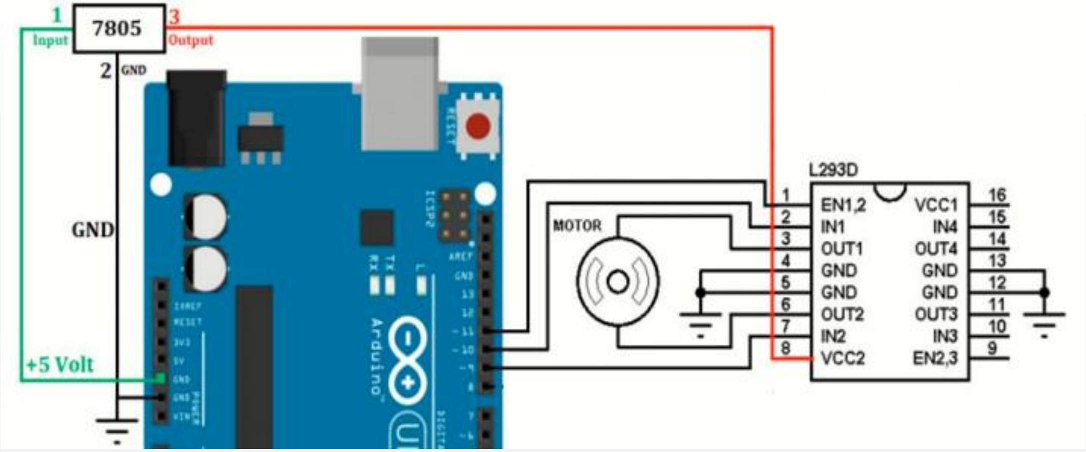






APPLICATION AND OPERATIONS OF L293D MOTOR CONTROLLER

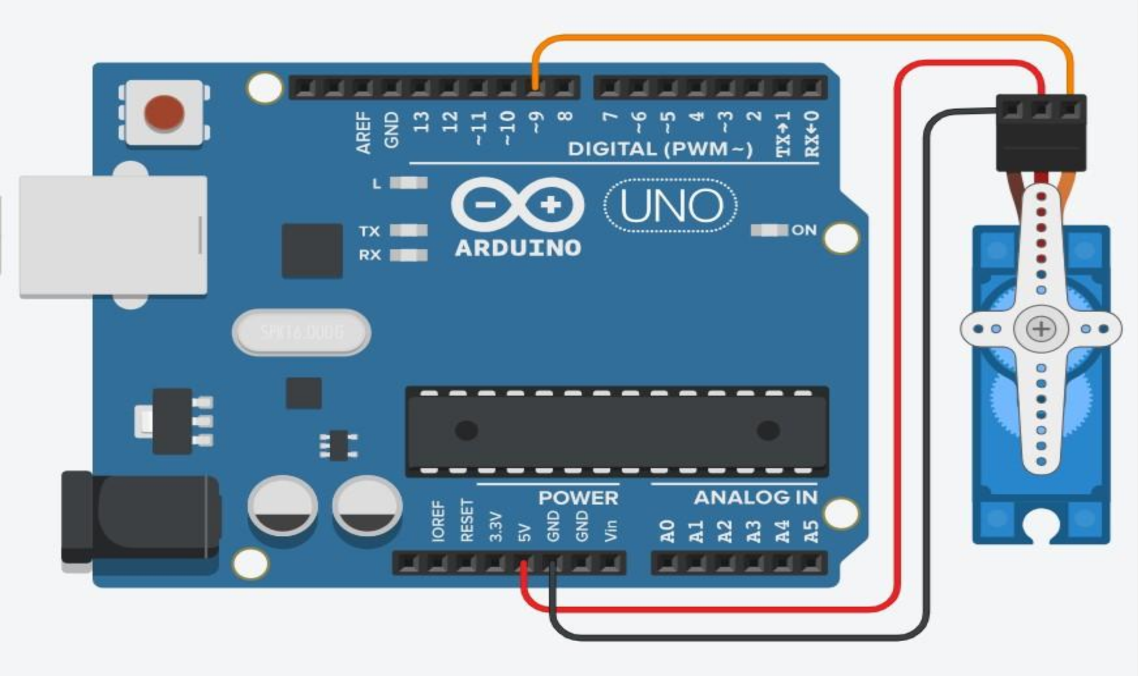
```
int enable1 = 11; // Assign pin 11 to Enable pin of L293D (used for speed control via PWM)
int in1 = 10;     // Assign pin 10 to Input 1 of L293D (controls motor direction)
int in2 = 9;      // Assign pin 9 to Input 2 of L293D (controls motor direction)

void setup()
{
  pinMode(enable1, OUTPUT); // Set enable pin as output
  pinMode(in1, OUTPUT);     // Set input pin 1 as output
  pinMode(in2, OUTPUT);     // Set input pin 2 as output
}

void loop()
{
  digitalWrite(enable1, HIGH); // Enable the motor (can also use analogWrite for speed control)
  digitalWrite(in1, HIGH);     // Set motor direction: forward
  digitalWrite(in2, LOW);      // in1 HIGH and in2 LOW → motor spins in one direction
}
```



INTRODUCTION TO SERVO MOTOR

- A **servo motor** is a type of actuator that allows **precise control of angular position**, typically within a range of **0° to 180°** (some allow 360° or more).
- **Key Features:**
  - Rotates to a specific angle and **holds** it
  - Operates using **Pulse Width Modulation (PWM)**
  - Contains an internal **motor + gear system + position sensor (potentiometer)**

INTRODUCTION TO SERVO MOTOR




```
#include <Servo.h>

Servo myServo;

void setup() {
  myServo.attach(9);
}

void loop() {
  for (int i = 0; i < 180; i++) {
    myServo.write(i);
    delay(10);
  }

  for (int i = 179; i > 0; i--) {
    myServo.write(i);
    delay(10);
  }
}
```

```
#include <Servo.h>

Servo myServo;

void setup() {
  myServo.attach(9);
}

void loop() {
  myServo.write(90);
  delay(1000);
  myServo.write(0);
  delay(1000);
}
```

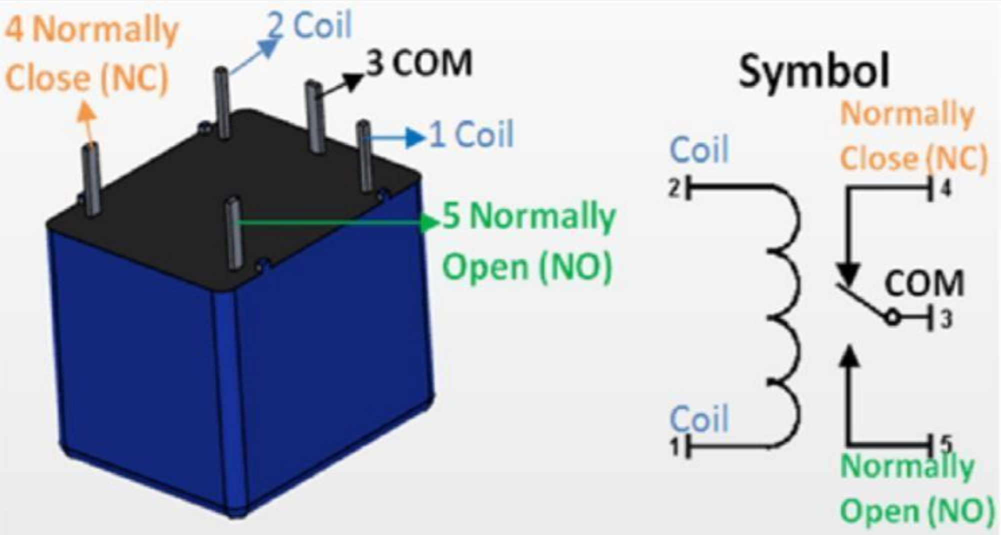
INTRODUCTION TO ELECTROMAGNETIC RELAY

What is an Electromagnetic Relay?

- An electromagnetic relay is an electromechanical switch that uses electromagnetism to open or close a high-power circuit using a low-power control signal — typically from a microcontroller like an Arduino.

How It Works:

- When a small current flows through the coil inside the relay, it creates a magnetic field.
- This magnetic field pulls a switch (armature) that closes or opens the connection in another circuit.
- It allows a low-power signal (e.g., from Arduino) to control high-power devices (e.g., fans, lights, pumps).

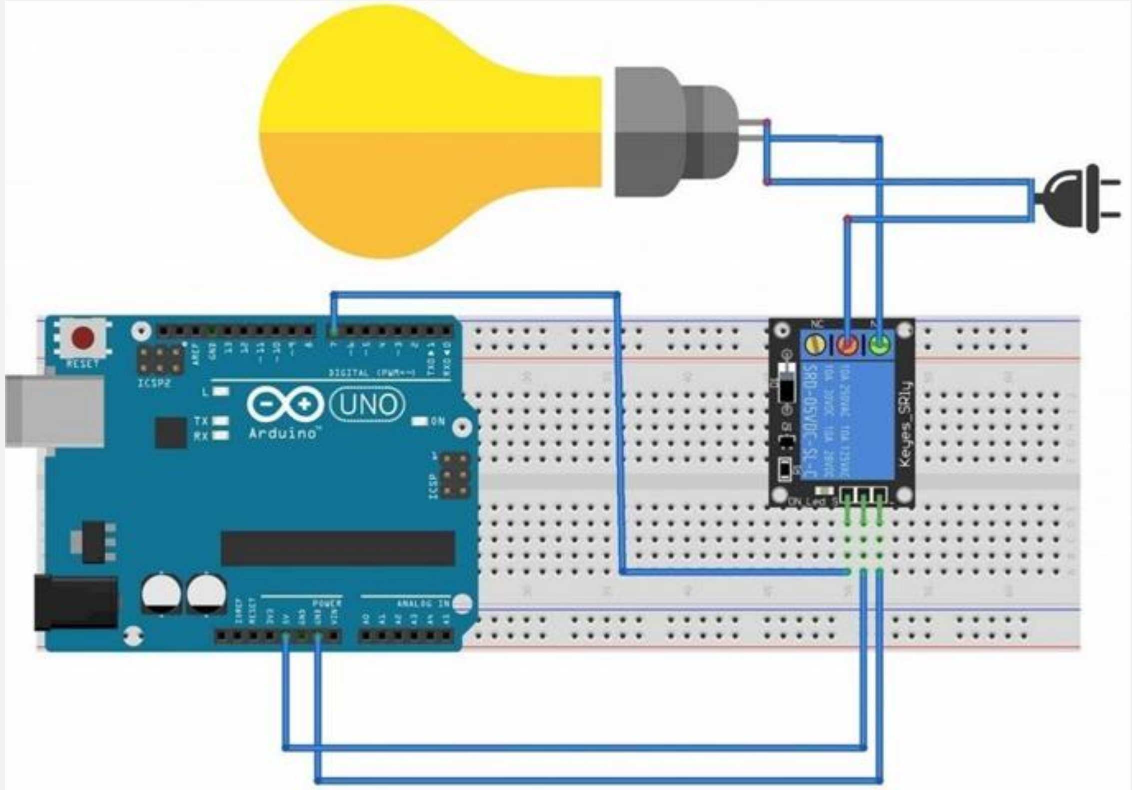


Applications of Relays:

- Turning **AC appliances** ON/OFF
- **Home automation**
- **Industrial control systems**
- **Water pumps, fans, lights, heaters**

INTRODUCTION TO ELECTROMAGNETIC RELAY

A **relay lets your Arduino control big devices** (like 220V fans or lights) using small signals. It acts like a **remote-controlled switch** powered by magnetism.



```
int RelayPin = 7;

void setup() {
	// Set RelayPin as an output pin
	pinMode(RelayPin, OUTPUT);
}

void loop() {
	// Let's turn on the relay...
	digitalWrite(RelayPin, LOW);
	delay(3000);

	// Let's turn off the relay...
	digitalWrite(RelayPin, HIGH);
	delay(3000);
}
```

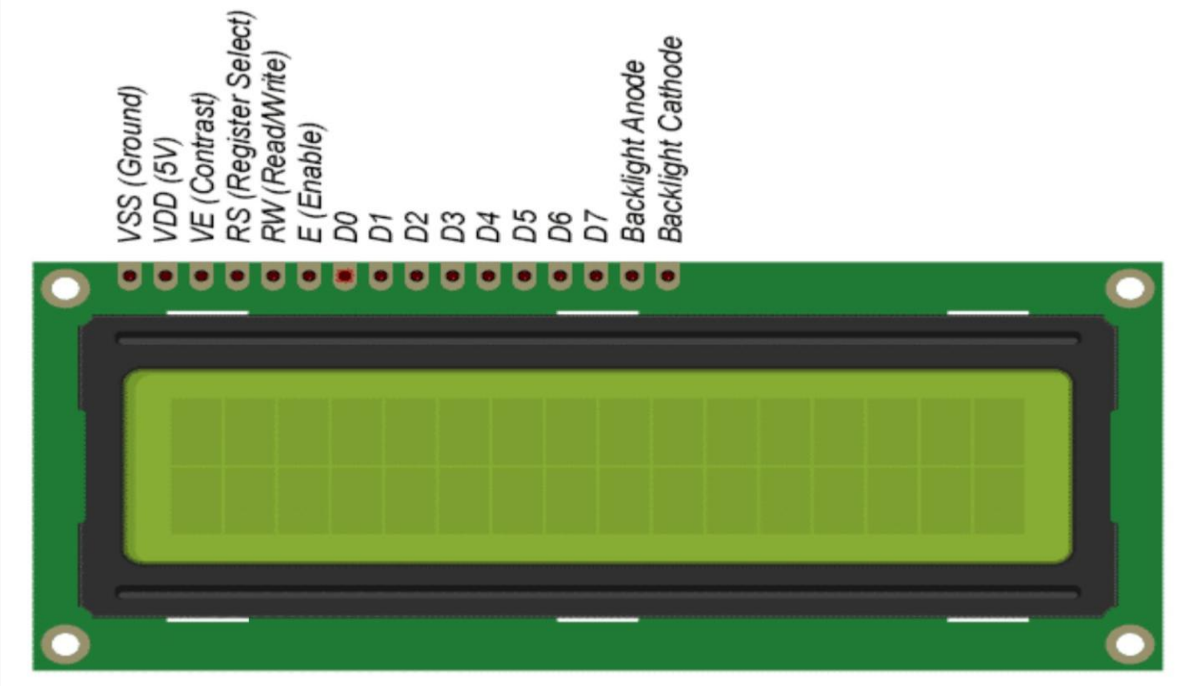
INTRODUCTION TO **16×2 LCD** (LIQUID CRYSTAL DISPLAY)

A **16×2 LCD** (Liquid Crystal Display) is a commonly used **character display module** that can show **16 characters per line**, across **2 lines** — making it ideal for Arduino projects where simple text output is needed.

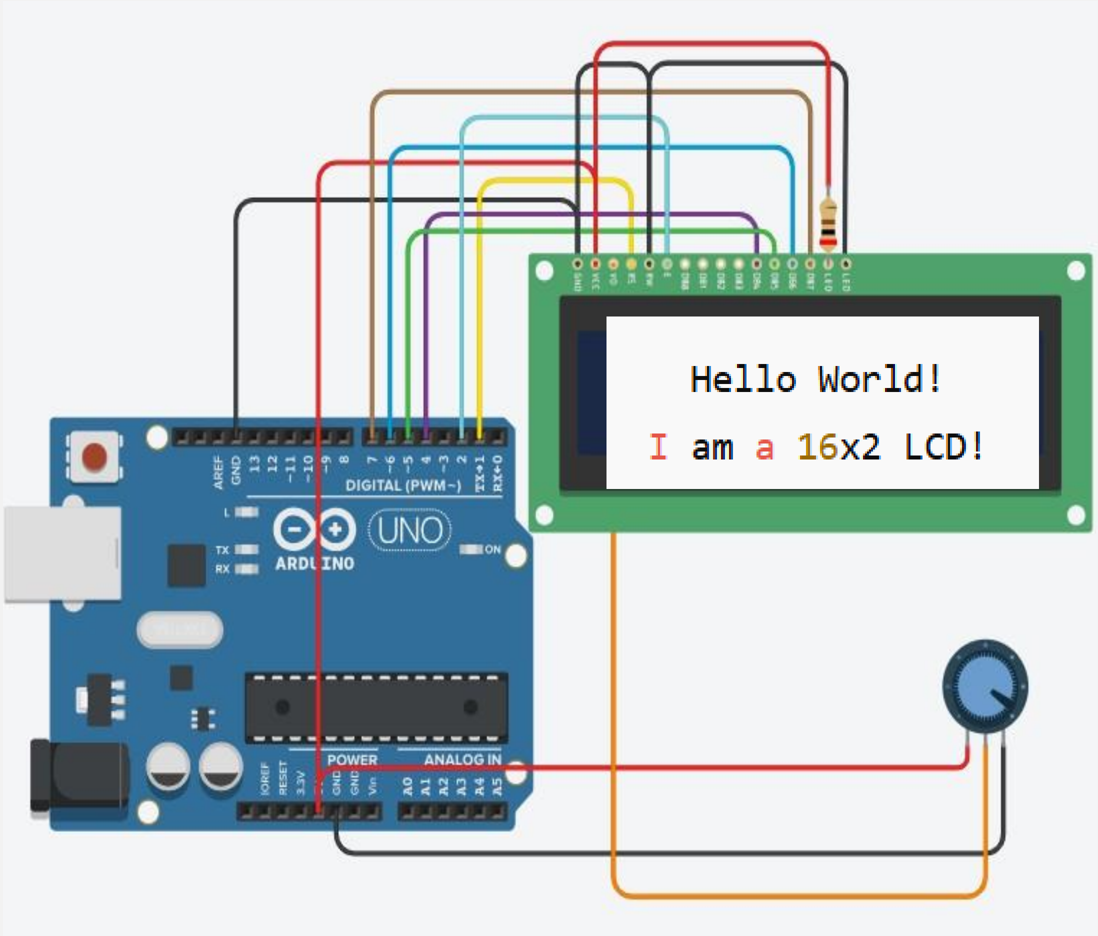
Key Features:

- Displays **letters, numbers, symbols**
- **16 characters × 2 rows**
- Uses **HD44780** controller (most common standard)
- Can be interfaced directly or via an **I2C adapter** (for fewer wires)

LCD Pin	Function	Connect To
1 (VSS)	Ground	Arduino **GND**
2 (VDD)	Power (5V)	Arduino **5V**
3 (V0)	Contrast	Middle pin of 10k pot
4 (RS)	Register Select	Arduino pin 1
5 (RW)	Read/Write	**GND** (Write only)
6 (E)	Enable	Arduino pin 2
11–14	D4 to D7	Arduino pins 4–7
15 (LED+)	Backlight +	5V (via 220Ω resistor)
16 (LED-)	Backlight –	GND



INTRODUCTION TO **16×2 LCD** (LIQUID CRYSTAL DISPLAY)




```
#include <LiquidCrystal.h>

LiquidCrystal lcd(1, 2, 4, 5, 6, 7);

void setup() {
  lcd.begin(16, 2);
  lcd.clear();
  lcd.setCursor(2, 0);
  lcd.print("Hello World!");
  lcd.setCursor(0, 1);
  lcd.print("I am a 16x2 LCD!");
}

void loop() {
  // do nothing
}
```

“Don’t make yourself down for anyone. Be the awkward, funny, intelligent, beautiful little weirdo that you are, don’t hold back”

THANK YOU

CEP Problem Using Arduino

CEP Problem Based on Embedded System and IoT

Q1.

How would you design a smart irrigation system for an agricultural field using appropriate microcontrollers, sensors, and actuators to monitor soil moisture levels, weather conditions, and plant growth? The system should automatically regulate water flow and trigger alerts when critical thresholds are crossed, ensuring optimal water usage and plant health. Develop a detailed schematic of the system and provide a comprehensive list of required components.

Points to Write

1. Overview of the Problem
2. System Requirement
3. Component Requirement
4. System Design
5. Software Design
    i. System Flow
    ii. Pseudo Code
6. Conclusion

CEP Problem Based on Embedded System and IoT

Solution:

Problem Overview:

The goal is to design a smart irrigation system that monitors soil moisture levels, weather conditions, and plant growth, automatically regulating water flow to ensure optimal water usage and plant health. The system will also trigger alerts when critical thresholds are crossed. Below is a detailed solution covering both hardware and software components.

CEP Problem Based on Embedded System and IoT

Solution:

1. System Requirements

The system should be capable of:

- **Monitoring soil moisture** to determine when watering is required.
- **Monitoring weather conditions** (rain, temperature, humidity) to adjust irrigation based on environmental conditions.
- **Monitoring plant growth** using relevant sensors to optimize watering based on plant maturity or health.
- **Controlling water flow** using solenoid valves to open or close irrigation lines.
- **Sending alerts** if soil moisture drops below a critical level or if weather conditions suggest extreme dryness or flooding.
- **Logging data** to cloud storage for analysis and long-term optimization of irrigation patterns.

CEP Problem Based on Embedded System and IoT

Solution:

2. Components Required

Microcontroller:

- **ESP32 or Arduino**: A microcontroller with built-in Wi-Fi for remote monitoring and control. The ESP32 is recommended for its better processing power, dual-core functionality, and built-in wireless capabilities.

Sensors:

- **Soil Moisture Sensor (Capacitive Type)**: Measures soil moisture to determine when to irrigate. Example: Capacitive Soil Moisture Sensor (YL-69).

- **Temperature and Humidity Sensor (DHT22 or SHT31)**: Monitors the ambient temperature and humidity to optimize irrigation based on weather conditions.

CEP Problem Based on Embedded System and IoT

Solution:

- **Rain Sensor**: Detects rainfall to prevent unnecessary watering.
- **Ultrasonic Distance Sensor** (optional): Monitors plant height/growth or could be replaced by more advanced sensors like NDVI (Normalized Difference Vegetation Index) sensors for plant health.
- **Weather Station** (optional): Integrates wind speed, rainfall, and UV sensors to optimize irrigation during extreme weather.

Actuators:

- **Solenoid Valve**: Controls the water flow in the irrigation system. It opens or closes based on signals from the microcontroller.
- **Relay Module**: Acts as a switch for high-power devices (solenoid valves or water pumps).

CEP Problem Based on Embedded System and IoT

Solution:

Power Supply:

- **12V or 24V DC Power Supply**: Powers the solenoid valves and the entire system. Solar power could be used for energy efficiency in remote fields.

- **5V Step-Down Converter**: To provide power to the microcontroller and sensors from the 12V/24V supply.

Communication:

- **Wi-Fi or LoRa Module**: Provides remote communication for data logging and alerting. ESP32 has built-in Wi-Fi, while LoRa is suitable for long-range applications.

Cloud Platform:

- **ThingSpeak/AWS IoT/MQTT Protocol**: For remote data logging and real-time alerting. The microcontroller sends sensor data to the cloud for analysis and triggers notifications via the cloud platform if required.

CEP Problem Based on Embedded System and IoT

Solution:

Power Supply:

- **12V or 24V DC Power Supply**: Powers the solenoid valves and the entire system. Solar power could be used for energy efficiency in remote fields.

- **5V Step-Down Converter**: To provide power to the microcontroller and sensors from the 12V/24V supply.

Communication:

- **Wi-Fi or LoRa Module**: Provides remote communication for data logging and alerting. ESP32 has built-in Wi-Fi, while LoRa is suitable for long-range applications.

Cloud Platform:

- **ThingSpeak/AWS IoT/MQTT Protocol**: For remote data logging and real-time alerting. The microcontroller sends sensor data to the cloud for analysis and triggers notifications via the cloud platform if required.

CEP Problem Based on Embedded System and IoT

Solution:

3. System Design

Main System Components:

1. **ESP32 Microcontroller** is the central brain that collects data from various sensors, processes it, and controls the solenoid valves via relays.

2. **Sensors**:

    - The **Soil Moisture Sensor** is placed near the plants and connected to the microcontroller's analog input to continuously monitor soil water content.

    - The **Temperature and Humidity Sensor (DHT22)** and **Rain Sensor** are placed in open-air locations to monitor environmental conditions.

CEP Problem Based on Embedded System and IoT

Solution:

1. **Water Flow Control**: The solenoid valves are connected to water pipes that distribute irrigation. The microcontroller, using a **Relay Module**, controls these valves based on sensor data.

2. **Power Supply**: A DC power supply unit is used to power the solenoid valves and sensors. The microcontroller is powered through a step-down converter.

CEP Problem Based on Embedded System and IoT

Solution:

4. Software Design

The software needs to perform the following tasks:

- **Read sensor data** for soil moisture, temperature, humidity, and rain status.
- **Process the sensor data** and decide when to water the plants.
- **Control solenoid valves** based on sensor thresholds and weather conditions.
- **Send alerts** if any parameters (like low soil moisture) exceed the predefined thresholds.
- **Log data to the cloud** for analysis and future optimization.

CEP Problem Based on Embedded System and IoT

Solution:

4.1 System Flow

1. **Sensor Data Collection**:
   - Soil moisture is measured in real-time.
   - Temperature, humidity, and rain sensors provide context for adjusting watering schedules based on environmental conditions.
2. **Decision Making**:
   - If the soil moisture falls below a certain threshold, and the weather sensors indicate no rain, the system opens the solenoid valves to irrigate the field.
   - If it's raining or soil moisture is adequate, irrigation is skipped.

CEP Problem Based on Embedded System and IoT

Solution:

3. Water Control:

- The relay module is triggered to open/close the solenoid valve for watering.
- Valves are automatically closed once soil moisture reaches a safe level.

4. Data Logging and Alerts:

- Data is logged to the cloud using an IoT platform (like ThingSpeak or AWS IoT).
- Alerts (via email, SMS, or app notification) are triggered if soil moisture or temperature reaches critical levels.

4.2 Pseudo Code

CEP Problem Based on Embedded System and IoT

Solution:

5. Conclusion

This smart irrigation system integrates sensors, actuators, and cloud connectivity to manage irrigation in an automated and efficient way. By monitoring soil moisture, weather conditions, and plant growth, the system optimizes water usage while ensuring plant health. Real-time alerts and data logging to the cloud make it robust, allowing remote monitoring and adjustments.

This system demonstrates how modern agricultural practices can be enhanced using IoT, real-time data, and automation to conserve water, improve crop yield, and reduce manual labor.

CEP Problem Based on Embedded System and IoT

Q2.

How would you design a smart energy management system for a commercial building using appropriate microcontrollers, sensors, and actuators to monitor and optimize energy consumption? The system should track occupancy, lighting, HVAC (Heating, Ventilation, and Air Conditioning) usage, and power loads from various appliances, automatically adjusting based on occupancy and time of day. It should also trigger alerts when energy usage exceeds certain limits or when equipment failures are detected. Develop a detailed schematic of the system and provide a comprehensive list of required components