Day 1

Theory

- **Resistor:** Limits current. V=IRV=IR
- **Capacitor:** Stores charge. Used for filtering/timing.
- **Diode:** One-way current. LED emits light.
- **Transistor:** Acts as a switch/amplifier (BJT: Base, Collector, Emitter).

Practical

- **Identify:** Use color codes (resistors), markings (caps, diodes)
- **Test:**
  - *Resistor:* Multimeter (Ω mode)
  - *Capacitor:* Capacitance mode
  - *Diode:* Diode mode (0.6–0.7V forward)
  - *Transistor:* Test junctions like two diodes

Day 2

Circuit Laws (Ohm's & Kirchhoff's)

Theory:

- **Ohm's Law**: V=IRV=IR
- **KCL**: Sum of currents at a node = 0
- **KVL**: Sum of voltages in a loop = 0

Practical:

- **Simulation**: Use Proteus/Multisim to build circuits, measure voltages/currents.
- **Real Circuit**: Use breadboard, resistors, DC supply, and multimeter to verify laws.

Day 3

Semiconductor Basics

Theory

- Diode: Allows current in one direction.
- BJT: Current-controlled switch (NPN/PNP).
- MOSFET: Voltage-controlled switch (n/p-channel).

Practical

- Diode Test: Forward = ~0.7V, Reverse = OL.
- BJT Test: Check B-E and B-C like diodes.
- MOSFET Test: Check G-D, G-S isolation; D-S after charging gate.

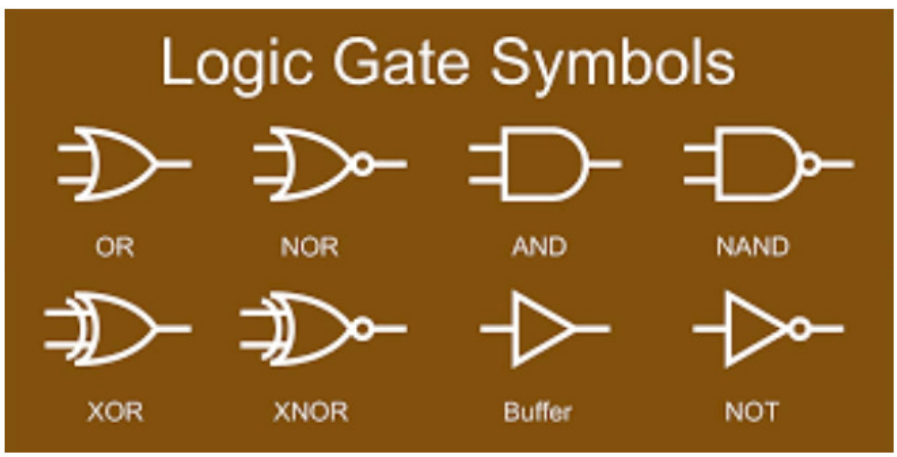
Day 4

Logic Gates

- AND: Output 1 if both inputs 1.
- OR: Output 1 if any input 1.
- NOT: Inverts input.
- NAND/NOR: Opposites of AND/OR.

Practical

- Simulate in TinkerCAD.
- Build with ICs on breadboard.



Day 5

Soldering & PCB Basics

Theory

- PCB Design Workflow:
    1. Create schematic (circuit diagram)
    2. Design PCB layout
    3. Generate Gerber files
    4. Send for fabrication

Practical

- Soldering:
    - Practice soldering components on perf board or sample PCB
    - Focus on clean joints, proper heating, and safety
- Design Tools:
    - KiCad / Eagle: Used to create schematics and PCB layouts
    - Demo: Place components, route traces, export files

Day 6

Arduino / Raspberry Pi Intro

Practical: Blink LED

- Arduino:
  - Program pin 13 to blink an LED (C/C++ in Arduino IDE).
  - Code: Turn pin HIGH for 1s, then LOW for 1s.
- Raspberry Pi:
  - Use GPIO pins to blink an LED (Python with RPi.GPIO library).
  - Code: Set GPIO pin, turn HIGH/LOW with delays.

Day 7

Sensor Interfacing

Practical Projects

- LDR (Light Sensor):
  - Varies resistance with light; use in auto-light systems.
- Ultrasonic Sensor:
  - Measures distance using echo time (e.g., obstacle detection).
- IR Sensor:
  - Detects objects or motion; used in line-following robots.

Day 8

Motor Control

Practical

- DC Motor:
  - Controlled via L293D H-bridge.
  - Use Arduino to set direction and speed (PWM).
- Servo Motor:
  - Controlled with PWM signal (0–180° rotation).
  - Signal sent from Arduino pin.

Day 9

Wireless Communication (IoT)

Practical Applications

- ESP8266 (Wi-Fi):
  - Enables internet connectivity for Arduino.
  - Common use: Send sensor data to a web server or cloud.
  - Can create a local web page to monitor/control devices.
- Bluetooth (HC-05 / ESP32):
  - Allows wireless communication with smartphones.
  - Used for controlling LEDs, motors, etc., via mobile apps (e.g., Bluetooth Terminal, MIT App Inventor).

Day 11

Theory

- Rectifiers:
  - Convert AC to DC.
  - Types: Half-wave, full-wave, bridge.
- SMPS (Switch Mode Power Supply):
  - Efficient DC conversion using high-frequency switching.
  - Used in chargers, power adapters.

Demo

- Show rectifier output using diodes + filter capacitor.
- SMPS module demo with input/output voltage check.

Day 12

IoT & Industry 4.0

Concept

- IoT: Connects devices to share data via the internet.
- Industry 4.0: Combines IoT, automation, and cloud for smart systems.

Case Study: Cloud Data Logging

- Platforms: ThingSpeak, Blynk
- Flow: Sensor → ESP8266/ESP32 → Wi-Fi → Cloud
- Use: Real-time data monitoring (e.g., temperature, light)

Applications

- Smart farming
- Factory monitoring
- Home automation

Day 13

PCB Fabrication & Troubleshooting

Activities

- Lab Visit (if available):
  - Observe PCB manufacturing process: etching, drilling, masking.
- Soldering Practice:
  - Hands-on with components on a PCB.
  - Focus on clean joints, avoiding bridges.

Troubleshooting

- Check for:
  - Loose connections
  - Solder bridges
  - Component orientation
  - Continuity with multimeter