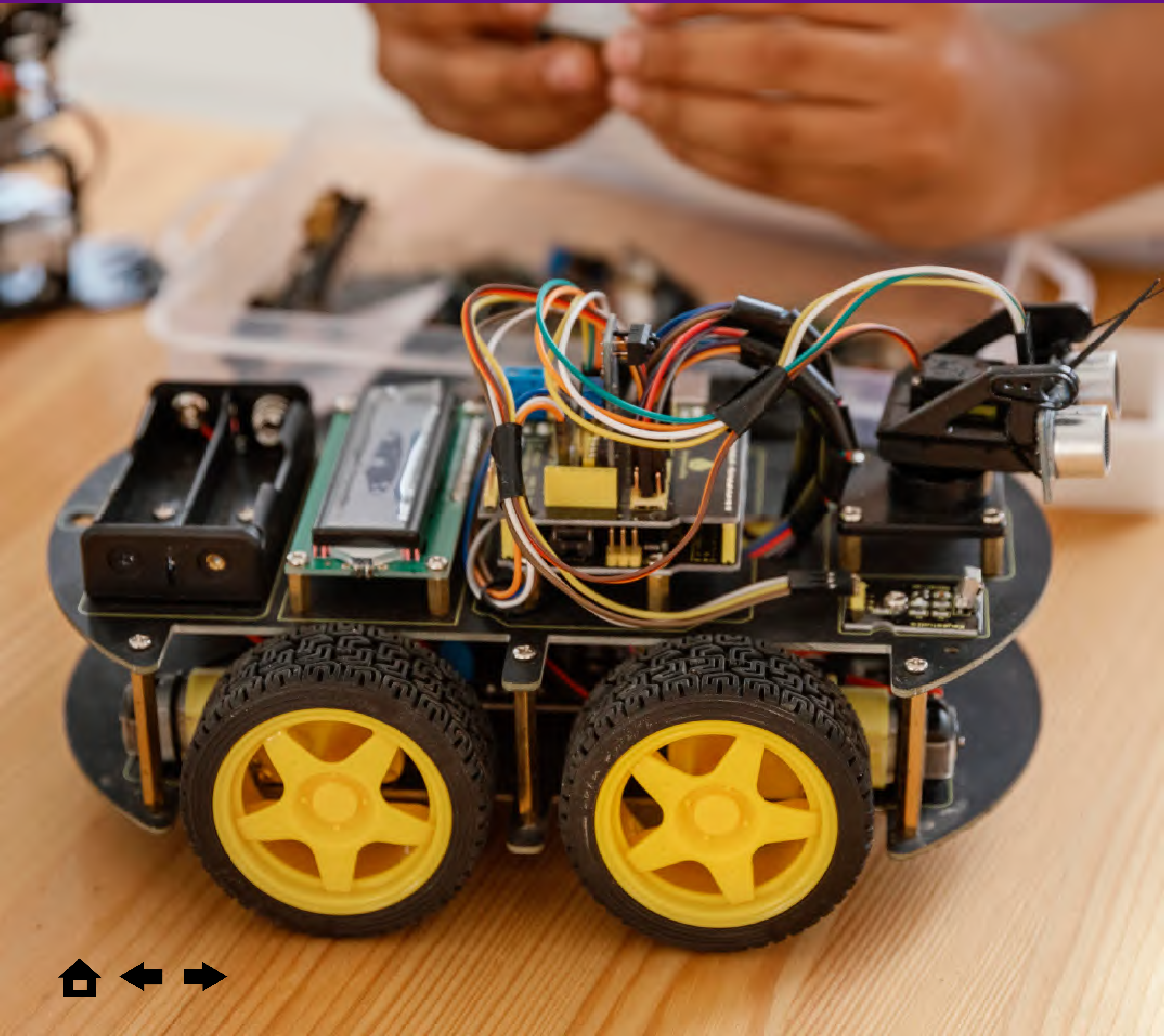
SECTION 4

DIGITAL AND ANALOGUE SYSTEM DESIGN 1



Robot Design Methodologies

Digital and Analogue System Design

Introduction

Welcome to the exciting world of electronic circuit design, a fundamental aspect of both digital and analogue systems. Electronic circuits are the backbone of modern technology, enabling everything from simple household gadgets to complex robots and sophisticated communication systems. Understanding the components of an electronic circuit and their functions would help you acquire knowledge on the various components of an electronic circuit. It is crucial for anyone interested in robotics, automation, or any field that relies on electronic devices to go through this indicator.

In robotics, electronic systems are essential for controlling and coordinating different components to complete specific tasks. Block diagrams are useful tools that provide a clear and simple way to represent these complex systems.

In this lesson you will learn that digital technology generates, stores, and processes data in terms of positive and nonpositive states, represented by the number 1 and the number 0. Data is transmitted or stored as a string of 0s and 1s, each referred to as a bit, and a byte is a string of bits a computer can address individually. Before the digital age, electronic transmission was limited to analogue technology, which conveyed data as electronic signals of varying frequency or amplitude added to carrier waves. This lesson also aims to critically analyse analogue and digital systems and their relation to discrete and continuous-time machine designs. Digital systems use binary digits (0s and 1s) and logic gates to process discrete signals, enabling precise control and decision-making in devices like computers and smartphones. Analog systems use components like resistors, capacitors, and operational amplifiers to interface with sensors and manage real-time control. Through hands-on activities, learners will build and analyse circuits, compare digital and analogue systems, and classify machines based on their inputs and outputs.

At the end of this section, you will be able to:

- Identify the components of an electronic circuit and their functions.
- Properly label and explain block and schematic diagram representation of electronic systems, system inputs and outputs.
- Assemble and test electronic circuits on a solderless breadboard using predesigned schematic diagrams.
- Critically analyse analogue and digital systems and observe how they relate to both discrete and continuous-time machine designs.

Key Ideas:

- Passive components such as resistors, capacitors, and inductors manage energy within circuits, crucial for voltage regulation, signal filtering, and energy storage. Active components like diodes, transistors, and integrated circuits play pivotal roles in signal amplification, current control, and digital logic operations, enabling complex functionalities in electronic systems. Specialised components such as relays and circuit breakers facilitate the control and protection of electrical circuits, while LEDs offer energy-efficient illumination. Understanding the functions and applications of these components, reinforced by analogies and hands-on practice, forms a solid foundation for effective electronic circuit design and troubleshooting.
- Block diagrams are visual representations that use simple blocks to depict different components of an electronic system and their connections, making it easier to understand complex electronic systems. These diagrams highlight system inputs and outputs, where inputs are the signals or data received by the system, and outputs are the actions or data generated by the system. In electronic systems, interconnected components like sensors and actuators play crucial roles; sensors gather environmental data (inputs), while actuators execute physical actions (outputs) based on control signals from the microcontroller. The microcontroller serves as the control block, processing inputs and generating outputs to manage the system. Schematic diagrams complement block diagrams by providing detailed symbolic representations of electronic circuits, showing precise connections between components. Understanding the flow of information, illustrated by arrows in block diagrams, is essential for grasping how different functional blocks interact and how the system operates as a whole.
- Implement electronics circuits and understand the layout of a breadboard and correctly assemble circuits using components like resistors, LEDs, power cells and transistors.
- Use a digital multimeter to measure voltage, current, and resistance in circuits, following safety precautions.
- Digital systems use binary digits (0s and 1s) for precise control and decision-making.
- Digital systems operate on discrete signals using binary digits.
- Logic gates (AND, OR, NOT, etc.) are the building blocks of digital circuits.
- Key components include microcontrollers, finite-state machines, timers, and pulse-width modulation.
- Analogue systems work with continuous signals, representing real-world quantities like sound or temperature.
- Analogue systems handle continuous signals and interface with real-world sensors.
- Key components include resistors, capacitors, and operational amplifiers (op-amps).
- These components perform functions like amplification, filtering, and signal processing.

Why is this important?

Electronic components are the building blocks of an electronic circuit. In building a house, bricks or blocks and other necessary materials are used just as a house is built from bricks, each with a specific role in the structure. Electronic circuits are constructed from components, each contributing to the overall operation. Knowing these components and their roles is the first step towards designing and troubleshooting electronic systems.

Practical Applications

Imagine you want to build a robot. To make it move, sense its environment, and interact with the world, you'll need to design circuits that control motors, process sensor data, and manage power. Each of these tasks requires a solid understanding of electronic components. By mastering this knowledge, you'll be equipped to bring your robotic creations to life and troubleshoot any issues that arise.

INTRODUCTION TO ELECTRONIC CIRCUIT COMPONENTS

Electronic circuits consist of various components that perform specific functions to control the flow of electrical signals. Some common electronic circuit components include resistors, capacitors, inductors, diodes, transistors, integrated circuits (ICs), relays, circuit breakers and LEDs.

Electronic circuits are the foundation of modern technology, enabling the functionality of countless devices and systems we rely on daily. From smartphones and computers to medical equipment and robotic systems, these circuits are integral to their operation. Understanding the components that make up electronic circuits is essential for anyone looking to delve into electronics, robotics, or any technology-related field.

What are Electronic Circuit Components?

Electronic circuit components are the building blocks of circuits, each serving a specific purpose to ensure the circuit functions correctly. These components can be classified into two main categories: Passive and Active components.

Passive Components	Active Components
Resistors	Diodes
Capacitors	Transistors
Inductors	Integrated Circuits
Circuit breaker	Relays
	Light Emitting Diodes

Passive components

Passive components are electronic components that do not require an external power source to operate. They cannot introduce energy into the circuit; instead, they consume, store, and release energy in the form of electrical or magnetic fields.

1. **Resistors** are passive components that limit and regulate the flow of electrical current. This is achieved by reducing the current which helps to control voltage levels in a circuit.

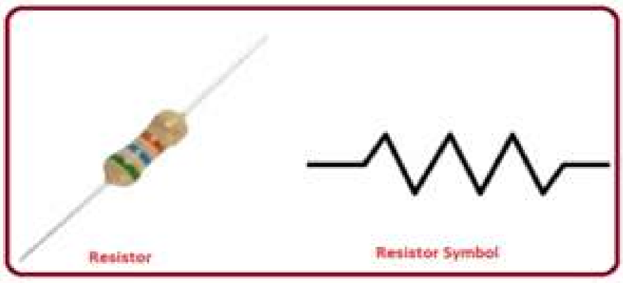


Fig. 4.1: An actual Resistor and a Resistor Symbol

The resistance of a resistor is measured in ohms (Ω). This resistance value determines how much the resistor will limit the current. Ohms are a unit of measurement that quantifies electrical resistance, named after the German physicist Georg Simon Ohm.

Resistors are often identified by a colour code system, which uses coloured bands to indicate their resistance value. This system consists of several coloured rings painted on the resistor, each colour representing a specific number (colours and their corresponding digits are: Black: 0, Brown: 1, Red: 2, Orange: 3, Yellow: 4, Green: 5, Blue: 6 Violet/Purple: 7, Grey: 8 White: 9).

Here's a brief guide to using the colour code system:

a. **First and Second Bands**: These represent the first two digits of the resistance value.

b. **Third Band**: This is the multiplier, which indicates the power of ten to multiply the first two digits by.

c. **Fourth Band (Tolerance Band)**: Usually on the right on its own. This indicates the tolerance of the resistor, which shows how accurate the resistance value is. (usually expressed as a +/- percentage, gold is +/- 5%, Silver +/- 10%, no colour +/- 20%).

For example, the resistance of a resistor with the colour bands brown, black, red close together on the left, and a gold band on its own on the right would be determined the following way:

Brown = 1, Black = 0 gives first two digits 10

Red = 2 gives 10^2

Therefore $10 \times 10^2 = 1000$

So, the resistance is 1000 Ω with a tolerance of ±5%

2. **Capacitors**: A capacitor that stores electrical charge. It can be used in filtering, coupling, and timing applications. They are essential in smoothing voltage fluctuations and blocking direct current.

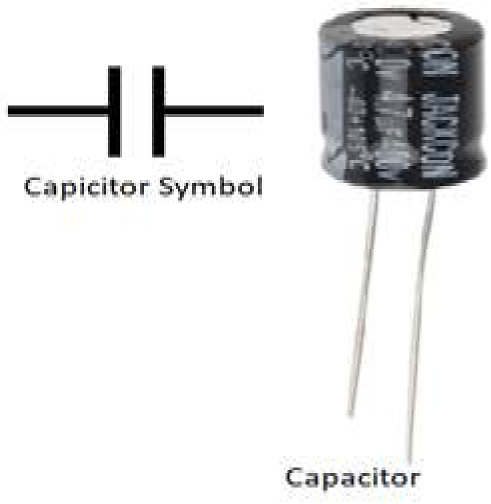


Fig. 4.2: Symbol of a capacitor and an actual capacitor

3. **Inductors**: Inductors store energy in the form of a magnetic field and are used in filtering and energy storage applications. They resist changes in current flow and play a crucial role in AC circuits.

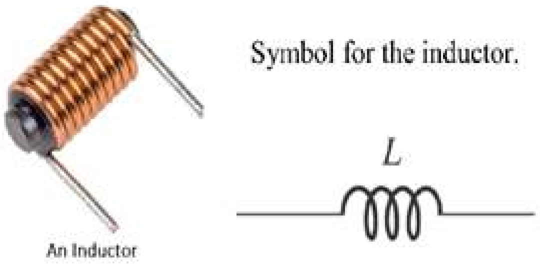


Fig. 4.3: An actual Inductor and the symbol for an inductor

4. **Circuit breaker**: A circuit breaker is an electrical switch designed to protect an electrical circuit from damage caused by excess current. It automatically interrupts the flow of electricity in a circuit when it detects a fault, such as a short circuit or an overload. Circuit breakers are designed to trip (open the circuit) when the current exceeds a certain threshold for a specified period. This helps prevent overheating of wires, damage to equipment, and electrical fires.

   In robotics, maintaining the reliability and safety of electrical systems is crucial for optimal performance and protection of components. A circuit breaker (CB) plays a vital role in this context by protecting the electrical circuits from damage caused by overcurrent or short circuits.

   The auto-resetting circuit breaker is especially beneficial in robotic applications. These breakers are designed to automatically reset after an overcurrent condition has been resolved, eliminating the need for manual intervention. An auto-resetting circuit breaker is ideal for situations where constant human monitoring and manual resetting are impractical or impossible, such as in autonomous or remotely operated robots.

Consider a robot equipped with an auto-resetting circuit breaker located in its Power Distribution Panel (PDP). The PDP is responsible for distributing electrical power to various subsystems of the robot, such as motors, sensors, and control units.

For instance, imagine an autonomous mobile robot used in a warehouse for material handling. If a sudden surge in current occurs in one of the motors due to a temporary obstruction, the auto-resetting circuit breaker in the PDP will trip to prevent damage. Once the obstruction is cleared and the overcurrent condition resolves, the breaker will reset, and the robot will continue its task seamlessly. This capability ensures that the robot can operate efficiently without requiring human workers to manually reset the breaker, thus maintaining productivity and operational continuity.

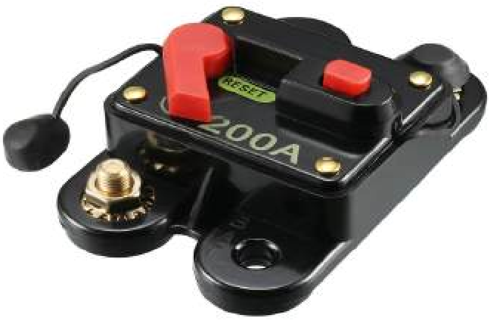


Fig. 4.4: A circuit breaker and a symbol of a circuit breaker

Active Components

Active components are electronic components that require an external power source to operate. They can introduce energy into the circuit and are capable of controlling current flow.

1. **Diodes**: Diodes are semiconductor devices that allow current to flow in one direction only. They are used in rectification, voltage regulation, and signal demodulation.

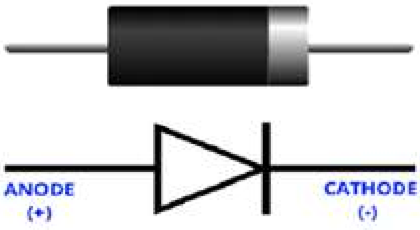


Fig. 4.5: An actual Diode and the symbol for a diode

2. **Transistors**: Transistors are active components that amplify or switch electronic signals. They form the building blocks of digital logic circuits and amplifiers.

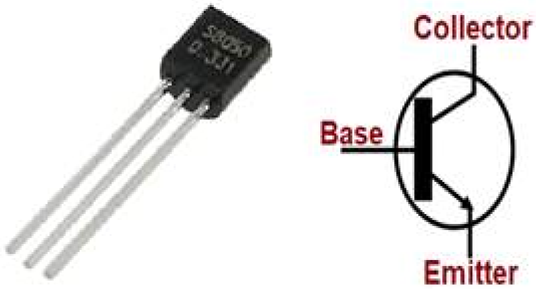


Fig. 4.6: A Transistor and a symbol of a transistor

3. **Integrated Circuits (ICs)**: ICs are complex assemblies of multiple electronic components integrated into a single package. They are used for various functions, such as microcontrollers, memory, and signal processing.

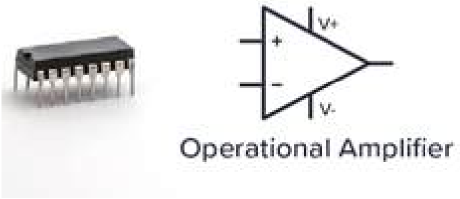


Fig 4.7: An Integrated Circuit and a symbol of an IC

4. **Relay**: A relay is an electrically operated switch that consists of a coil of wire around an iron core, an armature, and one or more sets of contacts. When an electrical current flows through the coil, it creates a magnetic field that attracts the armature, causing the contacts to close or open.

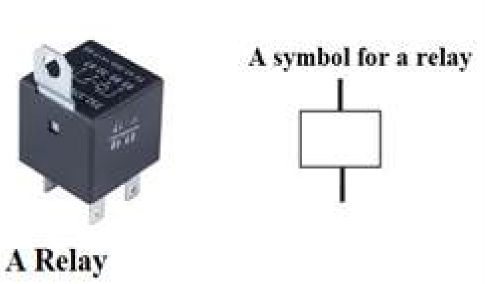


Fig. 4.8: An image of a relay and a symbol for a relay

Relays are simply switches that are used to control flow of voltage in circuits and operate electromechanically. They allow a low-power signal to control a high-power circuit, which is useful for applications where a low-power signal, such as from a microcontroller, needs to control a high-power device like a motor or a heater.

5. **LED (Light Emitting Diode)**: An LED is a semiconductor device that emits light when an electric current passes through it. It consists of a semiconductor

chip mounted on a reflector cup and encapsulated in a transparent or coloured epoxy resin.

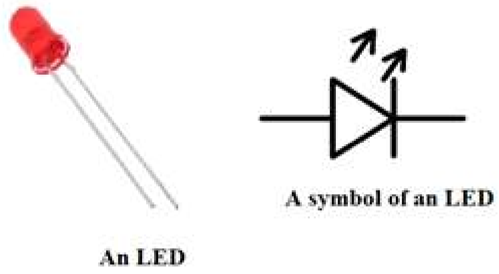
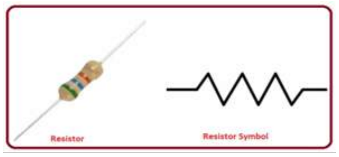
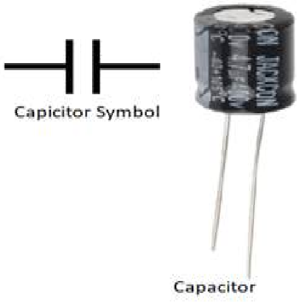
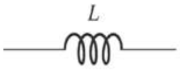
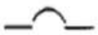
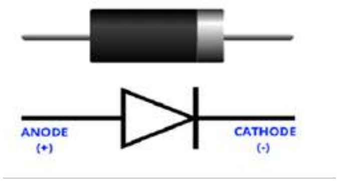


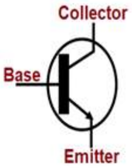
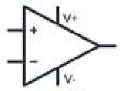
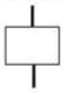
Fig. 4.9: An LED and a symbol of an LED

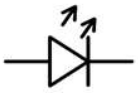
Summary table

Passive Components	Pictures/Symbol	Functions
Resistor		• Resists the flow of electric current, helping to control the amount of current that passes through a circuit. • Think of a resistor like a narrow section of a water pipe. It restricts the flow of water (current) through it. • **Real-World Example:** Used in LED circuits to limit the current, ensuring the LED doesn't burn out.
Capacitor		• Stores and releases electrical energy. They can filter out noise, smooth voltage fluctuations, and help in timing applications. • Imagine a capacitor as a water tank. It can fill up (store charge) and then release water (discharge) when needed. • **Real-World Example:** Found in power supplies to smooth out fluctuations in voltage.

Passive Components	Pictures/Symbol	Functions
Inductor	An Inductor Symbol for the inductor. 	- Stores energy in a magnetic field when electric current flows through them. They are used in filtering applications and to manage energy storage. - Picture an inductor as a coil in a water hose that can store some water pressure as energy. - **Real-World Example:** Used in transformers and radios for tuning and filtering signals.
Circuit Breaker	Circuit Breaker  Circuit Breaker Symbol	- Protect electrical circuits from damage caused by overloads or short circuits by automatically interrupting the flow of electricity when they detect a fault condition. - Think of a circuit breaker as a safety valve in a water system that shuts off the flow if the pressure gets too high, preventing pipe bursts and flooding. - Real-World Example: Circuit breakers are found in home electrical panels to protect wiring and appliances from overloads.

Active Components		
Diodes		- Allow current to flow in one direction only, acting as a one-way valve. - Think of a diode like a check valve in a plumbing system that only allows water to flow one way. - **Real-World Example:** Used in power supplies to convert AC to DC (rectification).

Active Components		
Transistors		• Can amplify electrical signals and act as switches. They are fundamental in digital circuits. • Imagine a transistor as a faucet. It can turn the flow of water on and off or control the amount of flow (amplification). • **Real-World Example**: Found in computer processors, amplifiers, and many other electronic devices.
Integrated Circuits	 Operational Amplifier	• Complex circuits with multiple components like transistors, resistors, and diodes on a single chip. They perform various functions from simple logic operations to complex microprocessor tasks. • Consider an IC as a small city, with all included components working together to perform complex tasks. • Real-World Example: It is used in nearly every electronic device including smartphones, computers, and digital watches.
Relays	A Relay A symbol for a relay 	• Act as electromechanical switches that allow a low-power electrical signal to control a high-power circuit by using an electromagnet to mechanically operate a switch. • Imagine a relay like a remote control for a gate. The remote control (low power) signals the gate's motor (high power) to open or close the gate, but the motor needs electricity to operate. • **Real-World Example:** Automotive Systems: Relays are used to control high-current devices like headlights, horns, and fuel pumps using low-power signals from the control switches.

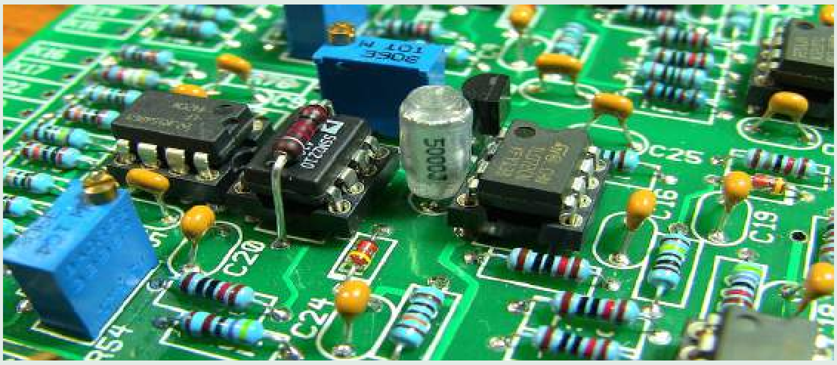
Active Components		
LEDs	An LED  A symbol of an LED	• Emit light when an electric current passes through them, serving as energy-efficient light sources. • Consider an LED like a small light bulb. It only lights up when connected to a power source. • Real-World Example: LEDs are used in household and commercial lighting fixtures for energy-efficient illumination.

Activity 4.1

1. Watch a simple video demonstration of some basic electronic components. Here A simple guide to electronic components.

2. Watch this simple video on identification of simple electronic components. Here Identification of Basic Electronics Components -Diodes, LED, Capacitors, Resistance, Transistors etc

3. The image is a printed circuit board with electronic components soldered on. Examine the image keenly.



4. After watching the video and examining the image provided above, perform the following tasks:
   a. List 10 electronic components you observed when you watched the video and on examining the picture
   b. Draw the circuit symbols of all components identified and listed
   c. Group the various components as passive or active
   d. State a function of each of the component listed
   e. Discuss your answers with a classmate

ELECTRONIC SYSTEMS

An electronic system is a collection of interconnected electronic components that work together to perform a specific function or set of functions. These components include devices like resistors, capacitors, transistors, integrated circuits, and microcontrollers. They work together to process signals, control processes, and manage information. For example, your smartphone is an electronic system that handles communication, internet browsing, photography, and many other tasks through its various interconnected parts.

In robotics, electronic systems are responsible for receiving input from sensors (like cameras, microphones, and touch sensors), processing this information using processors and microcontrollers, and then sending commands to actuators (like motors and servos) to perform actions. For instance, in a robotic arm, an electronic system processes the input from sensors that detect the position of the arm and sends commands to the motors to move the arm precisely. This allows robots to perform tasks like assembling parts, picking and placing objects, or even performing surgery with great accuracy.

Block Diagram Representation

In week five (5), our focus was on the use of control and logic diagrams and how flowcharts contribute to the control system design process. A flowchart breaks down a process into steps, just like a recipe does for cooking. For example, a flowchart for making a simple sandwich looks like the following:

1. Start: Begin the process.
2. Gather Ingredients: Bread, peanut butter, and jelly.
3. Spread Peanut Butter: Spread peanut butter on one slice of bread.
4. Spread Jelly: Spread jelly on the other slice of bread.
5. Assemble Sandwich: Put the slices together.
6. Serve: Serve the sandwich.
7. End: The process is complete.

Each step in the flowchart is represented by a box. Arrows between the boxes show the order of the steps, guiding you through the process from start to finish.

[image 01 - chart]

The recipe breaks down the process of cooking into steps, much like how flowcharts represent processes. This simple flowchart diagram demonstrates how breaking down a task into clear, sequential steps can help us understand and complete complex processes. In the same way, engineers use flowcharts to plan and manage tasks in robotics.

Block Diagrams

A block diagram is a visual map that simplifies an electronic system by breaking it into functional blocks connected by arrows. Each block represents a specific function or component, such as a microcontroller, sensor, or motor driver. The arrows show the direction of signal flow between these blocks, illustrating how the system operates. For example, a block might represent a microcontroller, sensor module, or motor driver. The arrows between blocks indicate the direction of information or signal flow between different subsystems. A block diagram could break down the elements of an audio system. It could show how the power supply unit interacts with the audio mixer and the audio power amplifier to project sound through a loudspeaker.

System Inputs and Outputs in Block Diagrams

Inputs and outputs are essential aspects of any electronic system and are represented in block diagrams

to illustrate the system's functionality and interactions with the external environment.

1. System inputs

Inputs represent the information or signals that enter the electronic system from external sources. These inputs can come from sensors, user interfaces, or other connected devices. In a block diagram, inputs are typically shown as arrows entering the respective blocks.

2. System outputs

Outputs represent the results, actions, or data generated by the electronic system and sent to external components or devices. Outputs can control motors, display information, or communicate data to other systems. In block diagrams, outputs are typically shown as arrows leaving the respective blocks.

Block Diagram Representation in Robotics

To illustrate the concept of block diagrams in robotics, let us consider a simple robotic system that includes three main functional blocks:

1. Sensing block

This block represents the sensors used in the robot to perceive the environment. Sensors could include cameras, ultrasonic sensors, or infra-red sensors.

2. Control block

The control block houses the microcontroller or microprocessor responsible for processing sensor data and making decisions. It takes inputs from the sensing block and generates control signals for the actuation block.

3. Actuation block

This block includes the actuators responsible for physical actions in the robot, such as motors or servos. The actuation block receives control signals from the control block and executes the required actions.

In building a robot that follows a line on the ground. A simple block diagram of such a system will consist of the following:

1. The sensing block
2. The control block
3. The actuation block

The table below describes each and how it works

Sensing Block	Control Block	Actuating Block
This block represents anything that gathers information about the robot's environment. In our example, it might be a light sensor that detects the difference between the dark line and the brighter floor. The sensor block sends this information (dark or light) as a signal to the next block.	This is the robot's brain! It receives the signal from the sensor block (dark or light) and decides what the robot should do next. It might be a microcontroller, like a small computer, programmed to tell the robot to turn its wheels if it goes off the line.	These are the parts that make the robot move based on the control block's instructions. In this case, it could be the motors that turn the robot's wheels. The control block sends a signal (turn left, turn right, or go straight) to the actuator block, which translates that signal into movement.

These blocks (sensing, control, and the actuation blocks) can be diagrammatically represented using shapes that represent individual blocks as shown below.

Robotic System Diagram

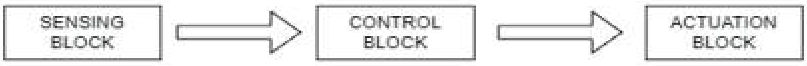


Fig. 4.10 shows the block diagram representation of the robotic system with appropriate labels for each block and arrows indicating signal flow between the blocks.

An arrow points from the Sensing Block to the Control Block, representing the flow of information from the sensors to the micro controller for processing. Another arrow points from the Control Block to the Actuation Block, symbolising the transmission of control signals from the microcontroller to the actuators, directing their actions. This Schematic block diagram provides a simplified representation of a robotic system, emphasising the flow of information from sensing to processing and control, leading to physical actions executed by the actuators. The block diagram format is an essential tool for understanding and designing complex electronic systems in robotics.

From the block diagram, one can explain the functions of the blocks as follow:

- The sensor block sees the environment (dark or light).
- The control block thinks about what to do (turn or go straight).
- The actuator block acts on those instructions (moves the wheels).
- The arrows between the blocks show the flow of information:
- Sensor block sends a signal (dark or light) to the control block.
- Control block sends a signal (turn or go straight) to the actuator block.

Schematic Diagram Representation

A schematic diagram is a simplified symbolic representation of an electronic circuit. It uses symbols to depict electronic components like resistors, capacitors, transistors, and integrated circuits (ICs). Lines connecting these symbols represent wires that carry electrical signals between components. By following the connections and understanding the symbols, you can visualize how the electronic system functions.

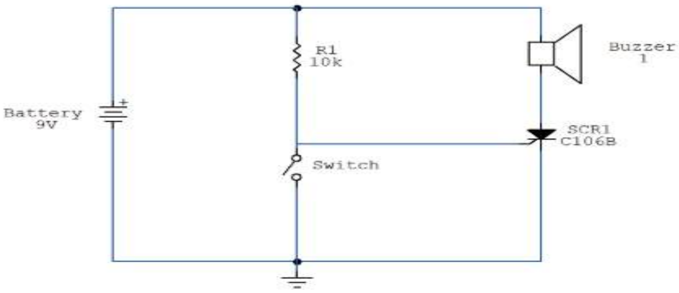


Fig. 4.11: A simple schematic diagram

A schematic diagram is like a detailed map for an electronic system. It shows how various components, like resistors, capacitors, and transistors, are connected to each other. Think of it as a blueprint that electricians and engineers use to understand and build electronic circuits.

In a schematic diagram, each component is represented by a unique symbol. For example, a resistor is shown as a zigzag line, while a capacitor is depicted as two parallel lines. These symbols are connected by lines that represent electrical wires.

By using schematic diagrams, you can easily see how electricity flows through a circuit, from the power source, through the components, and back again. This helps in troubleshooting and designing circuits because it provides a clear and precise way to understand the relationships and connections between different parts of the system.

In robotics, for example, a schematic diagram might show how sensors, processors, and actuators are wired together. This allows engineers to visualize how a robot's electronic systems work and ensure that all parts are connected correctly to perform tasks efficiently.

Components of a Schematic Diagram

- ### Symbols

  Schematic diagrams use standardised symbols (universally understood) to represent various electronic components such as resistors, capacitors, transistors, sensors, motors, and microcontrollers and any other electronic components that make up the electronic system.

- ### Lines and connections

  Lines in a schematic represent wires or conductive traces, while connections between components indicate how they are electrically linked. These lines represent the flow of electrical current in the electronic system.

- ### Labels and annotations

  Components and connections are often labelled with values, part numbers, or annotations to provide additional information.

Steps to Create Schematic Diagrams

1. **Use Standard Symbols:** Always use standardised symbols to ensure clarity and consistency in your schematics.
2. **Organise Neatly:** Arrange components and connections logically to enhance readability and understanding.
3. **Label Components:** Clearly label components, connections, and important nodes to aid comprehension.
4. **Document Changes:** Document any modifications or updates made to the schematic to maintain accuracy over time.

An Example Schematic Diagram

Using an Arduino buzzer circuit as an example, let us follow the steps outlined above to create a schematic diagram for the circuit.

1. **Use Standard Symbols:** We need symbols for the Arduino board, a buzzer, and connecting wires.
2. **Organise Neatly:** Place the Arduino board at the centre or on one side of your diagram. Position the buzzer near the Arduino, making sure there is enough space for connections.
3. **Label Components:** Label the Arduino, indicate the specific digital pin used (e.g., D8), and label the buzzer. Label the connecting wires and important nodes. For instance, the wire connecting D8 to the positive terminal of the buzzer should be labelled.
4. **Document Changes:** Keep track of any modifications. If you change the pin connection or add a resistor, note it down on the schematic.

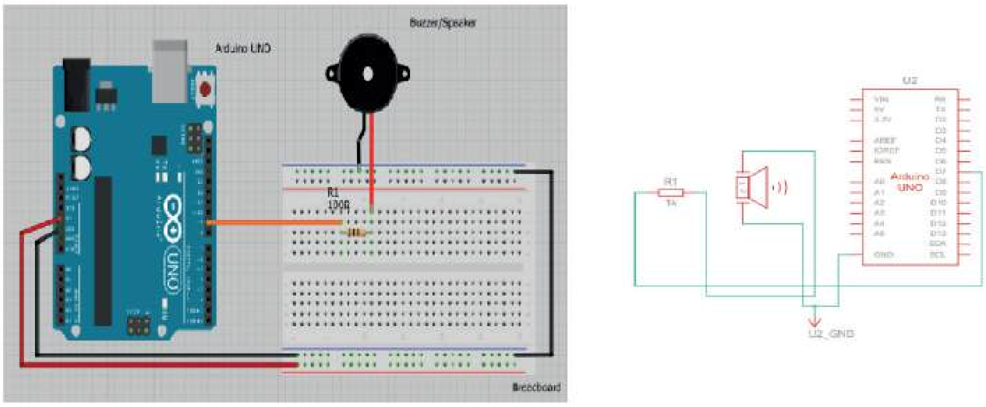


Fig. 4.12: Schematic diagrams of Arduino buzzer

Note: Some schematic diagrams might look more realistic, and some might just use symbols to represent various components as demonstrated below.

Interpreting Schematic Diagrams

1. Component Identification: Begin by identifying the symbols for each component in the schematic and understanding their functions.
2. Circuit Flow: Follow the flow of the circuit from the input to the output, tracing how signals or power move through the system.
3. Connection Understanding: Pay attention to how components are connected, and the paths signals take, including series and parallel connections.
4. Grounding: Identify the ground symbol and understand its role in providing a reference point for voltage levels.

Example Schematic Diagram and Interpretation:

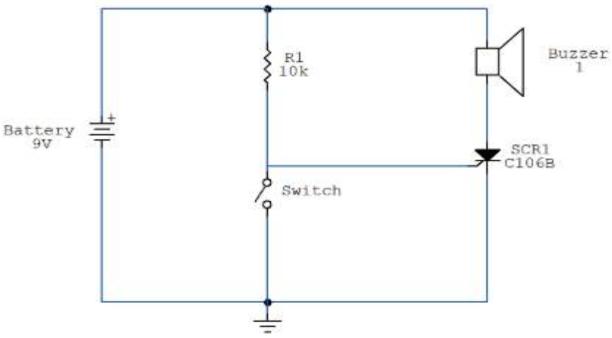


Fig. 4.13: A Buzzer Circuit

The schematic diagram provided above shows a simple buzzer circuit. Let's break down the components and their connections:

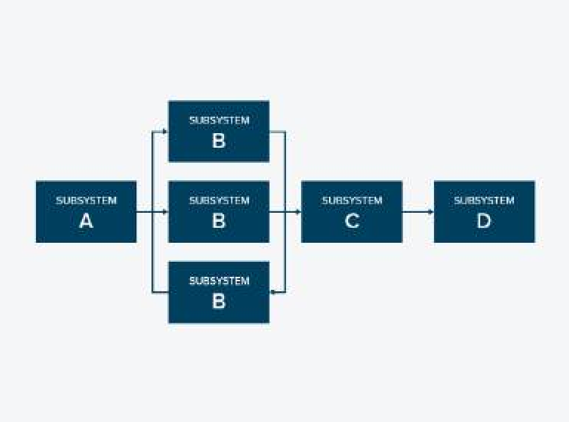
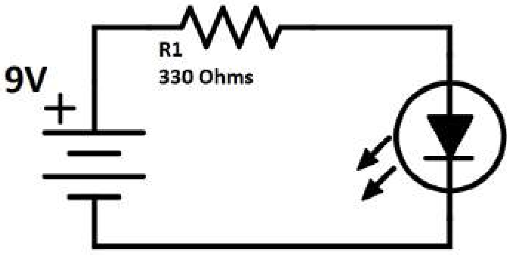
- **Battery (9V):** The battery is represented with two parallel lines of different lengths, and it acts as the power source for the circuit.

- **Resistor (R1, 10k ohms):** The zigzag line in the schematic diagram represents a resistor which has a value of 10k ohm. The function of the resistor in the schematic is to limit the current flowing through the circuit to protect other components.
- **Switch:** The switch is represented by an open gap with a line. The switch allows the circuit to be completed or broken manually.
- SCR (Silicon Controlled Rectifier, C106B): A diode with a gate lead represents a component known as the silicon-controlled rectifier. This diode with a gated lead which acts as a switch that is controlled by the gate signal.
- **Buzzer:** The speaker-like icon in the schematic is used to represent the component called the buzzer. The buzzer emits a sound when the circuit is activated.

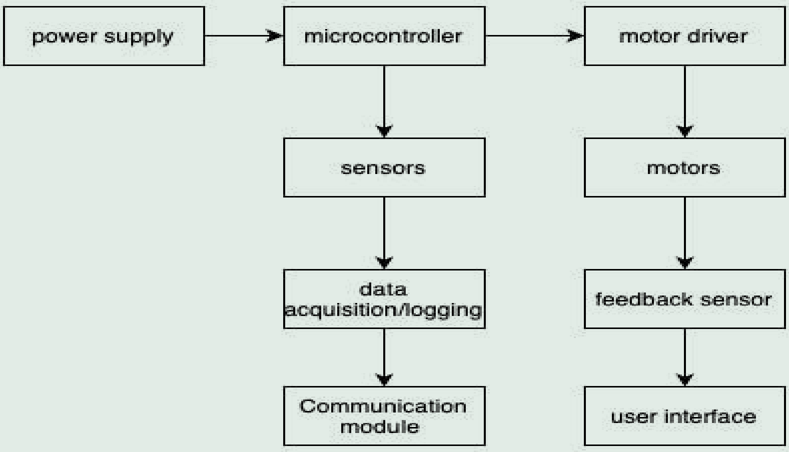
How the Circuit Works

- The 9V battery provides the necessary voltage for the circuit.
- The resistor (R1) ensures that the current remains within safe limits.
- When the switch is closed, it completes the circuit.
- The SCR allows current to flow to the buzzer when it receives a gate signal.
- The buzzer emits a sound when the SCR is activated and current flows through it.

Difference between Block Diagram and Schematic Diagram

Block Diagrams	Schematic Diagrams
Block diagrams use simple rectangular blocks to represent the main functional parts of a system and arrows to show the connections and flow of information between these parts. This provides a high-level overview of the system, helping to understand the structure and overall function without getting into the specifics.  SUBSYSTEM A SUBSYSTEM B SUBSYSTEM B SUBSYSTEM B SUBSYSTEM C SUBSYSTEM D	Schematic diagrams, on the other hand, provide a detailed representation of the system. They use standardised symbols to represent specific electronic components and show precise connections between them. This type of diagram serves as a blueprint for constructing the actual electronic circuit, detailing every component and its exact wiring. 

Activity 4.2



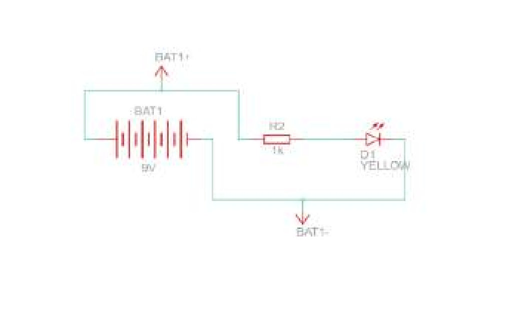
Analyse the block diagram (as shown above) of a robotic system and identify sensors as inputs that collect environmental data (e.g., temperature, distance) and actuators as outputs that execute physical actions (e.g., movement) based on commands from the microcontroller.

Activity 4.3

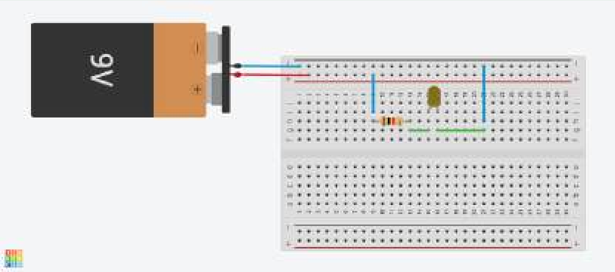
The diagram below is a schematic drawing of a simple electronic circuit. This simple circuit lights up an LED.

Examine this schematic carefully and perform the following tasks.

- Identify all the components in the schematics and write down their names.
- Trace connections between each component and take note of how each is connected.
- Explain how the circuit will function



Circuit schematic with component symbol



Circuit schematic with component icons

Activity 4.4

You may have encountered automatic doors at places like supermarkets, airports, and office buildings. These doors open automatically when you approach them and close after you pass through. But have you ever wondered how they work?

Here is a sample video on automatic door in operation:

ENOX AUTOMATIC SENSOR DOOR

Narrative on how the automatic door works

An automatic door operates by using sensors to detect motion or presence near the door. When someone approaches, the sensor sends a signal to a control unit, often a microcontroller, which processes the signal and activates a motor. The motor, driven by a motor driver, physically opens or closes the door. This system ensures convenient access without the need for manual intervention, enhancing convenience and accessibility in various environments such as supermarkets, airports, and office buildings.

To Do: Complete the following tasks to create a block diagram for the automatic door control system.

1. From the narrative above, Identify and list all the functional blocks (major components) used by the automatic door.
2. For each functional block/component listed, draw symbols to represent the component block and label them.
3. Connect the blocks with arrows to show power, data, and control flows.
4. Review and clearly label the complete diagram.

Activity 4.5

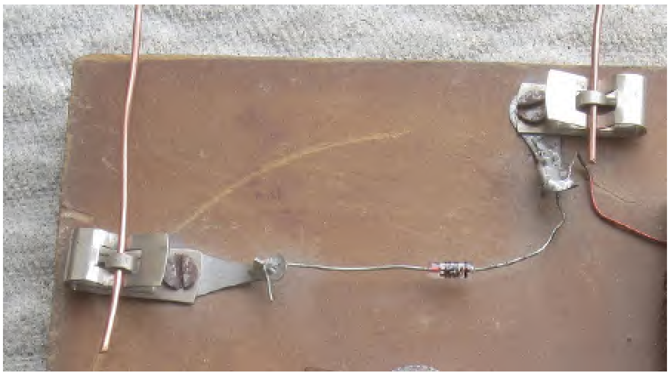
In your group, brainstorm ideas for a simple robot your group would like to build. Sketch a basic schematic diagram for your robot being sure to include the necessary components and connections.

Share your ideas, explaining the function of your robot with your class.

THE BREADBOARD

A breadboard, sometimes called a prototype board, or solderless board, is a board on which circuits can be implemented without the need to solder. The name **breadboards** came about from the fact that engineers used bread-cutting boards to assemble electronic circuits to implement and test their designs. The components were screwed to the bread board and wired with Fahnestock clips, as shown below.

PRESS
DOWN
HERE



The breadboard consists of rows and columns connected internally as shown in the diagram below. From left to right, the vertical strips by the red and blue vertical lines are called the **power rails** or buses. These are usually marked with red (positive) and blue or black (negative) lines. Power supplies are typically connected to the power rails from which power is supplied to other circuits being prototyped on the board. There are two sets of the power rails that sandwich terminal strips or component rails, and they are internally connected horizontally. The component rails are used in connecting components together.

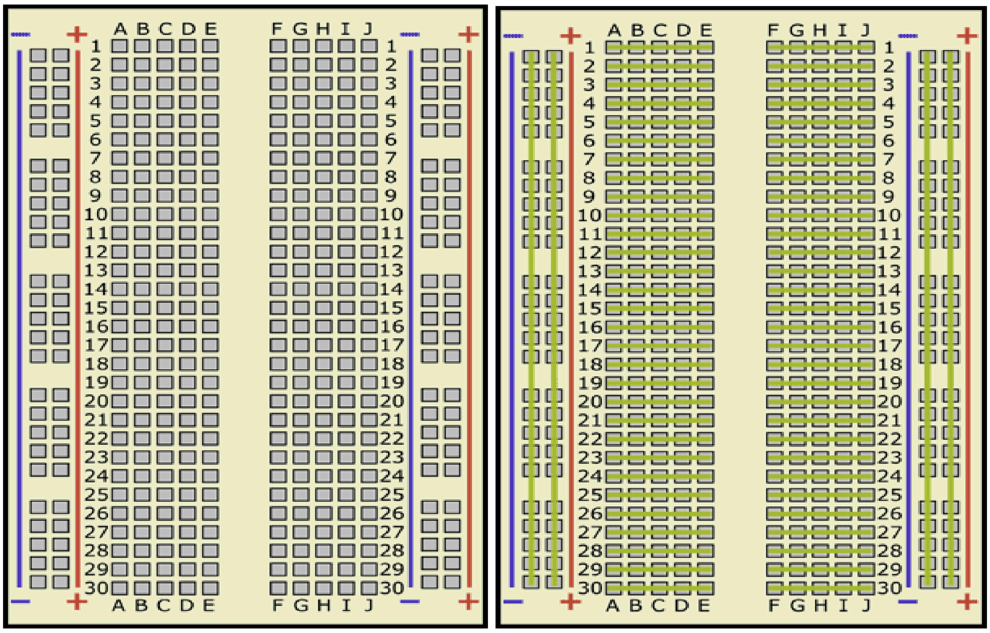


Fig. 4.14: *Solderless Breadboard*

How to Assemble Circuits on a Prototype Breadboard

To set up a circuit, start by inserting the components' leads into the breadboard's holes. Use jumper wires to connect components that are not in adjacent holes. Always double-check your connections before powering the circuit to avoid damaging the components. Consider the circuit below and observe how the simple LED circuit is set up on a breadboard.

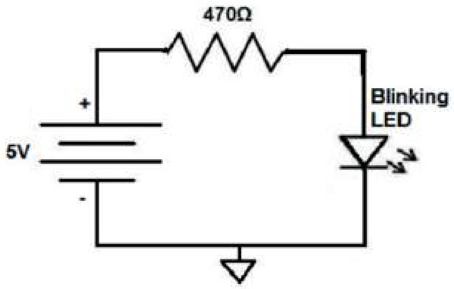
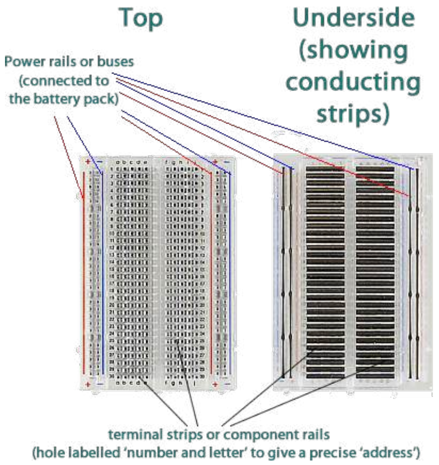


Fig. 4.15: Simple LED light flasher circuit

Materials

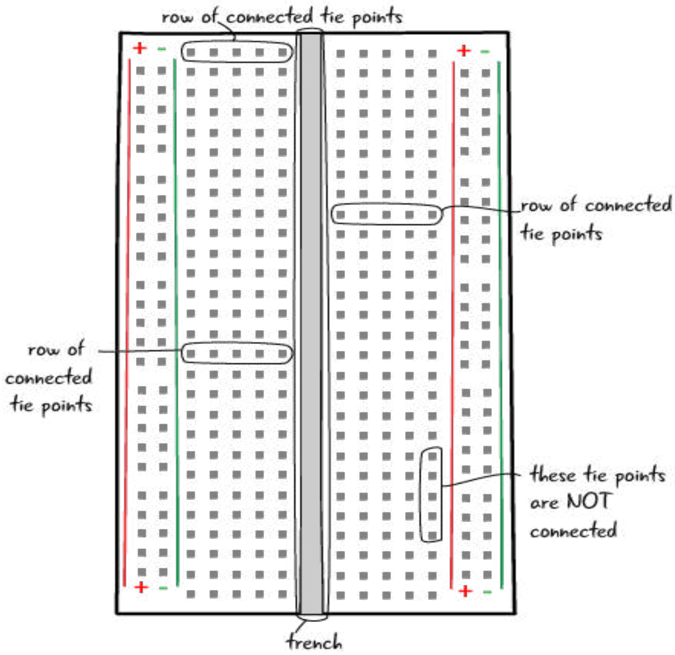
- 1 Breadboard
- 1 LED
- 1 470Ω Resistor


Top
Underside (showing conducting strips)
Power rails or buses (connected to the battery pack)
terminal strips or component rails (hole labelled 'number and letter' to give a precise 'address')

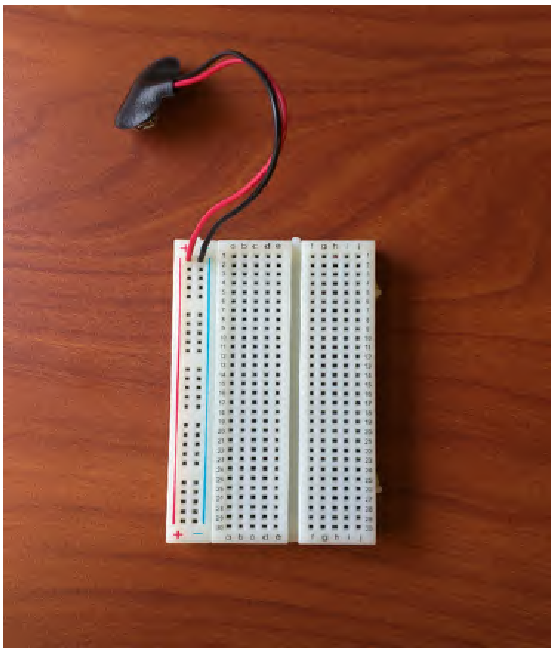
- 1 5V Power Supply (e.g., a 5V battery pack or a power adapter)
- Jumper Wires
- Digital Multimeter

Step-by-Step Guide to Assembling the Circuit

1. **Prepare the Breadboard**
   - Place the breadboard on a stable, flat surface.
   - Understand the layout: The breadboard has two main sections—the power rails and the terminal strips. Power rails are vertical, terminal rails are horizontal.

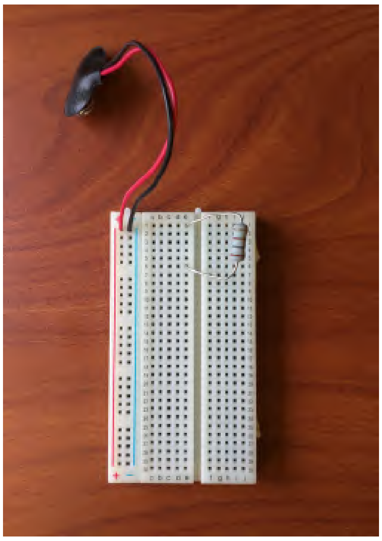

row of connected tie points
+ -
+ -
row of connected tie points
row of connected tie points
these tie points are NOT connected
+ -
+ -
trench

2. **Connect the Power Supply**
   - Connect the positive terminal (+) of the 5V power supply to the positive power rail on the breadboard.
   - Connect the negative terminal (-) of the 5V power supply to the negative power rail on the breadboard.

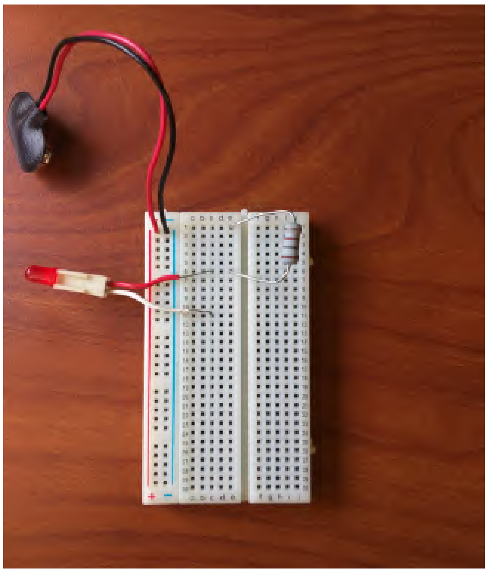


•

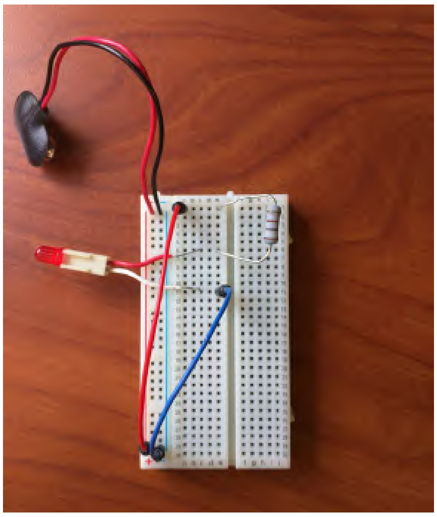
3. **Place the Resistor**
   - Using one set of terminal strips insert one wire of the 470Ω resistor into a hole in one of the rows in the terminal strip.
   - Insert the other wire of the resistor into a different row in the terminal strip.



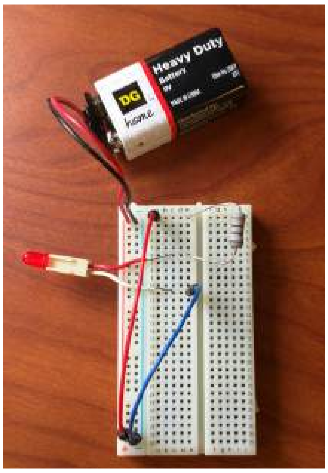
4. **Place the LED**
   - Identify the longer wire (anode, positive) and the shorter wire (cathode, negative) of the LED.
   - Insert the longer wire (anode) into the same row as the end of the resistor.
   - Insert the shorter wire (cathode) into a different row.



5. **Connect the Circuit**
   - Use a jumper wire to connect the row where the resistor is inserted to the positive power rail on the breadboard.
   - Use another jumper wire to connect the row where the cathode of the LED is inserted to the negative power rail on the breadboard.



6. **Double check** to ensure all connections are secure and that there are no loose wires.
7. **Turn on the power supply**. The LED should turn on if everything is connected correctly



Caution: When setting up a circuit on a breadboard, use a compatible power supply to prevent damage. Keep wires and component leads separated to avoid short circuits. Let hot components cool before handling to prevent burns. Discharge static electricity before touching components. Always turn off power before adjusting the circuit to avoid shocks. Use appropriate tools to handle components safely and protect the breadboard from damage.

How To Use A Multimeter

A Multimeter

A multimeter, also known as a volt-ohm-milliammeter (VOM), is a versatile measuring instrument used to measure multiple electrical properties. Typically, a multimeter can measure voltage, resistance, and current, allowing it to function as a voltmeter, ohmmeter, and ammeter. Many multimeters also offer additional measurement capabilities such as temperature and capacitance.

One fundamental use of a multimeter is testing circuit continuity, which is essential for fault finding in electrical circuits. When set to continuity mode, the multimeter checks if a circuit is complete and produces an audible sound if continuity is present. This feature helps users quickly identify breaks or shorts in the circuit, making troubleshooting more efficient.

Safety Precautions

Observe these safety precautions when using a multimeter:

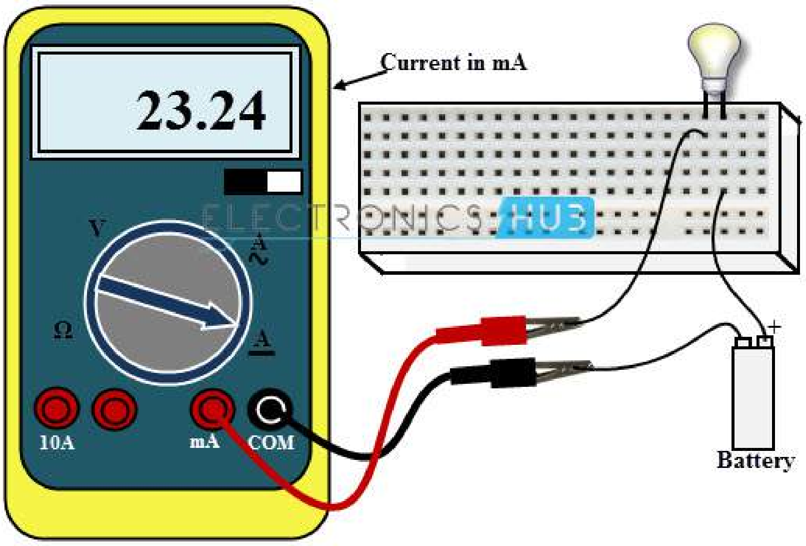
1. Use the correct voltage and current settings.
2. Ensure proper probe use (red for positive, black for negative).
3. Check for residual voltage before measuring.
4. Verify and replace fuses as needed.
5. Keep hands and multimeter dry.
6. Inspect test leads regularly for damage.

One way in which you can know how your circuit is behaving is to monitor the circuits. Circuits are monitored through measurement of electric properties components in the circuit exhibit. Some of those physical properties include voltage, current and resistance. To know the value of these quantities a multimeter is used to measure and display the values.

Steps to Use a Multimeter

Below are the steps involved in measuring electrical quantities of circuits using a multimeter.

1. **Prepare the Multimeter**
   - Turn the dial on the multimeter to the appropriate DC voltage setting (V with a straight line and dashed line below it). For a 5V circuit, a setting of 20V DC is usually appropriate.
2. **Turn on the Power Supply**
   - After setting up your multimeter, turn on the power supply to the breadboard circuit.
3. **Connect the Multimeter Probes**
   - **Red Probe (Positive)** Connect the red probe to the point in the circuit where you want to measure the voltage. Typically, this could be a point in the positive power rail or at the anode of a component like an LED.
   - **Black Probe (Negative)** Connect the black probe to the ground (GND) or the negative power rail of the breadboard.



https://www.electronicshub.org/current-measurement-using-multimeter/

Fig. 4.15: Sample of current measurement using multimeter

4. **Take the Measurement**
   - With the probes connected, the multimeter will display the voltage at the point of contact. Ensure that the probes are making good contact with the breadboard points to get an accurate reading.

5. **Read the Multimeter Display**
   - Note the voltage reading displayed on the multimeter. This will tell you the potential difference between the points you are measuring.

6. **Test Different Points (Optional)**
   - If you want to measure voltage across different components, move the red probe to different points in the circuit, keeping the black probe connected to ground.

7. **Turn Off the Power Supply:**
   - After taking your measurements, turn off the power supply before disconnecting the probes and making any adjustments to your circuit.

The step-by-step procedures listed below, and a video demonstration constitute an example of how to measure voltage across an LED.

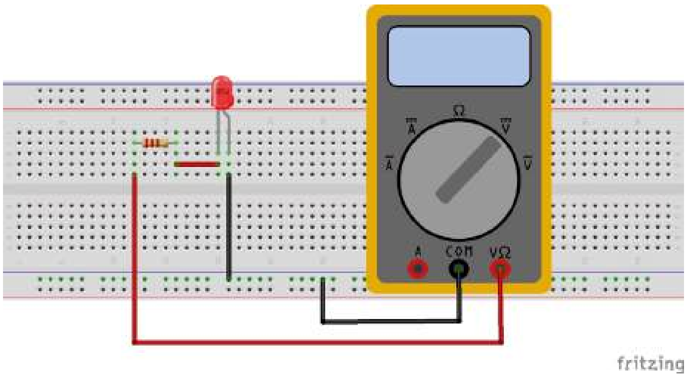
Measuring Voltage Across an LED

1. **Connect Power Supply**
   - Ensure your 5V power supply is connected to the breadboard.

2. **Set Up Multimeter**
   - Turn the multimeter to the 20V DC setting.

3. **Turn on the Power Supply**
   - Switch on the power supply to the circuit.

4. **Place Probes**
    - Touch the red probe to the anode (longer leg) of the LED.
    - Touch the black probe to the cathode (shorter leg) of the LED or to the ground rail.
5. **Read Voltage**
    - The multimeter will display the voltage drop across the LED.
6. **Turn Off Power Supply**
    - Once done, turn off the power supply.
7. Watch this video to see a practical example of how to measure voltage using a multimeter. How to Measure Voltage with a Multimeter

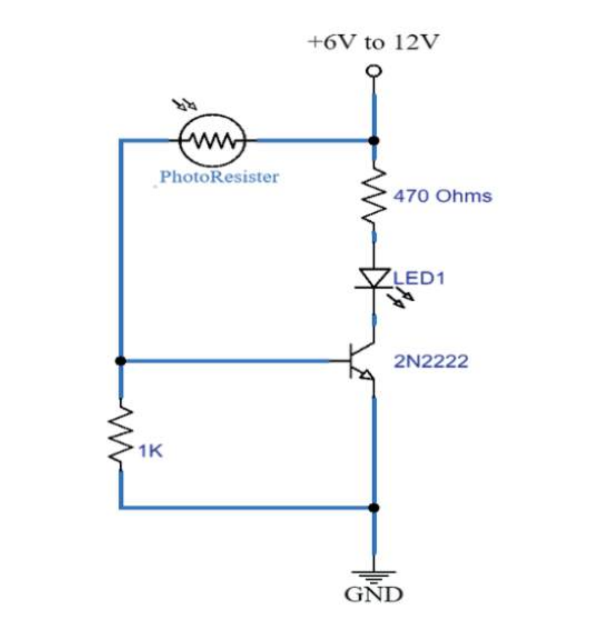


Safety Tips

- Always start with the multimeter set to a higher voltage range and then move to a lower range if needed.
- Avoid touching the metal parts of the probes while taking measurements to prevent electrical shock.
- Make sure the probes are securely connected to the multimeter and the circuit to avoid inaccurate readings or damage to the multimeter.

Activity 4.6

In this activity, you will implement the circuit diagram given below on a breadboard.



Precaution: Ensure to use the appropriate power rating and avoid short-circuits. Handle all electronic components with care. Seek assistance from instructors where necessary.

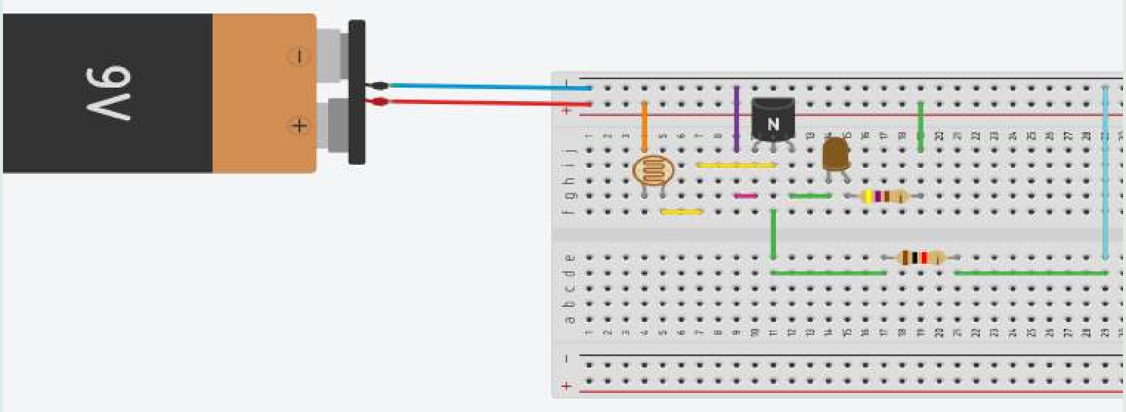
Light-Dependent LED Circuit with a Transistor

Materials Needed

- 1 Breadboard
- 1 Photoresistor
- 1 LED
- 1 NPN Transistor (2N2222)
- 1 470Ω Resistor
- 1 1kΩ Resistor
- 1 6V to 12V Power Supply (e.g., a battery pack or power adapter)
- Jumper Wires
- Digital Multimeter

Steps

1. **Place the Resistors on the Breadboard:**
   - Insert the 1kΩ resistor into the breadboard. One end should be in one row and the other end in a different row.
2. **Connect the Photoresistor:**
   - Insert the photoresistor into the breadboard so that one end is in the same row as the 1kΩ resistor. The other end should be in a new row.
3. **Connect the Transistor:**
   - Insert the transistor into the breadboard with its three terminals (collector, base, emitter) in separate rows.
   - Connect the junction of the 1kΩ resistor and the photoresistor to the base of the transistor.
4. **Connect the LED and 470Ω Resistor:**
   - Insert the LED into the breadboard, with the anode (longer leg) in one row and the cathode (shorter leg) in another row.
   - Connect one end of the 470Ω resistor to the row with the anode of the LED and the other end to a new row.
   - Connect the collector of the transistor to the row with the 470Ω resistor and the anode of the LED.
5. **Power Supply Connections:**
   - Connect the positive terminal (+) of the power supply to the row with the 470Ω resistor and LED anode.
   - Connect the negative terminal (-) of the power supply to the emitter of the transistor and to the row with the cathode of the LED.



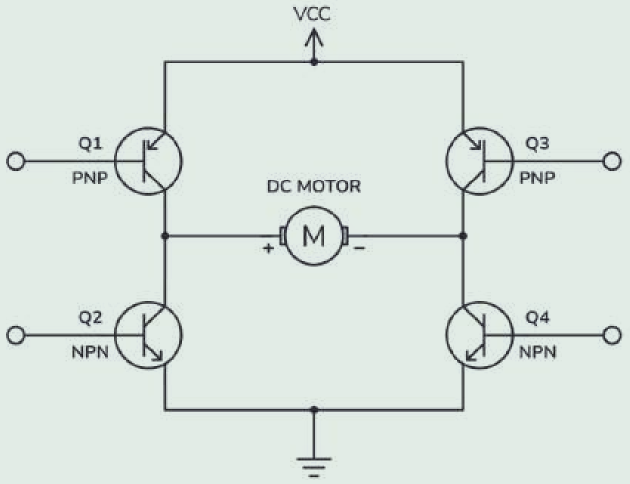
6. **Test the Circuit:**
   - Turn on the power supply. Cover and uncover the photoresistor to see the LED's response to changes in light.
7. **Use the Digital Multimeter:**
   - Measure voltage across the photodiode, resistors and LED in the circuit using the multimeter.
   - Compare voltage readings across different groups and discuss the results.

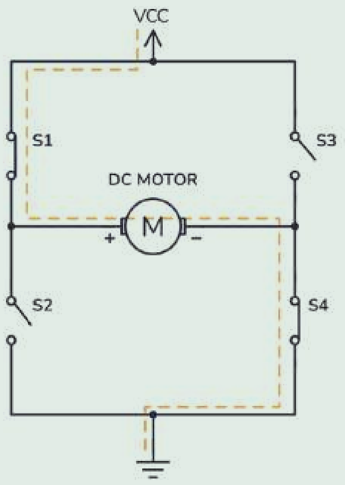
Activity 4.7

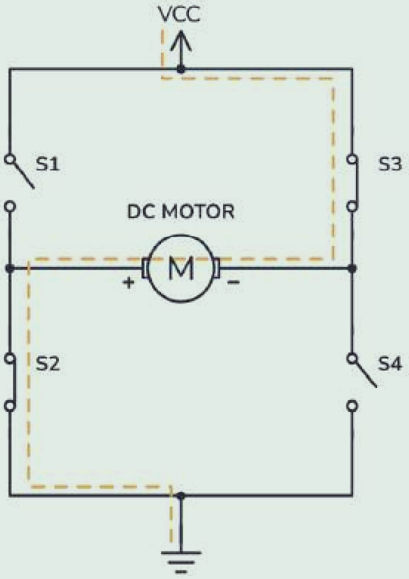
In this activity, you will implement a DC motor control circuit on a solderless breadboard. An H-bridge circuit will be used to control a DC motor, enabling you to alter the motor's spin direction from clockwise to counterclockwise and vice-versa. An H-bridge circuit allows you to control the flow of current, and the direction of this current flow determines the direction of the DC motor's spin.

While this activity presents an exciting challenge, the explanation can be improved for clarity. An H-bridge consists of four switches (usually transistors) that allow current to flow in either direction through the motor, thereby changing its spin direction. Each transistor in the H-bridge operates as a switch, controlled by signals that determine whether the current flows clockwise or counterclockwise. Additionally, this activity assumes that all learners have access to an Arduino Uno and a computer, which are necessary for controlling the H-bridge and motor.

As shown in the diagrams below,







Materials Needed

1. Arduino Uno
2. Breadboard
3. 2 NPN transistors (e.g., 2N2222)
4. 2 PNP transistors (e.g., 2N2907)
5. DC Motor
6. Resistors (1kΩ for base resistors)
7. Jumper wires
8. Power supply for the motor (e.g., 9V battery)

Circuit Assembly

1. **Prepare the Breadboard:**
   - Place the NPN and PNP transistors on the breadboard.
2. **Transistor Pin Configuration**
   - NPN Transistor (2N2222):
     - Emitter (E)
     - Base (B)
     - Collector (C)
   - PNP Transistor (2N2907):
     - Emitter (E)
     - Base (B)
     - Collector (C)
3. **Motor and Power Connections:**
   - Connect the motor terminals to the collectors of the NPN transistors.
   - Connect the emitters of the NPN transistors to ground.
   - Connect the emitters of the PNP transistors to the positive terminal of the motor power supply.
   - Connect the collectors of the PNP transistors to the motor terminals.
4. **Base Resistor Connections:**
   - Connect a 1kΩ resistor between each base of the transistors and the Arduino digital pins as follows:
     - Arduino pin 1 to the base of NPN transistor 1
     - Arduino pin 2 to the base of NPN transistor 2
     - Arduino pin 3 to the base of PNP transistor 1
     - Arduino pin 4 to the base of PNP transistor 2

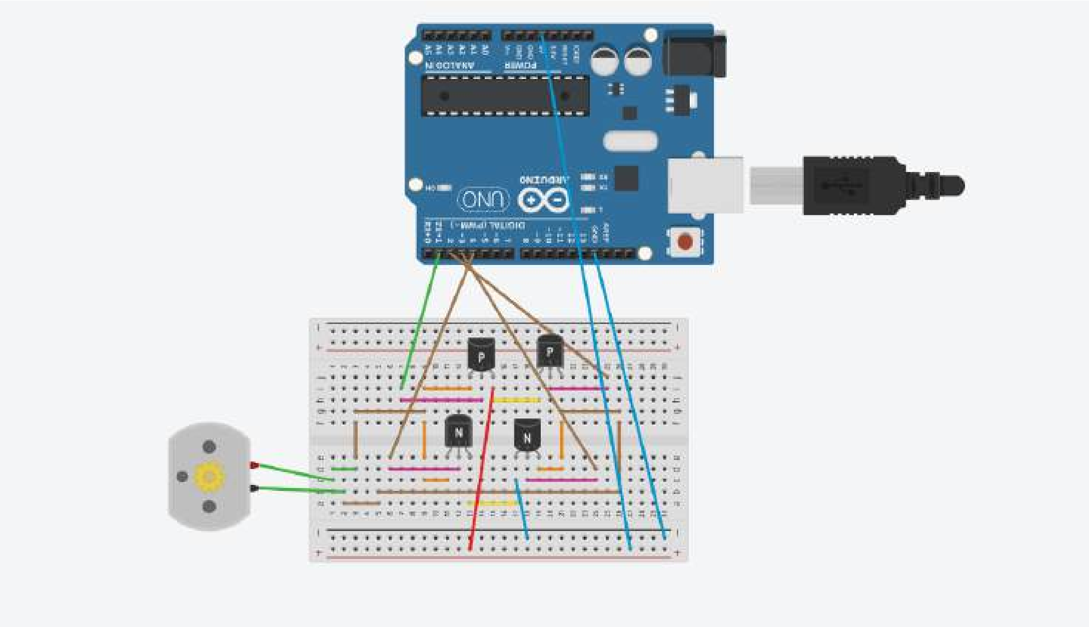
Arduino Code

```
// Define pin numbers
const int pin1 = 1;
const int pin2 = 2;
const int pin3 = 3;
const int pin4 = 4;
// Define the interval (20 seconds)
const unsigned long interval = 20000; // 20,000 milliseconds
void setup() {
  // Set pin modes
  pinMode(pin1, OUTPUT);
  pinMode(pin2, OUTPUT);
  pinMode(pin3, OUTPUT);
  pinMode(pin4, OUTPUT);
  // Start with pins 1 and 4 activated
  digitalWrite(pin1, HIGH);
  digitalWrite(pin4, HIGH);
  digitalWrite(pin2, LOW);
  digitalWrite(pin3, LOW);
}
void loop() {
  // Wait for 20 seconds
  delay(interval);
  // Deactivate pins 1 and 4
  digitalWrite(pin1, LOW);
  digitalWrite(pin4, LOW);
  // Activate pins 2 and 3
  digitalWrite(pin2, HIGH);
  digitalWrite(pin3, HIGH);
  // Wait for 20 seconds
  delay(interval);
  // Deactivate pins 2 and 3
  digitalWrite(pin2, LOW);
  digitalWrite(pin3, LOW);
  // Reactivate pins 1 and 4
  digitalWrite(pin1, HIGH);
  digitalWrite(pin4, HIGH);
}
```

Step-by-Step Guide

1. **Assemble the Circuit**
   - Follow the circuit assembly instructions above, ensuring that all connections are secure and correct.
   - Double-check the connections to make sure there are no shorts or incorrectly wired components.
2. **Upload the Code**
   - Open the Arduino IDE on your computer.
   - Copy and paste the provided code into the IDE.
   - Connect the Arduino Uno to your computer using a USB cable.
   - Select the correct board and port from the Tools menu.
   - Click the Upload button to upload the code to the Arduino.
3. **Test the Circuit**
   - Once the code is uploaded, the motor should start spinning in one direction.
   - After 20 seconds, the motor should reverse direction.
   - The motor will continue to alternate direction every 20 seconds.

Your circuit should look like the diagram below.



Activity 4.8

1. Turn on the power supply. Press the push buttons to see the motor's response.
2. Measure the voltage at different points, such as across the motor terminals and between the collector and emitter of the transistors.
3. Compare voltage readings across different groups and discuss the results.

4. Explain why the motor changes direction based on which button is pressed.

EXPLORING DIGITAL AND ANALOGUE SYSTEMS IN DISCRETE AND CONTINUOUS-TIME MACHINE DESIGN

The word ‘digital’ describes electronic technology that generates, stores and processes data in terms of positive and nonpositive states. Positive is expressed or represented by the number 1 and nonpositive by the number 0. Thus, data transmitted or stored with digital technology is described as a string of 0s and 1s. Each of these state digits is referred to as a bit; a string of bits a computer can address individually as a group is a byte. Before the digital age, electronic transmission was limited to analogue technology, which conveys data as electronic signals of varying frequency or amplitude added to carrier waves of a given frequency. Broadcast and phone transmission have conventionally used analogue technology.

A. Understanding Digital Systems

Digital Systems

Building on the concept of binary digits, digital systems are constructed using digital circuits designed to handle binary information. These systems operate on discrete signals that switch between well-defined states—such as on/off, true/false or voltage level zero/five. This discrete nature of digital signals ensures accuracy and reliability in data processing, which is critical for applications like computing, telecommunications, and automated control systems.

Digital and Binary

In the realm of digital systems, information is represented using binary digits. Each digit, or bit, can either be 0 or 1, and this binary representation is crucial for handling discrete signals. Understanding binary encoding helps clarify how digital systems manage data through discrete signals, which do not take on intermediate values, making them distinct from continuous signals.

Discrete Signals

Discrete signals, which can only assume specific, separated values - such as on/off or true/false - are central to digital systems. This characteristic differentiates them from continuous signals, which can take on a continuous range of values. Recognising the nature of discrete signals is essential for grasping the fundamentals of digital systems. In contrast, continuous signals, which we’ll explore in subsequent sections, are integral to analogue systems that operate with smoothly varying signals.

Logic Gates

The operation of digital systems is fundamentally based on logic gates, which execute essential logical operations by processing discrete signals. These gates are realised using electronic components such as diodes, transistors, and resistors. While the symbols for logic gates represent their logical functions, their actual implementation involves configuring transistors in specific ways to perform operations like AND, OR, NOT, and XOR. Understanding logic gates is crucial, as they form the building blocks of digital circuits, enabling the execution of complex computations and decision-making processes in digital systems.

Introduction to Logic Gates

Logic gates are fundamental components in digital electronics and play a crucial role in the field of robotics. They are the building blocks that allow robots to process information and make decisions. By understanding how logic gates work, you'll gain insights into how robots and other digital systems perform complex tasks using simple binary logic.

What Are Logic Gates?

A logic gate is an electronic circuit that performs a specific logical operation based on one or more input signals and produces a single logical output. These operations are based on Boolean algebra, which uses binary values (0 and 1) to represent true and false states. Each logic gate has a distinct function and produces a specific output based on its inputs. The behaviour of a logic gate is described by a Truth table.

Types of Logic Gates

There are several basic types of logic gates, each with its unique operation:

AND Gate

For the output of an AND gate to be true (1) all its inputs must be true (1). If any input is false (0), the output is false (0).

OR Gate

The OR gate outputs true (1) if at least one of its inputs is true (1). The output is false (0) only if all inputs are false (0).

NOT Gate

The NOT gate, also known as an inverter, outputs the opposite of its input. If the input is true (1), the output is false (0), and vice versa.

NAND Gate

The NAND gate is the inverse of the AND gate. It outputs false (0) only if all its inputs are true (1). Otherwise, the output is true (1).

NOR Gate

The NOR gate is the inverse of the OR gate. It outputs true (1) only if all its inputs are false (0). Otherwise, the output is false (0).

XOR Gate

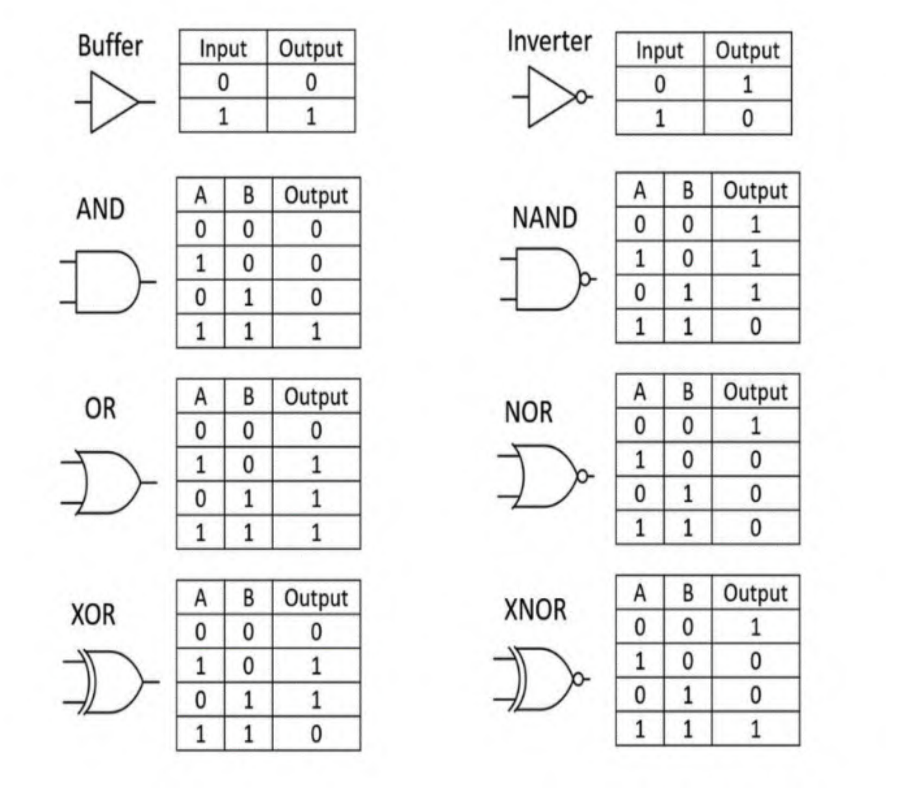
The XOR (exclusive OR) gate outputs true (1) if its inputs are different. If both inputs are the same, the output is false (0).

XNOR Gate

The XNOR (exclusive NOR) gate is the inverse of the XOR gate. It outputs true (1) if its inputs are the same. If the inputs are different, the output is false (0).

Truth Tables

A truth table is a simple way to represent the operation of a logic gate. It lists all possible input combinations and the corresponding output for each combination. The combinations as per the number of inputs (n) are determined using the relation 2^n. For example, if a gate has three inputs (n=3), the number of ways you can combine the inputs would be eight (2^3=2*2*2=8) as per the combination expression given. Here are the schematics and truth tables for the basic logic gates:



Buffer

Input	Output
0	0
1	1

Inverter

Input	Output
0	1
1	0

AND

A	B	Output
0	0	0
1	0	0
0	1	0
1	1	1

NAND

A	B	Output
0	0	1
1	0	1
0	1	1
1	1	0

OR

A	B	Output
0	0	0
1	0	1
0	1	1
1	1	1

NOR

A	B	Output
0	0	1
1	0	0
0	1	0
1	1	0

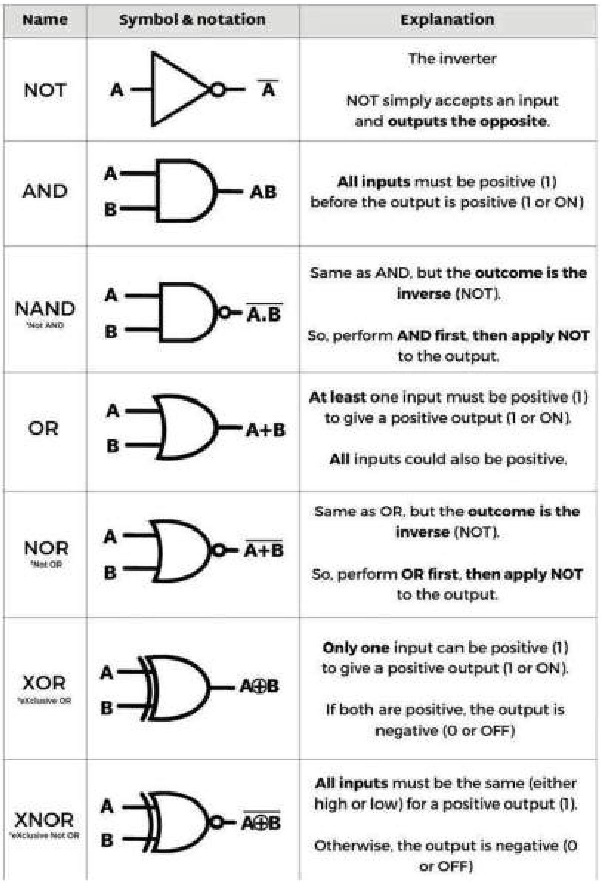
XOR

A	B	Output
0	0	0
1	0	1
0	1	1
1	1	0

XNOR

A	B	Output
0	0	1
1	0	0
0	1	0
1	1	1

Fig. 4.16: *Basic Logic Gates and their Truth Table*



Name	Symbol & notation	Explanation
NOT	A → $\overline{A}$	The inverter NOT simply accepts an input and **outputs the opposite**.
AND	A, B → AB	**All inputs** must be positive (1) before the output is positive (1 or ON)
NAND *Not AND	A, B → $\overline{A.B}$	Same as AND, but the **outcome is the inverse** (NOT). So, perform **AND first**, **then apply NOT** to the output.
OR	A, B → A+B	**At least** one input must be positive (1) to give a positive output (1 or ON). **All** inputs could also be positive.
NOR *Not OR	A, B → $\overline{A+B}$	Same as OR, but the **outcome is the inverse** (NOT). So, perform **OR first**, **then apply NOT** to the output.
XOR *eXclusive OR	A, B → $A \oplus B$	**Only one** input can be positive (1) to give a positive output (1 or ON). If both are positive, the output is negative (0 or OFF)
XNOR *eXclusive Not OR	A, B → $\overline{A \oplus B}$	**All inputs** must be the same (either high or low) for a positive output (1). Otherwise, the output is negative (0 or OFF)

Practical Application in Robotics

In robotics, logic gates are used to make decisions based on sensor inputs. For example, a robot might use an AND gate to move forward only if both the left and right sensors detect a clear path. Similarly, an OR gate could be used to trigger an alert if any of the multiple safety sensors detect a problem.

By combining different logic gates, you can create more complex circuits that allow a robot to perform sophisticated tasks, such as navigating through a maze, responding to voice commands, or interacting with its environment in real-time.

Understanding logic gates and their operations is a fundamental step in learning how robots and other digital systems think and operate. This knowledge will empower you to design and build your digital circuits, enabling you to create innovative robotic projects and explore the fascinating world of electronics and automation.

How to Design a Logic Circuit

Designing a logical circuit involves several systematic steps that ensure the desired functionality is achieved. Here is a detailed guide:

1. **Understand the Given Specifications**
   Before you start designing a digital circuit, it is important to understand what is required. This means knowing what the circuit is supposed to do and what kind of output it should produce. For instance, when you follow a recipe to prepare a

meal – you need to know the final dish you are making, and the steps needed to prepare it.

2. **Find the Number of Inputs and Outputs**
Next, you need to determine how many inputs and outputs your circuit will have. Inputs are the signals or data you give to the circuit, and outputs are what the circuit produces. This step is like listing the ingredients needed for your recipe – you need to know what you have to work with.

3. **Create a Truth Table**
Once you know the number of inputs, you can create a truth table. A truth table lists all possible combinations of inputs and the corresponding outputs for each combination. The columns in the table represent the inputs and outputs. This table helps you see how the circuit should behave for every possible input combination, similar to following the cooking steps to see how each ingredient affects the final dish.

4. **Draw the Circuit Diagram**
Finally, you draw the circuit diagram. This is a visual representation of your digital circuit, showing how the different components (like logic gates) are connected. Make sure your diagram matches the information in your truth table, with the correct relationships between inputs and outputs. This step is like arranging the ingredients on your plate, ensuring everything is in the right place to create the final dish.

Example 1

Steps to Design a Digital Circuit

In this example, we will design a digital circuit for a simple alarm system that turns on a light. The system is made up of two (2) sensors and the light will be switched on when either of two sensors detect motion.

Steps

1. Understand the Given Specifications

Before you start, you need to know what the circuit is supposed to do. In this case, the alarm system should turn on a light if either Sensor A or Sensor B detects motion. This means the output (the light) should turn on if either input (Sensor A or Sensor B) is triggered.

2. Find the Number of Inputs and Outputs

Determine the number of inputs and outputs for your circuit. Here, you have:

The number of Inputs = 2 (Sensor A and Sensor B)

And the number of outputs = 1 (the light)

3. Create a Truth Table

Next, create a truth table that shows all possible combinations of inputs and the corresponding output for each combination. This helps you understand how the circuit should behave.

Sensor A (INPUT)	**Sensor (INPUT)**	**Light (OUTPUT)**
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

In this table:

- 0 represents no motion detected.

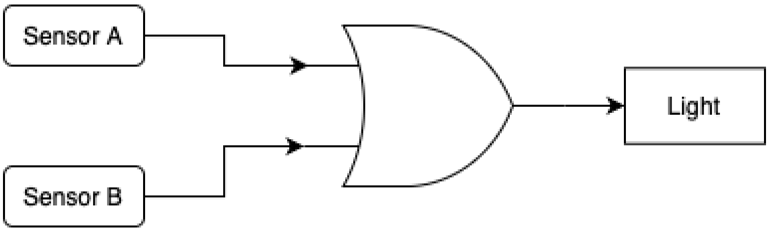
- 1 represents motion detected.

The light turns on (output 1) if either Sensor A or Sensor B detects motion.

4. Draw the Circuit Diagram

Finally, draw the circuit diagram using the information from the truth table. You can identify the correct logic gate to use from the table above to create the circuit. In this case, you need an OR gate because the light should turn on if one or both Sensors A and B, detects motion. The OR gate outputs true if any of its inputs are true.

1. INPUT: Connect Sensor A and Sensor B to the inputs of the OR gate.
1. OUTPUT: Connect the output of the OR gate to the light.



This circuit ensures that the light will turn on if either Sensor A or Sensor B detects motion, just as specified.

B. Understanding Analogue Systems

An electrical analogue represents the operation of logic gates, such as OR, AND, and NOT gates, using electrical circuits. In these circuits, selected components like resistors, diodes, and capacitors are used to mimic the behaviour of logic gates. For instance, an AND gate can be represented by a series connection of diodes, allowing current to flow and produce a high output voltage only when both inputs are high. Similarly, an OR gate can be represented by a parallel connection of diodes, producing a high output if at least one input is high. These components are configured in specific ways to perform logical operations, translating abstract binary logic into tangible electrical behaviours.

Analogue circuits work with continuous signals, unlike digital circuits that use binary digits (0 and 1. These continuous signals, such as varying voltage levels or current flows, represent real-world physical quantities. For example, analogue signals can convey information like audio, temperature readings, or voltage levels. Analogue circuits are crucial in many applications, especially in robotics, where they are used for sensor interfacing and real-time control. Sensors in robotics often measure physical quantities like light, sound, or temperature, producing analogue signals that must be processed by analogue circuits. These circuits convert the continuous signals into forms that digital systems can use, enabling precise control and interaction with the environment.

Assembling Analogue Circuits

The two basic steps involved in assembling analogue circuits from circuit schematics diagrams are; the selection of components, and the designing of the circuit.

1. Component Selection

This involves understanding the roles and functions of the three basic analogue circuit elements; resistors, capacitors and the operational amplifiers.

a. Resistors

Resistors are components that limit the flow of electrical current. Here are some key roles they play in analogue circuits.

i. **Current Limiting:** They prevent too much current from flowing in parts of the circuit, protecting other components.

ii. **Voltage Division:** Resistors can divide voltage in a circuit, which helps in regulating voltage levels.

iii. **Biassing:** They set the correct operating conditions for other components like transistors.

iv. **Load Resistance:** In amplifier circuits, resistors help in properly amplifying signals.

b. Capacitors

Capacitors store electrical charge and have several important uses in analogue circuits:

i. **Filtering:** They block certain frequencies and allow others to pass, cleaning up signals.

ii. **Timing Elements:** With resistors, determine the timing of signals in oscillators and timers.

iii. **Coupling:** Capacitors pass AC signals between parts of a circuit while blocking DC signals.

iv. **Energy Storage:** They store energy and help stabilise voltage during fluctuations.

c. Operational Amplifiers (Op-Amps)

Op-amps are versatile components used in many applications due to their ability to amplify signals:

i. **Amplification:** They increase the strength of weak signals, useful in audio devices and sensor readings.

ii. **Summing and Difference Amplification:** Op-amps can add multiple signals together or find the difference between two signals.

iii. **Voltage Follower:** They help maintain signal strength without loss.

iv. **Integrators and Differentiators:** With resistors and capacitors, op-amps can perform mathematical operations on signals.

v. **Comparator:** They compare two voltages and output a signal based on the comparison, useful in decision-making circuits.

These components (resistors, capacitors, and operational amplifiers) are vital components in analogue circuits. Each has unique functions that help in designing circuits for various applications, including robotics, where they play roles in sensors, filters, and signal processing. Understanding these components is essential for building effective analogue systems.

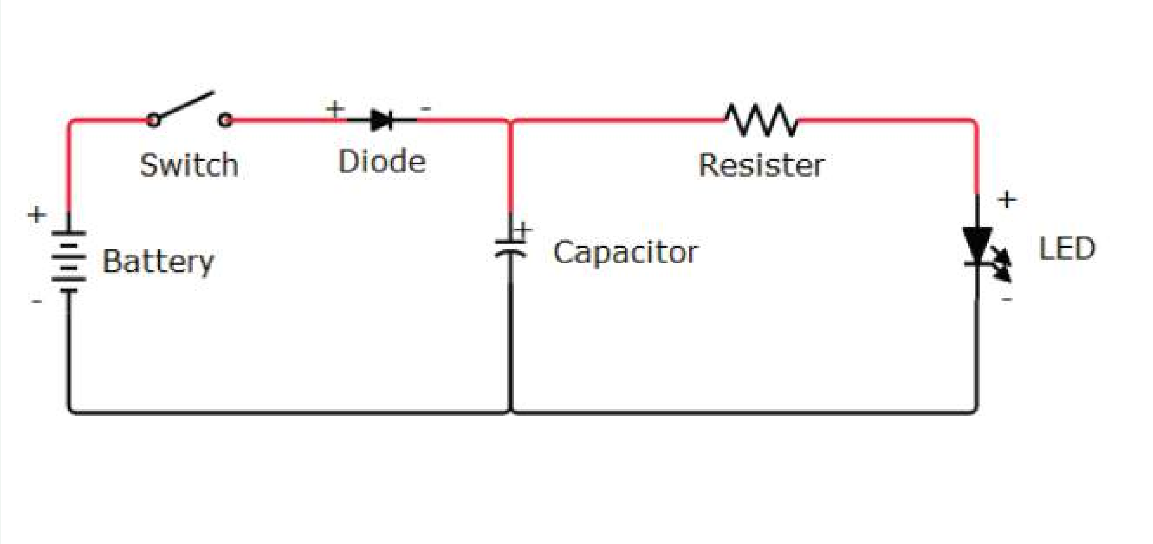
2. Circuit Design

Follow schematic diagrams to connect analogue components and create specific analogue functions.

Your teacher will guide you through the activity below to design an analogue circuit from the circuit schematics shown.

Activity 4.9

Implement the following schematics using the components found in the schematics on the breadboard.



1. **Gather the Components**:

   9V Battery, Switch, Diode, Resistor, Capacitor, LED, Breadboard, and Connecting wires.

2. **Understand the Schematic**:

   The schematic shows a circuit where a battery powers an LED through a switch, diode, resistor, and capacitor.

   The components should be connected as shown in the schematic to ensure correct operation.

3. **Set Up the Power Supply**:

   Connect the positive terminal of the 9V battery to the positive rail of the breadboard.

   Connect the negative terminal of the battery to the negative rail of the breadboard.

4. Place the switch on the breadboard. Connect one terminal of the switch to the positive rail of the breadboard using a wire.

5. **Insert the Diode**

   Connect the other terminal of the switch to the anode (positive side) of the diode.

   Connect the cathode (negative side) of the diode to an empty row on the breadboard.

6. **Add the Capacitor**

   Connect the same row (where the cathode of the diode is connected) to one wire of the capacitor.

   Connect the other wire of the capacitor to the negative rail of the breadboard.

7. **Connect the Resistor**

   Connect one wire of the resistor to the same row as the cathode of the diode and the lead of the capacitor.

   Connect the other wire of the resistor to an empty row on the breadboard.

8. **Insert the LED**

   Connect the anode (longer wire) of the LED to the same row where the resistor ends.

   Connect the cathode (shorter wire) of the LED to the negative rail of the breadboard.

9. **Complete the Circuit**

   Ensure all connections are secure and correct according to the schematic.

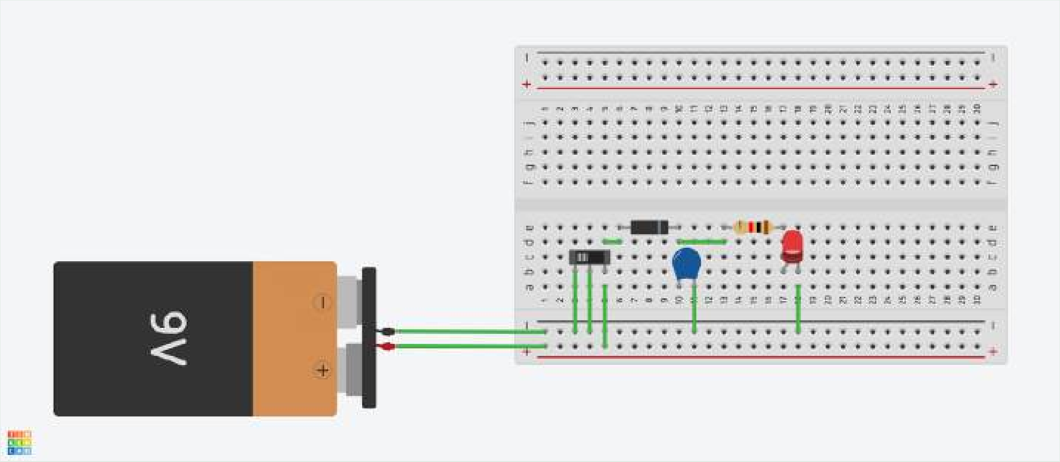
   Double-check that the positive and negative rails are connected properly to the battery.

10. **Test the Circuit**

    Close the switch to complete the circuit.

    The LED should light up if all connections are correct. If it doesn't, recheck all the connections ensuring the correct orientation of the diode and LED, and verify the connections with the schematic.

Your circuit implementation will look like the one below.



C. Discrete-Time Machine Design

Discrete-Time Signal

A discrete-time signal is a type of signal that is defined only at specific, separate points in time. Imagine taking a series of snapshots of a moving object at regular intervals. Each snapshot represents a value of the signal at a particular time. Unlike continuous signals, which are smooth and defined at every moment, discrete-time signals are like a series of steps or dots.

For instance, a clock ticking every second. Each tick represents a discrete point in time. If you were to record the temperature in a room every second, those recorded temperatures would form a discrete-time signal.

Discrete-Time System/Machine

A discrete-time system or machine processes discrete-time signals. It operates at specific intervals rather than continuously. These systems take the values of a discrete-time signal and use them to perform computations or make decisions at those specific times.

For example, consider a digital clock. It updates the time display every second. The system inside the clock reads the time signal (which is discrete because it updates at each second) and changes the display accordingly.

Digital systems play a crucial role in discrete-time machine design, where events occur at specific, well-defined intervals. Here is how digital systems manage these discrete events effectively:

1. **Signal Processing and Computation**
   Digital systems excel at processing discrete signals and performing complex computations. Sensors in discrete-time machines generate data, which is converted into digital form by analogue-to-digital converters (ADCs). In digital form, microcontrollers or digital signal processors (DSPs) can process the data using algorithms, filtering, and transformations to make decisions and control the machine's actions.

2. **Logic-Based Control**
   Digital logic gates and circuits are the backbone of discrete-time control systems. Logic gates (AND, OR, NOT, etc.) are used to evaluate conditions and determine the appropriate control actions based on predefined rules or algorithms. Decision-making processes, such as switching specific actuators or motors on or off, are accomplished through digital control logic.

Finite State Machines (FSMs)

FSMs are widely used in discrete-time machine design. FSMs enable machines to model complex behaviours and decision-making based on different states and transitions. Each state represents a specific condition or behaviour, and digital logic is used to determine state transitions based on inputs from sensors or user commands. The diagram provided illustrates how an FSM operates with states and transitions.

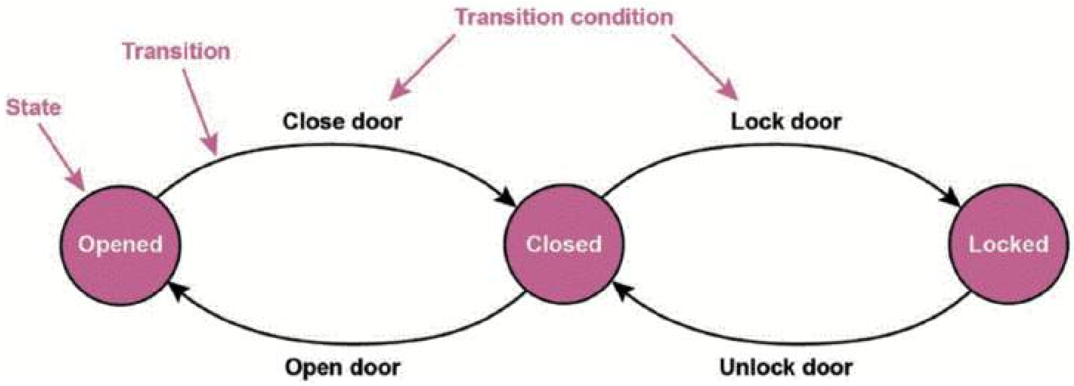


Fig. 4.17: illustration of a Finite State Machine

Timers and Clocks

Digital systems incorporate timers and clocks to control the timing of events in discrete-time machines. Timers can be programmed to trigger specific actions or tasks at predefined intervals, allowing precise control over the machine's operations. This is depicted in the FSM diagram above.

Pulse-Width Modulation (PWM)

PWM is a widely used digital control technique in discrete-time machine design. PWM signals control motor speeds, actuator positions, and light intensity by varying the duty cycle of the digital pulse. This technique is particularly useful in generating smooth analogue-like control signals from digital systems.

Digital Communication and Networking

Digital communication protocols, such as UART, SPI, and I2C, enable discrete-time machines to communicate with other devices or systems. Machines can exchange data with sensors, actuators, or central controllers, facilitating coordinated actions and distributed intelligence.

Digital Feedback Control

Digital feedback control loops ensure accurate and stable control of discrete-time machines. Sensors provide feedback on the machine's current state, which is processed digitally to calculate error signals and adjust control actions for precise regulation.

Programmability and Flexibility

Digital systems offer high programmability and flexibility, making it easier to modify, update, or adapt the control algorithms and behaviour of discrete-time machines. This flexibility enables machines to perform various tasks and adapt to changing environments efficiently.

Signal Processing

Digital systems process discrete signals using binary digits (0s and 1s) to control actions and computations. The processing of discrete signals involves manipulating binary data through logic gates and digital circuits. Digital systems use binary representation to encode information, where each binary digit (bit) can either be a 0 or a 1, representing two distinct states. By combining multiple bits, digital systems can represent more complex data, such as numbers, characters, or sensor readings.

Logic gates are the building blocks of digital circuits. By combining these gates in various configurations, digital circuits can perform complex computations and make decisions based on input conditions. Understanding how to use these gates is essential in creating systems that can solve problems and make decisions based on different inputs.

Digital Control

Microcontrollers and programmable logic controllers (PLCs) are key components in controlling digital systems, providing precise and efficient management. Here's how they work:

1. Real-time computation

Microcontrollers can run complex algorithms and perform real-time calculations. This allows them to process data from sensors, make decisions, and create control signals.

2. Sensor interface

Microcontrollers connect to various sensors to collect data from the environment. They convert the analogue signals from these sensors into digital data for processing.

3. Actuator control

Microcontrollers control actuators, like motors and solenoids, based on the control signals they compute. This ensures that the actuators respond precisely and at the right time.

4. Event-driven control

Microcontrollers can be programmed to react to specific events or conditions. This allows them to perform certain actions at specific times.

5. Feedback control

In closed-loop systems, microcontrollers receive feedback from sensors and adjust control signals as needed to keep the system operating correctly.

Programmable Logic Controllers (PLCs) in Discrete-Time Machine Control

Programmable Logic Controllers (PLCs) are special types of computers used to control machines and processes in factories and other automated environments. They are known for being very reliable and tough, making them perfect for use in challenging conditions. Here's a simplified explanation of the key features of PLCs and how they can be used:

1. **Modularity and Flexibility**
   PLCs are made up of different parts (modules) that can be easily added or removed. These modules allow PLCs to connect with different sensors (which detect changes in the environment) and actuators (which perform actions).

   PLCs can be easily reprogrammed for different tasks, making them very flexible. For example, the same PLC can be used to control different machines by simply changing the program.

2. **Ruggedness and Reliability**
   PLCs are built to withstand tough industrial conditions such as extreme temperatures, dust, and vibrations. This makes them very reliable and suitable for environments where other computers might fail.

3. **Distributed Control**
   PLCs can work together in a network to control complex processes. Multiple PLCs can communicate and coordinate tasks, making it possible to manage large systems efficiently.

4. **Time-Based Sequencing**
   PLCs operate on a fixed time cycle, meaning they can execute commands at precise intervals. This is important for tasks that require exact timing, such as controlling the steps in a manufacturing process.

5. **Fault Tolerance**
   PLCs have built-in systems to detect and fix errors. If something goes wrong, they can often correct the problem themselves or alert operators to take action, ensuring the system continues to operate smoothly.

How Microcontrollers and PLCs Work Together

In many systems, microcontrollers and PLCs (Programmable Logic Controllers) team up to control machines and processes efficiently. Each has its strengths and plays a specific role to achieve the best performance. Let's break down how they work together in a way that's easy to understand.

Microcontrollers are small computers that are really good at handling specific, low-level tasks. Here's what they do:

- **Motor Control**: They control motors that move parts of a machine.

- **Sensor Interfacing**: They connect to sensors that gather information (like temperature or position) from the environment.

Imagine a microcontroller as the part of your brain that controls your hand. It takes care of detailed tasks like picking up a pencil or pressing a button.

PLCs, on the other hand, are like the part of your brain that makes big decisions. They handle high-level control and coordination tasks, making sure everything works together smoothly:

- **Overall Coordination**: PLCs manage and coordinate different parts of the system.
- **Complex Processes**: They handle complex tasks that involve multiple steps and timing.

Think of a PLC as a conductor of an orchestra, ensuring all the musicians (microcontrollers) play in harmony.

When microcontrollers and PLCs work together, they create a powerful and efficient system. Here's how:

- **Division of Labour**: Microcontrollers handle the detailed work, like controlling individual motors and reading sensor data. PLCs focus on the big picture, coordinating multiple tasks and making higher-level decisions.
- **Optimised Resources**: By dividing tasks this way, the system uses its resources efficiently. Microcontrollers don't get overwhelmed with big-picture tasks, and PLCs don't get bogged down with details.
- **Enhanced Responsiveness**: The system responds quickly and accurately to changes. Microcontrollers can make fast adjustments to motors or sensors, while PLCs can update overall strategies as needed.
- **Simplified Control Architecture**: The overall design is simpler and easier to manage. Each part of the system has a clear role, making it easier to troubleshoot and improve.

D. Continuous -Time Machine Design

Continuous-time machine design involves systems that constantly adjust their inputs to keep their outputs stable and accurate. Unlike discrete-time systems that work with distinct events, continuous-time systems operate smoothly and continuously.

Continuous-time control systems work by making ongoing adjustments to input signals to control and change the system's output. Immediate reactions are essential in designing continuous-time machines. Devices like motors, valves, or heaters, known as actuators, receive control signals and make the necessary changes to the system.

Sensors monitor the system's output and send feedback to the controller. The controller then adjusts the control signals to achieve the desired results.

An analogue oscilloscope exemplifies a device that uses continuous-time principles in electronics. When connected to an electrical circuit, it constantly measures the input voltage at specific intervals and displays it as a waveform. These oscilloscopes are essential for testing, calibrating, and troubleshooting analogue circuits.

Key Components:

1. **Actuators:** Devices like motors, valves, or heaters that receive control signals and modify the system.
2. **Sensors:** Measure the system's output and provide feedback to the controller.
3. **Controller:** Adjusts the control signal to achieve the desired output.

Activity 4.10

Exploring Continuous-Time Machine Design

This activity is designed to help you understand how continuous-time systems work using sensors, actuators, and feedback control.

Materials Needed

1. Analogue sensors (e.g., temperature sensor, light sensor)
2. Actuators (e.g., small motor, heater, or LED)
3. Analogue oscilloscope (or a digital oscilloscope if available)
4. Power supply
5. Resistors and capacitors
6. Wires and breadboard

Steps to follow

1. **Introduction to Continuous-Time Systems**
   - Explain the key concepts of continuous adjustment, real-time reactions, and the role of sensors and actuators.
2. **Sensor Interfacing**
   - Connect a temperature sensor to the breadboard. Explain how the sensor converts temperature into an analogue signal (voltage).
   - Connect the sensor to the oscilloscope to observe the continuous voltage change as the temperature varies (e.g., by placing a warm object near the sensor).
3. **Real-Time Control Setup**
   - Connect an actuator (like a small motor) to the breadboard.
   - Use a simple analogue control circuit (like a transistor-based amplifier) to control the motor based on the sensor's output.
4. **Feedback Loop**
   - Create a feedback loop where the sensor measures the output (e.g., motor speed or temperature).

- Connect the sensor output to the controller, which adjusts the actuator to maintain the desired output.

5. **Observation and Analysis**
   - Use the oscilloscope to observe the continuous feedback loop in action.
   - Change the conditions (like temperature or light) and observe how the system reacts in real time.
6. **Discussion**
   - Discuss how continuous-time systems differ from discrete-time systems.
   - Explore real-world applications (like thermostats, light dimmers, and speed controllers).
7. **Hands-On Experiment**
   - Learners sit in groups and ensure each group has all the necessary components. The components will be provided by your teacher.
   - In your group build a simple continuous-time control system, such as a temperature control circuit using a heater and temperature sensor.

Example Experiment: Temperature Control System

To build a simple continuous-time temperature control system.

Materials Needed:

1. Temperature sensor (e.g., LM35)
2. Small heater (e.g., a resistor or heating element)
3. Transistor (e.g., TIP31)
4. Power supply (e.g., 12V battery or DC power supply)
5. Resistors, capacitors, and wires
6. Breadboard
7. Analogue oscilloscope

Steps to follow

1. **Circuit Design**
   - Connect the temperature sensor to the breadboard.
   - Use the transistor to control the heater based on the sensor's output.
   - Connect the sensor output to the base of the transistor through a resistor.
   - Connect the heater in series with the transistor and power supply.
2. **Building the Circuit**
   - Assemble the circuit on the breadboard.
   - Connect the sensor output to the oscilloscope to monitor the voltage changes.

3. **Testing and Observation**
   - Power the circuit and observe how the heater responds to temperature changes.
   - Use the oscilloscope to monitor the continuous feedback loop.
4. **Analysis and Discussion**
   - Discuss how the system maintains the desired temperature.
   - Explore how adjusting the sensor or control circuit affects the system's performance.

Analogue Oscilloscope

An analogue oscilloscope is an example of a continuous-time machine. It continuously samples the input voltage of an electrical circuit at specific time intervals, producing a waveform. This is useful for testing, calibrating, and troubleshooting analogue circuits.

Applications of Analogue Systems in Continuous-Time Machine Design:

1. **Sensor Interfacing:**
   - Analogue circuits convert continuous physical quantities (like temperature, pressure, or light) into electronic signals.
   - Example: A thermocouple sensor that measures temperature and converts it into a voltage signal.
2. **Real-Time Control:**
   - Analogue feedback control systems are used to make continuous adjustments based on the feedback from sensors.
   - Example: A thermostat controlling a heater, where the sensor continuously measures the temperature, and the controller adjusts the heater output to maintain a desired temperature.

Activity 4.11

Exploring Continuous-Time Control Systems

Understand how continuous-time control systems work by interfacing sensors and actuators using an analogue oscilloscope.

Materials:

- Analogue oscilloscope
- Thermocouple sensor (or any other sensor that provides a continuous signal)
- Heater or motor (as an actuator)
- Breadboard and connecting wires
- Power supply

Steps:

1. **Setup the Sensor and Actuator:**
   - Connect the thermocouple sensor to the breadboard.
   - Connect the output of the sensor to the analogue oscilloscope.
2. **Observe the Sensor Output:**
   - Turn on the oscilloscope and observe the waveform produced by the sensor.
   - Note how the waveform changes as the physical quantity (like temperature) changes.
3. **Implement Real-Time Control:**
   - Connect the actuator (heater or motor) to the system.
   - Use a simple analogue control circuit to connect the sensor and the actuator.
   - Set up the control circuit so that the actuator responds to changes in the sensor's output.
4. **Monitor and Adjust:**
   - Monitor the system using the oscilloscope.
   - Observe how the actuator's behaviour changes in response to the sensor's readings.
   - Adjust the control parameters (like gain or setpoint) and see how the system stabilises to the desired output.

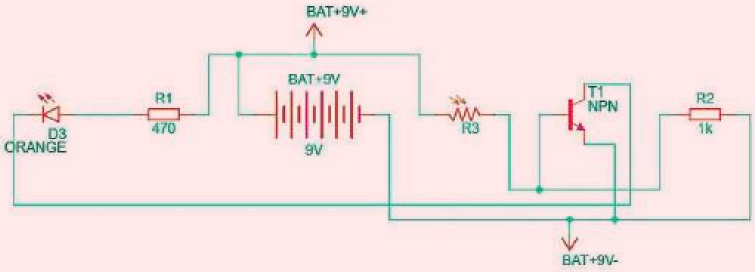
Discussions

1. What changes did you observe in the waveform when the physical quantity measured by the sensor changed?
2. How did the actuator respond to changes in the sensor's output?
3. What adjustments did you make to the control parameters, and how did they affect the system's stability?

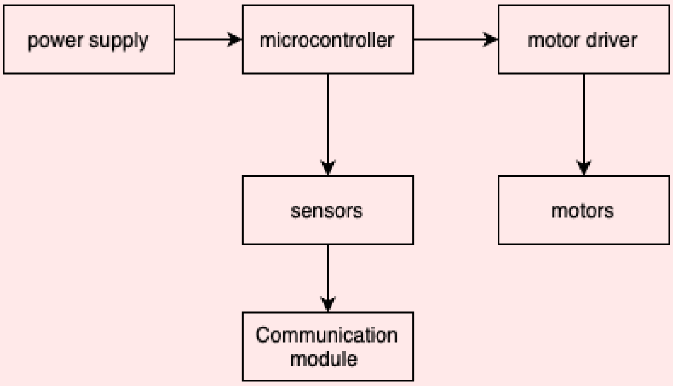
Review Questions

Review Questions 4.1

1. Which of the following components allows current to flow in one direction only?
   a. Resistor
   b. LED (Light Emitting Diode)
   c. Capacitor
   d. Transistor
2. A battery is an active component, while a resistor is a passive component.
   a. True
   b. False
3. A capacitor and an inductor are both passive electronic components; state the effect of these components on alternating current (AC) signals.
4. Consider the circuit diagram below and list the electronic components.



Review Questions 4.2



Question 1 and 2

The block diagram above represents a simple electronic system used in a robotic system.

1. What are the three main components represented in the block diagram?
2. Explain the purpose of using block diagrams in robotics.

3. Imagine you are tasked with designing a robotic system to automate complex tasks in a busy factory setting. In this scenario, why would using schematic diagrams be essential?
4. How are system inputs typically shown in a block diagram?

Review Questions 4.3

Question 1

Imagine you are working on a project to build a simple electronic circuit for a science fair. Why must you carefully follow the schematic diagram provided in the instructions when assembling your circuit?

Question 2

Your task is to build an electronic circuit for your school's science fair exhibition. Your teacher then gives you two options: use a solderless breadboard or traditional soldering to assemble your circuit.

I. Why might using the solderless breadboard be a better choice for your project?

II. List two safety precautions you will observe before starting and during the science fair project.

III. State the uses of the solderless breadboard in your project.

Review Questions 4.4

1. Imagine you are working on a project to build a simple light dimmer for your bedroom. This light dimmer will allow you to adjust the brightness of a lamp using a knob. The circuit includes a potentiometer (variable resistor) connected in series with a lamp and a battery. Turning the knob on the potentiometer changes its resistance, which controls the brightness of the lamp by allowing more or less current to flow through it.
   a. Is the circuit described in the scenario an analogue or digital circuit?
   b. State the function of the variable resistor in the light dimmer circuit described above.
2. Describe what analogue circuits are. What is the role of an operational amplifier in an analogue circuit?
3. Logic gates are the building blocks of digital circuits (True or False?). Mention any four (4) components found in digital circuits.

EXTENDED READING

1. Electronic components: Basic components, parts and function by *LoneStar Technologies*
https://www.lonestartech.tw/electronic-component-basic-components-parts-functions/#:~:text=Capacitors%3A%20Store%20and%20release%20electrical,for%20rectification%20and%20signal%20modulation.

2. Electronic Components by *GeeksforGeeks* | https://www.geeksforgeeks.org/electronic-components/

3. Makerspace Basic Electronics | https://www.makerspaces.com/basic-electronics/

4. Consider further reading on block diagram design and representation | https://www.mathworks.com/discovery/block-diagram.html

5. Consider further reading on schematic diagram design | https://www.thoughtco.com/what-is-a-schematic-diagram-4584811

6. The link provided below is a free, easy-to-use app for 3D design, electronics, and coding. Visit this platform and practice as many circuit simulations as possible.

   https://www.tinkercad.com/

7. Watch the following videos on how to practise circuit building simulation with Tinkercad

   a. Introduction to Tinkercad Circuits & Breadboarding - Part 1

   b. Intro to Tinkercad Circuits Part 2 - Varying the resistance of an LED circuit

8. Science Buddies. How to Use a Breadboard. https://www.sciencebuddies.org/science-fair-projects/references/how-to-use-a-breadboard

9. Juan P. B. Introduction to Digital Systems| http://dl.icdst.org/pdfs/files/1a0d1181a845ab8d63dba21fbd73f7f3.pdf

10. Ishita K. (2022). Analogue and Digital Systems. https://medium.com/@ishita.kadam20/analog-and-digital-systems-e077e3d6636b

REFERENCES

1. Drawing of block diagrams | www.apps.diagrams.net
2. Drawing of schematic designs | www.tinkercad.com
3. Multimeter - Wikipedia. https://en.wikipedia.org/wiki/Multimeter#:~:text=A%20multimeter%20(also%20known%20as,voltmeter%2C%20ohmmeter%2C%20and%20ammeter.
4. Beig, F. (2023, September 14). Simple light sensor circuit. Circuits DIY. https://www.circuits- diy.com/simple-light-sensor-circuit/
5. *HELPDESK_WJ (Waijung)*. DC Motors Control - HELPDESK_WJ (Waijung) - Aimagin Support. (n.d.). https://support.aimagin.com/projects/support/wiki/DC_Motors_Control
6. *Combinational Logic Circuits*. Digital and Analog Electronics Course. (n.d.). https:// electronics-course.com/combinational-logic
7. Kinzar Y. Digital. https://www.techtarget.com/whatis/definition/digital

ACKNOWLEDGEMENTS

List of Contributors

Name	Institution
Griffith Selorm Klogo	Kwame Nkrumah University of Science and Technology
Asare Boakye Ansah	Kumasi Technical University
Gershon Normenyo	Kwame Nkrumah University of Science and Technology
Nero Kofi Etornam Novor	Kwame Nkrumah University of Science and Technology
Samuel Quarm Olatunde	GES, Kumasi