Rete (4361106) - Problem Solver

Antigravity AI

June 4, 2026

Rete

First Edition: 2026

Author: Antigravity AI
Website: milav.in
YouTube: @milav.dabgar

To my students,
who constantly inspire me to find new analogies,
break down complex theories, and make learning
an engineered science.

Contents

Preface

About this Book

Welcome to **Computer Networks**! This textbook is meticulously designed to serve as a comprehensive problem solver and study companion. It provides a robust and intuitive foundation in the core principles of this subject, bridging the gap between theoretical models and practical applications. A unique feature of this book is its focus on decoding complex concepts using clear, step-by-step methodologies and real-world analogies.

Target Audience

This book is intended for students and professionals looking to master the concepts of Computer Networks. Whether you are revising for exams, seeking to solidify your fundamental understanding, or looking for practical examples to clarify abstract concepts, this book has been structured to support your learning journey. It serves as an accessible guide designed to remove the fear of complex topics, replacing it with an appreciation for the underlying mechanics.

Scope and Coverage

The coverage is divided logically to follow standard academic curriculums. Each chapter focuses on key topics, breaking them down into digestible sections:

- **Theoretical Insights** – Core concepts explained intuitively.
- **Method Blocks** – Standardized, step-by-step procedures for solving specific types of problems.
- **Worked Examples** – Fully solved problems that apply the methods in practice.

We hope this resource proves invaluable in your studies and helps you achieve excellence in **Computer Networks**. Happy learning!

Acknowledgments

Writing a comprehensive book that connects the fundamental forces of Chemistry with practical engineering applications requires more than just academic knowledge—it requires a community of support. I would like to express my sincere gratitude to everyone who contributed to the realization of this project.

Specially, I extend my heartfelt gratitude to the **Department of Technical Education (DTE), Government of Gujarat**, and **Government Polytechnic, Palanpur**, for providing an environment that fosters innovation, academic rigor, and continuous professional development.

I am deeply thankful to my family for their unwavering patience and support during the countless hours spent researching, formatting, and refining these chapters.

Finally, my deepest appreciation goes to my students. Your curiosity, challenging questions, and continuous feedback are the true driving force behind the unique analogies and streamlined structure of this book. This textbook was engineered for you.

Antigravity AI

About the Author

Milav Jayeshkumar Dabgar is an engineering educator and R&D professional with over 9 years of experience in electronics hardware development, embedded systems, AI/ML, and full-stack software engineering. He currently serves as a **Lecturer (Class - II) & Innovation Leader** at Government Polytechnic, Palanpur, under the Education Department of the Government of Gujarat.

Milav holds a Master of Engineering (ME) in Communication Systems from L.D. College of Engineering and a Bachelor of Engineering (BE) in Electronics & Communication. He is also currently pursuing a **BS in Data Science and Applications from IIT Madras**, where he has completed both the **Diploma in Programming** and the **Diploma in Data Science** with distinction.

Most recently, he completed the **AICTE QIP PG Certificate Program in Deep Learning from SVNIT Surat** with a **perfect 10/10 CGPA**, topping his batch.

A dedicated mentor, Milav has guided numerous student teams in securing over **INR 45 lakhs** in innovation funding, including INR 25 lakhs from Shark Tank India and INR 20 lakhs through iHub Gujarat, resulting in two student patents. He is recognized as an **NPTEL Evangelist** and **NPTEL Discipline Star** in Computer Science, reflecting his commitment to continuous learning and academic excellence.

His technical expertise spans from embedded systems (8051, STM32, Arduino) to modern web technologies (Next.js, Python, PostgreSQL) and Data Science (TensorFlow, PyTorch). Through this textbook, he aims to simplify complex Engineering Chemistry concepts by integrating relatable, real-world analogies drawn from modern tech and engineering landscapes.

Milav is also an active content creator, running a popular **YouTube channel** (@milav.dabgar) where he shares technical tutorials and academic resources. He maintains a personal blog and resource hub at milav.in and can be reached for professional collaborations on LinkedIn.

CHAPTER 1

Renewable Energy Systems and Applications

Renewable energy systems harness natural resources such as sunlight, wind, and water to generate power. This chapter focuses on the fundamental computational methods used to evaluate the power output, efficiency, and performance of various renewable energy systems.

1.1 Energy and Power Fundamentals

Methodology:

Basic Power and Energy Calculations To calculate the total energy produced or consumed over a period and to convert between different units of energy:

1. **Power and Energy Relationship:** Energy (E) is the product of Power (P) and Time (t).

$$E = P \times t$$

where P is in Watts (W) or kilowatts (kW), and t is in hours (h) or seconds (s).

2. **Unit Conversions:**

$$1 \text{ kWh} = 3.6 \times 10^6 \text{ Joules (J)}$$

3. **Overall Efficiency:** The efficiency (η) of a system is the ratio of useful energy output to total energy input.

$$\eta = \frac{\text{Energy Output}}{\text{Energy Input}} \times 100\%$$

Worked Example:

Calculate total energy and overall efficiency A renewable energy plant consumes 500 MJ of input resource energy to produce 40 kWh of electrical energy. Calculate the total electrical energy in Joules and the overall efficiency of the plant.

Solution:

1. **Convert electrical energy to Joules:**

$$\text{Energy Output} = 40 \text{ kWh} = 40 \times 3.6 \times 10^6 \text{ J} = 144 \times 10^6 \text{ J} = 144 \text{ MJ}$$

2. **Calculate overall efficiency:**

$$\eta = \frac{\text{Energy Output}}{\text{Energy Input}} \times 100\%$$

$$\eta = \frac{144 \text{ MJ}}{500 \text{ MJ}} \times 100\% = 28.8\%$$

The total electrical energy produced is 144 MJ and the overall efficiency is 28.8%.

1.2 Solar Energy Calculations

Methodology:

Solar Photovoltaic Power Output The power generated by a solar photovoltaic (PV) panel depends on the solar irradiance, the area of the panel, and its efficiency.

1. **Input Solar Power:**

$$P_{in} = G \times A$$

where G is the solar irradiance (W/m^2) and A is the surface area of the panel (m^2).

2. **Electrical Power Output:**

$$P_{out} = P_{in} \times \eta = G \times A \times \eta$$

where η is the efficiency of the solar panel (expressed as a decimal).

3. **Daily Energy Output:**

$$E_{daily} = P_{out} \times \text{Peak Sun Hours (PSH)}$$

Worked Example:

Calculate solar panel power and daily energy A solar PV panel has dimensions of 1.6 m × 1.0 m and an efficiency of 18%. It is installed in a location with a solar irradiance of 850 W/m^2 and receives 5.5 peak sun hours per day. Calculate the electrical power output and the daily energy produced.

Solution:

1. **Calculate the area of the panel:**

$$A = \text{length} \times \text{width} = 1.6 \times 1.0 = 1.6 \text{ m}^2$$

2. **Calculate the electrical power output:**

$$P_{out} = G \times A \times \eta$$

$$P_{out} = 850 \times 1.6 \times 0.18 = 244.8 \text{ W}$$

3. **Calculate the daily energy output:**

$$E_{daily} = P_{out} \times \text{PSH}$$

$$E_{daily} = 244.8 \text{ W} \times 5.5 \text{ h} = 1346.4 \text{ Wh} = 1.346 \text{ kWh}$$

The solar panel produces 244.8 W of power and 1.346 kWh of energy per day.

1.3 Wind Energy Calculations

Methodology:

Wind Turbine Power Equation The kinetic energy of wind is converted into mechanical and then electrical energy by a wind turbine. The power available in the wind is proportional to the cube of the wind speed.

1. **Power Available in Wind:**

$$P_{wind} = \frac{1}{2}\rho A v^3$$

where:

- ρ = Air density (approx. 1.225 kg/m^3 at sea level)

- A = Swept area of the rotor blades (πr^2 in m^2)
- v = Wind velocity (m/s)

2. **Actual Power Output (Betz Limit and Efficiency):** According to Betz's law, the maximum theoretical efficiency is 59.3%. The actual power output includes the coefficient of performance (C_p) and generator efficiency (η_g):

$$P_{out} = \frac{1}{2}\rho A v^3 \times C_p \times \eta_g$$

Worked Example:

Calculate wind turbine actual power output A wind turbine has a rotor diameter of 80 m. It operates in a wind speed of 12 m/s. The air density is 1.2 kg/m^3. The coefficient of performance (C_p) is 0.40 and the generator efficiency is 90%. Calculate the available wind power and the actual electrical power output in MW.

Solution:

1. **Calculate the swept area of the rotor:** Radius $r = \frac{80}{2} = 40$ m.

$$A = \pi r^2 = 3.1416 \times (40)^2 = 5026.5 \text{ m}^2$$

2. **Calculate the power available in the wind:**

$$P_{wind} = \frac{1}{2}\rho A v^3$$

$$P_{wind} = 0.5 \times 1.2 \times 5026.5 \times (12)^3 = 0.6 \times 5026.5 \times 1728 = 5,211,475 \text{ W} \approx 5.21 \text{ MW}$$

3. **Calculate the actual electrical power output:**

$$P_{out} = P_{wind} \times C_p \times \eta_g$$

$$P_{out} = 5.21 \text{ MW} \times 0.40 \times 0.90 = 1.8756 \text{ MW}$$

The available wind power is 5.21 MW and the actual electrical power output is 1.88 MW.

1.4 Hydroelectric Power Calculations

Methodology:

Hydro Power Output Hydroelectric power is generated by utilizing the potential energy of water stored at a height (head).

1. **Hydro Power Equation:**

$$P = \rho \times g \times Q \times H \times \eta$$

where:

- P = Electrical power output (W)
- ρ = Density of water (1000 kg/m^3)
- g = Acceleration due to gravity (9.81 m/s^2)
- Q = Water flow rate or discharge (m^3/s)
- H = Net head (effective height of water falling) (m)
- η = Overall efficiency of the turbine and generator (decimal)

Worked Example:

Calculate hydro plant power output A hydroelectric power plant has a net head of 50 m and a water flow rate of 20 m^3/s. If the overall efficiency of the turbine-generator system is 85%, calculate the power generated in MW.

Solution:

1. **Identify given parameters:** $\rho = 1000$ kg/m^3, $g = 9.81$ m/s^2, $Q = 20$ m^3/s, $H = 50$ m, $\eta = 0.85$.

2. **Apply the hydro power equation:**

$$P = \rho \times g \times Q \times H \times \eta$$

$$P = 1000 \times 9.81 \times 20 \times 50 \times 0.85$$

3. **Calculate the power:**

$$P = 1000 \times 9.81 \times 1000 \times 0.85 = 8,338,500 \text{ W}$$

$$P = 8.3385 \text{ MW}$$

The hydroelectric plant generates approximately 8.34 MW of power.

1.5 Plant Performance Metrics

Methodology:

Capacity Factor and Availability Factor Evaluating the performance of renewable energy systems requires an understanding of their operational metrics over a specific period (usually a year).

1. **Capacity Factor (CF):** The ratio of actual energy produced to the maximum possible energy produced if the plant operated at full rated capacity continuously.

$$\text{CF} = \frac{\text{Actual Energy Produced in a period}}{\text{Rated Capacity} \times \text{Total Hours in period}}$$

2. **Availability Factor (AF):** The percentage of time a power plant is available to generate power (i.e. not offline for maintenance or repairs).

$$\text{AF} = \frac{\text{Hours Available}}{\text{Total Hours in period}} \times 100\%$$

Worked Example:

Calculate capacity factor of a wind farm A 50 MW wind farm generates 153300 MWh of energy in a non-leap year. It was shut down for maintenance for 360 hours during the year. Calculate the Capacity Factor and the Availability Factor of the wind farm.

Solution:

1. **Calculate total hours in a non-leap year:**

$$\text{Total Hours} = 365 \text{ days} \times 24 \text{ hours/day} = 8760 \text{ hours}$$

2. **Calculate the Capacity Factor:** The maximum possible energy if operated at full capacity for the entire year:

$$\text{Max Energy} = 50 \text{ MW} \times 8760 \text{ h} = 438000 \text{ MWh}$$

$$\text{CF} = \frac{153300 \text{ MWh}}{438000 \text{ MWh}} = 0.35 = 35\%$$

3. **Calculate the Availability Factor:** The plant was available for $8760 - 360 = 8400$ hours.

$$\text{AF} = \frac{8400 \text{ h}}{8760 \text{ h}} \times 100\% = 95.89\%$$

The wind farm has a Capacity Factor of 35% and an Availability Factor of 95.89%.

CHAPTER 2

Smart Materials and Systems

This chapter focuses on computational and analytical problem-solving techniques for smart materials, including piezoelectric elements, thermoelectric generators, shape memory alloys, dielectric elastomers, and magnetostrictive materials. The emphasis is on the mathematical modeling of their actuation, energy generation, and deformation characteristics.

2.1 Piezoelectric Sensors and Actuators

Methodology:

Piezoelectric Sensor and Actuator Calculations Piezoelectric materials generate an electrical charge when subjected to mechanical stress (sensor mode) and undergo physical deformation when subjected to an electric field (actuator mode).

1. Charge and Voltage Generation When a force F is applied, the charge Q generated is:

$$Q = d_{33} \times F$$

where d_{33} is the piezoelectric charge constant (C/N) and F is the applied force (N).

The capacitance C of the piezoelectric element is:

$$C = \frac{\varepsilon_0 \varepsilon_r A}{t}$$

where ε_0 is the permittivity of free space (8.854×10^{-12} F/m), ε_r is the relative permittivity, A is the cross-sectional area (m^2), and t is the thickness (m).

The generated output voltage V is:

$$V = \frac{Q}{C}$$

2. Actuator Displacement When a voltage V is applied, the change in thickness Δt is:

$$\Delta t = d_{33} \times V$$

Worked Example:

Calculate Output Voltage for a Piezoelectric Disc A piezoelectric disc has a diameter of 20 mm and a thickness of 2 mm. The material properties are $d_{33} = 250 \times 10^{-12}$ C/N and relative permittivity $\varepsilon_r = 1500$. A mechanical force of 500 N is applied across its thickness. Calculate the generated charge, the capacitance of the disc, and the output voltage. Assume $\varepsilon_0 = 8.854 \times 10^{-12}$ F/m.

Step 1: Identify given parameters and calculate area Diameter $D = 20$ mm $= 0.02$ m

Thickness $t = 2$ mm $= 0.002$ m

Force $F = 500$ N

Area $A = \frac{\pi D^2}{4} = \frac{\pi \times (0.02)^2}{4} = 3.1416 \times 10^{-4}$ m^2

Step 2: Calculate the Capacitance

$$C = \frac{\varepsilon_0 \varepsilon_r A}{t} = \frac{(8.854 \times 10^{-12}) \times 1500 \times (3.1416 \times 10^{-4})}{0.002}$$

$$C = 2.086 \times 10^{-9} \text{ F} = 2.086 \text{ nF}$$

Step 3: Calculate the Generated Charge

$$Q = d_{33} \times F = (250 \times 10^{-12}) \times 500 = 1.25 \times 10^{-7} \text{ C}$$

Step 4: Calculate the Output Voltage

$$V = \frac{Q}{C} = \frac{1.25 \times 10^{-7}}{2.086 \times 10^{-9}} \approx 59.92 \text{ V}$$

2.2 Thermoelectric Energy Generation

Methodology:

Thermoelectric Generator Evaluation Thermoelectric generators (TEGs) utilize the Seebeck effect to convert temperature differences into electrical voltage.
1. Seebeck Voltage The open-circuit voltage V_{oc} generated by a TEG is proportional to the temperature difference ΔT:

$$V_{oc} = S \times \Delta T = S \times (T_h - T_c)$$

where S is the Seebeck coefficient (V/K), T_h is the hot-side temperature, and T_c is the cold-side temperature.
2. Figure of Merit (ZT) The efficiency of a thermoelectric material is characterized by the dimensionless Figure of Merit:

$$ZT = \frac{S^2 \sigma}{k} T$$

where σ is the electrical conductivity (S/m), k is the thermal conductivity (W/mK), and T is the average operating temperature $T = \frac{T_h + T_c}{2}$ in Kelvin.
3. Maximum Power Output For a generator with internal resistance R_{in}, maximum power is transferred to a load R_L when $R_L = R_{in}$. The maximum power output P_{max} is:

$$P_{max} = \frac{V_{oc}^2}{4R_{in}}$$

Worked Example:

Calculate Power Output and ZT for a Thermoelectric Material A thermoelectric generator module uses a material with a Seebeck coefficient $S = 220\ \mu$V/K, electrical conductivity $\sigma = 1.1 \times 10^5$ S/m, and thermal conductivity $k = 1.6$ W/mK. The module produces an open-circuit voltage of 4.5 V when operating between 150°C and 30°C and has an internal resistance of 2.0 Ω. Calculate the dimensionless Figure of Merit (ZT) of the material at the average temperature and the maximum power delivered by the module.
Step 1: Determine Temperatures in Kelvin Hot side $T_h = 150 + 273.15 = 423.15$ K
Cold side $T_c = 30 + 273.15 = 303.15$ K
Average temperature $T = \frac{423.15 + 303.15}{2} = 363.15$ K
Step 2: Calculate Figure of Merit (ZT) for the material The Seebeck coefficient is $S = 220 \times 10^{-6}$ V/K.

$$ZT = \frac{S^2 \sigma}{k} T$$

$$ZT = \frac{(220 \times 10^{-6})^2 \times (1.1 \times 10^5)}{1.6} \times 363.15$$

$$ZT = \frac{(4.84 \times 10^{-8}) \times (1.1 \times 10^5)}{1.6} \times 363.15$$

$$ZT = \frac{0.005324}{1.6} \times 363.15 \approx 0.0033275 \times 363.15 \approx 1.21$$

Step 3: Calculate Maximum Power Output of the module Given open-circuit voltage $V_{oc} = 4.5$ V and internal resistance $R_{in} = 2.0\ \Omega$.

$$P_{max} = \frac{V_{oc}^2}{4R_{in}} = \frac{(4.5)^2}{4 \times 2.0} = \frac{20.25}{8.0} \approx 2.53 \text{ W}$$

2.3 Shape Memory Alloys

Methodology:

SMA Actuator Mechanics Shape Memory Alloys (like Nitinol) can be deformed when cold and return to their pre-deformed shape when heated.

1. Recovery Stress When an SMA wire is prevented from recovering its shape during heating, it generates a recovery stress σ_r:

$$\sigma_r = E \times \varepsilon_r$$

where E is the Young's Modulus of the SMA in its high-temperature (Austenite) phase, and ε_r is the initial imparted strain.

2. Recovery Force and Work Done The recovery force F exerted by an SMA wire of cross-sectional area A is:

$$F = \sigma_r \times A$$

If the wire is allowed to contract against a constant force, the mechanical work done W is:

$$W = F_{load} \times \Delta L$$

where $\Delta L = \varepsilon_r \times L$ is the recoverable change in length.

Worked Example:

Calculate Recovery Force of a Nitinol Wire Actuator A Nitinol wire has a diameter of 0.5 mm and an initial length of 200 mm. It is stretched by 4% (strain $\varepsilon_r = 0.04$) at room temperature. It is then heated above its transformation temperature to its austenite phase while its ends are rigidly clamped to prevent any length change. The Young's Modulus of the austenite phase is 75 GPa. Calculate the recovery force exerted by the wire.

Step 1: Calculate the cross-sectional area Diameter $D = 0.5 \text{ mm} = 0.5 \times 10^{-3}$ m

$$A = \frac{\pi D^2}{4} = \frac{\pi \times (0.5 \times 10^{-3})^2}{4} \approx 1.963 \times 10^{-7} \text{ m}^2$$

Step 2: Calculate the recovery stress Young's Modulus $E = 75 \text{ GPa} = 75 \times 10^9 \text{ N/m}^2$.

$$\sigma_r = E \times \varepsilon_r = (75 \times 10^9) \times 0.04 = 3 \times 10^9 \text{ N/m}^2 = 3000 \text{ MPa}$$

Step 3: Calculate the recovery force

$$F = \sigma_r \times A = (3 \times 10^9) \times (1.963 \times 10^{-7}) = 588.9 \text{ N}$$

The wire exerts a recovery force of approximately 588.9 N.

2.4 Dielectric Elastomer Actuators

Methodology:

Dielectric Elastomer Actuation Dielectric Elastomer Actuators (DEAs) are a type of electroactive polymer that experience large strains when subjected to an electric field.

1. Electrostatic Maxwell Pressure When a voltage V is applied across a dielectric film of thickness t, it generates an electrostatic pressure P_e that squeezes the film:

$$P_e = \varepsilon_0 \varepsilon_r \left(\frac{V}{t}\right)^2 = \varepsilon_0 \varepsilon_r E^2$$

where ε_0 is the vacuum permittivity, ε_r is the relative permittivity, and E is the applied electric field.

2. Thickness Strain Assuming small strains and linear elastic behavior, the resulting compressive strain s_z in the thickness direction is:

$$s_z = -\frac{P_e}{Y} = -\frac{\varepsilon_0 \varepsilon_r V^2}{Y t^2}$$

where Y is the Young's Modulus of the elastomer. (The negative sign indicates compression).

Worked Example:

Calculate Strain in a Dielectric Elastomer Actuator An acrylic dielectric elastomer film has a thickness of 50 μm, a relative permittivity of 4.5, and a Young's Modulus of 1.2 MPa. A high-voltage signal of 2500 V is applied across the film. Calculate the electrostatic pressure generated and the resulting thickness strain. Assume $\varepsilon_0 = 8.854 \times 10^{-12}$ F/m.

Step 1: Calculate the electric field Thickness $t = 50 \times 10^{-6}$ m, Voltage $V = 2500$ V

$$E = \frac{V}{t} = \frac{2500}{50 \times 10^{-6}} = 5 \times 10^7 \text{ V/m}$$

Step 2: Calculate the electrostatic Maxwell pressure

$$P_e = \varepsilon_0 \varepsilon_r E^2 = (8.854 \times 10^{-12}) \times 4.5 \times (5 \times 10^7)^2$$

$$P_e = 3.9843 \times 10^{-11} \times 2.5 \times 10^{15} = 99607.5 \text{ Pa} \approx 99.6 \text{ kPa}$$

Step 3: Calculate the thickness strain

$$s_z = -\frac{P_e}{Y} = -\frac{99607.5}{1.2 \times 10^6} \approx -0.083 \text{ or } -8.3\%$$

The elastomer experiences a compressive strain of 8.3% in the thickness direction.

2.5 Magnetostrictive Materials

Methodology:

Magnetostrictive Strain Analysis Magnetostrictive materials change their shape when exposed to a magnetic field.

1. Joule Magnetostriction The strain λ produced by an applied magnetic field H is:

$$\lambda = \frac{\Delta L}{L_0}$$

In the linear operating region, strain is often modeled as proportional to the applied field:

$$\lambda = d_{33}^m \times H$$

where d_{33}^m is the magnetostrictive strain coefficient (m/A) and H is the magnetic field strength (A/m).

2. Blocked Force Generation Similar to thermal or SMA expansion, the blocked force $F_{blocked}$ generated if expansion is fully constrained is:

$$F_{blocked} = E \times A \times \lambda$$

where E is Young's Modulus and A is the cross-sectional area.

Worked Example:

Calculate Expansion and Blocked Force for a Terfenol-D Rod A rod made of Terfenol-D has a length of 100 mm and a diameter of 10 mm. It is subjected to a magnetic field that induces a magnetostrictive strain $\lambda = 1200 \times 10^{-6}$. The Young's Modulus of the material is 30 GPa. Calculate the free expansion of the rod and the maximum blocked force it could exert if completely confined.

Step 1: Calculate the free expansion Original length $L_0 = 100$ mm $= 0.1$ m

$$\Delta L = \lambda \times L_0 = (1200 \times 10^{-6}) \times 0.1 = 120 \times 10^{-6} \text{ m} = 120\ \mu\text{m}$$

Step 2: Calculate the cross-sectional area Diameter $D = 10$ mm $= 0.01$ m

$$A = \frac{\pi D^2}{4} = \frac{\pi \times (0.01)^2}{4} = 7.854 \times 10^{-5} \text{ m}^2$$

Step 3: Calculate the blocked force Young's Modulus $E = 30$ GPa $= 30 \times 10^9$ N/m^2

$$F_{blocked} = E \times A \times \lambda$$

$$F_{blocked} = (30 \times 10^9) \times (7.854 \times 10^{-5}) \times (1200 \times 10^{-6})$$

$$F_{blocked} = 2827.4 \text{ N} \approx 2.827 \text{ kN}$$

CHAPTER 3

Emerging Trends in Electronic Components

3.1 Introduction

This chapter focuses on the computational aspects of modern electronic components such as Surface Mount Devices (SMDs) and miniaturized power sources. We will cover the calculation of SMD values for resistors and capacitors, thermal management for high-density ICs, battery runtime estimation for IoT nodes, and component reliability metrics.

3.2 SMD Resistor Value Calculation

Methodology:

Decoding Standard SMD Resistor Codes Surface Mount Device (SMD) resistors typically use a 3-digit or 4-digit coding system to indicate their resistance value.

1. **Three-Digit Code**: The first two digits represent the significant figures. The third digit represents the multiplier (power of 10).
$$R = (D_1 \times 10 + D_2) \times 10^{D_3}\,\Omega$$
2. **Four-Digit Code**: The first three digits represent the significant figures. The fourth digit represents the multiplier.
$$R = (D_1 \times 100 + D_2 \times 10 + D_3) \times 10^{D_4}\,\Omega$$
3. **Letter 'R'**: Used as a decimal point for values less than $10\,\Omega$ or $100\,\Omega$.

Worked Example:

Calculate the resistance of SMD resistors with codes 473 and 1002 **Step 1: Decode the 3-digit code (473).**

- First digit $(D_1) = 4$
- Second digit $(D_2) = 7$
- Multiplier digit $(D_3) = 3$

$$R = 47 \times 10^3\,\Omega = 47,000\,\Omega = 47\,\text{k}\Omega$$

Step 2: Decode the 4-digit code (1002).

- First digit $(D_1) = 1$
- Second digit $(D_2) = 0$
- Third digit $(D_3) = 0$
- Multiplier digit $(D_4) = 2$

$$R = 100 \times 10^2\,\Omega = 10,000\,\Omega = 10\,\text{k}\Omega$$

Worked Example:

Calculate the resistance for SMD code 4R7 **Step 1: Identify the position of 'R'.** The letter 'R' acts as a decimal point. **Step 2: Substitute and write the value.**

$$R = 4.7\,\Omega$$

3.3 SMD Capacitor Value Decoding

Methodology:

Decoding SMD Ceramic Capacitor Values SMD capacitors often use a 3-digit code similar to resistors but the resulting value is in picofarads (pF).

1. The first two digits are the significant figures.
2. The third digit is the multiplier (number of zeros to add).
3. The final value is given in picofarads (pF).

$$C = (D_1 \times 10 + D_2) \times 10^{D_3}\,\text{pF}$$

4. Convert to nanofarads (nF) or microfarads (μF) by dividing by 10^3 or 10^6 respectively.

Worked Example:

Determine the capacitance for an SMD component marked 104 **Step 1: Extract the significant figures and multiplier.**

- Significant figures = 10
- Multiplier = 4

Step 2: Calculate the base value in pF.

$$C = 10 \times 10^4\,\text{pF} = 100,000\,\text{pF}$$

Step 3: Convert to standard units. To convert to nanofarads (nF), divide by 1000:

$$C = \frac{100,000}{1000}\,\text{nF} = 100\,\text{nF}$$

To convert to microfarads (μF), divide by 1,000,000:

$$C = 0.1\,\mu\text{F}$$

3.4 Thermal Management in Miniaturized Components

As components shrink (e.g. System-on-Chips and high-density LEDs), managing heat dissipation becomes crucial.

Methodology:

Junction Temperature Calculation To ensure a component does not exceed its maximum operating temperature the junction temperature (T_J) is calculated using thermal resistance values.

1. Identify power dissipation (P_D) in Watts.
2. Identify ambient temperature (T_A) in °C.

3. Sum the thermal resistances (θ) from the junction to ambient. Usually expressed in °C/W.

$$\theta_{JA} = \theta_{JC} + \theta_{CS} + \theta_{SA}$$

Where:

- θ_{JC} = Junction-to-Case thermal resistance
- θ_{CS} = Case-to-Sink thermal resistance
- θ_{SA} = Sink-to-Ambient thermal resistance

4. Calculate the Junction Temperature:

$$T_J = T_A + P_D \times \theta_{JA}$$

Worked Example:

Calculate junction temperature of a high-power SMD LED **Problem Statement:** An SMD LED dissipates 3 W of power. The ambient temperature is 30°C. The thermal resistances are: Junction-to-case is 4°C/W Case-to-sink is 1°C/W and Sink-to-ambient is 8°C/W. Calculate the junction temperature and determine if it is safe assuming the maximum allowed T_J is 100°C.

Step 1: Calculate total thermal resistance (θ_{JA}).

$$\theta_{JA} = 4 + 1 + 8 = 13°\text{C/W}$$

Step 2: Apply the junction temperature formula.

$$T_J = T_A + P_D \times \theta_{JA}$$

$$T_J = 30 + 3 \times 13$$

$$T_J = 30 + 39 = 69°\text{C}$$

Step 3: Evaluate safety. Since 69°C is less than the maximum limit of 100°C the thermal design is safe.

3.5 Battery Lifetime Estimation for Low-Power Devices

Emerging trends include low-power IoT devices operating on coin cell batteries.

Methodology:

Calculating Average Current and Battery Life IoT devices spend most of their time in sleep mode and wake up briefly to transmit data.

1. Determine the active current (I_{active}) and time spent active (t_{active}).
2. Determine the sleep current (I_{sleep}) and time spent sleeping (t_{sleep}).
3. Calculate the average current consumption (I_{avg}):

$$I_{avg} = \frac{(I_{active} \times t_{active}) + (I_{sleep} \times t_{sleep})}{t_{active} + t_{sleep}}$$

4. Calculate estimated battery life using battery capacity (C) and a derating factor (typically 0.7 to 0.8 to account for self-discharge and environment):

$$\text{Life (Hours)} = \frac{C\,(\text{mAh})}{I_{avg}\,(\text{mA})} \times \text{Derating Factor}$$

Worked Example:

Calculate battery life of a smart sensor node **Problem Statement:** A sensor node wakes up for 2 seconds to transmit data drawing 20 mA. It then sleeps for 58 seconds drawing 10 μA. The node is powered by a CR2032 coin cell with a 220 mAh capacity. Assume a derating factor of 0.8. Calculate the average current and estimated battery life in days.

Step 1: Standardize units to mA. $I_{active} = 20\,\text{mA}$
$I_{sleep} = 10\,\mu\text{A} = 0.01\,\text{mA}$
$t_{active} = 2\,\text{s}$, $t_{sleep} = 58\,\text{s}$
Total period $T = 60\,\text{s}$.

Step 2: Calculate Average Current (I_{avg}).

$$I_{avg} = \frac{(20 \times 2) + (0.01 \times 58)}{60}$$

$$I_{avg} = \frac{40 + 0.58}{60} = \frac{40.58}{60} = 0.6763\,\text{mA}$$

Step 3: Calculate Battery Life in Hours.

$$\text{Life (Hours)} = \frac{220}{0.6763} \times 0.8$$

$$\text{Life (Hours)} \approx 325.3 \times 0.8 = 260.24\,\text{Hours}$$

Step 4: Convert to Days.

$$\text{Life (Days)} = \frac{260.24}{24} \approx 10.84\,\text{Days}$$

3.6 Component Reliability and MTBF

As electronic systems grow more complex quantifying their reliability is essential.

Methodology:

Calculating Failure Rate and MTBF Mean Time Between Failures (MTBF) is a critical statistical value used to predict component reliability.

1. Calculate the total operating hours (Total Hours = Number of units × Test Duration).
2. Calculate the Failure Rate (λ):

$$\lambda = \frac{\text{Number of Failures}}{\text{Total Operating Hours}}$$

   (Often expressed in failures per million hours or FITs, where $1\,\text{FIT} = 10^{-9}$ failures/hour).

3. Calculate MTBF:

$$\text{MTBF} = \frac{1}{\lambda}$$

Worked Example:

Calculate MTBF and FIT rate for a batch of ICs **Problem Statement:** A manufacturer tests 5000 units of a new microcontroller for 2000 hours. During this test 4 units fail. Calculate the failure rate in FITs and the MTBF in hours.

Step 1: Calculate total operating hours.

$$\text{Total Hours} = 5000\,\text{units} \times 2000\,\text{hours} = 10,000,000\,\text{hours}$$

Step 2: Calculate Failure Rate (λ).

$$\lambda = \frac{4}{10,000,000} = 4 \times 10^{-7}\,\text{failures/hour}$$

Step 3: Convert to FITs.

$$\text{FIT} = \lambda \times 10^9 = 4 \times 10^{-7} \times 10^9 = 400\,\text{FITs}$$

Step 4: Calculate MTBF.

$$\text{MTBF} = \frac{1}{\lambda} = \frac{1}{4 \times 10^{-7}} = 2,500,000\,\text{hours}$$

CHAPTER 4

Advanced Trends in Computing Applications

This chapter focuses on the computational and interfacing aspects of modern computing trends, specifically utilizing platforms like the Raspberry Pi for emerging electronic applications. The core topics include designing interface circuits for general-purpose input/output (GPIO), analog-to-digital conversion for sensor integration, serial data transmission, and system power management.

4.1 LED Interfacing and Current Limiting Resistor Calculation

When interfacing a Light Emitting Diode (LED) with a Raspberry Pi or any modern microcontroller, it is essential to limit the current flowing through the LED to prevent damage to both the LED and the GPIO pin. Raspberry Pi GPIO pins typically operate at 3.3V and can source a maximum of around 16mA per pin safely.

Methodology:

Calculating the Current Limiting Resistor To find the appropriate resistor value for safe LED operation, use Ohm's Law along with the voltage characteristics of the LED.
Step 1: Identify the operating voltage of the GPIO pin (V_{GPIO}). For the Raspberry Pi, $V_{GPIO} = 3.3$ V.
Step 2: Identify the forward voltage drop of the LED (V_f) and the desired operating current (I_d).
Step 3: Calculate the required resistance (R) using the formula:

$$R = \frac{V_{GPIO} - V_f}{I_d}$$

Step 4: Select the nearest standard resistor value that is equal to or slightly greater than the calculated R to ensure safe operation.

Worked Example:

Interfacing a Red LED with Raspberry Pi A Red LED with a forward voltage (V_f) of 1.8V is to be interfaced with a Raspberry Pi GPIO pin. The desired current for optimal brightness without exceeding the pin's safe limit is 10mA. Calculate the exact and practical standard value for the current limiting resistor.
Solution:

1. **Given Data:**
   - GPIO Voltage, $V_{GPIO} = 3.3$ V
   - Forward Voltage, $V_f = 1.8$ V
   - Desired Current, $I_d = 10$ mA $= 0.010$ A
2. **Apply the formula:**

$$R = \frac{V_{GPIO} - V_f}{I_d}$$

$$R = \frac{3.3 - 1.8}{0.010}$$

$$R = \frac{1.5}{0.010} = 150\,\Omega$$

3. **Conclusion:** The exact required resistance is $150\,\Omega$. Since $150\,\Omega$ is a common standard resistor value (e.g., in the E24 series), a $150\,\Omega$ resistor should be connected in series with the LED.

4.2 Interfacing Analog Sensors using ADC

Unlike many standard microcontrollers, the Raspberry Pi does not possess built-in Analog-to-Digital Converters (ADCs). To interface analog sensors (such as analog temperature or humidity sensors), an external ADC chip like the 10-bit MCP3008 is frequently used over the SPI protocol.

Methodology:

ADC Resolution and Voltage Calculation To convert the digital value received from the ADC back into a meaningful analog voltage, you must determine the ADC's step size.

Step 1: Determine the reference voltage (V_{REF}) connected to the ADC and its resolution in bits (n).

Step 2: Calculate the step size (Resolution Voltage, ΔV):

$$\Delta V = \frac{V_{REF}}{2^n}$$

Step 3: Calculate the analog sensor voltage (V_{sensor}) using the digital value (D_{out}) read from the ADC:

$$V_{sensor} = D_{out} \times \Delta V$$

Worked Example:

Reading an LM35 Temperature Sensor via 10-bit ADC An MCP3008 (10-bit ADC) is connected to a Raspberry Pi with a reference voltage of 3.3V. An LM35 temperature sensor is connected to channel 0. If the Raspberry Pi reads a digital value of 210 from the ADC, compute the sensor's output voltage and the corresponding temperature. (Note: The LM35 outputs 10 mV/°C).

Solution:

1. **Given Data:**
   - Reference Voltage, $V_{REF} = 3.3$ V
   - Resolution, $n = 10$ bits (Total steps $= 2^{10} = 1024$)
   - Digital Output, $D_{out} = 210$
   - LM35 scale factor = 10 mV/°C = 0.01 V/°C
2. **Calculate the step size (ΔV):**

$$\Delta V = \frac{3.3}{1024} \approx 0.0032227 \text{ V (or 3.223 mV)}$$

3. **Calculate the sensor voltage (V_{sensor}):**

$$V_{sensor} = 210 \times 0.0032227 \approx 0.6768 \text{ V} = 676.8 \text{ mV}$$

4. **Calculate the temperature (T):**

$$T = \frac{V_{sensor}}{\text{Scale Factor}} = \frac{0.6768}{0.01} = 67.68°\text{C}$$

5. **Conclusion:** The sensor output voltage is approximately 0.6768 V, which corresponds to an environmental temperature of 67.68°C.

4.3 Temperature Measurement using an NTC Thermistor

Thermistors are non-linear temperature sensors widely used in modern IoT environmental systems. To accurately compute the temperature from a Negative Temperature Coefficient (NTC) thermistor, the Beta (β) parameter equation is utilized.

Methodology:

Thermistor Resistance and Temperature Conversion The thermistor is typically placed in a voltage divider circuit with a known series resistor (R_s). Assuming the thermistor is connected between the supply voltage (V_{in}) and the ADC input, and the series resistor (R_s) is connected between the ADC input and Ground:
Step 1: Read the analog voltage drop across the series resistor (V_{out}) using an ADC.
Step 2: Calculate the current resistance of the thermistor (R_T) using the voltage divider formula:

$$R_T = R_s \times \left(\frac{V_{in}}{V_{out}} - 1 \right)$$

Step 3: Calculate the temperature in Kelvin (T) using the Beta equation:

$$T = \frac{1}{\frac{1}{T_0} + \frac{1}{\beta} \ln\left(\frac{R_T}{R_0}\right)}$$

Where T_0 is the nominal temperature (usually 298.15 K, equivalent to 25°C), R_0 is the nominal resistance at T_0, and β is the material constant.
Step 4: Convert the temperature to Celsius:

$$T_{Celsius} = T - 273.15$$

Worked Example:

Calculating Temperature from NTC Thermistor Data An NTC thermistor has a nominal resistance of 10 kΩ at 25°C (298.15 K) and a β value of 3950 K. It is used in a voltage divider setup with a 10 kΩ series resistor and a 3.3 V supply. If the voltage measured across the series resistor by the ADC is 1.65 V, calculate the temperature of the environment.
Solution:

1. **Given Data:**
   - Supply Voltage, $V_{in} = 3.3$ V
   - Measured Voltage, $V_{out} = 1.65$ V
   - Series Resistor, $R_s = 10$ kΩ $= 10000\,\Omega$
   - Nominal Resistance, $R_0 = 10000\,\Omega$
   - Nominal Temperature, $T_0 = 298.15$ K
   - Beta constant, $\beta = 3950$ K
2. **Calculate Thermistor Resistance (R_T):**

$$R_T = 10000 \times \left(\frac{3.3}{1.65} - 1 \right) = 10000 \times (2 - 1) = 10000\,\Omega$$

3. **Calculate Temperature in Kelvin (T):**

$$T = \frac{1}{\frac{1}{298.15} + \frac{1}{3950} \ln\left(\frac{10000}{10000}\right)}$$

   Since $\ln(1) = 0$:

$$T = \frac{1}{\frac{1}{298.15} + 0} = 298.15 \text{ K}$$

4. **Convert to Celsius:**

$$T_{Celsius} = 298.15 - 273.15 = 25°\text{C}$$

5. **Conclusion:** Because the measured resistance exactly matches the nominal resistance (10 kΩ), the ambient temperature is at the nominal temperature of 25°C.

4.4 Serial Data Transmission Timing for Peripheral Interfacing

When interfacing modern modules (such as GPS, GSM, or Bluetooth) with a Raspberry Pi via UART (Universal Asynchronous Receiver-Transmitter), calculating the data transmission time is critical for preventing software buffer overflows and managing system latency.

Methodology:

Calculating UART Transmission Time Data transmitted via UART is framed with start bits, stop bits, and optional parity bits around the payload data bits.
Step 1: Determine the total number of bits per frame:

$$\text{Bits per frame} = \text{Start Bit} + \text{Data Bits} + \text{Parity Bit (if any)} + \text{Stop Bit(s)}$$

Step 2: Calculate the time required to transmit a single bit (T_{bit}) based on the configured Baud Rate:

$$T_{bit} = \frac{1}{\text{Baud Rate}}$$

Step 3: Calculate the transmission time for one single frame (T_{frame}):

$$T_{frame} = T_{bit} \times \text{Bits per frame}$$

Step 4: Calculate the total transmission time (T_{total}) for a given message size (measured in bytes or frames):

$$T_{total} = T_{frame} \times \text{Total Frames}$$

Worked Example:

Estimating Transmission Time for GPS NMEA Strings A GPS module sends a 50-byte NMEA data string to a Raspberry Pi over UART. The communication protocol is configured at a baud rate of 9600 bps, 8 data bits, no parity, and 1 stop bit. Calculate the time required to transmit the complete 50-byte string.
Solution:

1. **Determine Bits per frame:**
   - Start Bit = 1
   - Data Bits = 8
   - Parity Bit = 0
   - Stop Bit = 1

$$\text{Bits per frame} = 1 + 8 + 0 + 1 = 10 \text{ bits/frame}$$

2. **Calculate the time for one bit (T_{bit}):**

$$T_{bit} = \frac{1}{9600} \approx 0.00010417 \text{ seconds (or } 104.17\,\mu\text{s)}$$

3. **Calculate the time for one frame (T_{frame}):**

$$T_{frame} = 104.17\,\mu\text{s} \times 10 = 1041.7\,\mu\text{s} \approx 1.042 \text{ ms}$$

4. **Calculate Total Transmission Time (T_{total}):** Since each character is 1 byte (1 frame), the total number of frames is 50.

$$T_{total} = 1.042 \text{ ms} \times 50 = 52.1 \text{ ms}$$

5. **Conclusion:** It will take approximately 52.1 ms for the entire 50-byte NMEA string to be completely transmitted to the Raspberry Pi over the UART interface.

4.5 Power Budgeting and Battery Life Estimation for Remote Nodes

With the rise of edge computing, deploying Raspberry Pi devices (such as the Raspberry Pi Zero or Pico) in remote locations using battery power requires accurate power budgeting to estimate operational lifespan.

Methodology:

Estimating Battery Life for Duty-Cycled Systems Remote systems typically operate in duty cycles, waking up to process and transmit data (Active State) and then returning to a low-power mode (Sleep State) to conserve energy.

Step 1: Identify the current consumption in the active state (I_{active}) and the duration of the active state (T_{active}).

Step 2: Identify the current consumption in the sleep state (I_{sleep}) and the duration of the sleep state (T_{sleep}).

Step 3: Calculate the Average Current (I_{avg}) consumed over one complete duty cycle:

$$I_{avg} = \frac{(I_{active} \times T_{active}) + (I_{sleep} \times T_{sleep})}{T_{active} + T_{sleep}}$$

Step 4: Calculate the estimated battery life in hours considering efficiency:

$$\text{Battery Life (hours)} = \frac{\text{Battery Capacity (mAh)}}{I_{avg} \text{ (mA)}} \times \eta$$

Where η is the battery efficiency factor (typically 0.7 to 0.85 depending on chemistry, aging, and temperature).

Worked Example:

Estimating Battery Life of a Raspberry Pi Pico Sensor Node A Raspberry Pi Pico is deployed as a remote environmental sensor node powered by a 2000 mAh Li-Po battery. The device wakes up for 2 seconds to read sensors and transmit data, drawing 45 mA. It then sleeps for 58 seconds, drawing only 1.5 mA. Assuming a battery efficiency of 80% ($\eta = 0.8$), estimate the battery life in days.

Solution:

1. **Given Data:**
   - $I_{active} = 45$ mA
   - $T_{active} = 2$ s
   - $I_{sleep} = 1.5$ mA
   - $T_{sleep} = 58$ s
   - Battery Capacity = 2000 mAh
   - Efficiency $\eta = 0.8$
2. **Calculate Average Current (I_{avg}):** The total cycle time is $T_{active} + T_{sleep} = 2 + 58 = 60$ s.

$$I_{avg} = \frac{(45 \times 2) + (1.5 \times 58)}{60}$$

$$I_{avg} = \frac{90 + 87}{60} = \frac{177}{60} = 2.95 \text{ mA}$$

3. **Calculate Total Battery Life in Hours:**

$$\text{Life}_{hours} = \frac{2000}{2.95} \times 0.8 \approx 677.96 \times 0.8 \approx 542.37 \text{ hours}$$

4. **Convert Battery Life to Days:**

$$\text{Life}_{days} = \frac{542.37}{24} \approx 22.6 \text{ days}$$

5. **Conclusion:** The sensor node is estimated to operate for approximately 22.6 days on a single charge before requiring a battery replacement or recharge.

4.6 Logic Implementation for Burglar Alarm Systems

An essential application of Raspberry Pi GPIO is developing automation and security systems, such as a Burglar Alarm utilizing a Passive Infrared (PIR) motion sensor and a buzzer. While primarily a logic problem, calculating polling delays and managing software states systematically is vital for maintaining low CPU utilization.

Methodology:

Algorithm for PIR-based Alarm System The logic algorithm dictates how the Raspberry Pi will poll the PIR sensor and trigger the output device (such as a Buzzer or LED).
Step 1: Configure the GPIO pin connected to the PIR sensor as a digital `INPUT`.
Step 2: Configure the GPIO pin connected to the Buzzer as a digital `OUTPUT`.
Step 3: In a main loop, continuously read the digital state of the PIR sensor.
Step 4: Apply logic conditions based on the read state:

- If the PIR state is `HIGH` (Motion detected):
  1. Set the Buzzer GPIO to `HIGH`.
  2. Apply a blocking delay (t_{alarm}) for the desired alarm sounding duration.
  3. Set the Buzzer GPIO to `LOW` to reset the alarm.
- If the PIR state is `LOW` (No motion):
  1. Ensure the Buzzer GPIO remains `LOW`.

Step 5: Implement a short loop delay (t_{poll}) at the end of the main loop to prevent excessive CPU polling overhead.

Worked Example:

Determining Polling Overhead CPU Time A Raspberry Pi runs a Python script for a PIR alarm system. The main loop takes 0.5 ms to execute the read and logic check instructions. To save CPU cycles, a polling delay (t_{poll}) of 100 ms is added at the end of each loop iteration. Calculate the percentage of time the CPU spends executing the alarm loop versus resting in the delay.
Solution:

1. **Given Data:**
   - Execution Time (T_{exec}) = 0.5 ms
   - Polling Delay (T_{poll}) = 100 ms

2. **Calculate Total Cycle Time (T_{cycle}):**

$$T_{cycle} = T_{exec} + T_{poll} = 0.5 + 100 = 100.5 \text{ ms}$$

3. **Calculate CPU Execution Percentage:**

$$\text{Execution } \% = \left(\frac{T_{exec}}{T_{cycle}}\right) \times 100$$

$$\text{Execution } \% = \left(\frac{0.5}{100.5}\right) \times 100 \approx 0.4975\%$$

4. **Calculate Idle/Rest Percentage:**

$$\text{Rest } \% = 100\% - 0.4975\% = 99.5025\%$$

5. **Conclusion:** By introducing a 100 ms polling delay, the CPU only spends approximately 0.5% of its time checking the PIR sensor. This frees up significant resources for other computational background tasks while keeping the system highly responsive to motion events.

Formulae

Appendix: Key Formulae

This appendix provides a quick reference to the key symbolic formulae and mathematical relationships discussed in this book.

Unit 1: Energy Sources & Power

- **Wind Power (Total):**

  $$P_{wind} = \frac{1}{2}\rho A v^3$$

  Where ρ is air density, A is swept area (πr^2), and v is wind velocity.

- **Wind Turbine Output Power:**

  $$P_{out} = P_{wind} \times C_p \times \eta_g = \frac{1}{2}\rho A v^3 \times C_p \times \eta_g$$

  Where C_p is the coefficient of performance and η_g is generator efficiency.

- **Solar Power Input & Output:**

  $$P_{in} = G \times A, \quad P_{out} = P_{in} \times \eta = G \times A \times \eta$$

  Where G is solar irradiance, A is area, and η is efficiency.

- **Solar Daily Energy:**

  $$E_{daily} = P_{out} \times \text{Peak Sun Hours (PSH)}$$

- **Hydroelectric Power:**

  $$P = \rho \times g \times Q \times H \times \eta$$

  Where ρ is water density, g is acceleration due to gravity, Q is flow rate, H is head, and η is efficiency.

- **Capacity Factor (CF) & Availability Factor (AF):**

  $$\text{CF} = \frac{\text{Actual Energy Produced}}{\text{Rated Capacity} \times \text{Total Hours}}, \quad \text{AF} = \frac{\text{Hours Available}}{\text{Total Hours}} \times 100\%$$

- **Energy & Power Relationship:**

  $$E = P \times t$$

Unit 2: Sensors & Actuators

- **Electrostatic Pressure (P_e) and Strain (s_z):**

  $$P_e = \varepsilon_0 \varepsilon_r \left(\frac{V}{t}\right)^2 = \varepsilon_0 \varepsilon_r E^2$$

  $$s_z = -\frac{P_e}{Y} = -\frac{\varepsilon_0 \varepsilon_r V^2}{Y t^2}$$

  Where $\varepsilon_0 \varepsilon_r$ is permittivity, E is electric field (V/t), and Y is Young's modulus.

- **Piezoelectric Equations:**

$$\Delta t = d_{33} \times V, \quad Q = d_{33} \times F$$

Where d_{33} is the piezoelectric coefficient, V is voltage, Q is charge, and F is force.

- **Capacitance (C) and Voltage (V):**

$$C = \frac{\varepsilon_0 \varepsilon_r A}{t}, \quad V = \frac{Q}{C}$$

- **Magnetostrictive Strain (λ):**

$$\lambda = d_{33}^m \times H = \frac{\Delta L}{L_0}$$

Where d_{33}^m is the piezomagnetic coefficient and H is the magnetic field.

- **Mechanical Stress (σ_r) & Blocked Force ($F_{blocked}$):**

$$\sigma_r = Y \times \lambda, \quad F_{blocked} = Y \times A \times \lambda$$

(Note: Y or E represents Young's modulus).

- **Thermoelectric Seebeck Voltage (V_{oc}) & Figure of Merit (ZT):**

$$V_{oc} = S \times \Delta T$$

$$ZT = \frac{S^2 \sigma}{k} T$$

Where S is the Seebeck coefficient, σ is electrical conductivity, and k is thermal conductivity.

- **Maximum Power Transfer:**

$$P_{max} = \frac{V_{oc}^2}{4R_{in}}$$

Unit 3: Components & Reliability

- **Average Current (I_{avg}) & Battery Life:**

$$I_{avg} = \frac{(I_{active} \times t_{active}) + (I_{sleep} \times t_{sleep})}{t_{active} + t_{sleep}}$$

$$\text{Life (Hours)} = \frac{\text{Battery Capacity (mAh)}}{I_{avg}\,\text{(mA)}} \times \text{Derating Factor}$$

- **Failure Rate (λ), MTBF, and FIT:**

$$\lambda = \frac{\text{Number of Failures}}{\text{Total Operating Hours}}$$

$$\text{MTBF} = \frac{1}{\lambda}, \quad \text{FIT} = \lambda \times 10^9$$

- **Thermal Management (T_J and θ_{JA}):**

$$\theta_{JA} = \theta_{JC} + \theta_{CS} + \theta_{SA}$$

$$T_J = T_A + P_D \times \theta_{JA}$$

Where T_J is junction temperature, T_A is ambient temperature, P_D is power dissipated, and θ values are thermal resistances.

- **Component Values (Resistors & Capacitors):**

$$R = (D_1 \times 10 + D_2) \times 10^{D_3}\,\Omega \quad \text{or} \quad R = (D_1 \times 100 + D_2 \times 10 + D_3) \times 10^{D_4}\,\Omega$$

$$C = (D_1 \times 10 + D_2) \times 10^{D_3}\,\text{pF}$$

Unit 4: Interfacing & Communication

- **ADC Resolution and Sensor Voltage:**

$$\Delta V = \frac{V_{REF}}{2^n}$$

$$V_{sensor} = D_{out} \times \Delta V$$

Where ΔV is the step size, V_{REF} is reference voltage, n is ADC bits, and D_{out} is the digital value.

- **Thermistor Resistance & Temperature (Steinhart-Hart):**

$$R_T = R_s \times \left(\frac{V_{in}}{V_{out}} - 1 \right)$$

$$T = \frac{1}{\frac{1}{T_0} + \frac{1}{\beta} \ln \left(\frac{R_T}{R_0} \right)}$$

Where T is in Kelvin. Conversion to Celsius: $T_{Celsius} = T - 273.15$.

- **LED Current Limiting Resistor:**

$$R = \frac{V_{GPIO} - V_f}{I_d}$$

Where V_{GPIO} is output voltage, V_f is LED forward voltage, and I_d is desired current.

- **Serial Communication Timing:**

$$\text{Bits per frame} = \text{Start} + \text{Data} + \text{Parity} + \text{Stop Bit(s)}$$

$$T_{bit} = \frac{1}{\text{Baud Rate}}, \quad T_{frame} = T_{bit} \times \text{Bits per frame}$$

$$T_{total} = T_{frame} \times \text{Total Frames}$$

- **Execution Metrics:**

$$T_{cycle} = T_{exec} + T_{poll}$$

$$\text{Execution \%} = \left(\frac{T_{exec}}{T_{cycle}} \right) \times 100\%$$