جامعة الإسكندرية
ALEXANDRIA
UNIVERSITY
كليـة الحاسبات وعلـوم البيـانات

Internet of Things: Theory and Applications

Module 2: ESP32 based IoT Systems – Part A

Introduced by: Eng. Mohamed Hatem

Teaching Assistant at the Faculty of Computers and Data Science in Alexandria University

Graduate of Nanotechnology and Nano-electronics Engineering Program at the Zewail City of Sciences, Technology, and Innovation

Graduate Student Member at IEEE

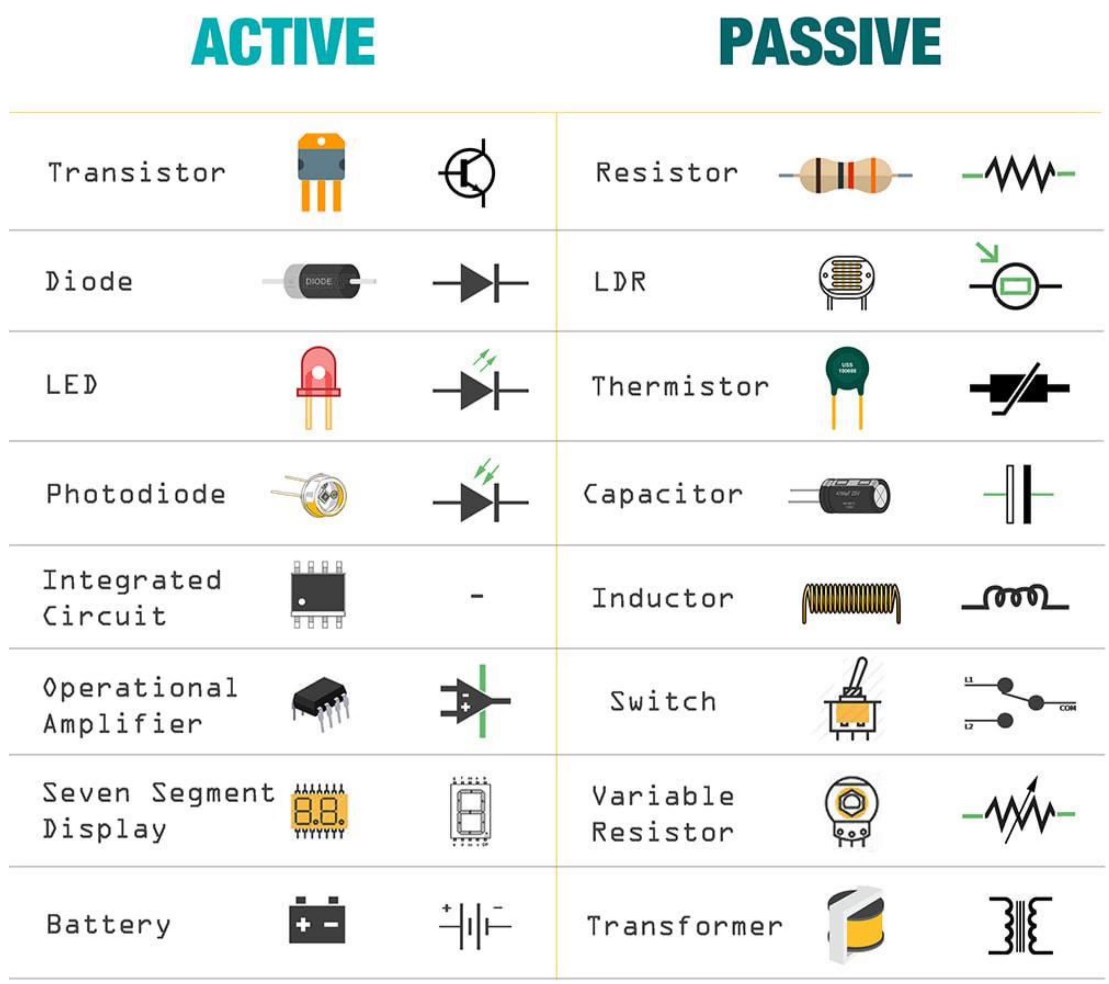
Session Outline:

- Section 0: Introduction to Electrical, Electronics & Communications Engineering
- Section 1: Introduction to Embedded Systems
- Section 2: ESP32 based IoT & Embedded Systems
- Section 3: ESP32 Basic Practice Projects
- Section 4: Assignments

Introduction to Electrical, Electronics & Communications Engineering

Internet of Things: Theory and Applications

Active & Passive Components in Circuits and Electronics



Ohms Law

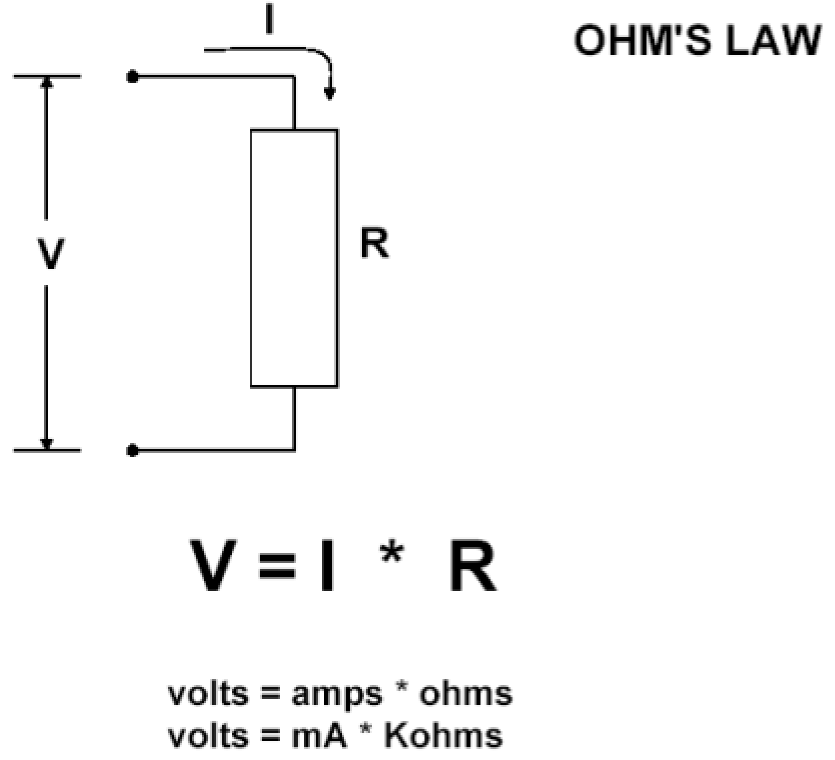
- Ohms Law: Ohm's law states that the current flowing through a conductor between two points is directly proportic to the voltage across the two points, provided the temperature and other physical conditions remain constant.

Equation:

V = I * R

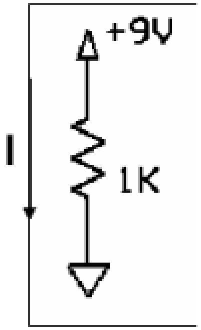
Voltage (volts) =

Current (amperes) * resistance (ohms)

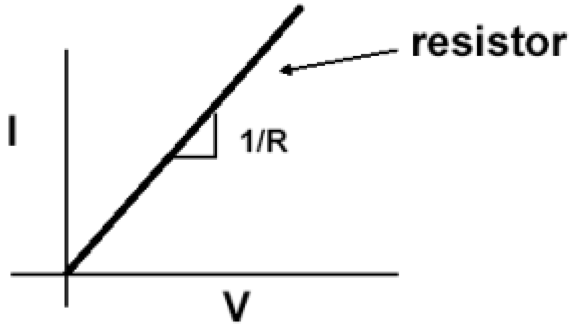


Ohms Law Verification

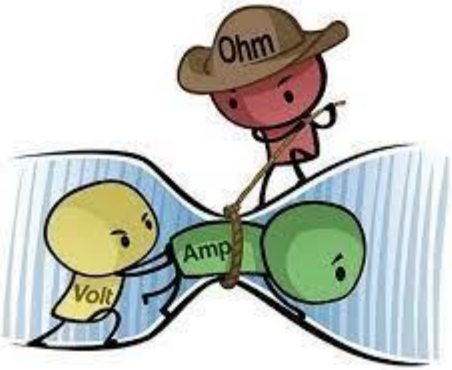
- To verify ohms law, you need to get a power source such as a battery and a resistance and measure using the AVO meter the voltage across the resistance and the current passing through it then divide them by each other and compare the calculated resistance and the actual resistance.



$$I = V/R = 9V/1K = 9mA = .009A$$



Volt & Current

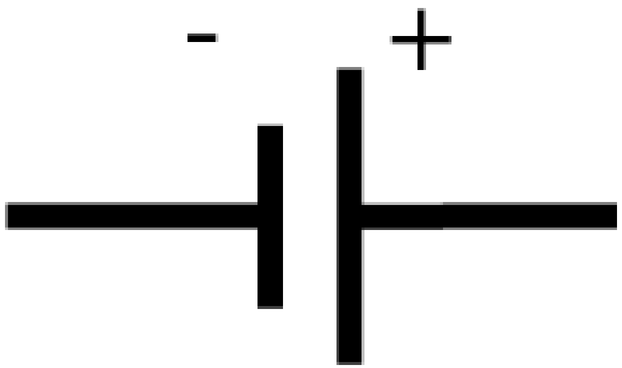


Difference Between Voltage and Current

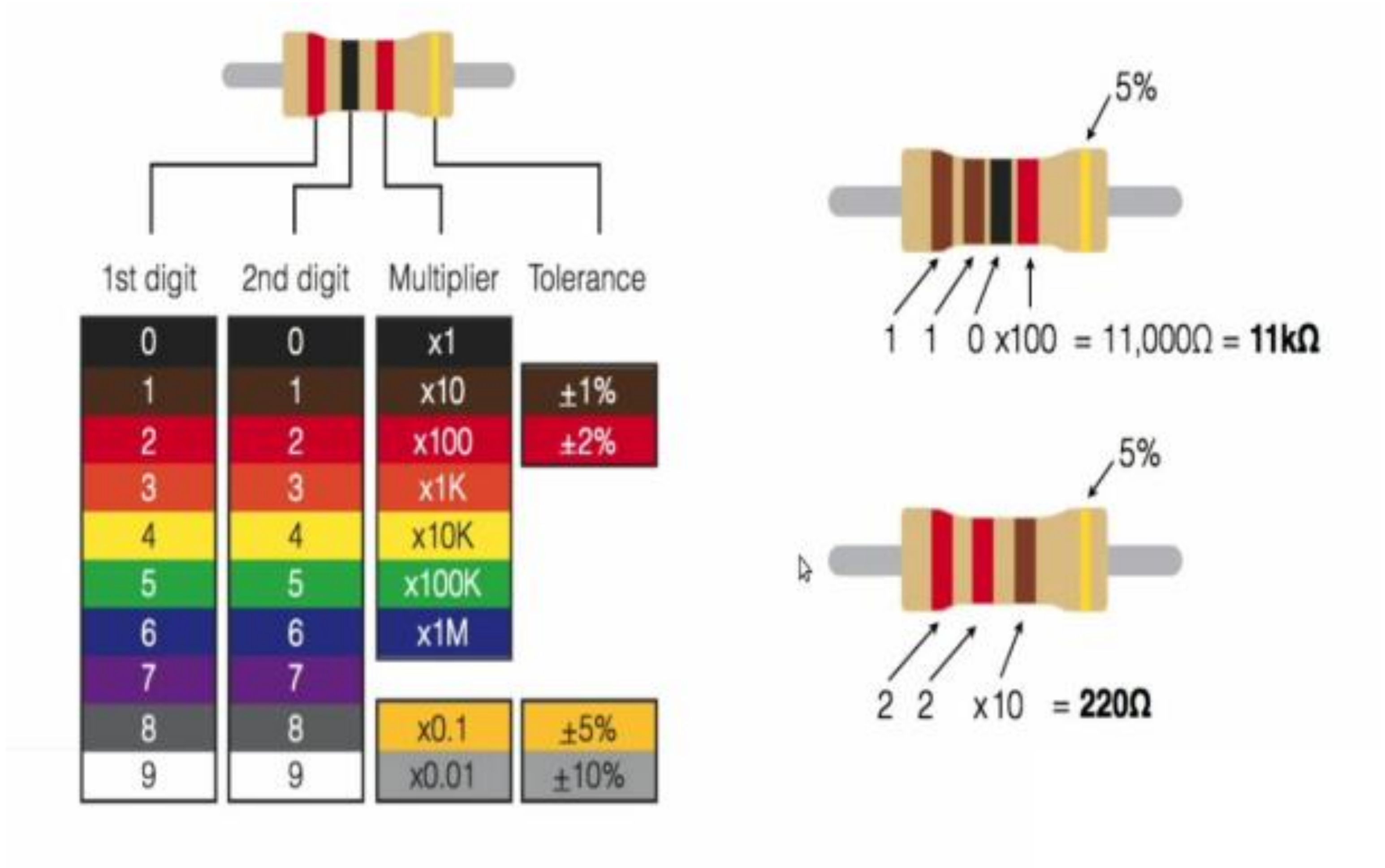
Voltage
Voltage is the potential difference between two points in an electric field.
Symbol of voltage is " V "
SI Unit of voltage is volt.
Its formula is V= W/q
Voltage remains same in Parallel combination.

Current
Current is the rate of flow of charges between two points caused by voltage.
Symbol of current is " I ".
SI Unit of current is Ampere.
Its formula is I = Q/t
Current remain s same in series circuit.

Power Source: Battery & Battery Cap



Get a Resistance and Read its Actual Value



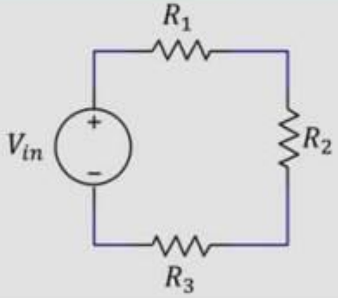
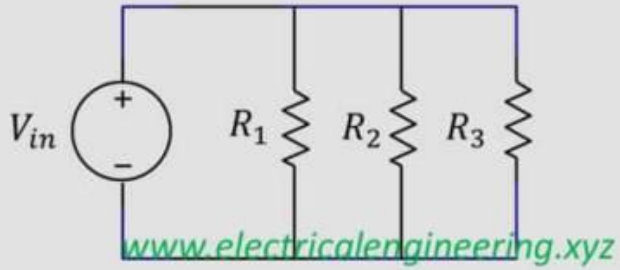
Common Resistors for in Embedded Systems

Resistor	Colors	Rating
	red, red, brown	2, 2, x10 = **220Ω**
	green, blue, brown	5, 6, x10 = **560Ω**
	brown, black, red	1, 0, x100 = 1,000Ω = **1kΩ**
	yellow, purple, red	4, 7, x100 = 4,700Ω = **4.7kΩ**

Common Resistors for in Embedded Systems

Resistor	Colors	Rating
	brown, black, orange	1, 0, x1,000 = 10,000Ω = **10kΩ**
	brown, black, green	1, 0, x100,000 =1,000,000Ω = **1MΩ**
	brown, black, blue	1, 0, x1,000,000 =10,000,000Ω = **10MΩ**

Series and Parallel Connections

	Series	Parallel
How it looks		
Voltage	$V_{in} = V_1 + V_2 + V_3$	$V_{in} = V_1 = V_2 = V_3$
Current	$I_{series} = I_1 = I_2 = I_3$	$I_{in} = I_1 + I_2 + I_3$
Resistance	$R_{eq} = R_1 + R_2 + R_3$	$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$
Features	If one components burns current becomes inactive	If one component burns current stops only through that branch rest part works fine

Pull up and Pull Down Resistors

On most microcontrollers, pull-up and pull-down resistors are commonly used to ensure reliable and stable digital input signals. They are used in conjunction with digital pins to set a default logic level when the pin is not actively driven.

Pull-up Resistors

- Pull-up resistors: A pull-up resistor is connected between a digital pin and the voltage supply (usually Vcc or 3.3V). It "pulls up" the voltage level of the pin when it is not actively driven, creating a default HIGH logic level.
- When the pin is in an open or high-impedance state (not actively driven), the pull-up resistor provides a path to Vcc, maintaining the pin at a HIGH logic level. When the pin is actively driven LOW, it overcomes the pull-up resistor's resistance, bringing the pin to a LOW logic level.
- Pull-up resistors are often used in scenarios where a button or switch is connected to a digital pin. When the button is not pressed, the pull-up resistor keeps the pin at a HIGH logic level. Pressing the button connects the pin to ground, overriding the pull-up resistor and bringing the pin to a LOW logic level.

Pull-down Resistors

- Pull-down resistors: A pull-down resistor is connected between a digital pin and ground (GND). It "pulls down" the voltage level of the pin when it is not actively driven, creating a default LOW logic level.
- Similar to pull-up resistors, pull-down resistors provide a path for the current to flow when the pin is in an open or high-impedance state. This ensures that the pin is held at a LOW logic level. When the pin is actively driven HIGH, it overcomes the pull-down resistor's resistance, bringing the pin to a HIGH logic level.
- Pull-down resistors are commonly used in scenarios where a switch or sensor is connected to a digital pin. When the switch is not active or the sensor is not triggering, the pull-down resistor keeps the pin at a LOW logic level. When the switch is closed or the sensor is triggered, it connects the pin to Vcc, overriding the pull-down resistor and bringing the pin to a HIGH logic level.

Pull up and Pull Down Resistors

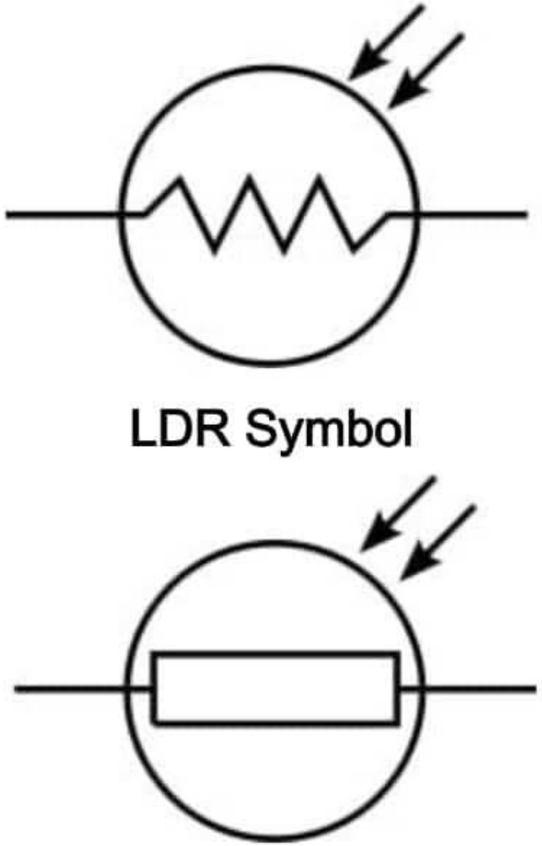
<u>**Calculating actual values for pull-up resistors:**</u>

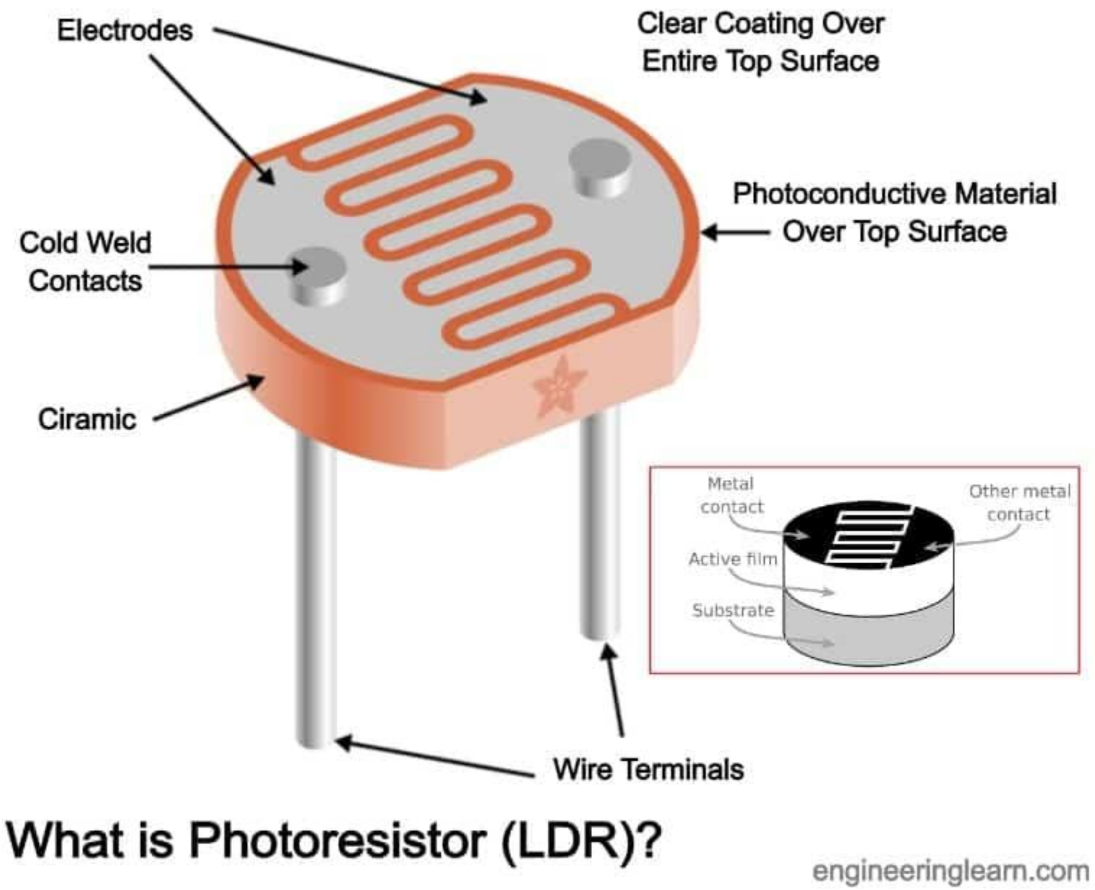
- For pull up resistors, voltage refers to the source voltage minus minimum voltage accepted as high. While the current refers to the maximum current sunk by the logic pins.
- Formula for calculating actual value for pull-up resistors:
- R pull-up = (V supply – $V_{H(min)}$) / I_{sink}

<u>**Calculating actual values for pull-down resistors:**</u>

- For pull-down resistors, there's a slight change to the formula, though it's still based on Ohms law. Voltage will know be referred to logic Low, while the current refers to the maximum current sourced by the digital pin.
- Formula for calculating actual value for pull-down resistors:
- R pull-down = ($V_{L(max)}$ – 0) / I_{source}

Light Dependent Resistor (LDR)





Light Dependent Resistor (LDR)

- **Introduction:**
  - The LDR sensor, also known as a light-dependent resistor or photocell, is a passive electronic component that changes its resistance based on the amount of light falling on its surface.
  - It is commonly used to detect and measure light intensity in various applications.
- **Working Principle:**
  - The LDR sensor is made of a semiconductor material whose resistance decreases with an increase in the intensity of light falling on it.
  - When exposed to light, the photons in the incident light cause the semiconductor material's electrons to move, leading to a decrease in resistance.
  - In darkness or low-light conditions, the resistance of the LDR sensor is high, and it decreases as the light intensity increases.

Light Dependent Resistor (LDR)

- **Features and Applications:**
  - **Features:**
    - Wide range of resistance values depending on the type of LDR sensor.
    - Non-polarized component, meaning it can be connected in any direction.
    - Low-cost and easily available.
  - **Applications:**
    - Automatic lighting control systems.
    - Light-sensitive alarms and security systems.
    - Day-night sensing applications.
    - Camera exposure control.
    - Solar tracking systems.

Light Dependent Resistor (LDR)

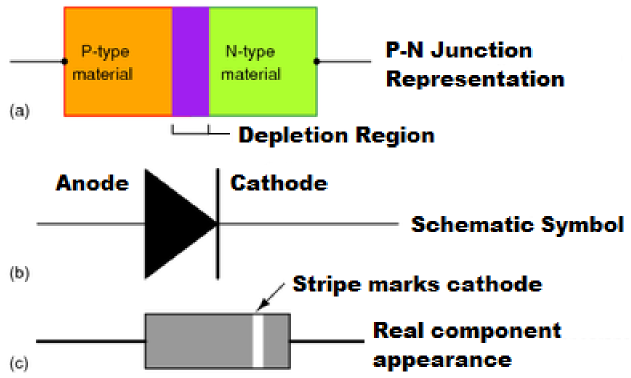
- **Connection:**
  - Connect one terminal of the LDR sensor to the positive supply voltage.
  - Connect the other terminal of the LDR sensor to one end of a resistor.
  - Connect the other end of the resistor to the ground (GND) or negative supply voltage.
- **Voltage Divider Circuit:**
  - The LDR sensor and the resistor form a voltage divider circuit.
  - The output voltage from the junction between the LDR sensor and the resistor varies based on the light intensity.
- **Readings and Calibration:**
  - Measure the output voltage using an analog-to-digital converter (ADC) or analog input of a microcontroller.
  - Calibrate the LDR sensor by mapping the output voltage to the corresponding light intensity using reference measurements.

Light Dependent Resistor (LDR)

- **Threshold Detection:**
  - Set threshold values for different light intensity levels.
  - Use the readings from the LDR sensor to determine if the light intensity crosses the set thresholds.
  - Trigger actions or control other components based on the light intensity detected.
- **Considerations:**
  - Protect the LDR sensor from direct exposure to intense light sources that may damage it.
  - Shield the sensor from ambient light to prevent false readings.
  - Account for response time and sensitivity variations between different LDR sensor models.
- *Note: The LDR sensor's resistance and sensitivity may vary depending on factors such as the wavelength of light and environmental conditions. It is essential to refer to the sensor's datasheet for specific details and characteristics.*

Semiconductor Diodes

- **Definition:** A diode is a two-terminal electronic component made of a semiconductor material, typically silicon or germanium. It allows current to flow in only one direction while blocking it in the opposite direction.
- **P-N Junction:** The basic structure of a semiconductor diode consists of a P-N junction formed by joining a P-type (positive) semiconductor region and an N-type (negative) semiconductor region. The P-region has an excess of holes, while the N-region has an excess of electrons.
- **Forward Bias:** When a positive voltage is applied to the P-region and a negative voltage to the N-region, the diode is said to be forward-biased. In this state, the diode allows current to flow easily from the P-side (anode) to the N-side (cathode) with a low voltage drop.
- **Reverse Bias:** When a positive voltage is applied to the N-region and a negative voltage to the P-region, the diode is reverse-biased. In this state, the diode blocks the current flow, acting as an insulator.



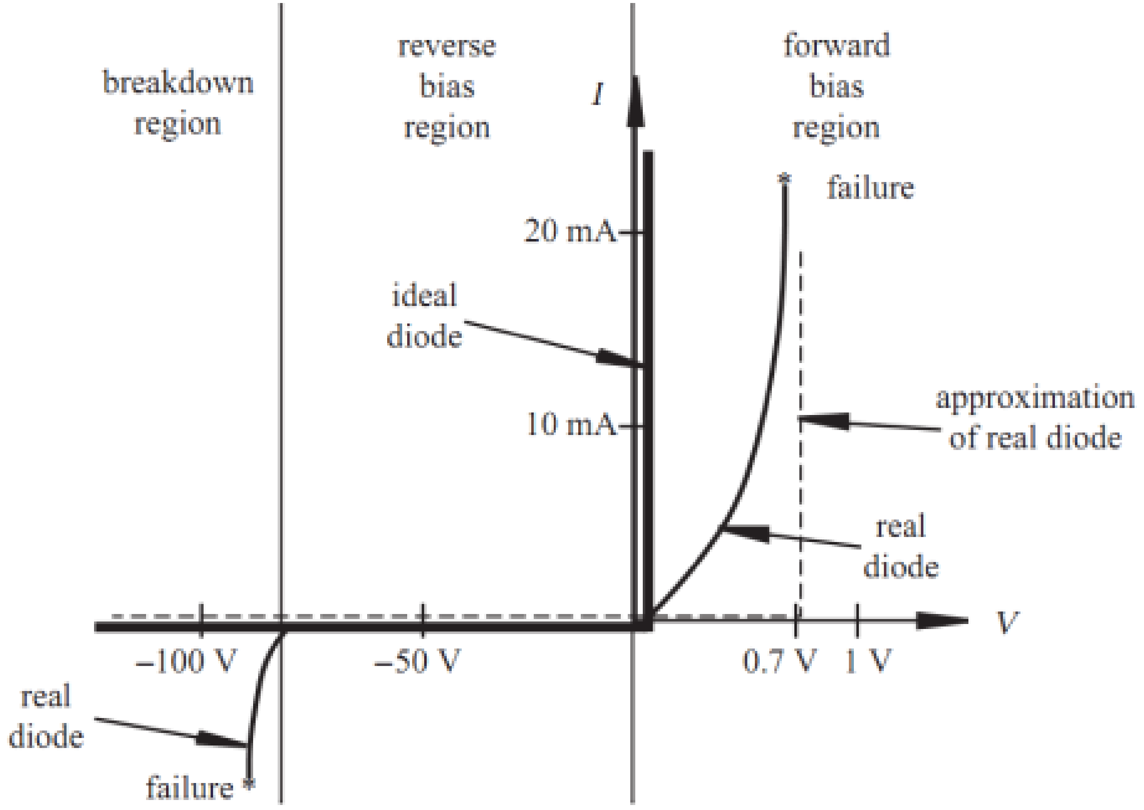
Semiconductor Diodes

- **Diode Characteristics:**
  - Forward Voltage Drop: When a diode is forward-biased, it exhibits a voltage drop typically around 0.7 volts for silicon diodes and 0.3 volts for germanium diodes.
  - Reverse Leakage Current: Even in reverse bias, a small amount of current, called reverse leakage current, can flow through the diode.
- **Applications:**
  - Rectification: Diodes are commonly used as rectifiers to convert alternating current (AC) to direct current (DC) by allowing current flow in one direction.
  - Signal Demodulation: Diodes are used in radio and communication systems for demodulating signals.
  - Voltage Regulation: Zener diodes, a special type of diode, are used for voltage regulation and protection against voltage spikes.
  - Switching: Diodes can be used as electronic switches in digital circuits to control the flow of current.

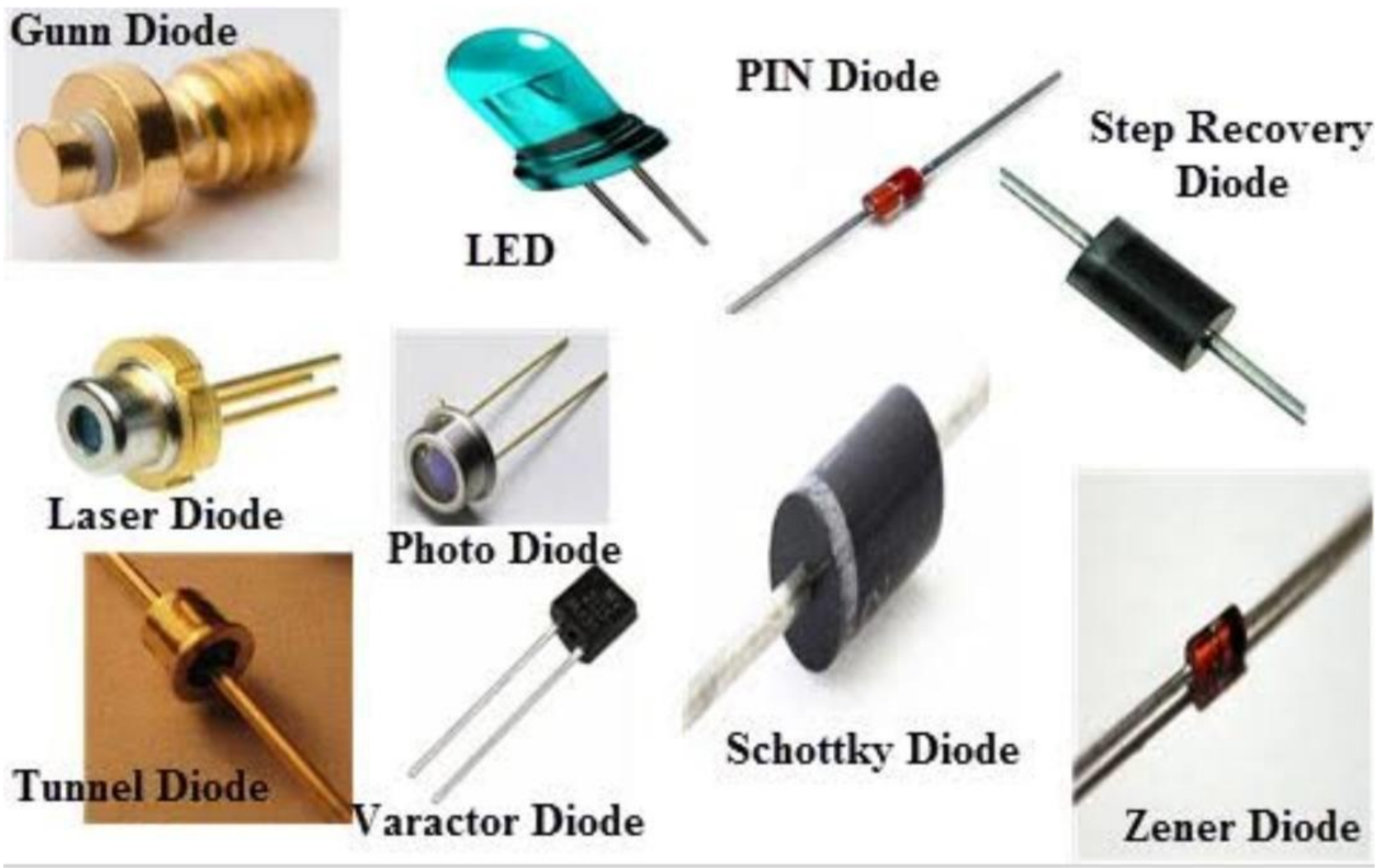
Semiconductor Diodes

- **Types of Diodes:**
  - Schottky Diode: A diode with a low forward voltage drop and fast switching characteristics.
  - Light-Emitting Diode (LED): A diode that emits light when forward-biased.
  - Photodiode: A diode that converts light energy into electrical current.
  - Varactor Diode: A diode whose capacitance changes with applied voltage, used in tuning and frequency control applications.
  - Avalanche Diode: A diode designed to operate in the avalanche breakdown region, used in voltage clamping and protection circuits.

Diode Characteristics

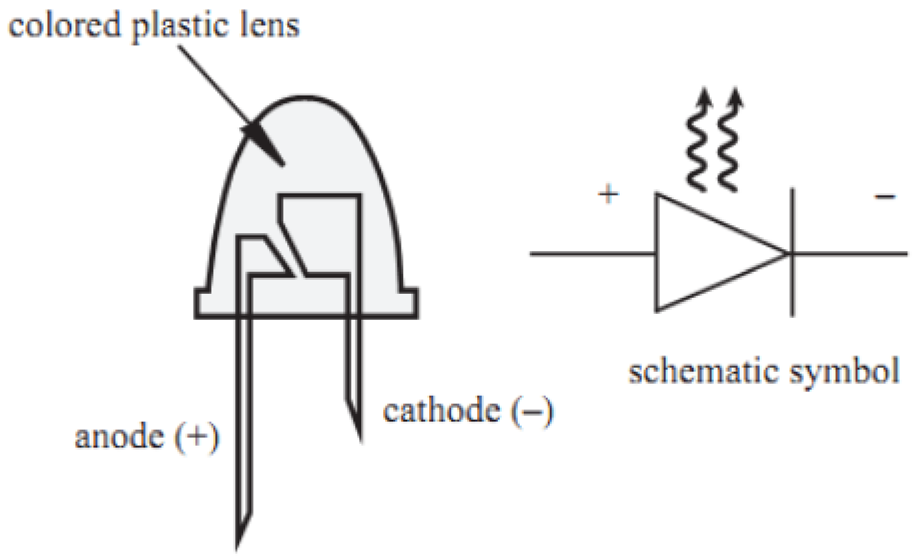


Types of Diodes



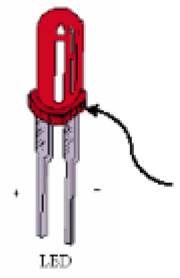
Light Emitting Diode (LED)

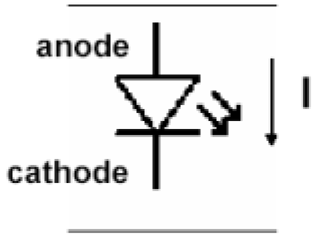
- The working principle of a light-emitting diode (LED) is based on **a phenomenon called electroluminescence.** When a forward voltage is applied to an LED, current flows through the semiconductor material, which consists of layers of different materials with varying electron properties.
- As the current passes through the LED**, it encounters a junction where electrons from the higher energy conduction band recombine with electron holes from the lower energy valence band.** This recombination process releases energy in the form of photons, which are the particles of light.

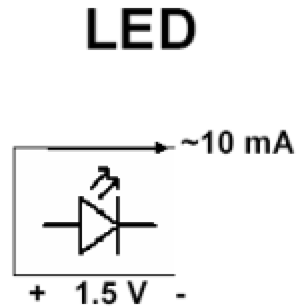


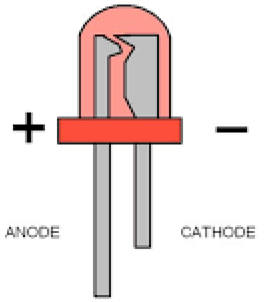
Light Emitting Diode (LED)

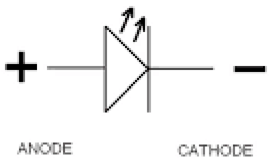
- The specific materials used in the LED determine the color of the emitted light. Different semiconductor materials, such as gallium arsenide (GaAs) or gallium nitride (GaN), result in different colors of light, such as red, green, blue, or white.
- The emitted light **<u>is highly directional</u>**, thanks to the LED's design and packaging, which helps focus the light in a specific direction. The intensity of the light is directly related to the current passing through the LED, allowing for brightness control by adjusting the current.
- *Overall, the working principle of an LED involves the conversion of electrical energy into light energy through the recombination of electrons and electron holes in the semiconductor material.*











Practice Circuit

Select the Right Resistor for the LED in a 5v DC Power Supply

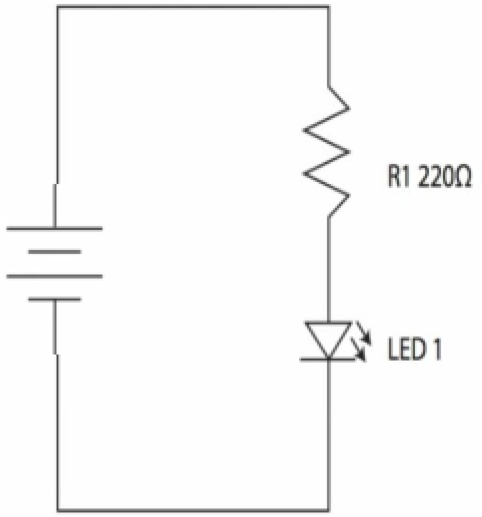
5-volt circuit

Our LEDs use about 2 volts each

20 mA

- **R = V ÷ I**

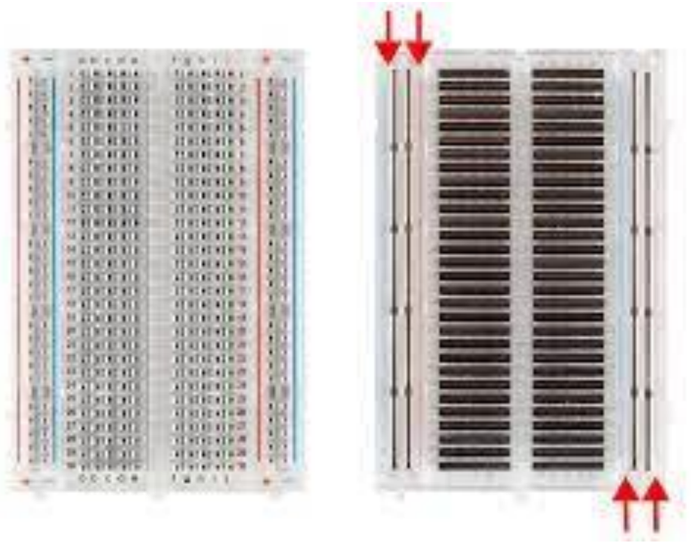
$$R = \frac{3V}{0.02A} = 150\Omega$$



NB: The cathode of an LED is the terminal through which current flows into the device, while the anode is the terminal through which current exits the device.

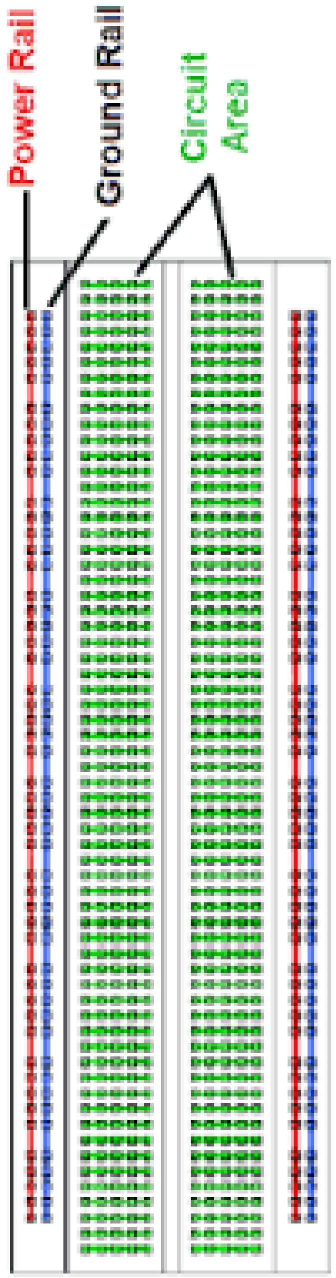
Solderless Breadboard

- **<u>Introduction:</u>**
  - A solderless breadboard is a device used for prototyping and experimenting with electronic circuits.
  - It provides a convenient way to connect electronic components without the need for soldering.
- **<u>History:</u>**
  - The solderless breadboard was first developed in the 1970s by Ronald J. Portugal, an engineer at E & MP Electronics.
  - It was designed as a tool to simplify circuit prototyping and testing for engineers and hobbyists.
- **<u>Working Principle:</u>**
  - The solderless breadboard consists of a grid of interconnected metal clips or sockets.
  - These clips allow electronic components to be inserted and connected without soldering.
  - The clips provide electrical continuity, allowing components to be easily interconnected to form circuits.

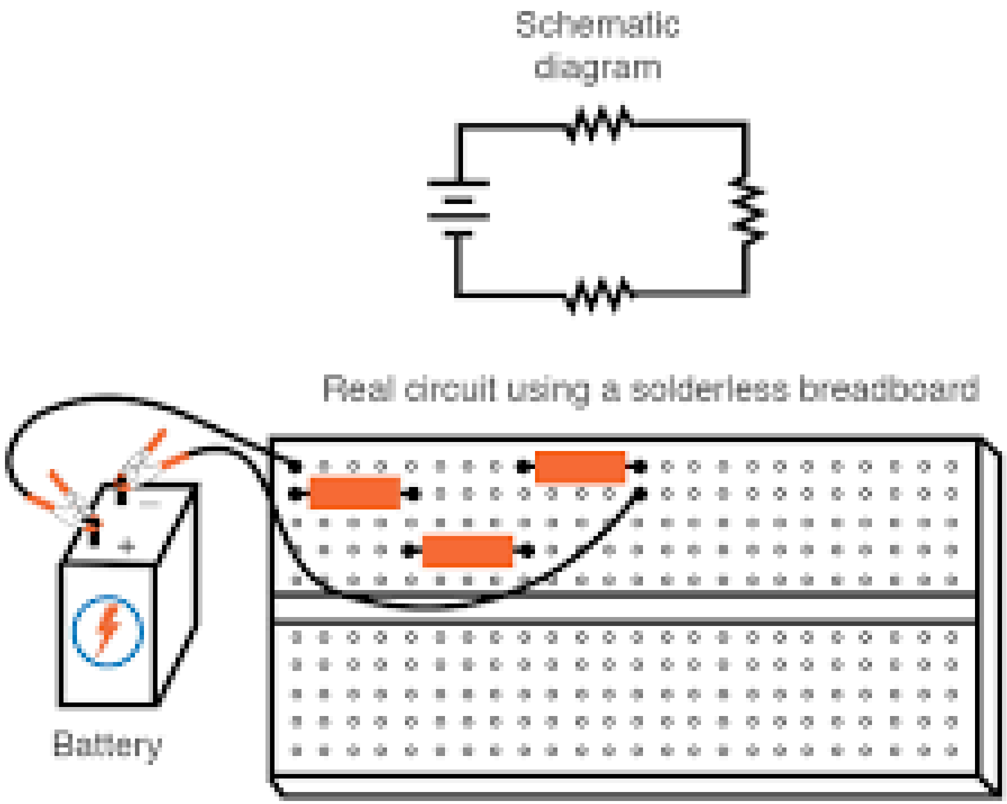


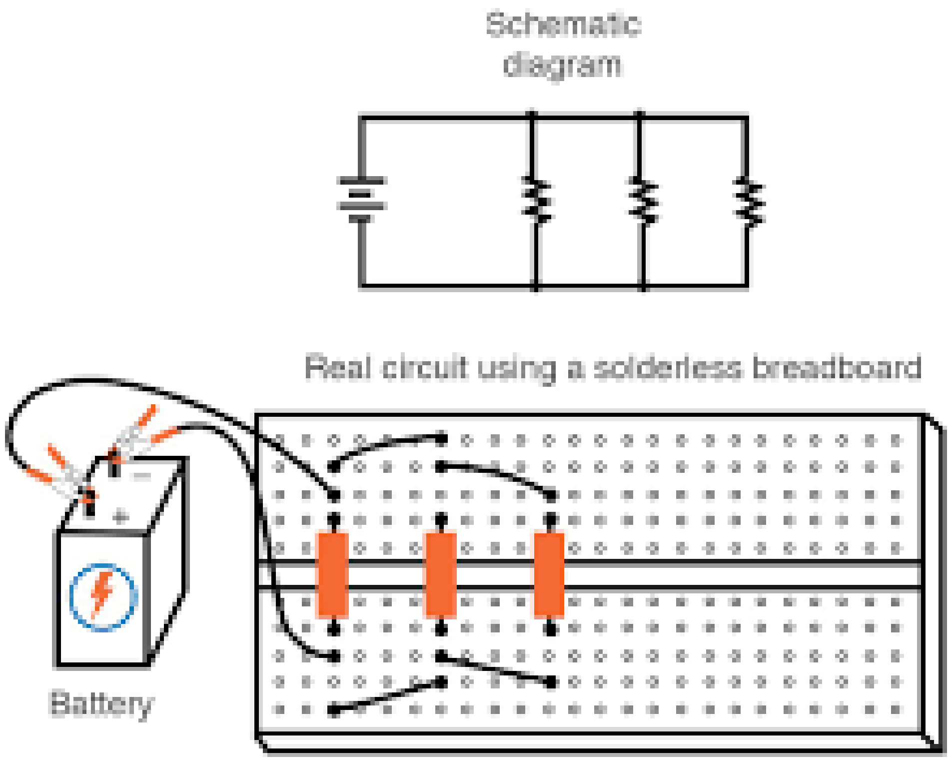
Using a Breadboard

- **<u>Step 1: Power Rails</u>**
  - Identify the power rails on the breadboard. They are typically labeled as + (positive) and - (negative).
  - Connect the positive and negative terminals of your power source (such as a battery or power supply) to these rails.
- **<u>Step 2: Component Placement</u>**
  - Insert your electronic components into the breadboard, such as resistors, capacitors, and integrated circuits.
  - Make sure to orient the components correctly, matching the polarity if applicable.
- **<u>Step 3: Interconnection</u>**
  - Use jumper wires to connect the components on the breadboard.
  - Insert one end of the jumper wire into the desired clip/socket of one component and the other end into another component's clip/socket to create connections.
- **<u>Step 4: Testing and Prototyping</u>**
  - Once all the components are connected, apply power to the breadboard.
  - Test the circuit by observing the desired behavior, such as LED illumination or signal output.

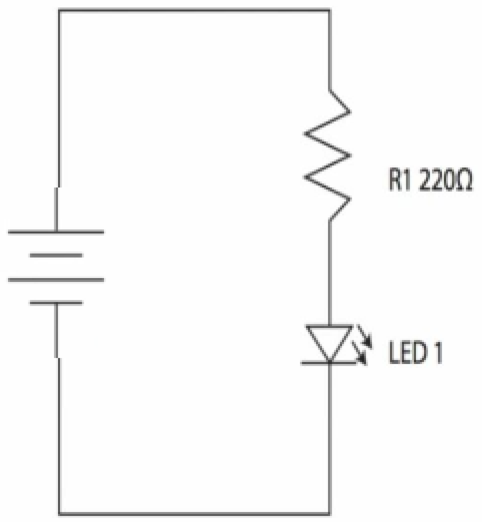


Series & Parallel Connections on Breadboard





Try it!



Other Components for Today

Push buttons

Potentiometers

Resistors

LEDs

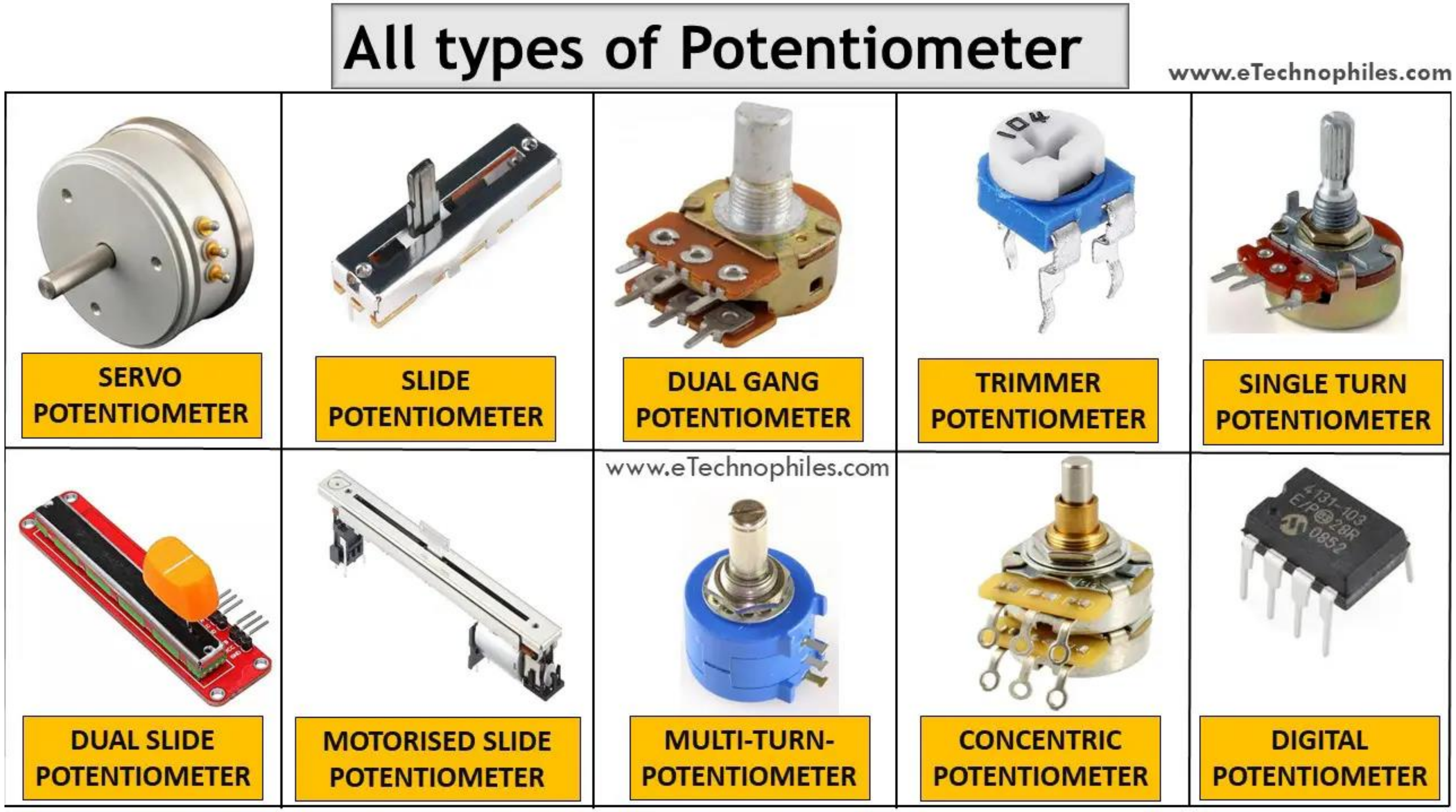
Potentiometers

- **Introduction:**
  - A potentiometer, also known as a variable resistor or pot, is an electronic component used to control the voltage in a circuit.
  - It consists of a resistive element and a movable contact or wiper that can be adjusted to vary the resistance.
- **Working Principle:**
  - The resistive element of a potentiometer is typically a long, thin strip or a coiled wire made of a resistive material.
  - The movable contact, connected to a sliding mechanism, can be moved along the resistive element.
  - As the wiper moves, the effective length of the resistive element through which current flows changes, resulting in a change in resistance.
- **Features and Applications:**
  - **Features:**
    - Adjustable resistance over a specific range.
    - Linear or logarithmic response depending on the potentiometer type.
    - Available in various resistance values and power ratings.
  - **Applications:**
    - Volume control in audio devices.
    - Brightness control in lighting systems.
    - Control of motor speed in robotics.
    - Sensing and calibration in measurement equipment.
    - Setting reference voltages in electronic circuits.

Potentiometer

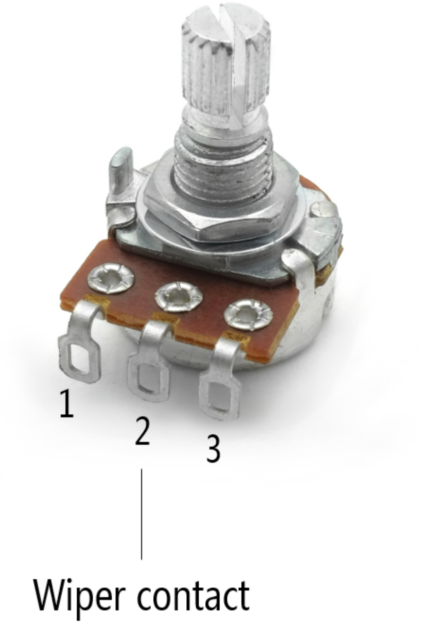
- **Connection:**
  - Connect one end of the potentiometer to the positive supply voltage.
  - Connect the other end of the potentiometer to the ground (GND) or negative supply voltage.
  - Connect the wiper terminal of the potentiometer to the desired circuit point where variable resistance is required.
- **Voltage Divider Circuit:**
  - The potentiometer acts as a voltage divider, dividing the input voltage based on the position of the wiper.
  - The output voltage at the wiper terminal can be adjusted by moving the wiper along the resistive element.
- **Resistance Adjustment:**
  - Rotate or slide the potentiometer's control knob or lever to change the position of the wiper.
  - This adjustment varies the resistance between the wiper terminal and the fixed end terminals, controlling the output voltage.
- **Sensing and Calibration:**
  - Measure the voltage at the wiper terminal using an analog-to-digital converter (ADC) or analog input of a microcontroller.
  - Calibrate the potentiometer by mapping the output voltage to the desired parameter or range of values.
- **Considerations:**
  - Choose the appropriate potentiometer type (linear or logarithmic) based on the application.
  - Consider the power rating of the potentiometer to ensure it can handle the expected current.
  - Avoid excessive mechanical force on the potentiometer to prevent damage to the wiper or resistive element.
- ***Note: The precise characteristics and behavior of potentiometers may vary based on the specific type and quality of the component. It is advisable to consult the potentiometer's datasheet or specifications for detailed information.***

Types of Potentiometers

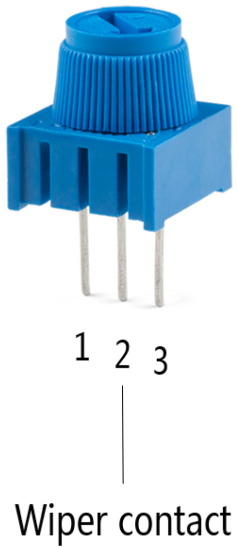


Wiper Based Contact Potentiometer

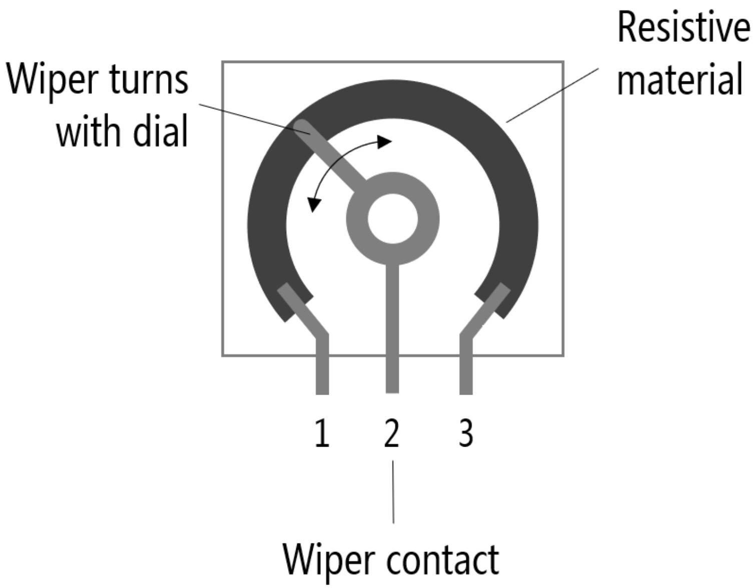
PANEL MOUNT POT 10KΩ



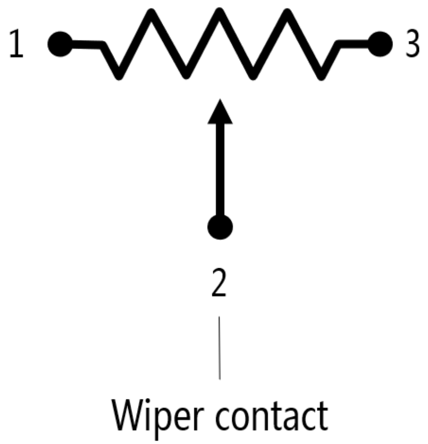
TRIM POT10KΩ



FUNCTIONAL DIAGRAM



SCHEMATIC SYMBOL



Connecting Potentiometer

1.<u>Fixed Terminal:</u>

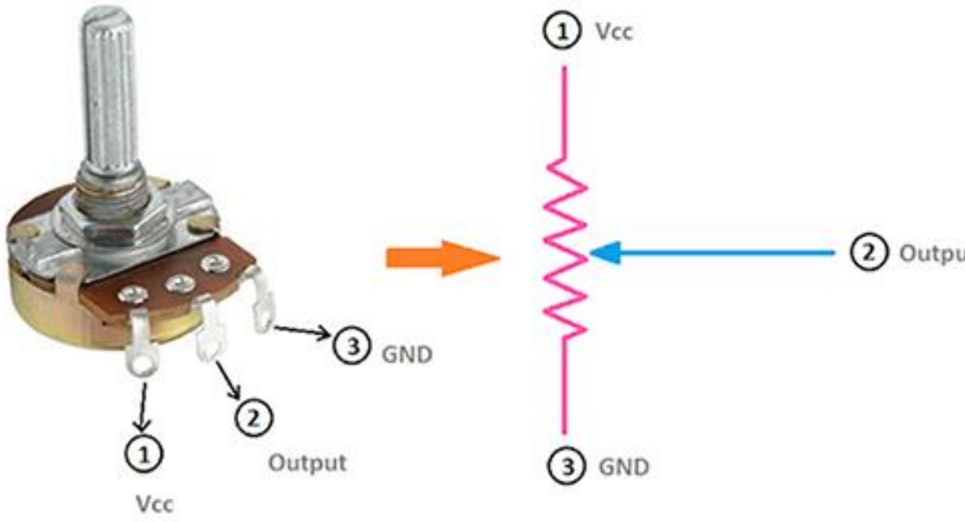
1. The fixed terminal is one of the outermost terminals of a potentiometer.
2. It provides a constant reference voltage or resistance value.
3. The fixed terminal is connected to one end of the resistive element.

2.<u>Wiper Terminal:</u>

1. The wiper terminal is the middle terminal of the potentiometer.
2. It is connected to a movable contact or wiper that slides along the resistive element.
3. As the wiper moves, it establishes electrical contact with different positions on the resistive element.

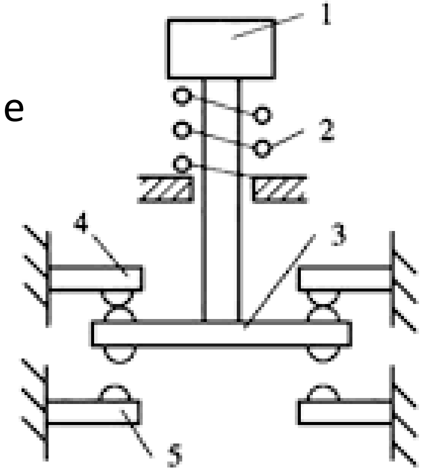
3.<u>Variable Terminal:</u>

1. The variable terminal is the third terminal of the potentiometer.
2. It is connected to the wiper terminal and provides an adjustable output voltage or resistance.
3. By adjusting the wiper position, the resistance or voltage at the variable terminal can be controlled.



Push Button

- **Common Terminal:**
  - The common terminal, also known as the "COM" terminal, is one of the outermost terminals of a push button.
  - It is the common connection point for the button.
  - In the resting state, the common terminal is not connected to any other terminal.
- **Normally Open Terminal (NO):**
  - The normally open terminal, also known as the "NO" terminal, is one of the inner terminals of a push button.
  - It is not connected to the common terminal when the button is not pressed.
  - When the button is pressed, the normally open terminal gets connected to the common terminal, creating an electrical connection.
- **Normally Closed Terminal (NC):**
  - The normally closed terminal, also known as the "NC" terminal, is the other inner terminal of a push button.
  - It is connected to the common terminal when the button is not pressed.
  - When the button is pressed, the normally closed terminal loses connection with the common terminal, breaking the electrical connection.

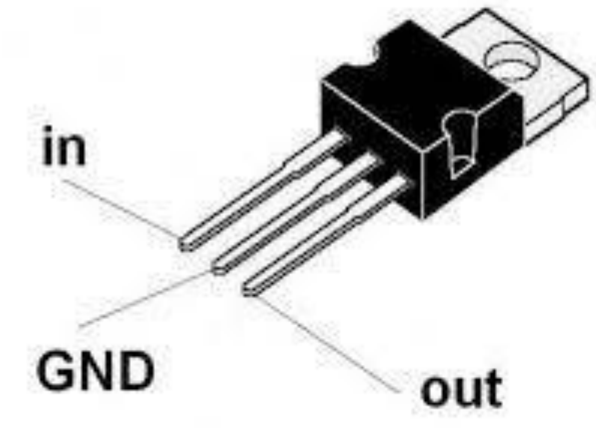


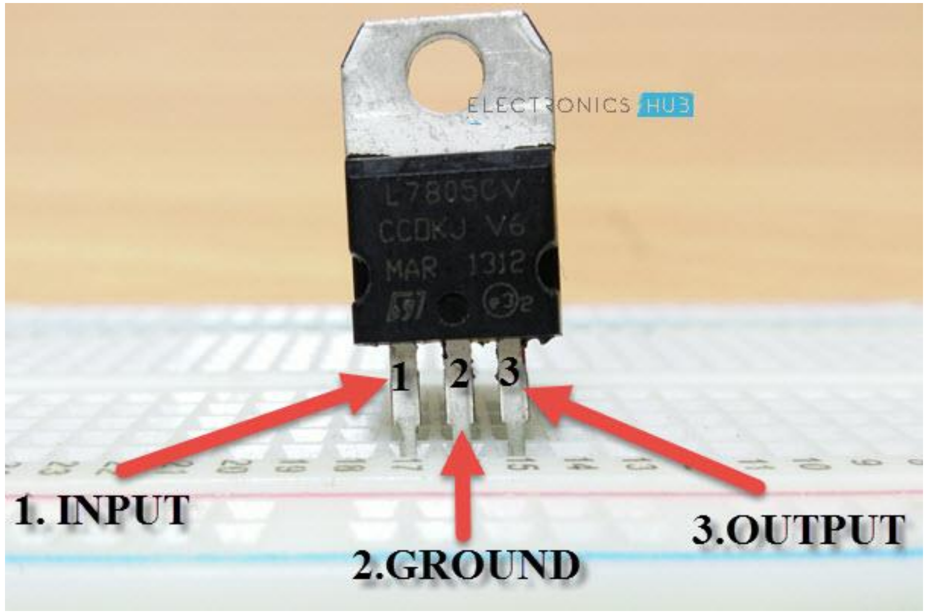
Push Button

- In its resting state, a push button is open, and the common terminal is not connected to either the normally open or normally closed terminal.
- When the button is pressed, it closes the circuit between the common terminal and one of the other terminals, either the normally open or normally closed terminal, depending on the type of push button.
- **Applications:**
- Push buttons are commonly used as input devices in various electronic systems and circuits.
- They are used for tasks such as signaling, switching, toggling, or triggering specific functions.
- Examples include power on/off switches, doorbell buttons, control buttons on electronic devices, and momentary switches in digital systems.

Voltage Regulator

- **Input Terminal (VIN):**
  - The input terminal, also known as the VIN terminal, is the terminal where the unregulated input voltage is applied.
  - It is usually denoted as "VIN" or "V-IN" on the voltage regulator.
  - The input terminal receives the higher voltage from the power source or a previous stage in the circuit.
- **Output Terminal (VOUT):**
  - The output terminal, also known as the VOUT terminal, is the terminal where the regulated output voltage is obtained.
  - It is usually denoted as "VOUT" or "V-OUT" on the voltage regulator.
  - The output terminal provides a stable and regulated voltage to power the components or devices connected to it.
- **Ground Terminal (GND):**
  - The ground terminal, denoted as "GND" or "G", is the reference point for the voltage regulator's operation.
  - It serves as the common ground reference for both the input and output sides of the regulator.
  - The ground terminal is typically connected to the circuit's ground or the negative terminal of the power supply.

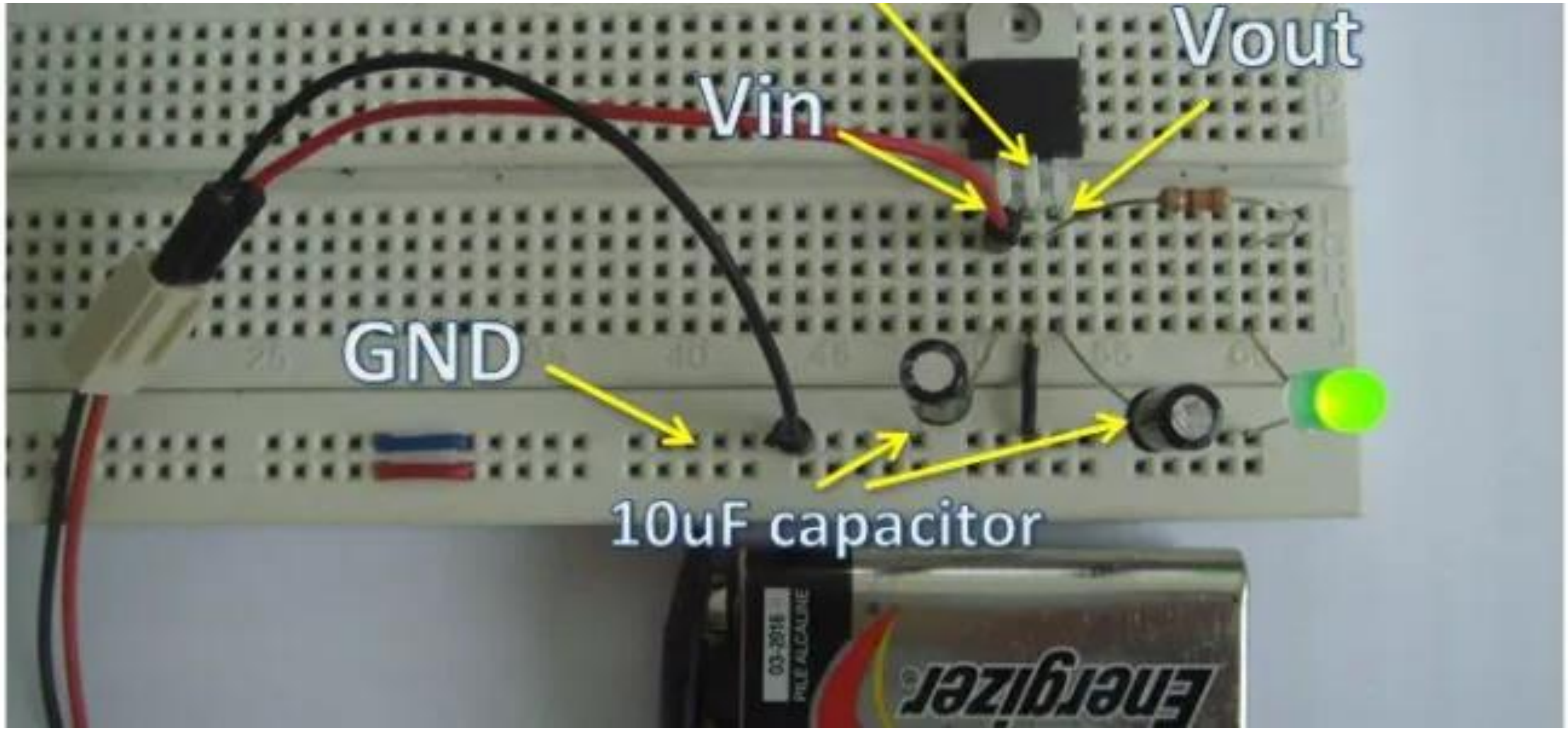




Voltage Regulator

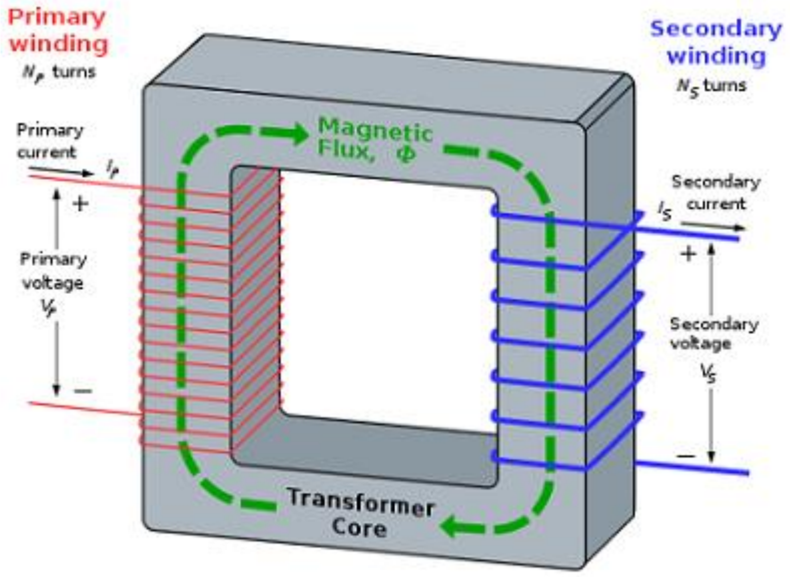
- **Working Principle:**
- A voltage regulator is an electronic component designed to maintain a steady output voltage regardless of fluctuations in the input voltage or load conditions.
- The input voltage, applied to the VIN terminal, undergoes internal regulation processes to generate a stable output voltage at the VOUT terminal.
- The voltage regulator compares the input voltage to a reference voltage and adjusts its internal circuitry to regulate the output voltage to the desired value.
- **Applications:**
- Voltage regulators are extensively used in electronic circuits and systems to ensure a consistent and stable power supply.
- They are employed in various applications, including power supplies, voltage-sensitive components, microcontrollers, sensors, and analog circuitry.
- Voltage regulators play a crucial role in protecting sensitive components from voltage variations and maintaining reliable operation.

Example on Voltage Regulator 9v to 5v Conversion



Transformers

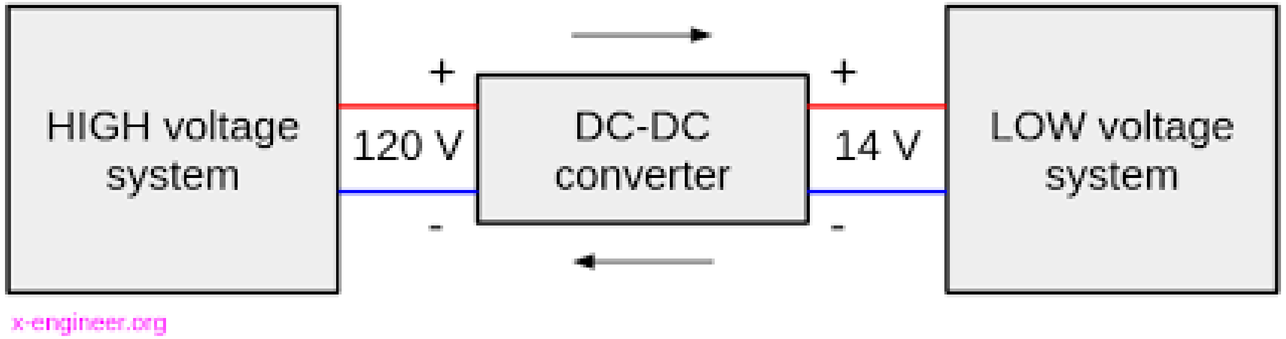
- **<u>Primary Terminal:</u>**
  - The primary terminal, also known as the primary winding, is the input side of the transformer.
  - It is usually denoted as "P" or "Pri" and is connected to the power source that supplies alternating current (AC) voltage.
  - The primary winding consists of coils of wire and is wound around a core material, typically made of iron or ferrite.
- **<u>Secondary Terminal:</u>**
  - The secondary terminal, also known as the secondary winding, is the output side of the transformer.
  - It is usually denoted as "S" or "Sec" and is connected to the load or device that requires a different voltage level.
  - The secondary winding is also made up of coils of wire wound around the same core material as the primary winding.
- **<u>Common Terminal:</u>**
  - The common terminal, sometimes referred to as the center tap, is a terminal that is shared between the primary and secondary windings.
  - It is denoted as "C" or "CT" and is connected to the center point of the winding coils.
  - The common terminal allows for different configurations of the transformer, such as center-tapped or split-phase connections.



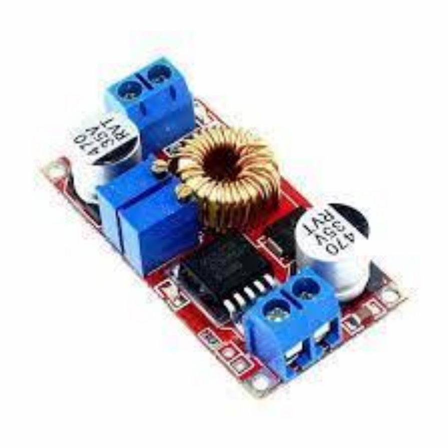
Transformers

- **<u>Working Principle:</u>**
- Transformers are electrical devices that transfer electrical energy between two or more coils of wire through electromagnetic induction.
- The primary winding is connected to an AC power source, which creates a changing magnetic field around the core.
- This changing magnetic field induces an alternating current in the secondary winding, resulting in a different voltage level at the secondary terminal.
- The ratio of the number of turns in the primary and secondary windings determines the voltage transformation of the transformer.
- **<u>Applications:</u>**
- Transformers are widely used in various electrical and electronic systems, including power distribution networks, electrical appliances, and electronic devices.
- They are employed to step-up or step-down voltages for efficient transmission, distribution, and utilization of electrical energy.
- Transformers are crucial components in power supplies, audio equipment, electric motors, and many other applications.

DC-DC Convertors



- **<u>Input Terminal:</u>**
  - The input terminal of a DC-DC converter is where the DC power source is connected.
  - It is denoted as Vin and usually connected to a DC voltage source, such as a battery or power supply.
- **<u>Output Terminal:</u>**
  - The output terminal is where the converted DC voltage is available for the load or device.
  - It is denoted as Vout and connected to the load or other components that require a specific DC voltage level.
- **<u>Ground Terminal:</u>**
  - The ground terminal, often labeled as GND or COM, serves as the reference point for the DC-DC converter's circuit.
  - It is connected to the common ground potential and provides a stable reference voltage for the converter's operation.

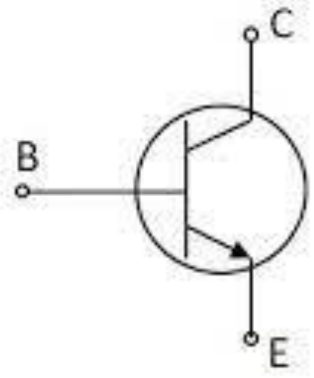


DC DC Convertors

- DC-DC converters, also known as voltage regulators, are electronic devices that convert one DC voltage level to another.
- They operate based on the principle of switching regulation, using high-frequency switching circuits to control the voltage conversion.
- The input voltage is first transformed into an oscillating waveform, which is then rectified and filtered to obtain a stable DC voltage.
- This voltage is then further processed through a switching mechanism to regulate the output voltage according to the desired level.
- The switching action allows for efficient voltage conversion, typically achieved through techniques like pulse-width modulation (PWM).
- **Applications:**
- DC-DC converters are widely used in various electronic systems and devices that require different DC voltage levels.
- They are essential components in power supplies, battery chargers, voltage regulators, and DC motor control circuits.
- DC-DC converters are commonly used in portable electronic devices, such as smartphones, laptops, and tablets, to provide different voltage levels for different components within the device.
- They are also utilized in automotive electronics, renewable energy systems, telecommunications equipment, and many other applications where efficient voltage conversion is necessary.

Transistor: BJT

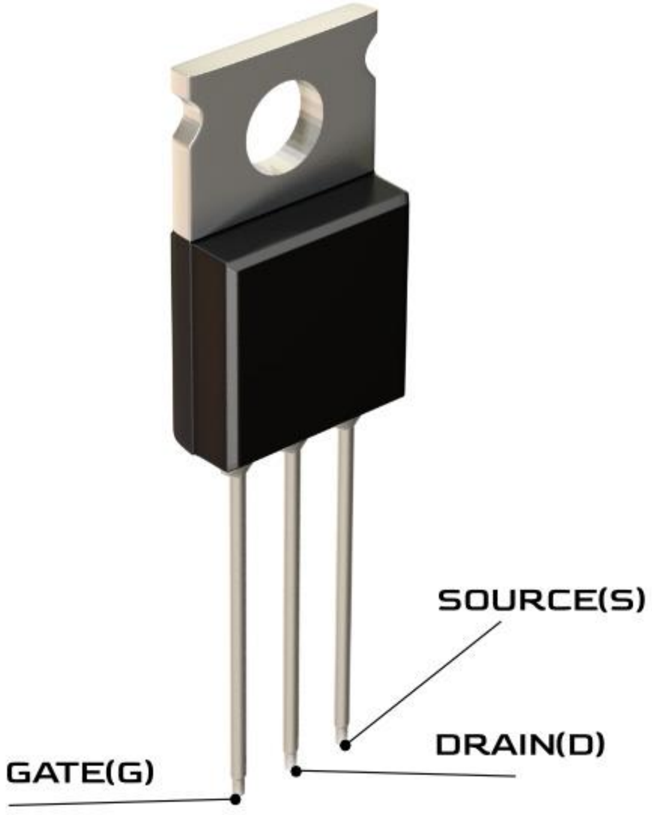
- **Base Terminal (B):**
  - The base terminal is one of the three terminals of a bipolar junction transistor (BJT).
  - It is responsible for controlling the transistor's operation by regulating the current flow between the other two terminals.
  - The base terminal is denoted as B and is typically connected to the control signal or input source.
- **Collector Terminal (C):**
  - The collector terminal is another terminal of the BJT transistor.
  - It serves as the main current-carrying terminal and collects the majority of current flowing through the transistor.
  - The collector terminal is denoted as C and is connected to the load or output of the transistor circuit.
- **Emitter Terminal (E):**
  - The emitter terminal is the third terminal of the BJT transistor.
  - It is responsible for emitting or injecting majority charge carriers (electrons or holes) into the transistor's base region.
  - The emitter terminal is denoted as E and is usually connected to the ground or common reference potential.



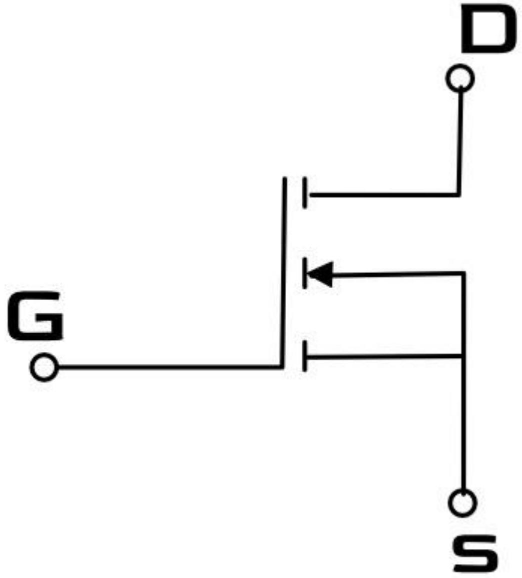
Transistor: BJT

- **<u>Working Principle:</u>**
- The BJT transistor operates based on the principle of current amplification and control.
- It consists of two pn-junctions: the base-emitter junction (BE) and the base-collector junction (BC).
- In an NPN transistor, the base is made of P-type material, and the emitter and collector are made of N-type material.
- When a small current is applied to the base-emitter junction, it causes a larger current to flow between the collector and emitter terminals.
- The base current controls the flow of current between the collector and emitter, allowing the transistor to act as an amplifier or switch.
- **<u>Applications:</u>**
- BJT transistors are widely used in electronic circuits for various applications.
- They are commonly employed in amplifiers, oscillators, signal processing circuits, and switching applications.
- BJT transistors can be used as audio amplifiers, radio frequency (RF) amplifiers, voltage regulators, and logic gates.
- They are also utilized in power control circuits, motor control circuits, and in digital and analog electronics.

Transistor: MOSFET



MOSFET



SYMBOL OF MOSFET

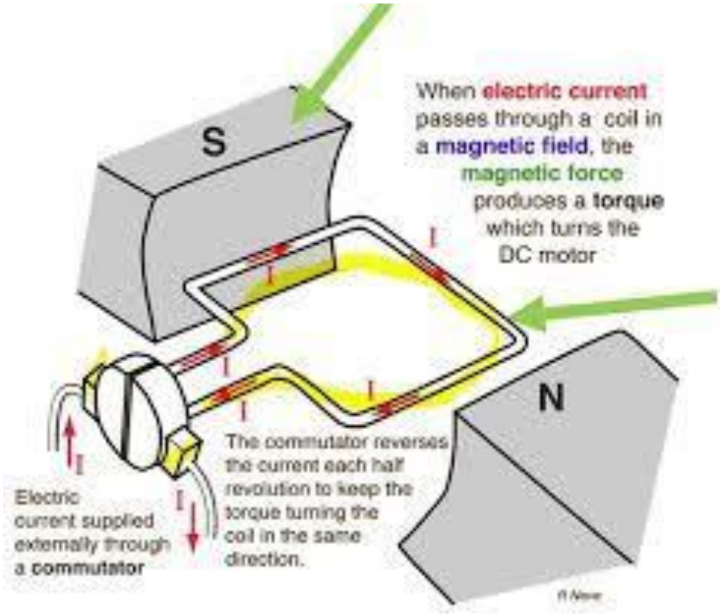
Transistor: MOSFET

- **<u>Gate Terminal (G):</u>**
  - The gate terminal is one of the three terminals of a MOSFET.
  - It controls the conductivity of the transistor by regulating the electric field in the channel region.
  - The gate terminal is denoted as G and is typically connected to the control signal or input source.
- **<u>Drain Terminal (D):</u>**
  - The drain terminal is another terminal of the MOSFET.
  - It serves as the main current-carrying terminal and collects the majority of current flowing through the transistor.
  - The drain terminal is denoted as D and is connected to the load or output of the transistor circuit.
- **<u>Source Terminal (S):</u>**
  - The source terminal is the third terminal of the MOSFET.
  - It acts as the source of the majority charge carriers (electrons or holes) for the transistor's channel region.
  - The source terminal is denoted as S and is typically connected to the ground or common reference potential.

Transistor: MOSFET

- **Working Principle:**
- The MOSFET transistor operates based on the principle of voltage-controlled conductivity.
- It consists of a metal gate electrode separated from the channel region by a thin insulating layer (oxide layer).
- By applying a voltage to the gate terminal, an electric field is formed in the channel region, controlling the flow of current.
- In an N-channel MOSFET, the channel is formed by N-type semiconductor material between the source and drain regions.
- By varying the gate voltage, the conductivity of the channel can be modulated, allowing current to flow between the source and drain terminals.
- **Applications:**
- MOSFETs are widely used in electronic circuits for various applications due to their high switching speed and low power consumption.
- They are commonly employed in power amplifiers, motor control circuits, switching regulators, and audio amplifiers.
- MOSFETs are extensively used in digital logic circuits, such as in microprocessors, memory chips, and integrated circuits.
- They are also utilized in power electronics applications, such as in DC-DC converters, voltage regulators, and power supplies.

DC Motors



S
N
When electric current passes through a coil in a magnetic field, the magnetic force produces a torque which turns the DC motor
The commutator reverses the current each half revolution to keep the torque turning the coil in the same direction.
Electric current supplied externally through a commutator
I

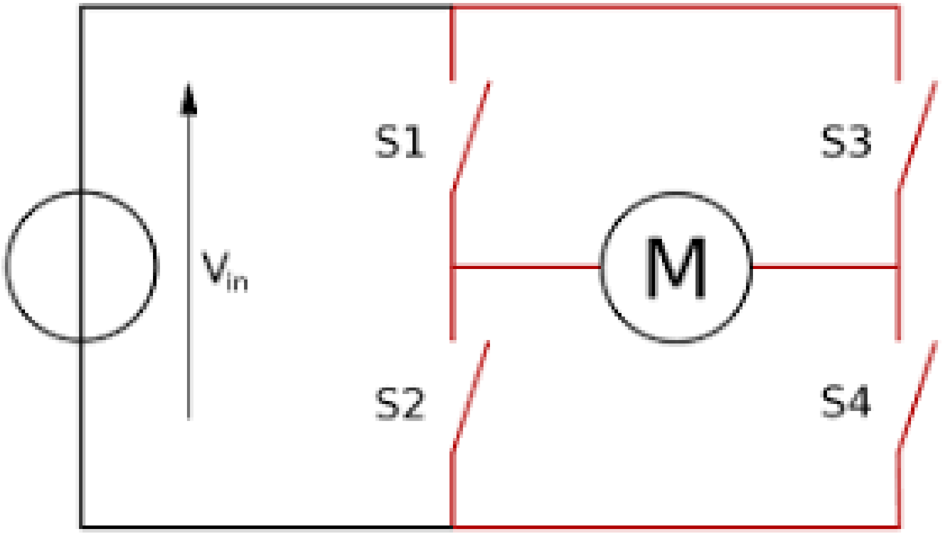
DC Motors

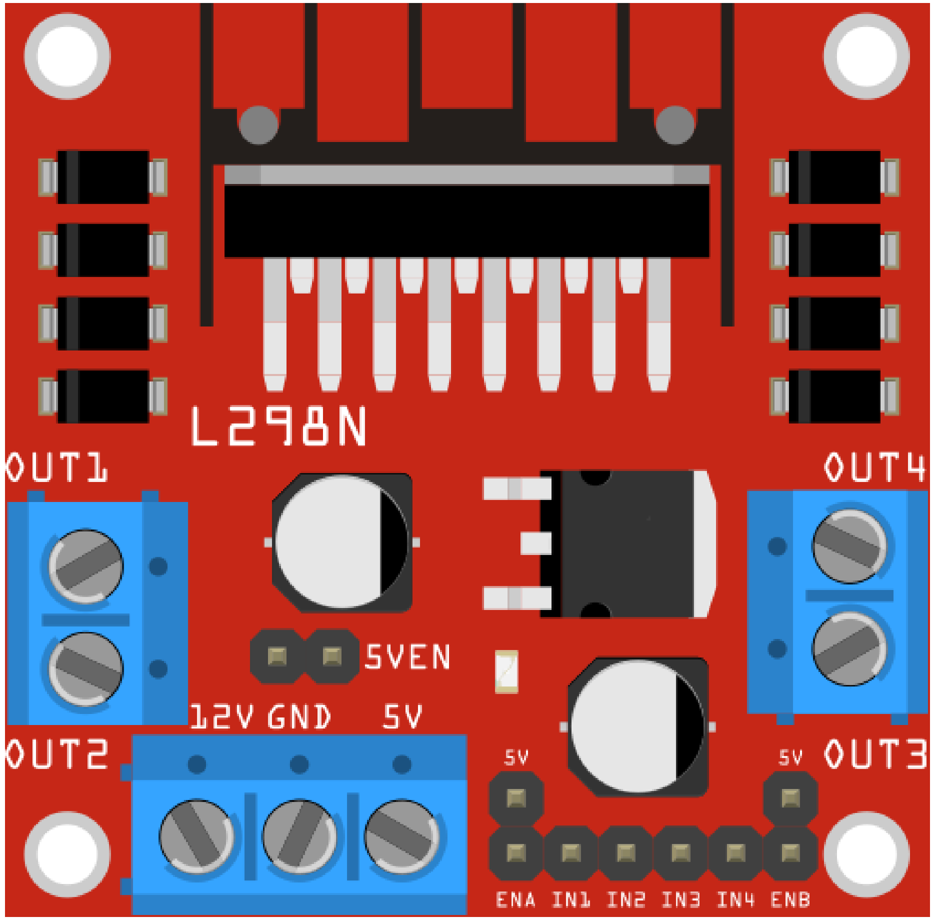
- **Introduction:**
  - DC (Direct Current) motors are widely used in various applications for converting electrical energy into mechanical motion.
  - They operate based on the principle of electromagnetic induction, utilizing the interaction between a magnetic field and current-carrying conductors.
- **Working Principle:**
  - DC motors consist of two essential components: a stator and a rotor.
  - The stator contains a stationary set of magnets that create a magnetic field.
  - The rotor, connected to the motor shaft, consists of a coil of wire (armature) that carries current.
  - When a current flows through the armature, it interacts with the magnetic field, resulting in a rotational force (torque) on the rotor.
- **Types of DC Motors:**
- **a. Brushed DC Motors:**
  - In brushed DC motors, the current flows through a commutator and brushes, which help in reversing the current direction through the armature coil.
  - The commutator ensures that the polarity of the electromagnet changes at the right time to maintain continuous rotation.
  - These motors are relatively simple and widely used in low-cost applications.
- **b. Brushless DC Motors (BLDC):**
  - Brushless DC motors eliminate the need for brushes and commutators.
  - Instead, they use electronic commutation through a motor controller to switch the current direction in the windings at the right time.
  - BLDC motors are more efficient, have longer lifespans, and require less maintenance compared to brushed DC motors.
  - They are commonly used in applications demanding higher efficiency and precise control.

DC Motors

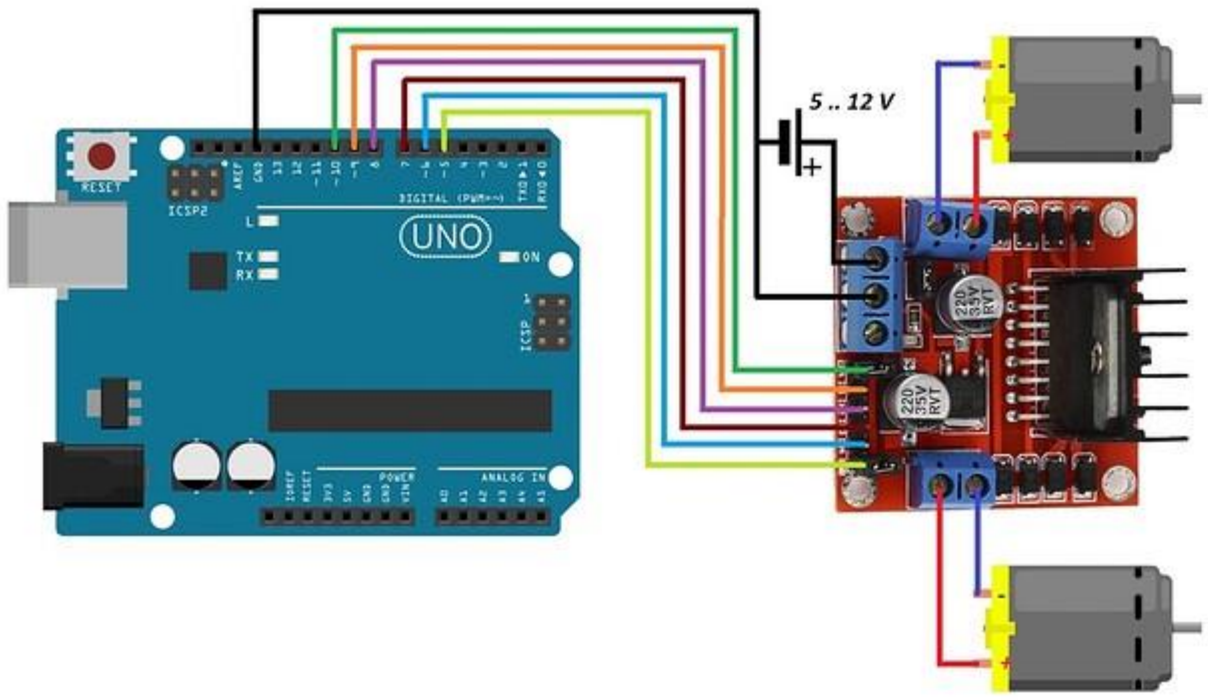
- **Operation and Control:**
  - DC motors can be operated in various configurations, such as with H-bridges, motor drivers, or motor control circuits.
  - By controlling the voltage, current, and polarity applied to the motor, its speed, torque, and direction can be controlled.
  - Speed control can be achieved by adjusting the input voltage or using pulse-width modulation (PWM) techniques.
- **Applications:**
  - DC motors find applications in a wide range of fields, including robotics, automation, electric vehicles, home appliances, and industrial machinery.
  - They are used in fans, pumps, conveyors, electric vehicles, CNC machines, printers, and many other devices that require rotational motion.

H-bridge


S1
S3
V_{in}
M
S2
S4


L298N
OUT1
OUT4
5VEN
12V GND 5V
OUT2
5V
5V
OUT3
ENA IN1 IN2 IN3 IN4 ENB

H-bridge & Motors



H-bridge

1. **Introduction:**
   - An H-bridge is an electronic circuit configuration commonly used to control the direction and speed of DC motors.
   - It consists of four switches (transistors or MOSFETs) arranged in the shape of an "H," hence the name H-bridge.
2. **Working Principle:**
   - The H-bridge allows the current to flow through the motor in both directions, enabling bidirectional control.
   - By selectively turning on/off the switches, different combinations of currents can be applied to the motor.
   - The four switches are typically labeled as Q1, Q2, Q3, and Q4, forming two pairs on the left and right sides of the H-bridge.

H-bridge

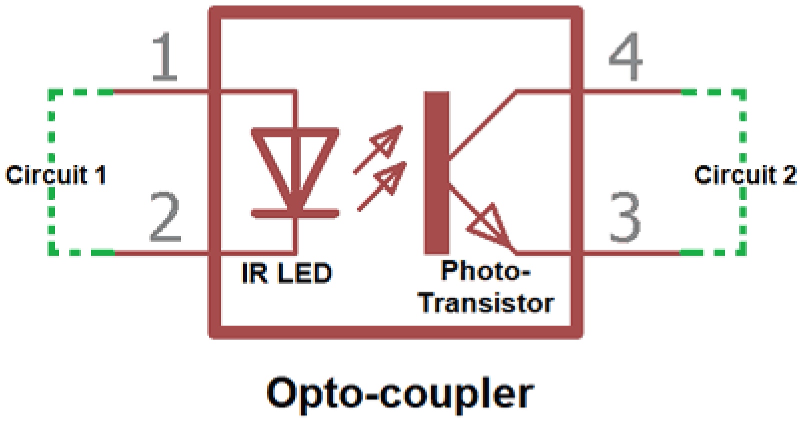
3. **<u>Operation Modes:</u>**

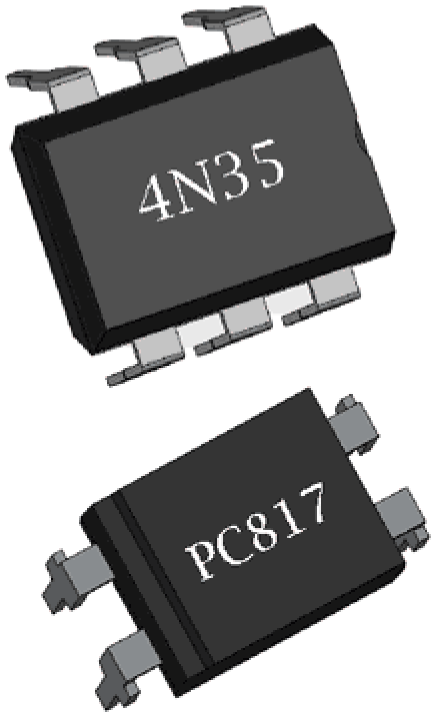
- •Forward Operation: In this mode, switches Q2 and Q3 are turned ON, while switches Q1 and Q4 are turned OFF. This allows current to flow from the positive supply through the motor, resulting in forward motion.
- •Reverse Operation: In this mode, switches Q1 and Q4 are turned ON, while switches Q2 and Q3 are turned OFF. This allows current to flow from the negative supply through the motor, causing reverse motion.
- •Coast/Brake Operation: To stop the motor, all switches are turned OFF, creating an open circuit. The motor coasts to a halt due to inertia.
- •Regenerative Braking: By rapidly switching the H-bridge, the motor's momentum can be used to regenerate power back into the power supply.

H-bridge

4. **<u>Control Signals:</u>**
   - To control the H-bridge, a control circuit is required to provide appropriate signals to the switches.
   - Using a microcontroller or other control devices, the switches are driven in a complementary manner to achieve the desired motor direction and speed.
5. **<u>Applications:</u>**
   - H-bridges are commonly used in robotics, electric vehicles, and various motor control applications.
   - They are utilized for controlling DC motors, stepper motors, and servo motors.
   - H-bridges enable precise control of motor speed, direction, and torque, making them suitable for applications that require motor control versatility.

Opto-couplers





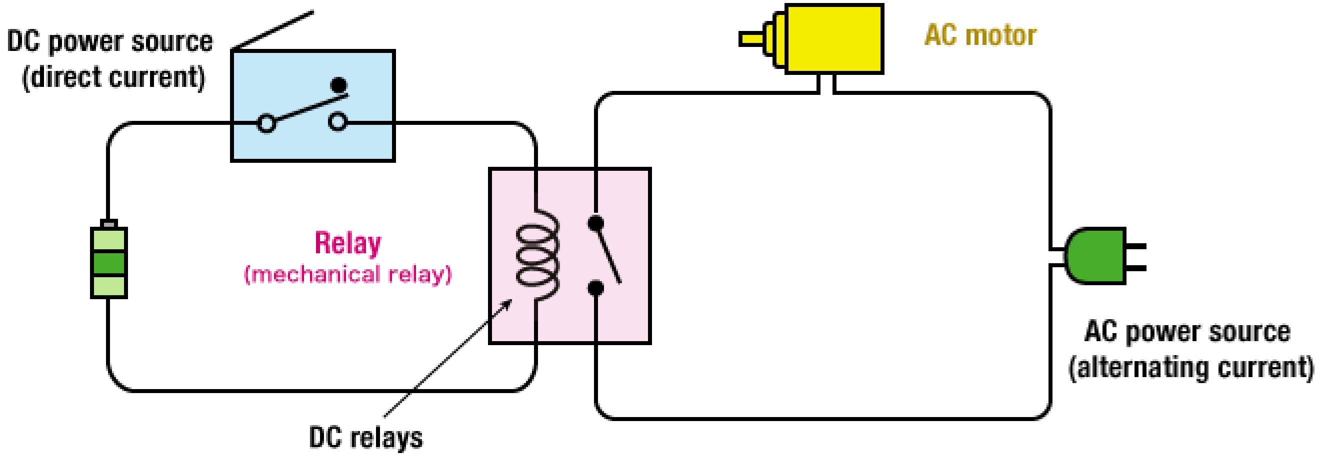
Opto-couplers

- **Introduction:**
  - Optocouplers, also known as optoisolators, are electronic components used to transmit signals between two isolated circuits using light.
  - They provide electrical isolation and protect sensitive components from voltage spikes, noise, and other disturbances.
- **Working Principle:**
  - An optocoupler consists of an LED (Light Emitting Diode) and a photodetector (usually a phototransistor or a photodiode) enclosed in a single package.
  - When current flows through the LED, it emits light.
  - The emitted light is then detected by the photodetector, which generates a corresponding electrical signal.
- **Isolation and Protection:**
  - Optocouplers are used to electrically isolate two circuits while allowing signal transmission between them.
  - The input circuit (LED side) and output circuit (photodetector side) are physically separated, preventing direct electrical contact.
  - This isolation protects sensitive components from voltage surges, ground loops, and noise.

Opto-couplers

- **Signal Transmission:**
  - To transmit a signal, an input signal is applied to the LED, which emits light.
  - The emitted light activates the photodetector, producing an output signal.
  - The output signal replicates the input signal and can be used to control or monitor the connected circuit.
- **Applications:**
  - Optocouplers have a wide range of applications in electronic systems, including:
    - Isolation of high-voltage circuits from low-voltage circuits.
    - Signal transmission in noisy environments to prevent interference.
    - Protection of microcontrollers and sensitive components from voltage spikes.
    - Level shifting and signal conditioning.
    - Feedback and control systems.
    - Motor control and power electronics.
    - Switching power supplies and inverters.

Relays


DC power source
(direct current)
AC motor
Relay
(mechanical relay)
AC power source
(alternating current)
DC relays


Normally Closed
Common
Normally Open
Ground
Vcc
Signal Pin
SONGLE
10A 250VAC 10A 125VAC
10A 30VDC 10A 28VDC
SRD-05VDC-SL-C

Relays

- **Introduction:**
  - Relays are electromechanical devices used to control the switching of electrical circuits.
  - They act as electrically controlled switches that open or close contacts to control the flow of current in a separate circuit.
- **Construction:**
  - A relay consists of a coil, an armature, and one or more sets of contacts.
  - The coil is an electromagnet that generates a magnetic field when an electrical current passes through it.
  - The armature, typically a movable iron core, is attracted or repelled by the magnetic field produced by the coil.
  - The contacts are metal conductors that open or close when the armature is activated.
- **Working Principle:**
  - When current flows through the coil, it creates a magnetic field that attracts or repels the armature.
  - The movement of the armature causes the contacts to either close or open, depending on the relay type.
  - Normally Open (NO) relays have contacts that are open when the coil is not energized and close when the coil is energized.
  - Normally Closed (NC) relays have contacts that are closed when the coil is not energized and open when the coil is energized.
- **Switching Function:**
  - Relays are used to control the switching of high-power or high-voltage circuits using a low-power control signal.
  - They provide electrical isolation between the control circuit and the switched circuit, enhancing safety and protection.
  - The control signal can come from a variety of sources, such as microcontrollers, sensors, or other electronic devices.

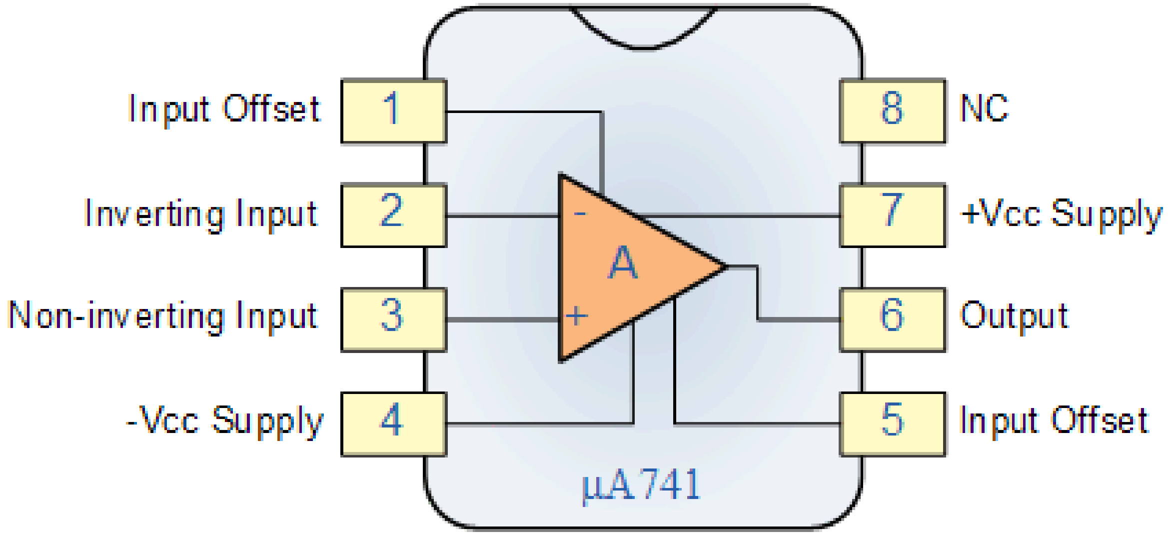
Relays

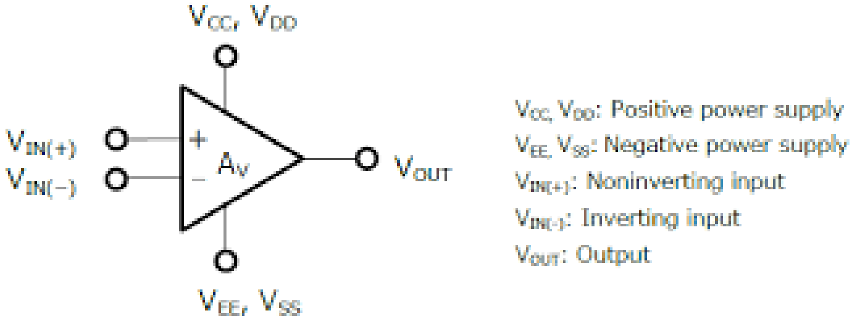
Applications:

Relays have numerous applications in different industries, including:

- Industrial automation and control systems.
- Home automation and smart systems.
- Automotive electronics.
- Power distribution and protection.
- Motor control and starters.
- HVAC (Heating, Ventilation, and Air Conditioning) systems.
- Lighting control.
- Alarm and security systems.

OpAmp


Input Offset
Inverting Input
Non-inverting Input
-Vcc Supply
1
2
3
4
8
7
6
5
NC
+Vcc Supply
Output
Input Offset
A
μA741


V_{CC}, V_{DD}
$V_{IN(+)}$
$V_{IN(-)}$
A_V
V_{OUT}
V_{EE}, V_{SS}
V_{CC}, V_{DD}: Positive power supply
V_{EE}, V_{SS}: Negative power supply
$V_{IN(+)}$: Noninverting input
$V_{IN(-)}$: Inverting input
V_{OUT}: Output

OpAmp

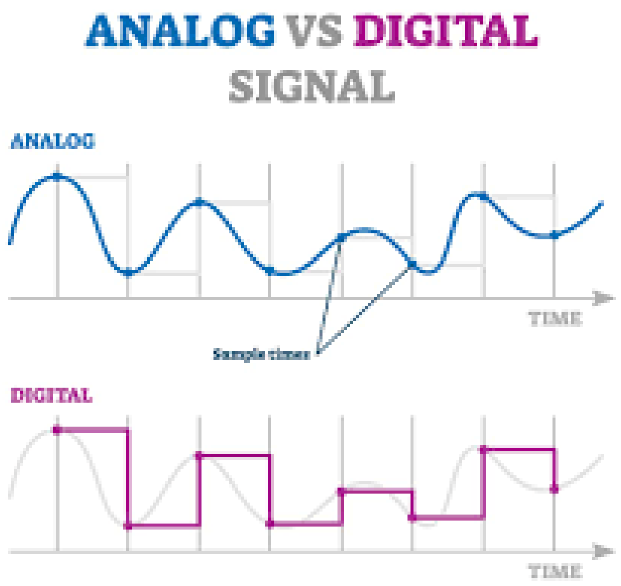
- **Operational Amplifiers (Op-Amps):**
- Operational Amplifiers, or op-amps, are versatile electronic devices used for amplification and signal processing in analog circuits.
- They consist of differential input stages, gain stages, and output stages, and are widely available as integrated circuits (ICs).
- **Working Principle:**
- Op-amps operate based on the concept of differential amplification.
- The voltage difference between the two input terminals (inverting and non-inverting) is amplified to produce the output voltage.
- Op-amps have a high gain, allowing them to amplify small input signals to a larger output signal.
- **Applications:**
- Differentiation:
  - Op-amps can be used for differentiation, where they amplify the rate of change of an input voltage.
  - This is commonly used in applications such as frequency modulation (FM) demodulation, signal processing, and wave shaping.
- Integration:
  - Op-amps are also used for integration, where they integrate an input voltage over time.
  - This finds applications in analog computing, analog filters, and time-dependent circuits.
- Summing Amplifier:
  - Op-amps can be configured as summing amplifiers to combine multiple input signals into a single output signal.
  - This is useful in audio mixing consoles, analog signal processing, and analog computation.

OpAmp

- **Advantages of Op-Amps:**
- High gain and accuracy: Op-amps offer high amplification and precise signal processing capabilities.
- Versatility: They can be configured for various amplification and signal processing tasks.
- Compatibility: Op-amps are compatible with a wide range of electronic components and can be easily integrated into circuits.
- Stability and reliability: They provide stable operation and have well-defined specifications.
- Low output impedance: Op-amps have low output impedance, allowing them to efficiently drive loads.
- Cost-effective: Op-amps are available at affordable prices and offer excellent performance for their cost.

Analog vs Digital

- **<u>Analog Signals:</u>**
- Concept: Analog signals are continuous, time-varying signals that represent information using varying voltage or current levels. They can take on any value within a range.
- Usage: Analog signals are used to represent real-world phenomena that are inherently continuous, such as audio, temperature, pressure, and voltage measurements.
- **Examples:**
  - Audio signals in music, speech, or any sound recording.
  - Temperature readings from a thermometer.
  - Voltage levels from sensors measuring light intensity or distance.

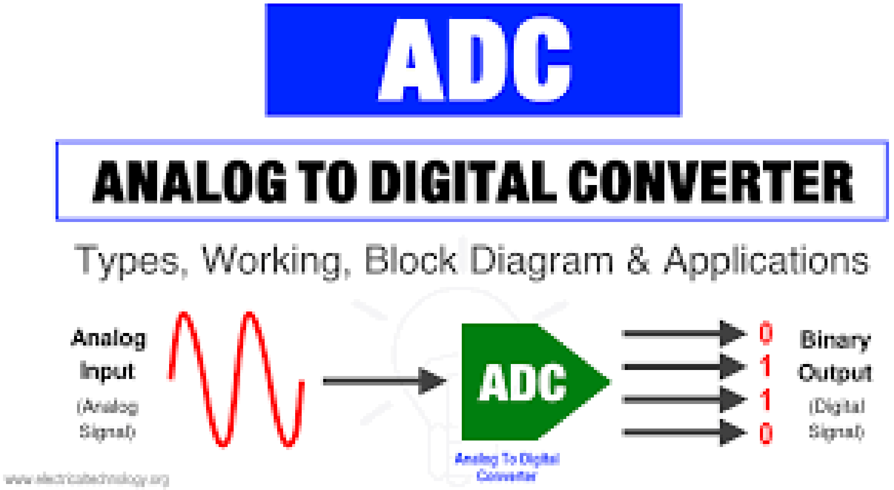


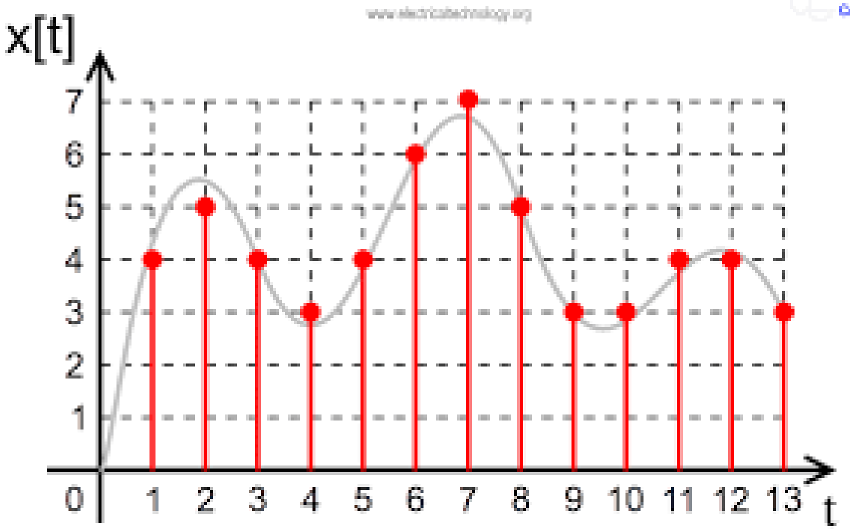
- **Digital Signals:**
- Concept: Digital signals are discrete, binary signals that represent information using only two distinct states, typically represented as "0" and "1." They have fixed voltage or current levels that correspond to these states.
- Usage: Digital signals are used in digital communication, computing, and control systems where information needs to be precisely encoded and transmitted in a discrete manner.
- **Examples:**
  - Binary data transmitted over computer networks.
  - Digital audio files encoded as "0s" and "1s."
  - Control signals in digital electronics, such as on/off commands.

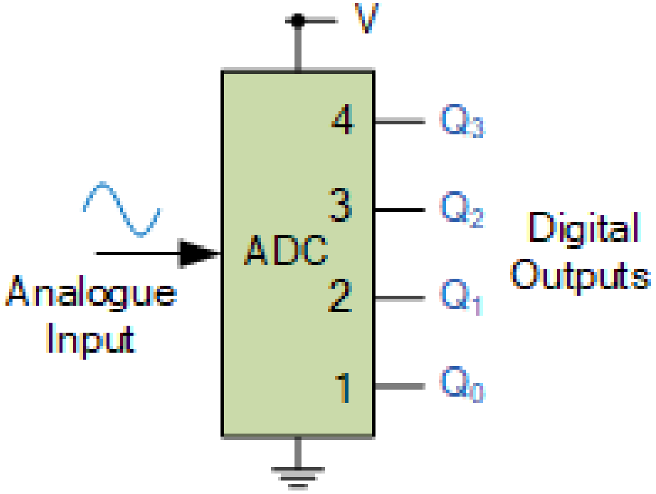
Analog vs Digital

- **<u>Comparison:</u>**
- Representation: Analog signals represent continuous information, while digital signals represent discrete information.
- Precision: Analog signals can have infinite precision since they can take on any value within a range, while digital signals have finite precision based on the number of bits used.
- Noise Immunity: Digital signals are more immune to noise and interference compared to analog signals since they rely on distinct voltage levels.
- Transmission and Storage: Digital signals can be easily transmitted and stored using binary encoding, while analog signals require specialized transmission and storage methods.
- Processing: Digital signals can be easily processed, manipulated, and analyzed using digital circuits and algorithms, while analog signals require specialized analog circuits for processing.
- Conversion: Analog signals can be converted to digital signals using analog-to-digital converters (ADCs), and digital signals can be converted back to analog using digital-to-analog converters (DACs).

Analog to Digital Convertor







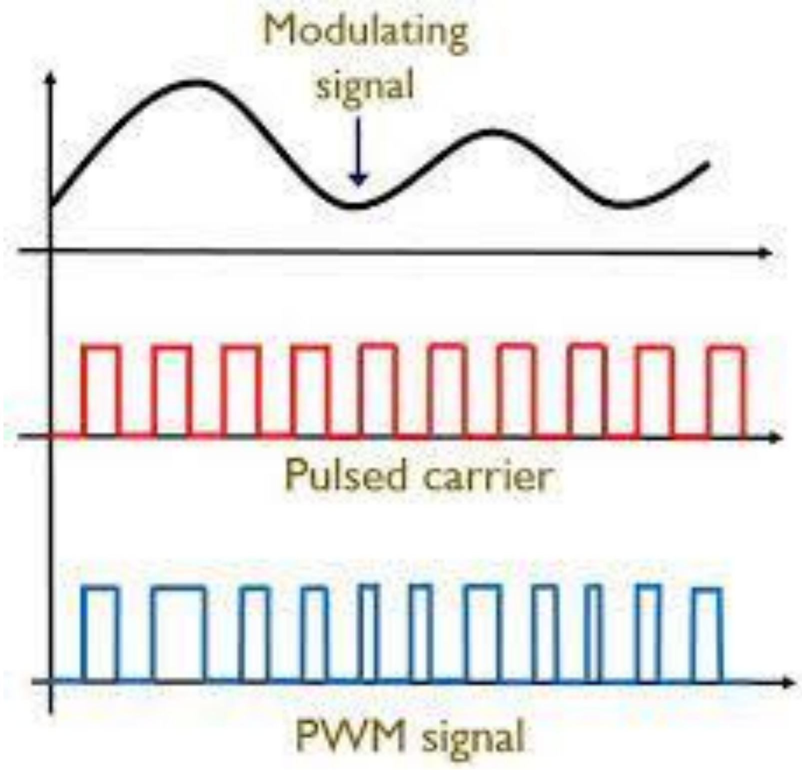
Analog to Digital Convertor

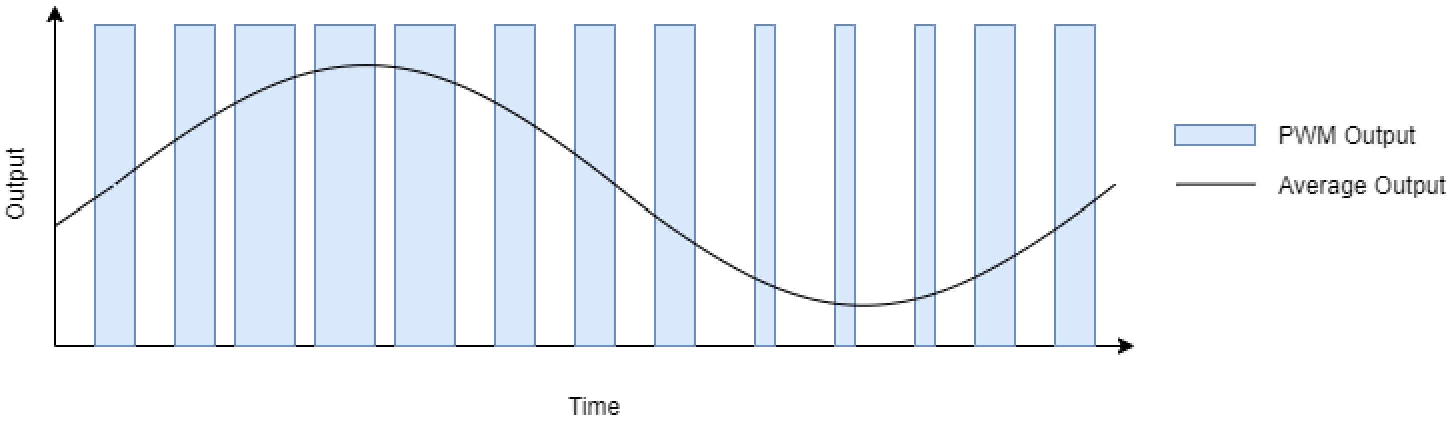
- Concept: An analog-to-digital converter (ADC) is an electronic device that converts analog signals into digital representations. It measures the continuous voltage or current levels of an analog signal and assigns discrete digital values to represent those levels.
- Usage: ADCs are used in various applications where analog signals need to be processed, analyzed, or transmitted digitally. They are commonly found in data acquisition systems, communication systems, instrumentation, and control systems.
- Examples:
  - Converting analog audio signals into digital format for digital audio processing, storage, or transmission.
  - Measuring analog sensor outputs (e.g., temperature, pressure, light intensity) and converting them into digital values for further analysis or control.
  - Digitizing analog video signals for image processing, compression, or transmission.

Analog to Digital Convertor

- **Working Principle:**
- Sampling: The analog signal is sampled at regular intervals to capture its voltage or current level at discrete points in time.
- Quantization: Each sample is assigned a discrete digital value based on its corresponding voltage or current level. The resolution of the ADC determines the number of possible digital values, typically represented by binary codes.
- Encoding: The digital values are encoded and represented in binary format, usually as binary numbers.
- Conversion: The encoded binary data is processed by internal circuitry within the ADC to convert the analog signal into a digital representation.
- Output: The digital representation of the analog signal is made available as output, typically in parallel or serial digital format, for further processing or transmission.

Pulse Width Modulation (PWM)





Pulse Width Modulation (PWM)

- Concept: Pulse Width Modulation (PWM) is a modulation technique used to encode information in the form of a pulsing signal. It involves varying the width or duration of the pulses while keeping the frequency constant. PWM is commonly used to control the power output or simulate analog signals using digital devices.
- Usage: PWM is widely used in applications where precise control of power, speed, or intensity is required. It is commonly used in motor speed control, LED dimming, audio amplification, and many other applications.
- Examples:
  - Controlling the speed of a DC motor by varying the duty cycle of the PWM signal.
  - Dimming the brightness of an LED by adjusting the duty cycle of the PWM signal.
  - Generating audio signals by modulating the pulse width of the PWM signal.
  - Power regulation in switch-mode power supplies by controlling the duty cycle of the PWM signal.

Pulse Width Modulation (PWM)

- **Working Principle:**
- Signal Generation: A PWM signal is generated by toggling a digital signal (usually a square wave) between high and low states at a fixed frequency.
- Duty Cycle: The duty cycle represents the ratio of the pulse width (duration of the high state) to the total period of the PWM signal. It determines the average power delivered or the level of control applied.
- Analog Simulation: By varying the duty cycle, the PWM signal can simulate different analog values. A higher duty cycle corresponds to a higher average voltage or power output, while a lower duty cycle corresponds to a lower average voltage or power output.
- Filtering: In applications where PWM is used to simulate analog signals, such as audio amplification, a low-pass filter is often employed to smooth out the PWM signal and reconstruct the original analog waveform.

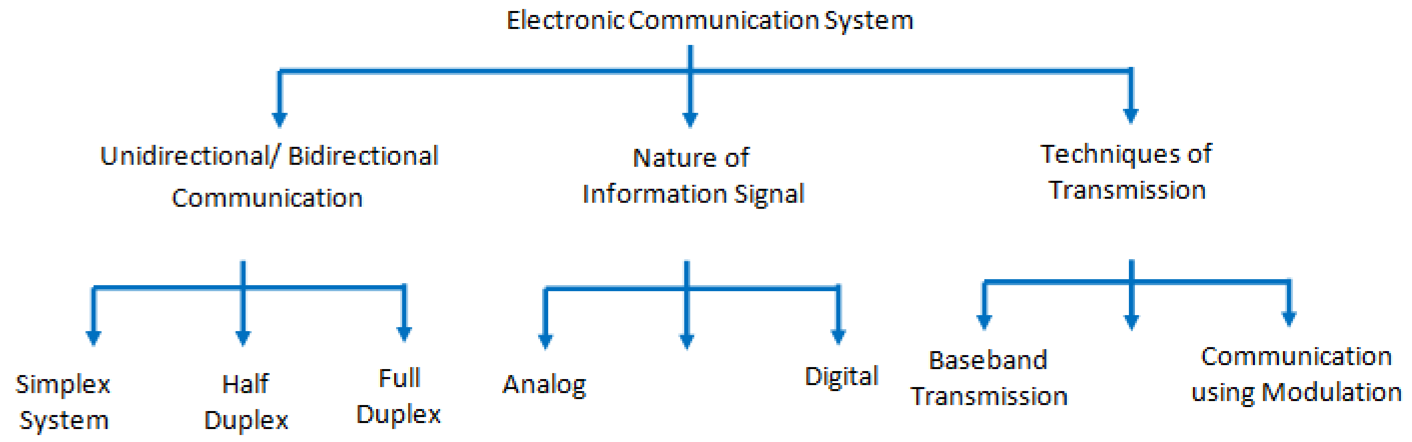
Communication in Embedded Systems

- Embedded systems often require communication capabilities to interact with other devices, exchange data, and perform various tasks. Here are some common communication types used in embedded systems:
- **Serial Communication:**
    - UART (Universal Asynchronous Receiver-Transmitter): It is a popular asynchronous serial communication protocol used for point-to-point communication between two devices.
    - SPI (Serial Peripheral Interface): It is a synchronous serial communication protocol typically used for short-distance communication between a microcontroller and peripheral devices.
    - I2C (Inter-Integrated Circuit): It is a multi-master, multi-slave synchronous serial communication protocol used for communication between ICs on the same board.

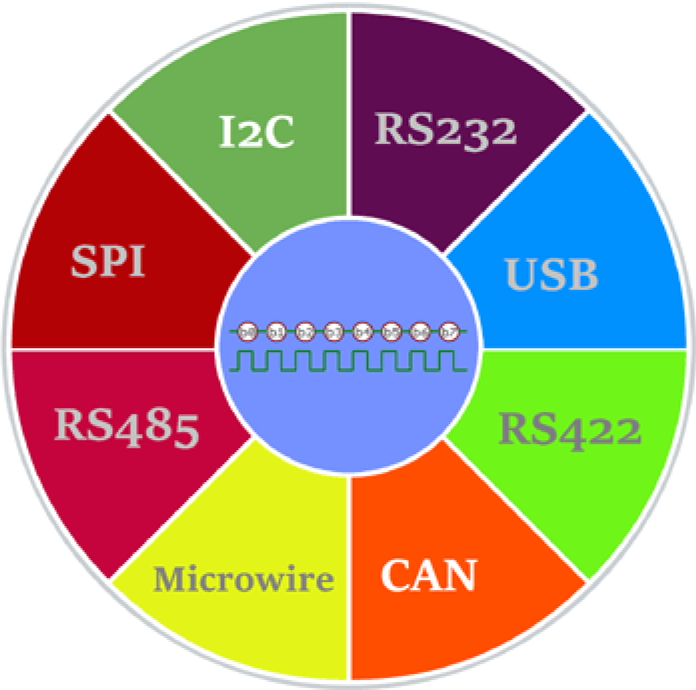
Communication in Embedded Systems

- **Parallel Communication:**
  - GPIO (General Purpose Input/Output): GPIO pins can be configured as parallel communication interfaces to exchange data between microcontrollers or between microcontrollers and other devices.
  - Parallel Bus: It is a collection of multiple data lines used for high-speed parallel communication between devices.
- **Wireless Communication:**
  - Wi-Fi (Wireless Fidelity): Allows devices to connect and communicate over a wireless network, enabling remote access, data transfer, and internet connectivity.
  - Bluetooth: Provides short-range wireless communication between devices, commonly used for wireless audio streaming, device pairing, and data transfer.
  - Zigbee: A low-power wireless communication protocol suitable for applications requiring low data rates, long battery life, and mesh networking capabilities.
  - RFID (Radio Frequency Identification): Enables wireless identification and tracking of objects using radio frequency signals.

Communication Systems



Communication Systems



Serial Communication Protocols

Communication in Embedded Systems

- **Ethernet**:
  - Ethernet: Enables wired network communication, commonly used for connecting embedded systems to local area networks (LAN) or the internet.
- **CAN (Controller Area Network):**
  - CAN: A robust and reliable serial communication protocol designed for use in automotive and industrial applications, allowing communication between multiple devices over a single bus.
- **USB (Universal Serial Bus):**
  - USB: Provides a standardized interface for connecting various peripherals to embedded systems, allowing data transfer and device control.
- **MQTT (Message Queuing Telemetry Transport):**
  - MQTT: A lightweight messaging protocol suitable for machine-to-machine communication, often used in IoT applications to exchange data between embedded devices and cloud services.
- These communication types offer different features, data rates, ranges, and power requirements, allowing embedded systems to communicate with other devices efficiently based on their specific requirements and constraints. The choice of communication type depends on factors such as distance, speed, power consumption, and the nature of the application.

جامعة الإسكندرية
ALEXANDRIA
UNIVERSITY
كليـة الحاسبات وعلـوم البيـانات

Section 1: Introduction to Embedded Systems

Internet of Things: Theory and Applications

What are Embedded Systems?

- Computing systems embedded within electronic devices.
- An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, often with real-time computing constraints.
- Combination of computer hardware & software, that is specifically designed for a particular kind of application device.
- All components must use the memory optimally.

Embedded Software

- Each software component execution speed must be optimum
- Software must have controlled complexity and must be thoroughly tested and debugged for errors

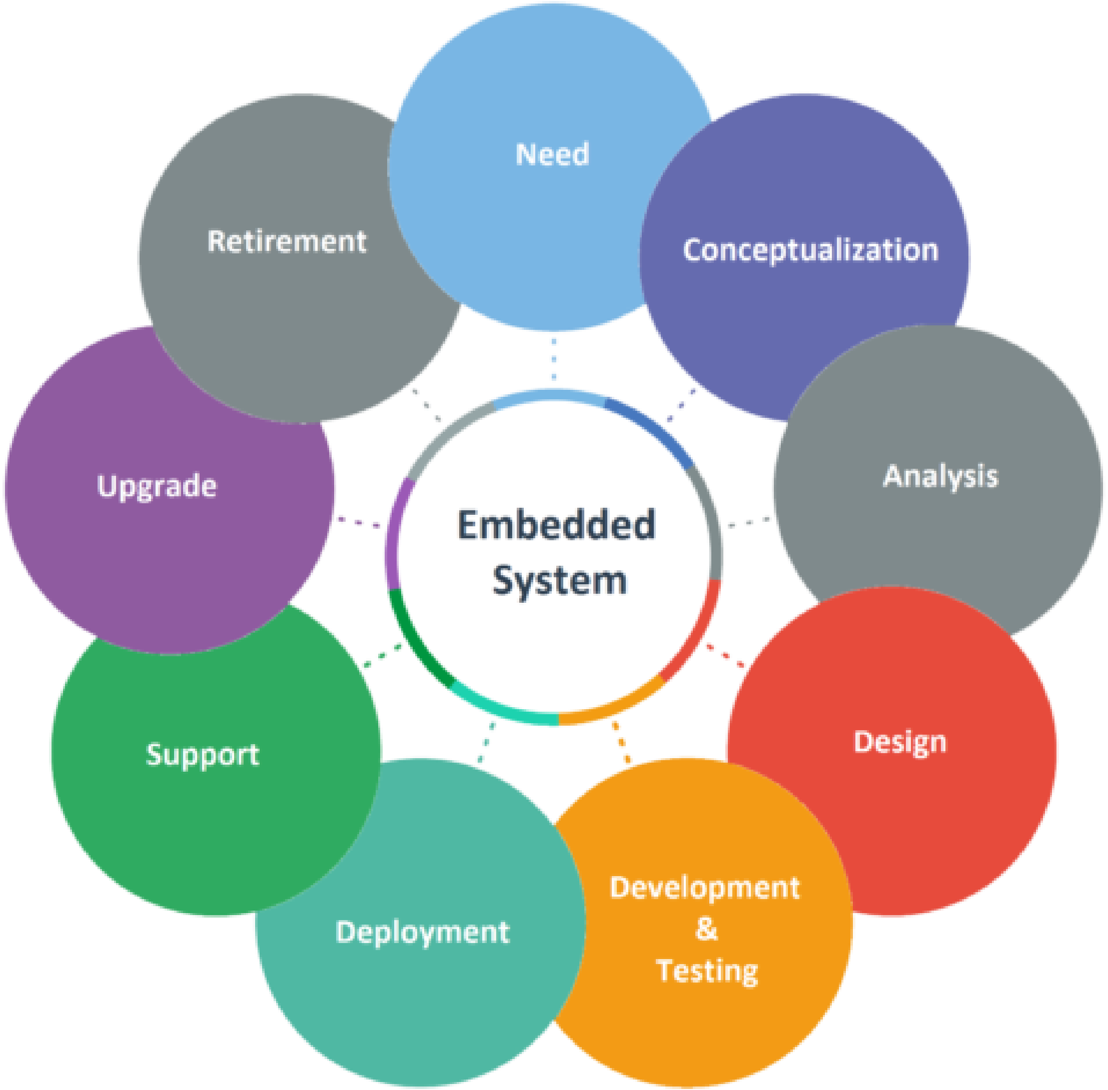
Features of Embedded Systems

- Embedded Systems are the **modern compacted devices** with multifunction capabilities.
- An embedded system performs **pre-defined tasks**, unlike a general-purpose personal computer.
- An embedded system is **a programmed hardware device**. A programmable hardware chip is the platform and it is programmed with particular applications.
- Embedded systems are not always standalone devices. Many embedded systems consist of small, computerized parts within a larger device that serves a more general purpose.

Features of Embedded Systems

- The **program instructions** written for embedded systems are referred to **as firmware.**
- The program stored in **read-only memory or Flash memory** chips.
- They run with limited computer hardware resources: **little memory, small or non-existent keyboard and/or screen.**

Life Cycle of Embedded Systems



History of Embedded System

Date	Event	Comments
1947	1st transistor	Bell Labs
1951	1st solid state transistor	Bell Labs
1958	1st IC	Jack Kilby (MSEE ’50) @TI Winner of 2000 Nobel prize
1971	1st microprocessor	Intel (calculator market)
1974	Intel 4004	2300 transistors
1978	Intel 8086	29K transistors
1989	Intel 80486	1M transistors
1995	Intel Pentium Pro	5.5M transistors
2008	Intel Montecito	1.7B transistors

Characteristics of Embedded Systems

Special Purpose	Tightly Constrained	Reactive and Real Time	Hardware and Software Co-exist
Reliability	Maintainability	Efficiency (Cost-Wight-Energy-Run-time)	Safety & Security

Specialties of Embedded Systems

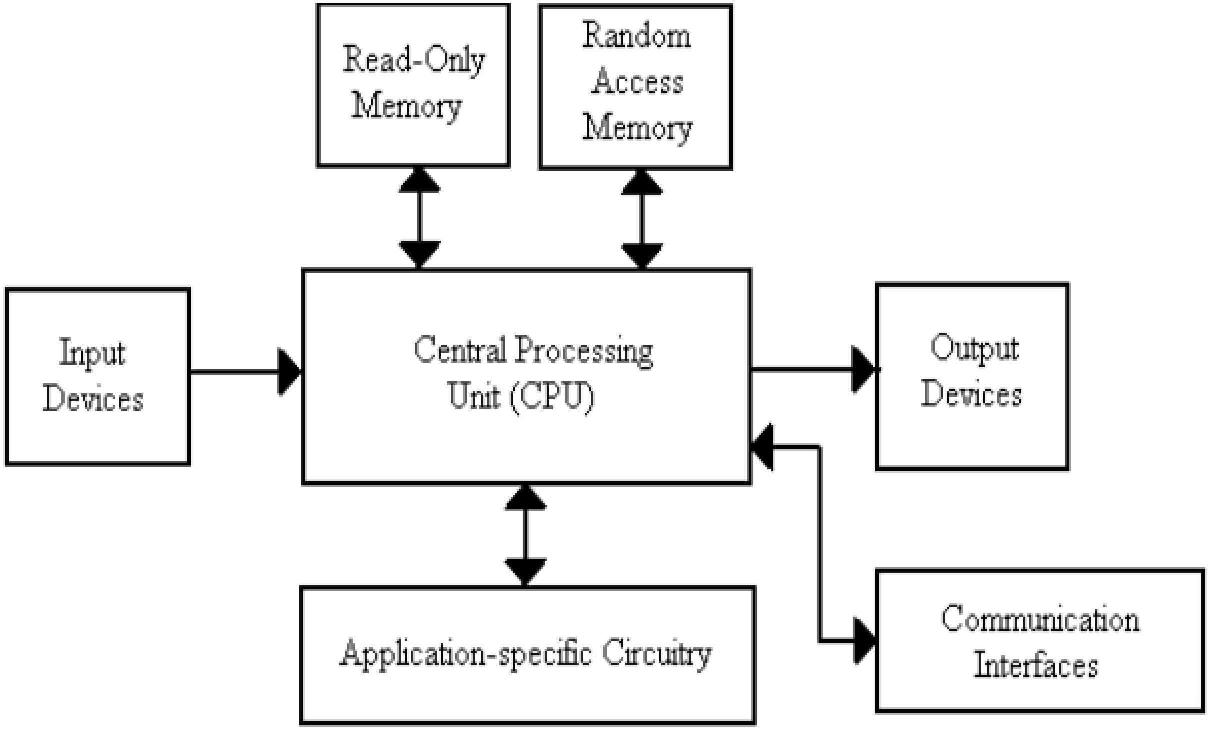
- High performance
- Low power consumption
- Low cost
- Small size
- High software upgradation capability

Classification of Embedded Systems

Based on functionality and performance requirements, embedded systems are classified as :

1. Stand-Alone Embedded Systems
2. **Real-Time Embedded Systems**
3. Networked Information Appliances
4. Mobile Devices

Block Diagram of Embedded System



Main Parts of Any Embedded System

- Microprocessor / Microcontroller System
- Memory (and Off chip)
- Sensors On-chip
- Converters (A-D and D-A)
- Actuators
- Software
- Communication path with the interacting environment

Computer Brain/Control Unit

- The brain is often a Microprocessor or Microcontrollers.
- A Processor is the heart of the Embedded System

Types of Processors

1. General Purpose Processor (GPP)
   A. Microprocessor (uP)
   B. Microcontroller (uC)
   C. Embedded Processor (EP)
   D. Digital Signal Processor (DSP)
2. Application Specific System Processor (ASSP)
3. Multi Processor System using GPPs

About Micro-controller

- It has a CPU in addition to a fixed amount of **RAM,ROM and other peripherals all on a single chip.**
- Other peripherals like **TIMER,ADC,DAC,I/P-O/P PORTS, Interrupt Controllers**
- It is also called as a **mini computer** or computer on single chip.
- Micro-Controllers systems are considered embedded systems
- Example:- Motorola 6811, Intel 8051, PIC 16X, AVR Atmega, ARM Cortex, etc …
- NB: The microcontroller chip used in the ESP32 is the **Tensilica Xtensa LX6**

Definition of a Microcontroller

- A microcontroller is a single silicon chip with memory and all Input/output peripherals on it.
- Hence a microcontroller is also popularly known as a single chip computer.

A Micro-controller has

- Central Processing Unit (CPU)
- Memory
- ROM ==> Read Only Memory
- RAM ==> Random Access Memory
- Input Devices
- Output Devices
- Communication interfaces
- Application Specific Circuitry

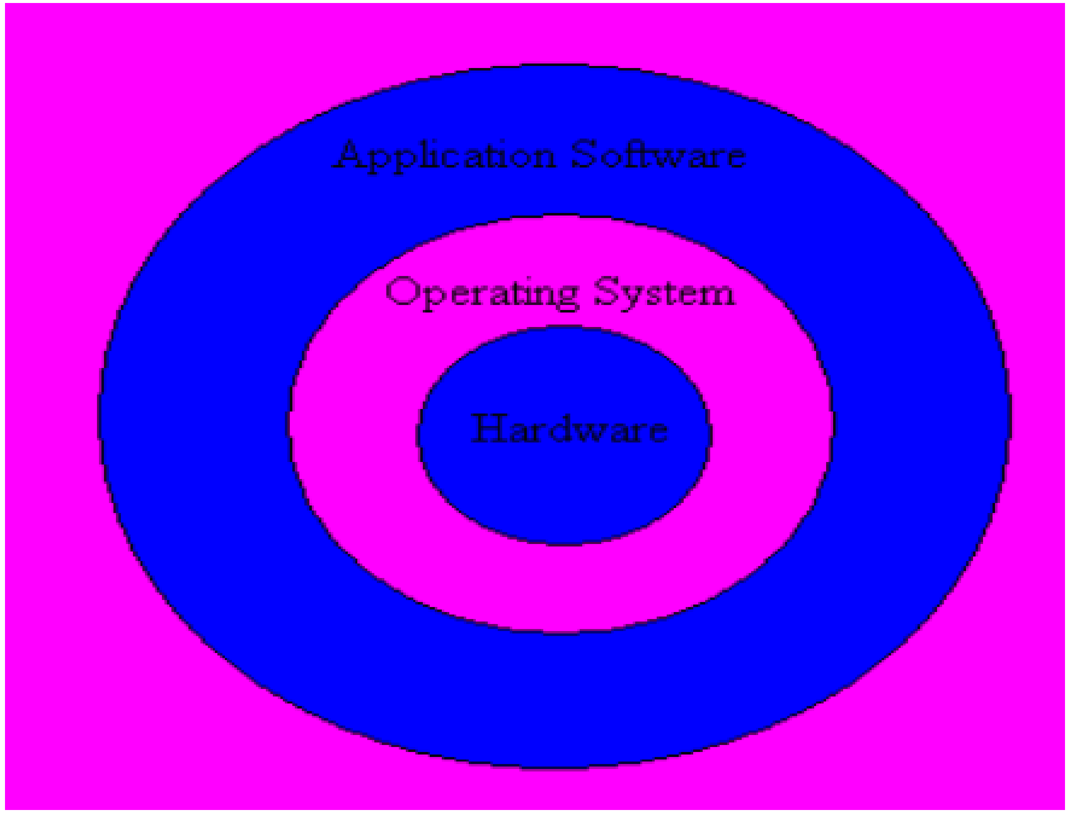
Main Features of Micro-controller

- Arithmetic and logic unit
- Memory for storing program
- EEPROM for nonvolatile data storage
- RAM for storing variables and special function registers
- Input/output ports
- Timers and counters
- Analog to digital converter
- Circuits for reset, Brown out, power up, serial programming, debugging
- Instruction decoder and a timing and control unit
- Serial communication port

System on Chip & System in Package

- Integrating all components of a computer or other electronic system into a single integrated circuit (chip).
- A typical application is in the area of embedded systems.

Layered Architecture of Embedded Systems



Embedded System Development Process

- Requirements (Application and UI)
- Select Microcontroller and H/W
- Design Hardware for your Application
- Draw Flow Chart and Write Software Code
- Write your HEX Code to Microcontroller
- Test your Prototype Product
- Start your Product Production

Micro-controller Selection

- Microcontrollers can classified like
  - 8 bit microcontroller
  - 16 bit microcontroller
  - 32 bit microcontroller
  - DSPs
- ***NB: The ESP32 microcontroller is a 32-bit microcontroller, not a DSP (Digital Signal Processor). It features a 32-bit Tensilica Xtensa LX6 microprocessor core, which is a highly configurable and extensible architecture designed for embedded systems.***

Embedded Systems Programming

- Assembly language was the pioneer for programming embedded systems till recently.
- Nowadays there are many more languages to program these systems.
- Some of the languages are C, C++, Ada, Forth, and Java together with its new enhancement
J2ME.
- The presence of tools to model the software in UML, SDL is sufficient to indicate the maturity of embedded software programming.

Embedded Systems Programming

- The majority of software for embedded systems is still done in C language.
- Recent survey indicates that approximately % of the embedded software is still being done in C language.
- C++ is also increasing its presence in embedded systems.
- As C++ is based on C language, thus providing programmer the object oriented methodologies to reap the benefits of such an approach.
- Mainly, procedure-oriented C is used in most embedded systems programming
- Object-Oriented Programming C++ and Java languages may also be used in embedded systems programming.
- Embedded programming is such that methods to optimize the system memory requirements are also used.

Embedded Systems Programming

- C is very close to assembly programming and it allows very easy access to underlying hardware.
- A huge number of high quality compilers and debugging tools are available for the C language.
- Nowadays in order to gain the extra benefits of C++ in addition to avoiding buggy execution, experts are doing efforts to identify a subset of C++ that can be used in embedded systems and this subset is called “Embedded C++
- *NB: The programming language used for programming the ESP32 using the Arduino framework is a variant of C+*

Micro-controller Main Manufacturers

- ATMEL
- MICROCHIP
- TEXAS INSTRUMENTS
- INTEL
- FREESCALE
- ***NB: The Tensilica Xtensa LX6 is a microprocessor core in ESP32 developed by Tensilica, a company that specializes in customizable and extensible processor solutions. Tensilica was acquired by Cadence Design Systems in 2013. Therefore, Cadence Design Systems is the manufacturer of the Tensilica Xtensa LX6 core.***

جامعة الإسكندرية
ALEXANDRIA
UNIVERSITY
كليـة الحاسبات وعلـوم البيـانات

Section 2:
ESP32 based IoT & Embedded Systems

Internet of Things: Theory and Applications

Arduino Based Embedded Systems

- Introduced in 2005 as a project for students at the Interaction Design Institute Ivrea in Ivrea,
  Italy**, Arduino is a single board microcontroller.**
- An Arduino board consists of an **Atmel 8-bit AVR microcontroller** with complementary components to facilitate programming and incorporation into other circuits [2].
- **Arduino can sense the environment** by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators.

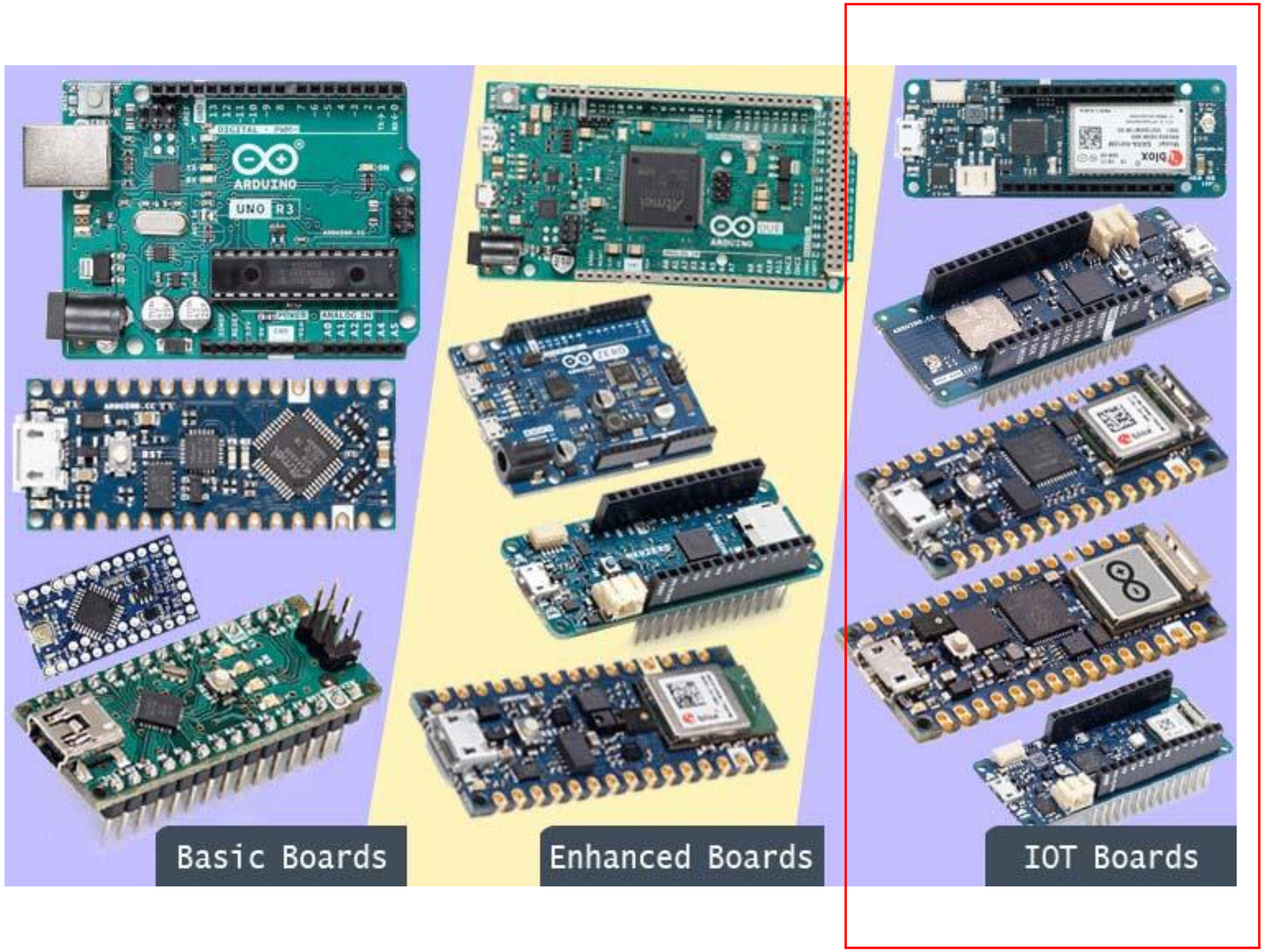
What is an Arduino?

• The boards can be assembled or purchased eassembled; the open-source IDE can be downloaded for free.
• The **Arduino programming language is very simple** and follows C like syntax.
• Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Processing).

Arduino Based Embedded Systems

- An open-source hardware platform based on an Atmel AVR 8-bit microcontroller and a C++ based IDE.
- Over 300000 boards have been manufactured.
- Arduino Due is based on a 32-bit ARM Cortex.

Types of Arduino Boards



ESP32 Based Embedded Systems

- The ESP32 was developed by Espressif Systems and was first released in 2016.
- It integrates various components including a dual-core Xtensa LX6 microprocessor, Wi-Fi and Bluetooth modules, GPIO pins, ADC, SPI, I2C, UART, and more.
- The ESP32 can be used with a wide range of sensors such as temperature sensors, humidity sensors, motion sensors, and many others to gather data from the environment.
- It can also integrate with cloud platforms like Firebase, allowing for data storage, real-time updates, and remote control of devices.

ESP32 Based Embedded Systems

- The programming syntax for the ESP32 is based on C++, similar to the Arduino programming language. It also supports other programming languages such as MicroPython and JavaScript.
- The ESP32 has gained significant popularity and has sold millions of units worldwide.
- It is compatible with various development platforms and IDEs, including the Arduino IDE, ESP-IDF (Espressif IoT Development Framework), PlatformIO, and more.
- The ESP32's versatility, low power consumption, and extensive features have made it a popular choice for a wide range of IoT applications, including home automation, robotics, environmental monitoring, and wearable devices.

Advantages of ESP32

- **Dual-Core Processor:** The ESP32 features a powerful dual-core Xtensa LX6 microprocessor, which allows for efficient multitasking and handling complex tasks.
- **Wi-Fi and Bluetooth Connectivity:** The ESP32 has built-in Wi-Fi and Bluetooth modules, providing seamless wireless connectivity for IoT applications. It supports various Wi-Fi protocols (e.g., 802.11 b/g/n) and Bluetooth standards (e.g., Bluetooth Low Energy).
- **Low Power Consumption:** The ESP32 is designed to be energy-efficient, enabling battery-powered and energy-conscious applications. It offers multiple low-power modes and features that help conserve power.
- **Extensive I/O and Peripheral Support:** With a wide range of GPIO pins, SPI, I2C, UART, ADC, DAC, and other peripherals, the ESP32 provides flexibility in connecting and controlling external devices and sensors.
- **Rich Development Ecosystem:** The ESP32 benefits from a thriving development ecosystem with comprehensive documentation, community support, and numerous libraries and examples available. It is compatible with popular development platforms and IDEs.
- **Cost-Effective Solution:** The ESP32 offers a cost-effective solution for IoT projects, providing a powerful set of features at an affordable price point. Its competitive pricing makes it an attractive choice for both hobbyists and commercial applications.
- **Versatility and Scalability:** The ESP32 is highly versatile, suitable for a wide range of IoT applications. Its scalability allows it to be used in projects of various sizes, from small prototypes to large-scale deployments.

Technical Specifications of ESP32

- Microcontroller: Tensilica Xtensa LX6 dual-core processor
  - Main frequency: Up to 240 MHz
  - Computing power: 600 DMIPS
- Memory:
  - Internal SRAM: 520 KB
  - External SPIRAM: Up to 16 MB
- Wireless Connectivity:
  - Wi-Fi: 802.11 b/g/n, supporting WPA/WPA2 and WAPI
  - Bluetooth: Bluetooth v4.2 BR/EDR and BLE (Bluetooth Low Energy)

Technical Specifications of ESP32

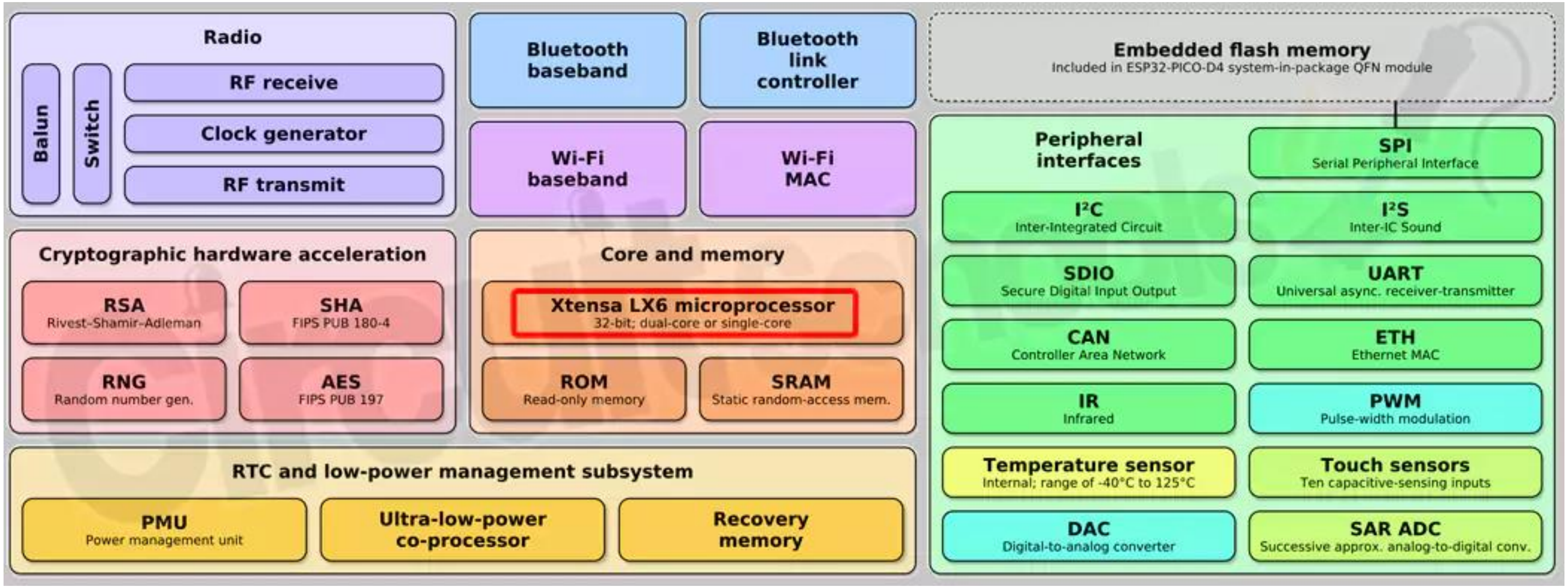
- Operating Voltage: 2.2V to 3.6V
- Digital I/O Pins: 34 GPIO pins (including capacitive touch sensors)
- Analog Inputs: 12-bit SAR ADC with up to 18 channels
- **Interfaces:**
  - UART
  - SPI
  - I2C
  - I2S
  - PWM
  - SD/SDIO/MMC Host Controller
  - IR Remote Controller (TX/RX)

Technical Specifications of ESP32

- **<u>Peripherals:</u>**
  - 4 × SPI interfaces
  - 2 × I2C interfaces
  - 3 × UART interfaces
  - 2 × I2S interfaces
  - 16 × PWM channels
  - 10 × capacitive touch sensors
  - SD/SDIO/MMC host controller
  - Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- **<u>Security:</u>**
  - WPA/WPA2 personal and enterprise security modes
  - Cryptographic hardware acceleration: AES, SHA-2, RSA, and ECC
- Operating Temperature: -40°C to +125°C
- Development Platforms: Arduino IDE, ESP-IDF (Espressif IoT Development Framework)

Layout of ESP32

ESP32 Block Diagram



How to Program the ESP32 on Arduino IDE

- Get an ESP32 board and USB cable
- Download the Arduino environment using this link: https://www.arduino.cc/en/software

(Download version 1.8 is preferred)

- Go to File > Preferences and post in the Additonal Boards URL the following: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json then Ok.
- Go to Tools > Board > Boards Manager > then search for esp32 by espressif systems and install it

How to Program the ESP32 on Arduino IDE

- Then, Go to Tools > Boards “DOIT ESP32 DEVKIT V-1” > select the DOIT ESP32 DEVKIT V1
- Connect the ESP32 to the Device
- Then, Go to the Tools > Ports > Select the Port which is the ESP32 is connected on.
- Then, you can start coding!

Arduino Syntax on Arduino IDE

- Arduino programs run in form of sketches.
- Main Structure of Arduino Program is composed of two main functions with no return which is the void setup and the void loop functions.

Arduino Syntax on Arduino IDE

- **serial.println(value);**
  – Prints the value to the Serial Monitor on your computer
- **pinMode(pin, mode);**
  – Configures a digital pin to read (input) or write (output) a digital value
- **digitalRead(pin);**
  – Reads a digital value (HIGH or LOW) on a pin set for input
- **digitalWrite(pin, value);**
  – Writes the digital value (HIGH or LOW) to a pin set for output

Arduino Syntax on Arduino IDE

- Analog Write using analogWrtie & ledc Fundtion:
  - Syntax: ledc(pin, value);
  - Example: analogWrite(9, 127);
  - Description: Writes an analog value (PWM) to the specified pin.
- Analog Read:
  - Syntax: analogRead(pin);
  - Example: int value = analogRead(A0);
  - Description: Reads the analog value from the specified analog pin (0-1023).

Arduino Syntax on Arduino IDE

- Serial Communication:
  - Syntax: Serial.begin(baudRate);
  - Example: Serial.begin(9600);
  - Description: Initializes the serial communication with the specified baud rate.
- Serial Print:
  - Syntax: Serial.print(value);
  - Example: Serial.print("Sensor value: "); Serial.println(sensorValue);
  - Description: Prints the specified value to the serial monitor.

- ledcSetup() function:

- First argument: PWM channel number (0 to 15). This specifies the channel you want to configure.
- Second argument: PWM frequency in Hertz (Hz). This sets the frequency at which the PWM signal will operate.
- Third argument: PWM resolution in bits (1 to 16). This defines the number of bits used to represent the PWM duty cycle, affecting its precision.
- ledcAttachPin() function:

- First argument: GPIO pin number to which you want to attach the PWM channel.
- Second argument: PWM channel number (0 to 15). This specifies the channel to be attached to the GPIO pin.
- ledcWrite() function:

- First argument: PWM channel number (0 to 15). This specifies the channel you want to control.
- Second argument: PWM duty cycle value. The value ranges from 0 to the maximum value based on the resolution set in ledcSetup(). For example, if the resolution is set to 8 bits (0-255), the duty cycle value should be within that range.

Arduino Syntax on Arduino IDE

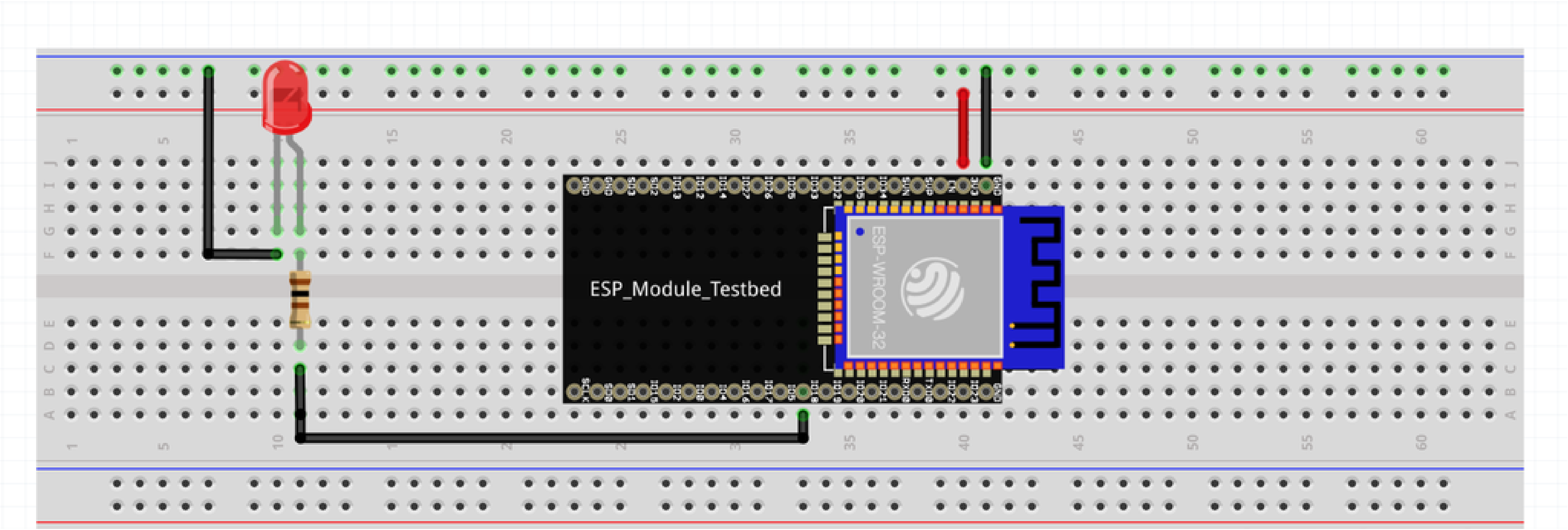
- If Statement:
  - Syntax: if(condition) { /* code */ }
  - Example: if(sensorValue > 500) { digitalWrite(LED_PIN, HIGH); }
  - Description: Executes the code block inside the if statement if the condition is true.
- For Loop:
  - Syntax: for(initialization; condition; increment) { /* code */ }
  - Example: for(int i = 0; i < 5; i++) { digitalWrite(LED_PIN, HIGH); delay(500); digitalWrite(LED_PIN, LOW); delay(500); }
  - Description: Executes the code block inside the for loop for a specified number of times.
- Functions:
  - Syntax: returnType functionName(parameters) { /* code */ }
  - Example: int add(int a, int b) { return a + b; }
  - Description: Defines a reusable block of code with the specified function name, parameters, and return type.

جامعة الإسكندرية
ALEXANDRIA
UNIVERSITY
كليـة الحاسبات وعلـوم البيـانات

Section 3:
ESP32 Basic Practice Projects

Internet of Things: Theory and Applications

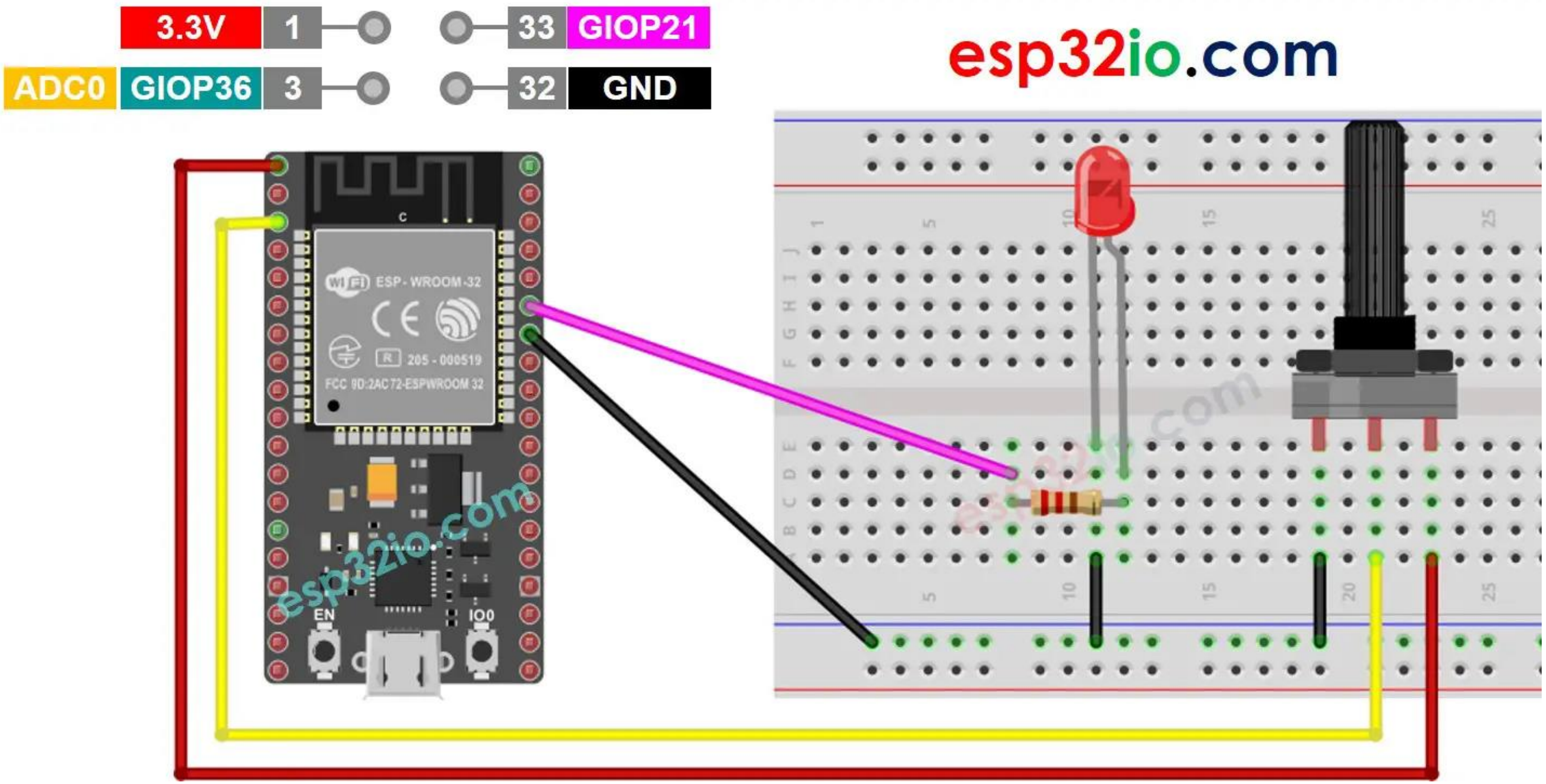
Blinking LED: Circuit Schematic



Blinking LED: Code

```
const int ledPin = 5;
void setup() {
  // setup pin 5 as a digital output pin
  pinMode (ledPin, OUTPUT);
}
void loop() {
  digitalWrite (ledPin, HIGH);   // turn on the LED
  delay(500);    // wait for half a second or 500 milliseconds
  digitalWrite (ledPin, LOW);   // turn off the LED
  delay(500);    // wait for half a second or 500 milliseconds
}
```

Fading LED: Schematic



Fading LED: Code

```
// ESP32 LED Control with Potentiometer
#include <Preferences.h>

const int ledPin = 33;        // GPIO033 pin connected to the LED
const int potentiometerPin = 35;  // GPIO03 pin connected to the potentiometer

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(potentiometerPin, INPUT);
  Serial.begin(115200);
}
```

Fading LED: Code

```
void loop() {
  // Read the value from the potentiometer
  int potValue = analogRead(potentiometerPin);

  // Map the potentiometer value to the LED brightness range
  int ledBrightness = map(potValue, 0, 4095, 0, 255);
  Serial.println(ledBrightness);

  // Set the LED brightness using PWM
  analogWrite(ledPin, ledBrightness);
  delay(10);
}
```

جامعة الإسكندرية
ALEXANDRIA
UNIVERSITY
كليـة الحاسبات وعلـوم البيـانات

Section 4: Assignment

Internet of Things: Theory and Applications

Assignment: Theoretical

- Write down a brief description of the Tensilica Xtensa LX6 Daul core microprocessor.
- Compare micro-controllers to micro-processors with at least 5 different comparisons.
- Compare digital signals to analog signal with at least 5 different comparisons.

ESP32 Projects Assignments:

- A student is tasked with designing and implementing an ESP32-based system that includes a bar graph display, a push button-controlled LED, and an LDR (Light Dependent Resistor) light sensor connected to a buzzer and LED alarm. The system should provide visual and audible indications based on the ambient light levels and user inputs. The specific requirements are as follows:

1- Bar Graph Display:

- Design a bar graph display using multiple LEDs to visually represent a range of values.
- The bar graph should consist of a series of LEDs that light up sequentially based on a specific condition or input.
- Implement a function that maps the input values to the LED bar graph, ensuring that each LED corresponds to a specific range or threshold.

2- Push Button-Controlled LED:

- Connect a push button to the ESP32, which can toggle an LED on and off.
- Pressing the button should turn the LED on, and releasing the button should turn it off.
- Implement a debouncing mechanism to handle any potential glitches or bounces when the button is pressed or released.

3- LDR Light Sensor with Buzzer and LED Alarm:

- Connect an LDR light sensor to the ESP32 to measure ambient light levels.
- When the light level goes below a certain threshold, activate a buzzer to emit an audible alarm sound.
- Simultaneously, turn on an LED to provide a visual indication of the alarm state.
- Once the light level rises above the threshold, the buzzer should stop, and the LED should turn off.

Bonus in any of the previous 3 problems

- By incorporating TinkerCAD for simulation and Fritzing for the circuit schematic, the student will enhance their design process by providing a realistic simulation experience and a visual representation of the circuitry. These tools will further support the documentation and presentation of the system, allowing for a more comprehensive and visually appealing demonstration of the ESP32-based system's functionality and design.

Bonus Projects

- Use analog output IR sensor to measure the proximity around you with LED and buzzer Alarm.
- Use LCD module to print your name on it.
- Use Ultrasonic sensor to measure any distance and print measured distance on Arduino serial with LED and Buzzer Alarm
- Use PIR sensor and print measured distance on Arduino serial with LED and Buzzer Alarm
- Use Flame sensor and print measured values on Arduino serial with LED and Buzzer Alarm
- Connect with GPS module and identify your location and print it on the Arduino IDE serial.

Double Bonus Project

- Read any analog sensor data from any sensor and print it on LCD with LED and Buzzer alarm given upon any threshold point and alarm message.

Ideas for Bonus Projects

1. Line follower/Path follower
2. Obstacles Avoider
3. Automatic car parking
4. Driverless car
5. Quad copter
6. Water-level detection in soil
7. Surveillance System
8. Dancing/ Funny Robot
9. Smart phone Garage Door Opener
10. Intrusion alarm
11. Thermostat
12. Balance multirotor motor using arduino & acceleromter
13. Email notifier
14. LED Matrix Control
15. Maze Solver Robot

Thank you