Automation Technologies

Dr Bikash Sah

Universität Siegen / IAS Interconnected Automation Systems

Content

1. An initial overview of of the course
2. Sensor Technologies
3. Signal Processing and Conditioning
4. Processor Technologies in Automation Systems
5. Industrial Communication in Automation Systems

Table of contents

Power Electronics Devices and Components

Dr Bikash Sah

Universität Siegen

IAS Interconnected Automation Systems

Course Overview

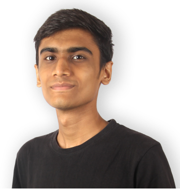
The course will be in 6 modules.

- Module I: Introduction to Automation Technologies.
- Module II: Sensor Technologies.
- Module III: Signal Conditioning.
- Module IV: Processors in Automation Systems.
- Module V: Controllers and Communication.
- Module VI: Actuators and Motion Systems.
- Module VII: Testing and Validation.
- Module VIII: Industrial Case Studies.

Pattern of class:
Day: Tuesday, every week
Time: 8 am to 10 am (theory) and 10 am to 12 am (tutorial) (ideally but it can change)
Holidays: 23 Dec 2025, 30 Dec 2025, and 6 Jan 2026.

The teaching team

Bikash Sah



Tatsat Baldaniya

Pawel Malicki

Contact

- Email: see chair's homepage
- Offices: H-A building, 4th floor
- Individual appointments on request (remote or personally)
- Multiple relevant courses are offered by the Chair. Check link!

Necessary Prior Knowledge for this Course

Students are expected to have a basic understanding of:

- **Basic physics and laws governing systems:** for example, Newton's law of motion, electricity, dimensional consistency, etc.
- **Mathematical background:** algebra, complex numbers (phasors), basic calculus and elementary differential equations.
- **Basic signal theory:** Fourier series, Laplace transform, transfer functions, step/impulse response.
- **General systems thinking:** reading block diagrams, input-output viewpoints, feedback/feedforward concepts.
- *No advanced programming is required—any code used will be introduced as needed and tied to real automation tasks.*

Topics not covered (addressed in other courses):

- Machine learning in automation.
- Advanced control theory (e.g., optimal, robust, nonlinear control), robotics, etc.

Recommended reading

- Weyrich, Michael. Industrial Automation and Information Technology. Berlin, Germany: Springer, 2024.
- Daniel E.. Kandray. Programmable Automation Technologies: An Introduction to CNC, Robotics and PLCs. Industrial Press, 2010.
- Hering, Ekbert, and Gert Schönfelder. Sensors in Science and Technology. Springer Fachmedien Wiesbaden, 2022.
- Woods, Roger, John McAllister, Gaye Lightbody, and Ying Yi. FPGA-based implementation of signal processing systems. John Wiley & Sons, 2008.
- Bartelt, Terry LM. Industrial automated systems: instrumentation and motion control. Delmar Learning, 2010.

Module I: Introduction to Automation Technologies

Submodule: Introduction

What is *Automation*?

Working definition for this course

Automation is the purposeful application of mechanical, electrical, and computer technologies to reduce human involvement in task performance-without necessarily removing humans from the loop.

Highly interdisciplinary, it encompasses: combine elements of sensor technology, actuator technology, control technology, communication technology, real-time software, and data science, among other things.

Key nuances

- **Displace vs. replace:** automation often reallocates human effort to higher-value tasks rather than eliminating it.
- **Hardware and software co-design:** improvement may be purely software (e.g., CAD workflows) or mechatronic (robotic cells).
- **Beyond factories:** retail checkout, medical devices, energy systems, mobility-all are automation domains.

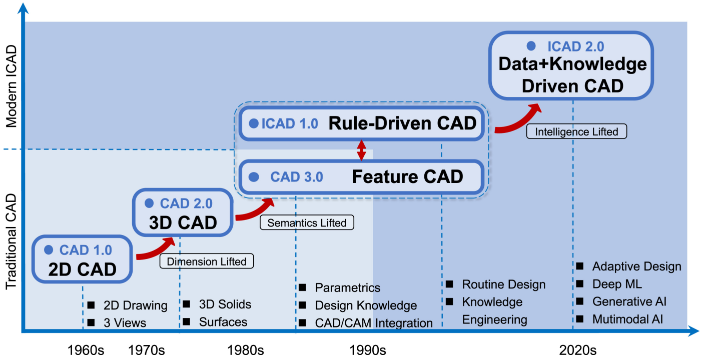
Examples of Automation

- *Self-checkout:* barcode + payment automation reduces repetitive cashier tasks while keeping supervision.

Self checkout machines (Source: from Self checkout using NCR Fastlane machines, CC BY 2.0)

Examples of Automation

- *Computer Aided Design (CAD) evolution:* from manual drafting to 3D solid modeling, numerical control code, and auto-documentation.



Historical development of CAD (Source: Zou, Qiang, Yingcai Wu, Zhenyu Liu, Weiwei Xu, and Shuming Gao. ”Intelligent CAD 2.0.” Visual Informatics 8, no. 4 (2024): 1-12.)

Why Now? Value, Impact, and Trends

What automation accomplishes today

- Enables **flexible value networks**: rapid changeovers, mass customization, and supply-chain visibility.
- Integrates **heterogeneous technologies** across disciplines; links physical assets to **digital representations**.
- Improves **quality, safety, and energy efficiency**; turns data into decisions.

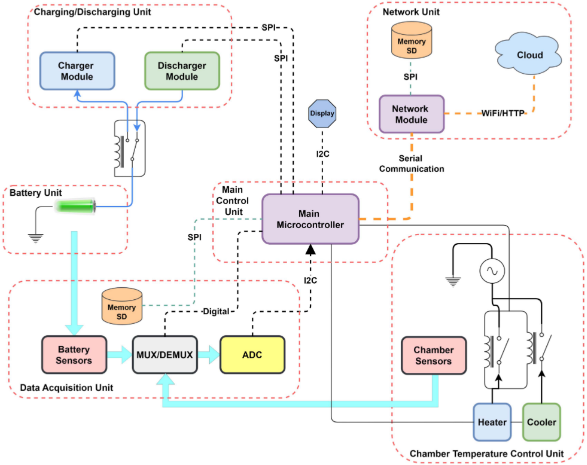
Where it is going

- From real-time control of single machines to **orchestration** of whole cells/lines and ecosystems.
- **Autonomous capabilities**: perception, context understanding, and bounded autonomy with human oversight.
- **Sustainability and circularity**: traceability, footprint accounting, and lifecycle optimization.

Why Now? Value, Impact, and Trends

Practical life example

Battery cycler cell: recipe engine (e.g., CC-CV-rest), interlocks, data historian, remote diagnostics, and energy-aware scheduling.



Automated cell test set-up (Source: Mulpuri et. al.,IEEE Trans. on Industry Appl., vol. 61-5)

Nature and Scope of Automation

Automation can occur anywhere humans perform structured tasks.

- Includes entertainment (remote controls), offices (data entry), transportation, and healthcare.
- Combines mechanical, electrical, and computer elements-modern systems are hybrid and interlinked.
- Machines themselves can contain smaller automated subsystems, leading to hierarchical automation.

Essence: Automation evolves continually, driven by technology that reduces human effort, improves precision, and enhances reliability across diverse applications.

Automation Technology Today and Tomorrow

Current accomplishments : as per the Society for Measurement and Automation Technology within the Association of German Engineers (VDI/VDE) outlook[1]:

- Enables **flexible value networks** and digital connectivity.
- Makes technology more accessible and user-friendly.
- Integrates mechanical, electrical, and information technologies.
- Connects **real-world physical elements** with their **digital representations**.

Emerging directions

- Intelligent, networked products capable of autonomous action.
- Sustainability and circular economy supported by digital transparency.
- Need for clear standards, security, and qualified human oversight.

[1] Adamczyk et. al., Automation 2025—Hypotheses and fields of action. (in German) VDI/VDE Society for Measurement and Control, 2015

Self-Perception of Automation Technology

A multidisciplinary field:

- Integrates **mechanical, electrical, computer, and information engineering**.
- Engineers act as **system integrators**-coordinating diverse components into a coherent whole.

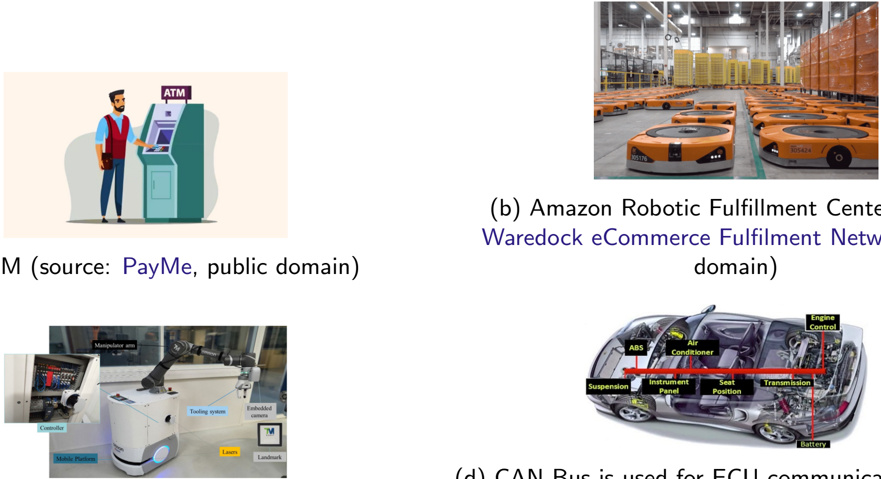
Historical evolution:

- From mechanical aids to networked, software-defined systems.
- Increased productivity and flexibility through information technology.

Future perspective:

- Toward **autonomous systems** that perceive, decide, and act.
- Humans remain central-for supervision, creativity, and ethical responsibility.

Examples of automation technologies (2)



(a) ATM (source: PayMe, public domain)

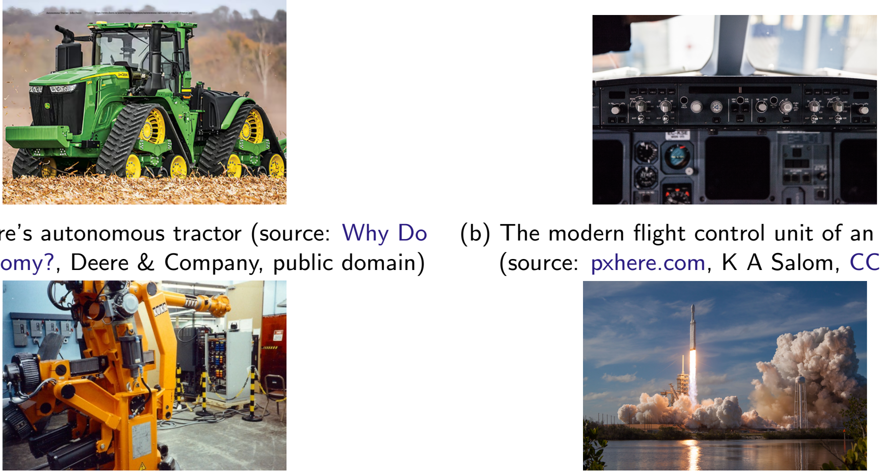
(b) Amazon Robotic Fulfillment Center (source: Waredock eCommerce Fulfilment Network, public domain)

(c) PLC Controller used in Mobile Robots (source: Ghodsian et. al.)

(d) CAN Bus is used for ECU communication in cars (source: Wikimedia Commons, Stephen St. Michael, Introduction to CAN (Controller Area Network))

Examples of automation technologies in various domains

Examples of automation technologies (1)



(a) John Deere's autonomous tractor (source: Why Do I Need Autonomy?, Deere & Company, public domain)

(b) The modern flight control unit of an Airbus A340 (source: pxhere.com, K A Salom, CC BY 2.0)

(c) Factory robots (source: Wikimedia Commons, A. Reinhold, CC BY-SA 4.0)

(d) SpaceX Automated Guidance System (source: Wikimedia Commons, CC0)

Examples of automation technologies in various domains

Automation - example case of manufacturing systems

Automation in manufacturing can appear in:

- **Manufacturing support systems** - e.g., Computer aided design (CAD), computer aided manufacturing (CAM), computer-integrated manufacturing (CIM) *, and material requirement planning (MRP).
- **Manufacturing systems** - the actual production and material-handling equipment.

*****Computer-integrated manufacturing (CIM)** links both levels by connecting support systems directly with factory automation for a fully integrated enterprise.

Focus of this section:

Automation that directly reduces the level of human participation in manufacturing processes.

Three principal types of automation:

1. Fixed automation
2. Programmable automation
3. Flexible automation

Fixed Automation

Definition: Equipment designed to perform a specific sequence of operations for one product or family of products.

Key characteristics

- High **production rates**, low to high product complexity.
- Very limited product variety; reconfiguration requires major setup time.
- High initial cost but low unit cost at large volumes.
- Typical of flow-line and assembly systems.

Examples:

- Transfer lines for engine blocks.
- Automated bottling or packaging lines.
- Semiconductor assembly lines.

Technology note: Modern fixed systems may embed PLCs or computer control for limited adaptability, but mechanical configuration remains largely fixed.

Programmable Automation

Definition: Automation that allows the operating sequence or equipment configuration to be changed through program modification.

Key characteristics

- Handles **moderate product variety** and **medium production volumes**.
- Reprogramming enables new tasks, but setup/changeover time can be long.
- Combines mechanical flexibility with electronic control.
- Core element of almost all modern automated systems.

Enabling technologies:

- **CNC (Computer Numerical Control)** - programmable motion and tool paths.
- **Robotic technology** - reprogrammable manipulators for various tasks.
- **PLC (Programmable Logic Control)** - discrete logic sequencing and interlocks.

Used in: Machining, welding, assembly, and process industries.

Flexible Automation

Definition: An extension of programmable automation with **no lost time for changeover** between products.

Key characteristics

- Capable of producing different parts in any order with little or no manual intervention.
- Combines features of fixed and programmable automation.
- Uses integrated computer control, sensors, and communication networks to recognize and adjust to part variations automatically.
- Supports **high product variety** and **moderate to high production rates**.

Examples:

- Flexible manufacturing systems (FMS).
- CNC machining cells with automatic tool changers and pallet systems.
- Robotized assembly lines with vision feedback.

Comparison of Automation Types

Type	Product Complexity	Product Variety	Production Volume	System Type
Fixed	Low–High	None	High	Flow-line
Programmable	High	Hard	Moderate	All types
Flexible	Low	Hard	High	Flow-line

Observations:

- Boundaries between categories are **not rigid**.
- Modern plants often combine all three types.
- **Programmable automation** has become the *core enabler* of flexibility across manufacturing domains.

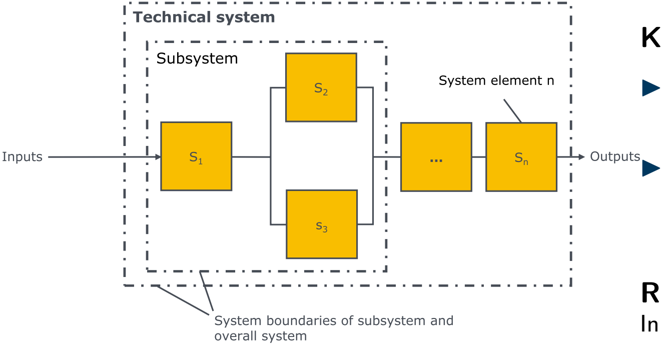
Key Takeaways

- Automation types differ primarily in **variety**, **volume**, and **changeover capability**.
- **Fixed** → highest rate, least flexibility.
  Programmable → moderate rate, adaptable via reprogramming.
  Flexible → high variety, minimal setup delay.
- Programmable and flexible automation underpin **modern Industry 4.0** and **reconfigurable manufacturing**.
- Engineers must select the automation type based on economic trade-offs between productivity, product mix, and lifecycle adaptability.

Technical System — Definition & Structure

Definition

A **technical system** is characterized by its **inputs**, **outputs**, **function**, and **structure**. It comprises several **subsystems** arranged hierarchically and interacting with each other.



System and subsystem of a technical system (source: Weyrich, M., 2024. Industrial Automation and Information Technology. Berlin, Germany, Springer.)

Key points

- Each subsystem has a specific function (mapping of input → output).
- Abstraction helps describe and model complex installations independent of their concrete hardware.

Reading the diagram

Inputs are processed through subsystems $S_1, \ldots, S_n$ within defined system boundaries to produce outputs. Subsystems can themselves contain elements and control logic.

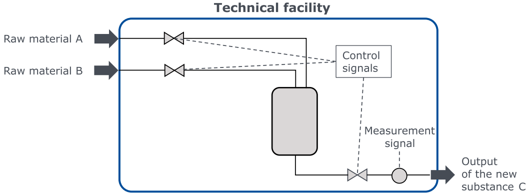
Technical Facility — Concrete Realization

Definition

A **technical facility** is a concrete collection of equipment, devices, and machinery that work together to achieve a purpose. It is a physical implementation of one or more technical systems.

Example (process plant excerpt)

- Raw materials A and B; valves, piping, reactor; sensors and actuators.
- **Control signals** command actuators; **measurement signals** report states.
- Output is a new substance/product C.



Example of a technical facility (adapted from Lauber et. al.)

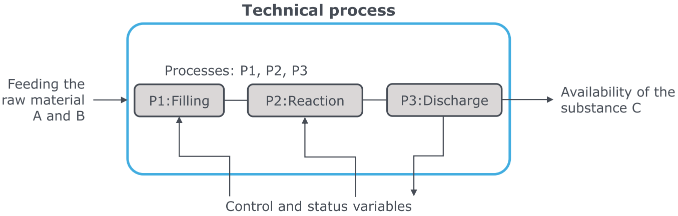
Technical Process — Functional Abstraction

Definition (DIN IEC 60050-351 / VDI Glossary)

A **technical process** is the complete set of interacting operations by which **matter, energy, or information** is transformed, transported, or stored.

Why abstract?

- Real-time information processing and distributed integration require a **function-oriented** view.
- Focus on **subprocesses**, their interrelations, and **command/state variables**.


Technical process
Processes: P1, P2, P3
Feeding the raw material A and B
P1:Filling
P2:Reaction
P3:Discharge
Availability of the substance C
Control and status variables

Example sequence

P1: Filling P2: Reaction P3: Discharge

Inputs: inflow A & B Output: product C
Measured/commanded: valve openings, flow rates, temperatures, levels.

Relating the Terms in Practice

From concrete to abstract and back

1. **Technical system** — abstract structure of subsystems and functions (what maps to what).
2. **Technical facility** — physical realization (hardware: valves, pumps, reactors, sensors).
3. **Technical process** — functional chain of operations (filling → reaction → discharge) with **inputs/outputs** and **state & command variables**.

Signal viewpoints

- *Inputs/Outputs*: material/energy/information entering or leaving the process.
- *Command variables*: actuator setpoints (e.g., valve opening).
- *State variables*: measured quantities (e.g., flow rate, temperature, level).

Takeaway

Use *system* to reason about structure, *facility* to reason about hardware, and *process* to reason about operations and signals—then connect all three with software and communication to realize automation.

What is an *Automaton*? (DIN IEC 60050-351)

Definition (concise)

A self-acting artificial system whose behaviour is governed by given decision rules (discretely or continuously). Its output variables are generated from its input and state variables.

Implications

- Autonomy arises from **rules/relationships** linking inputs, states, and outputs.
- Can operate in discrete-event fashion (sequencing) or continuous-time regulation.
- An **automaton** is the basic building block of automation technology.

From devices to systems

- Modern automation focuses on **orchestrating subsystems** that cooperate to perform tasks.
- Interest shifts from isolated functions to **how functions are connected**.

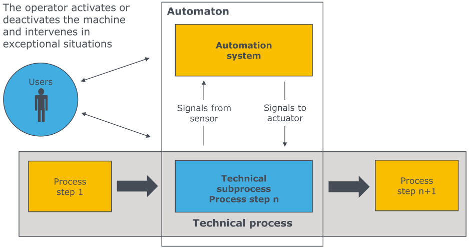
Automaton = Technical Process + Automation System + User

Constituent parts

- **Technical process**: the physical/chemical/information transformation (process step n and neighbours).
- **Automation system**: sensors, controllers, actuators implementing the decision rules.
- **User (human operator)**: activates/deactivates, monitors, and intervenes in exceptions.

Signal view

- *Signals from sensors* → internal state/measurement.
- *Signals to actuators* → commands to influence the process.



The operator activates or deactivates the machine and intervenes in exceptional situations
Automaton
Automation system
Users
Signals from sensor
Signals to actuator
Process step 1
Technical subprocess Process step n
Technical process
Process step n+1

Technical system with automation and human in the loop (source: Weyrich, M., 2024. Industrial Automation and Information Technology. Berlin, Germany: Springer.)

Role of the human

A special **monitoring** function remains; the user decides **when/where** to employ the automaton within a defined context.

Degree of Automation and Taylorian Background

Degree of automation (DIN IEC 60050-351)

Proportion of automatic functions relative to the entire set of functions of a system or facility.

Taylorism and early industrial automation

- Production decomposed into **detailed tasks** via labour studies and management.
- Enables direct comparison of human vs machine for **time, quality, and cost**.
- Example: **welding robot** executes a specified seam; quality becomes more reproducible, but setup/programming costs arise.

Limits: Taylor's "vicious circle"

- Increasingly specialized subprocesses ⇒ complex dependencies and inflexibility.
- Changing functions requires significant **engineering effort**.

Towards Flexible Automata

Raising applicability without full reprogramming

- Add **sensing** to compensate disturbances (e.g., thermal distortion during welding).
- Use feedback to adapt within the **dedicated function** while avoiding time-consuming reprogramming.

Modern direction

- Combine **software** with **standardized & modular systems** to increase **flexibility and applicability**.
- Move from isolated automata to **integrated automated systems** with coordinated functions.

Design takeaway

- Specify the *context of use*, the *degree of automation*, and the *intervention rules* for the human.
- Architect sensor-decision-actuation loops that allow **bounded autonomy** and graceful user override.

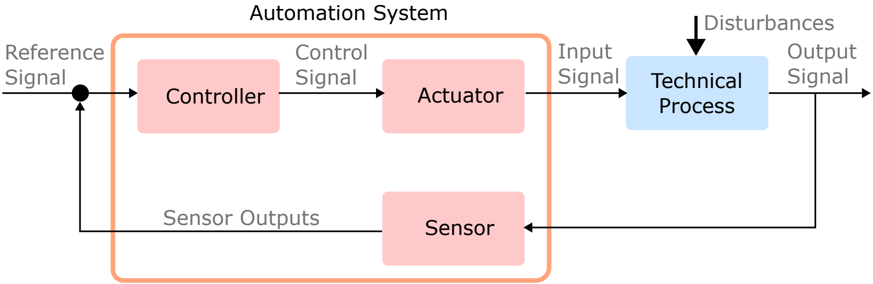
Summary for Practice

- An **automaton** = automation system + technical process, supervised by a **user**.
- **Degree of automation** quantifies how much is automatic; it varies by task and context.
- Historical Taylorism enabled productivity but also rigidity; modern systems target **flexibility** through sensing, software, and modularity.
- Effective automation design balances **autonomy** with **human oversight** and clear exception handling.

Fundamental Components of Automation Systems

Core concept: Automation systems acquire information from the technical process, process it, and influence the process again—closing the feedback loop.

- Three essential elements to form a **closed-loop system**:
  1. **Sensor** — measures physical variables from the process.
  2. **Controller** — processes measurement data and generates control commands.
  3. **Actuator** — converts control signals into physical action on the process.



Functional blocks in a feedback control loop.

The Feedback Control Loop

Operating principle

- **Reference signal:** desired setpoint from operator or automation program.
- **Controller:** computes the control signal based on the error (difference between reference and actual output).
- **Actuator:** applies control effort (force, torque, motion, etc.) to the process.
- **Sensor:** measures actual output and feeds it back to the controller.
- **Feedback:** compensates for **disturbances** to maintain system stability and accuracy.

Modern trends

- Physical proximity of components no longer mandatory—virtualized and networked control is possible via industrial communication.
- Data-driven feedback enables adaptive and remote control architectures.

Actuators: Converting Signals into Action

Definition: Actuators transform control signals into mechanical or other physical process variables.

Forms of actuation

- **Electromechanical:** DC, stepper, and servo motors; often coupled with gears and sensors for precise positioning.
- **Fluid systems:** Pneumatic and hydraulic drives—convert pressure or flow into linear or rotary motion.
- **Thermal or chemical:** Generate heat, flow, or chemical reactions to influence process variables.

Key developments

- Integrated electronics and control software → smart servo drives.
- Environmentally friendly pneumatics and efficient hydraulic systems.
- Specialized actuators: piezo, magnetostrictive, or shape-memory drives.

Sensors: The Sensory Organs of Automation Systems

Role: Sensors detect the state of elements in a technical process and convert physical or chemical quantities into signals.

Classification by perception principle:

Sense	Perception	Examples
Vision	Light, contours	Camera, optical systems
Hearing	Sound	Microphone, ultrasound
Smell	Fragrance	Gas or chemical analyzer
Taste	Ingredients	Chromatograph
Touch	Force, shape, heat	Probe, strain gauge, thermometer

Key trends:

- Self-calibration and diagnostic features.
- Data fusion from multiple sensing principles.
- Integration into sensor networks with edge intelligence.

Advanced Sensor Concepts and Data Fusion

Modern enhancements:

- **Intelligent sensors:** perform preprocessing, self-diagnosis, and error correction.
- **Data fusion:** combine multiple measurement principles to improve robustness and accuracy.
- **Human–machine integration:** intuitive visualization and configuration interfaces for non-experts.

Emerging technologies

- **Sensor networks:** distributed measurement and shared data processing.
- **Cognitive sensors:** capable of perception and context-aware decision making.

Outlook:

- Sensors are evolving toward **autonomous subsystems** that enable perception, reasoning, and adaptation—key enablers of **Industry 4.0**.

Control at the Center of Automation

Terminology (used precisely in this course)

- **Open-loop control:** controller output does *not* use the measured process output.
- **Feedforward control (open-loop with disturbance model):** known/disturbance inputs are measured or estimated and compensated *without* comparing to the output.
- **Closed-loop (feedback) control:** measured output is compared to a reference; the error drives the control action.

Key idea

Real systems often *combine* feedforward (anticipate known effects) with feedback (reject unknown disturbances) for improved performance.

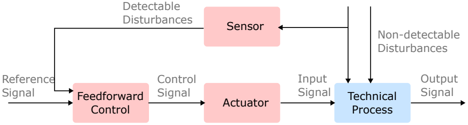
Feedforward Control with Disturbance Detection

Structure & assumptions

- Relationships between inputs, outputs, and disturbances are known at design time.
- Disturbances that are **detectable/measurable** can be injected into a compensator to pre-empt their effect.
- Unknown or non-detectable disturbances still pass to the output.

Pros/Cons

- ✓ Fast response, no sensor delay in the loop, can avoid unnecessary actuation.
- × Model dependence; cannot correct modelling errors or unmeasured disturbances.



Feedforward with sensor for disturbance detection

Closed-Loop (Feedback) Control & Practical Example

Closed-loop principle

- Measure output → compare with reference → control action reduces error.
- Rejects unknown disturbances; tolerates modelling errors.

Example: room thermostat

- *Feedback only:* keeps temperature at setpoint; sudden window opening (disturbance) causes controller to drive heater hard, wasting energy.
- *With feedforward sensor logic:* detect rapid temperature drop due to ventilation, **temporarily disable** heating (open-loop action), then resume feedback control after window closes.

Lesson: combine **feedback** (robustness) with **feedforward** (anticipation) when disturbance pathways are known.

Choosing Between Control Types

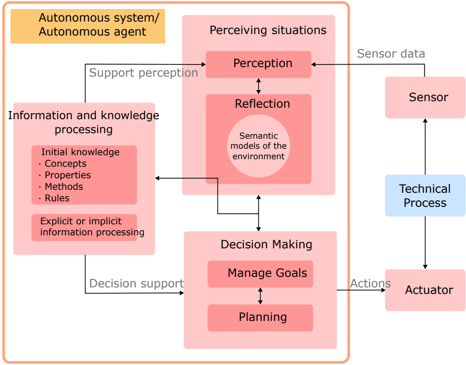
Aspect	**Open-Loop (pure)**	**Feedforward (open-loop w/ disturbance)**	**Closed-Loop (feedback)**
Uses output measurement?	No	No (uses disturbance measurement/model)	**Yes** (error-driven)
Handles unknown disturbances	Poor	Limited (only measured/known)	**Good**
Model dependence	Low–Moderate	**High** (needs accurate path)	Moderate
Response speed	Fast (no sensor delay)	**Very fast** (anticipatory)	Slower (measurement + computation)
Steady-state accuracy	Sensitive to drift	Sensitive to model mismatch	**High** with integral action
Typical use	Simple sequences	Known load/ambient effects	General regulation, robustness

Practical guideline: Use *feedback as the backbone*; add *feedforward* when disturbance paths are known and measurable.

Autonomous System: Architecture at a Glance

Main parts (contrasted with an automaton)

- **Perception**: acquire and interpret sensor data, resolve ambiguities.
- **Reflection**: build/update semantic models of the environment; maintain a runtime world model.
- **Decision-making**: goal management and action planning under constraints.
- **Information & knowledge processing**: initial knowledge (concepts, properties, methods, rules) and learned relations.
- **Sensors/Actuators**: interact with the technical process (measurements and actions).
- **User**: supervises, defines context and boundaries ("guard rails").



"Autonomous system architecture".

Roles of the Cognitive Components

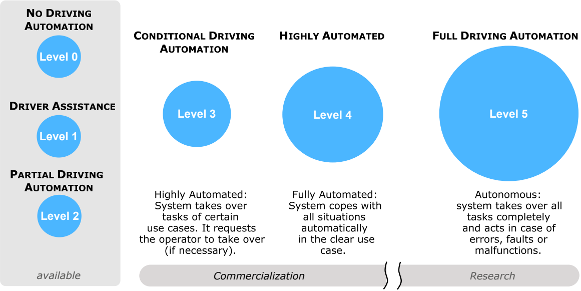
- **Perception** — interprets sensor data, extracts relevant information from complex/partial observations.
- **Reflection** — maintains a realistic runtime model; feeds planning with up-to-date context.
- **Goal management** — selects appropriate objectives for the current situation.
- **Planning** — chooses and sequences actions; enables actuators to influence the process.
- **Information/knowledge processing** — augments initial knowledge via learning new relationships (AI/ML).

Practical stance

- AI/ML is key for perception and pattern recognition, but open questions remain for deduction/inference; near-term use is in *well-defined contexts*.
- Many real deployments are **supervised autonomy**: the system recognizes limits and prompts human intervention when needed.

Levels of Driving Automation (SAE) — Analogy for Autonomy

- **L0-L2:** No/partial automation; driver assistance.
- **L3 "Highly Automated":** System handles certain use cases; requests human takeover if necessary (commercially available).
- **L4 "Fully Automated":** System manages all situations within a *clear use case*; human may intervene via teleoperation in exceptions (active research/prototypes).
- **L5 "Autonomous":** System handles all tasks and acts in case of errors/faults — long-term research goal.



Highly Automated: System takes over tasks of certain use cases. It requests the operator to take over (if necessary).

Fully Automated: System copes with all situations automatically in the clear use case.

Autonomous: system takes over all tasks completely and acts in case of errors, faults or malfunctions.

(SAE levels.)

Transfer to industry

Best near-term fit is in **specialized, well-structured applications** (e.g., mining vehicles, harvesting, intralogistics cells).

Opportunities, Limits, and Responsibilities

Why autonomy in industry

- Addresses labor shortages; increases availability; copes with unforeseen events; supports productivity and safety.

Limits & open challenges

- Handling rare/novel situations; robustness under uncertainty; verification and validation of learning components.
- Ethical & legal issues: accountability when decisions are made by the system; need for societal discourse and risk management.

Design takeaway

- Engineer **guard rails**: clear operating domain, fail-safe behaviors, and human-in-the-loop supervision.
- Combine perception-reflection-planning with measurable objectives; log decisions for auditability.

Module I: Introduction to Automation Technologies

Submodule: Safety and open source tools

What is Mechatronics?

Origin & sense of the term

- "Mecha" (mechanism) + "tronics" (electronics): products integrate electronics/software *intimately* with mechanisms—boundaries blur.
- A **mechatronic system**: a mechanical machine under electronic (incl. software) control.
- **Mechatronics (discipline)**: all subject matter needed to *specify and develop* such systems—modelling/simulation of the machine, sensors/actuators, and the control system.

Canonical scope (per text)

- **Mechanical systems**: elements, machines, precision mechanics
- **Electronic systems**: micro/power electronics, sensor & actuator tech
- **Information technology**: systems theory, automation, software engineering, AI

Modern view: mechatronics is a *holistic* system-design activity, not just "add electronics to mechanics."

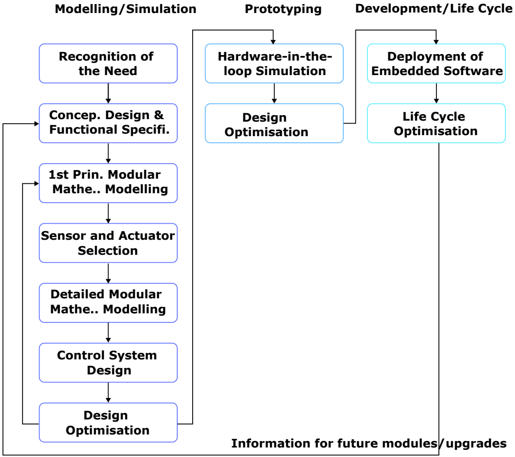
Generic Mechatronic System Design

Block perspective

- Mechanical system ↔ Sensors/Actuators
- Electrical/electronic system (power + interface)
- A/D & D/A conversion around the control computer
- Computer system executes control/software functions

Implication

- Co-design across mechanics–electronics–software is mandatory; signals bridge domains.

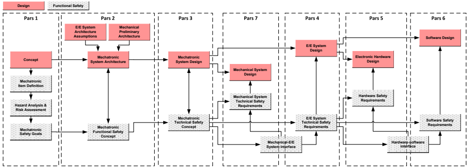

Modelling/Simulation
Prototyping
Development/Life Cycle
Recognition of the Need
Concep. Design & Functional Specifi.
1st Prin. Modular Mathe.. Modelling
Sensor and Actuator Selection
Detailed Modular Mathe.. Modelling
Control System Design
Design Optimisation
Hardware-in-the-loop Simulation
Design Optimisation
Deployment of Embedded Software
Life Cycle Optimisation
Information for future modules/upgrades

Generic Mechatronic System Design

Mechatronic Design Process (from the text)

Main lanes

- **Modelling/Simulation**: recognize need → concept & functions → first-principle modular models → sensor/actuator selection → detailed models → control design → optimisation.
- **Prototyping**: HIL (hardware-in-the-loop) simulation → design optimisation.
- **Deployment/Lifecycle**: embedded software deployment → lifecycle optimisation → info for future modules/upgrades.

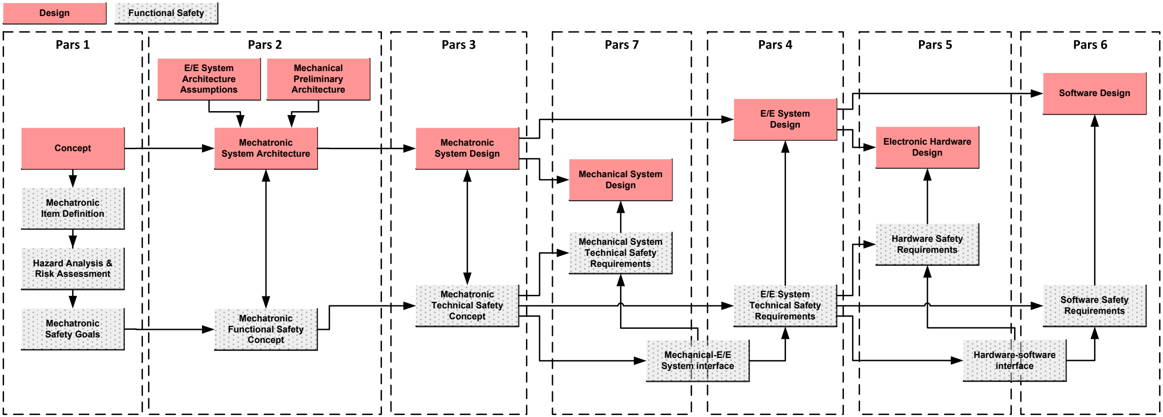

Design
Functional Safety
Pars 1
Pars 2
Pars 3
Pars 7
Pars 4
Pars 5
Pars 6
Concept
Mechatronic Item Definition
Hazard Analysis & Risk Assessment
Mechatronic Safety Goals
E/E System Architecture Assumptions
Mechanical Preliminary Architecture
Mechatronic System Architecture
Mechatronic Functional Safety Concept
Mechatronic System Design
Mechatronic Technical Safety Concept
Mechanical System Design
Mechanical System Technical Safety Requirements
Mechanical-E/E System interface
E/E System Design
E/E System Technical Safety Requirements
Electronic Hardware Design
Hardware Safety Requirements
Hardware-software interface
Software Design
Software Safety Requirements

Mechatronic Design Process

What the figure emphasizes

Prominent role of modelling & simulation; translation of control models into embedded software follows in deployment.

Mechatronic Design Process (from the text)


Design
Functional Safety
Pars 1
Pars 2
Pars 3
Pars 7
Pars 4
Pars 5
Pars 6
Concept
Mechatronic Item Definition
Hazard Analysis & Risk Assessment
Mechatronic Safety Goals
E/E System Architecture Assumptions
Mechanical Preliminary Architecture
Mechatronic System Architecture
Mechatronic Functional Safety Concept
Mechatronic System Design
Mechatronic Technical Safety Concept
Mechanical System Design
Mechanical System Technical Safety Requirements
Mechanical-E/E System interface
E/E System Design
E/E System Technical Safety Requirements
Electronic Hardware Design
Hardware Safety Requirements
Hardware-software interface
Software Design
Software Safety Requirements

Mechatronic Design Process

System-Level Modelling & Systems Engineering Notes

Why systems engineering matters

- Mechatronic design integrates multiple engineering disciplines from *concept* stage—beyond traditional siloed flows.
- Activities align with ISO 15288 ideas: stakeholder needs, system requirements, architecture, detailed design, and analysis.

SysML for concept design

- Addresses the lack of an early mechatronic concept process by enabling **system-level models** that treat all domains equitably—including *non-material* (software) components.
- Mechatronic systems are **systems-of-systems**; SysML diagrams capture dependencies among subsystems originating from different disciplines.

Takeaway

Start with system-level models; iterate models → HIL → deployment; carry forward information for upgrades.

Hierarchy of Safety Requirements

Motivation:

- To develop a **safety argument** for a *mechatronic system* that integrates both mechanical and electrical/electronic (E/E) subsystems.
- Based on the ISO 26262 hierarchy, safety requirements are cascaded through several levels of abstraction.

Safety hierarchy (ISO 26262):

- **Safety Goals** — derived from *item definition*.
- **Functional Safety Requirements** — applied to the *preliminary architecture*.
- **Technical Safety Requirements** — mapped to *system design*.
- **Hardware Safety Requirements** — assigned to hardware design.
- **Software Safety Requirements** — assigned to software design.

Challenge: No universally accepted equivalent hierarchy yet exists for *mechatronic or mechanical* systems — motivating the need for a unified model.

Functional Safety Concept

Definition:

- Derived from ISO 26262 — ensures that the system maintains a safe state even under foreseeable faults.
- In a **mechatronic context**, this extends beyond E/E components to include mechanical subsystems.

In Mechatronic Systems:

- The **Mechatronic Functional Safety Concept** includes both control logic and the physical plant (mechanical system).
- Mechanical aspects formerly referred to as *external measures* are now internalized into the concept.
- The concept remains independent of detailed E/E and mechanical designs but defines required safe states and control strategies.

Purpose: Provide a unified architecture that integrates *mechanical and electrical safety assumptions* early in design.

Technical Safety Concept

Definition:

- Refines functional safety requirements into detailed, implementable *technical requirements*.
- Establishes how the safety goals are realized through both E/E and mechanical design decisions.

In Mechatronic Systems:

- The **Mechatronic Technical Safety Concept** links the functional concept with physical system implementation.
- It includes mappings between:
  - Mechanical design parameters
  - E/E subsystem performance limits
  - Interfaces at the **Mechanical-E/E boundary**
- These relationships define how both domains interact to achieve required safety integrity.

Purpose: Ensure traceability from high-level safety goals down to specific mechanical and E/E design choices.

Hardware Safety and Integration

Hardware Safety Requirements:

- Derived from the technical safety concept; specify measurable reliability and fault tolerance targets.
- Include quantitative metrics such as fault rates, diagnostic coverage, and safe failure fraction.

Integration Challenges in Mechatronic Systems:

- Hardware safety applies not only to electronic hardware but also to:
  - Electromechanical interfaces (e.g., actuators, sensors, converters)
  - Mechanical structures critical to safe operation
- Co-validation of mechanical and E/E subsystems is essential to verify safe performance.

Takeaway: Mechatronic safety demands a **unifying design ontology** linking functional, technical, and hardware domains — ensuring traceability from system goals down to component implementation.

Introduction to HARA for Mechatronic Systems

Purpose of HARA- Hazard Analysis and Risk Assessment:

- Identify and evaluate potential hazards that can lead to unsafe conditions in a mechatronic system.
- Derive safety goals and requirements systematically from system-level behavior.

Applied Example: 4CAS (Four Corner Air Suspension System)

- 4CAS raises or lowers vehicle body using air springs at each wheel.
- Sensors measure ride height; ECU adjusts pressure via electromechanical valves.
- Typical functions:
  - Maintain level body at selected height.
  - Adjust vehicle height for comfort, aerodynamics, or off-road conditions.

HARA Objective: Link mechanical (pneumatic, structural) and E/E (electronic, software) subsystems within a unified safety argument.

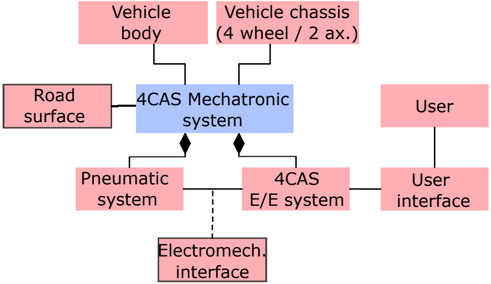
System Architecture Context - 4CAS

System Components:

- **Vehicle Body & Chassis** — dynamic mass and structural reference.
- **Pneumatic System** — air springs, valves, compressor.
- **4CAS E/E System** — ECU, sensors, software control.
- **Electromechanical Interface** — connects mechanical and electrical domains.
- **User Interface** — in-cabin and remote commands.

System Features:

- Closed-loop control at each corner.
- Cross-linking for rough terrain handling



High-level 4CAS Block Diagram.

HARA Framework for Mechatronic Systems

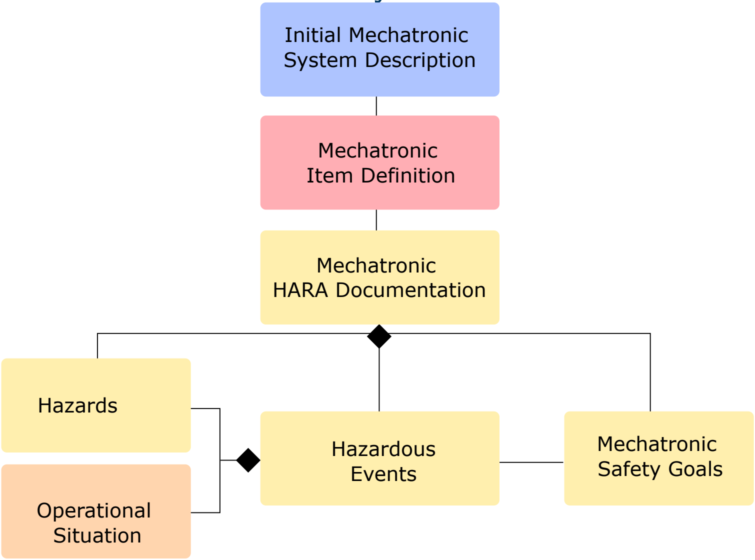
Pars 1: Mechatronic Item Definition and HARA

- Defines system boundaries and operational context.
- Identifies hazards and operational scenarios.
- Evaluates hazardous events to establish **Mechatronic Safety Goals**.

Key Steps:

1. **Initial System Description** — captures mechanical and E/E domains.
2. **Mechatronic Item Definition** — functional behavior, interfaces, and assumptions.
3. **Mechatronic HARA Documentation** — hazards, operational situations, and hazardous events.

HARA Framework for Mechatronic Systems


Initial Mechatronic System Description
Mechatronic Item Definition
Mechatronic HARA Documentation
Hazards
Operational Situation
Hazardous Events
Mechatronic Safety Goals

HARA Design Description

Hazard Identification and Safety Goal Derivation

Conceptual Flow (from the figure and text)

- Hazards arise from both **mechanical** (e.g., over-pressure, spring failure) and **E/E** (e.g., sensor fault, ECU error) domains.
- Each operational situation is evaluated to define corresponding **Hazardous Events**.
- Risk levels (severity, exposure, controllability) determine the **ASIL** (Automotive Safety Integrity Level).
- From hazardous events, **Mechatronic Safety Goals** are derived — spanning mechanics, electronics, and software.

Example (4CAS):

- *Hazard:* Loss of air pressure in one corner → vehicle tilt → loss of control.
- *Safety Goal:* Maintain stable body height and prevent rapid tilt under fault conditions.

Outcome: The HARA provides the foundational safety reasoning upon which functional, technical, and hardware safety concepts are built.

Safety Concepts in Mechatronic Systems

Hierarchy of Safety Concepts (ISO 26262-inspired):

1. **Functional Safety Concept (Pars 2)** — defines system-level safe behavior and responses.
2. **Technical Safety Concept (Pars 3)** — allocates safety requirements to system elements.
3. **Hardware/E/E Technical Safety Concept (Pars 4)** — addresses hardware, software, and interface implementation.

Integration Challenge:

- Mechanical and E/E systems must be unified under one safety argument.
- Functional → Technical → Hardware flow ensures traceability from hazard to implementation.

4CAS Context Example: Air suspension must maintain vehicle height even if a sensor or valve fails — this defines the top-level *Functional Safety Goal*.

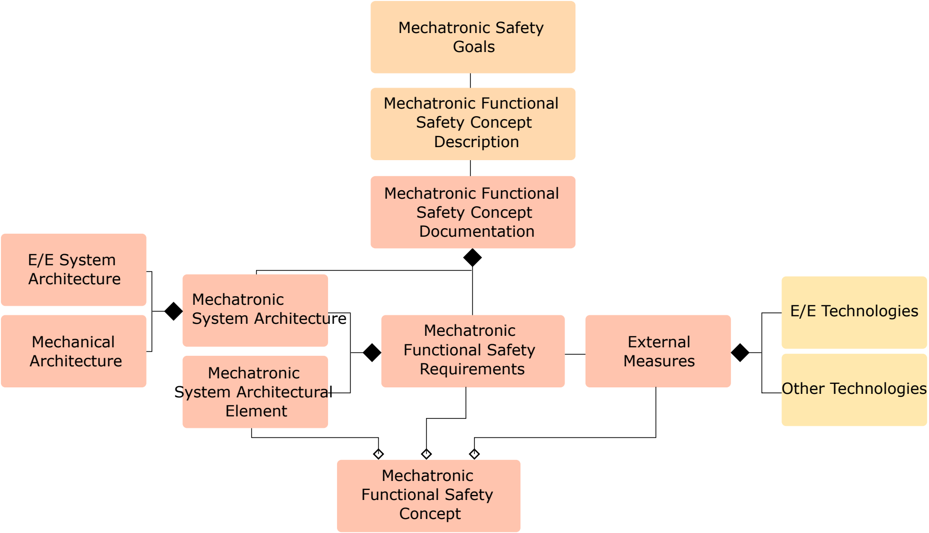
Functional Safety Concept (Pars 2)

Objective: Translate the *Mechatronic Safety Goals* from HARA into concrete *functional behaviors* that guarantee safety.
Key Components:

- **Mechatronic System Architecture** = Mechanical + E/E Architecture.
- **Functional Safety Requirements** are assigned to both mechanical and electronic elements.
- **External Measures** still exist for effects outside system scope (e.g., backup mechanical stops).

Example (4CAS): If an air-pressure sensor fails, the ECU must detect the fault and switch to a safe mode — e.g., lock ride height until service.

Functional Safety Concept (Pars 2)


Mechatronic Safety Goals
Mechatronic Functional Safety Concept Description
Mechatronic Functional Safety Concept Documentation
E/E System Architecture
Mechanical Architecture
Mechatronic System Architecture
Mechatronic System Architectural Element
Mechatronic Functional Safety Requirements
External Measures
E/E Technologies
Other Technologies
Mechatronic Functional Safety Concept

Mechatronic Functional Safety Concept Diagram.

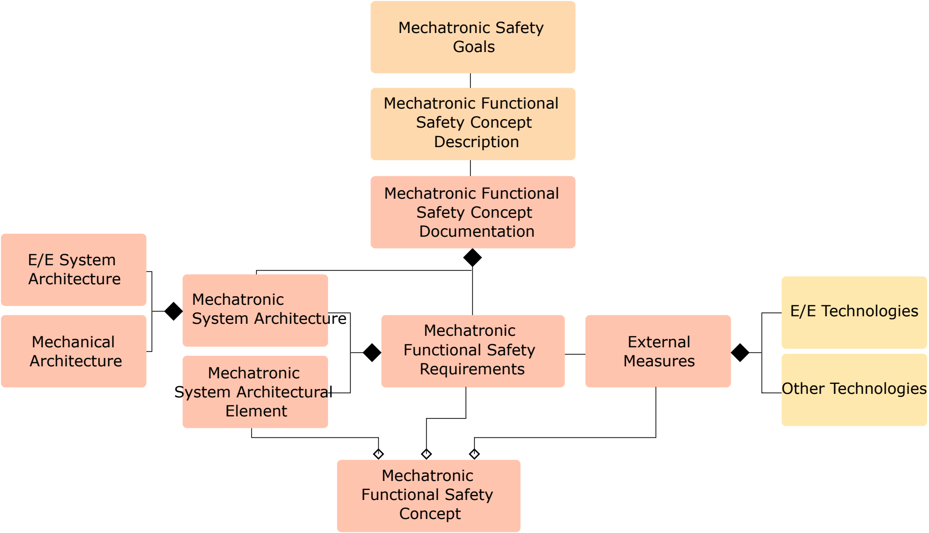
Technical Safety Concept (Pars 3)

Objective: Refine functional requirements into implementable, verifiable **technical safety requirements**.
Key Aspects:

- Extends design to explicit **mechanical elements** (e.g., actuators, springs).
- Defines **interfaces** between E/E and mechanical subsystems.
- Allocates safety requirements across:
  - **E/E System Elements:** controllers, sensors, software.
  - **Mechanical System Elements:** valves, cylinders, mounts.

Example (4CAS): Ensure valve response time and ECU actuation timing meet defined limits so that pressure adjustments remain stable under faults.

Technical Safety Concept (Pars 3)


Mechatronic Safety Goals
Mechatronic Functional Safety Concept Description
Mechatronic Functional Safety Concept Documentation
E/E System Architecture
Mechanical Architecture
Mechatronic System Architecture
Mechatronic System Architectural Element
Mechatronic Functional Safety Requirements
External Measures
E/E Technologies
Other Technologies
Mechatronic Functional Safety Concept

Mechatronic Technical Safety Concept Diagram.

Hardware / E/E Technical Safety Concept (Pars 4)

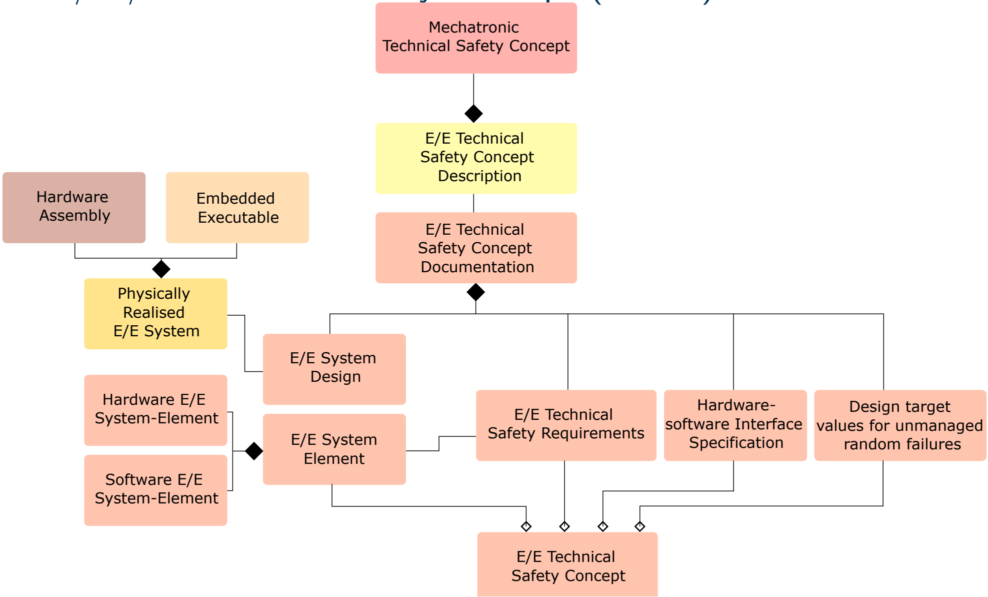
Objective: Detail how E/E and software components implement and verify technical safety requirements.

Elements:

- **Hardware Software Interface Specification** — defines signal, timing, and fault-reaction behavior.
- **Design Targets:** fault tolerance, diagnostic coverage, random failure rates.
- **Physically Realised System:** Hardware + Embedded Executable + Mechanical Interface.

Example (4CAS): ECU performs real-time diagnostics of height sensors; if the sensor drifts, the controller uses redundant input or holds last safe pressure.

Hardware / E/E Technical Safety Concept (Pars 4)


Mechatronic Technical Safety Concept
E/E Technical Safety Concept Description
Hardware Assembly
Embedded Executable
E/E Technical Safety Concept Documentation
Physically Realised E/E System
E/E System Design
Hardware E/E System-Element
E/E System Element
Software E/E System-Element
E/E Technical Safety Requirements
Hardware-software Interface Specification
Design target values for unmanaged random failures
E/E Technical Safety Concept

Relationship Between Concepts

Cascading Structure:

1. **Functional Safety Concept** defines "*what*" must be safe.
2. **Technical Safety Concept** defines "*how*" it will be made safe.
3. **Hardware/E/E Concept** defines "*how it is implemented and verified*".

4CAS Example Summary:

- **Functional:** Keep vehicle level under all driving modes.
- **Technical:** Ensure consistent pneumatic pressure control, even with sensor or valve degradation.
- **Hardware:** Verify electrical and mechanical subsystems meet reliability and timing specifications.

Outcome: These layers ensure complete traceability from hazard $\rightarrow$ goal $\rightarrow$ design $\rightarrow$ verification, across both mechanical and electronic domains.

Open-Source Automation: Tools and Benefits

What is Open-Source Automation?

- **Definition:** Use of freely available, community-driven hardware and software for automation, control, and data management.
- **Goal:** Promote transparency, interoperability, and cost-effective experimentation without vendor lock-in.

Typical Open-Source Ecosystem

Layer	Example Tools / Platforms
Field / Sensors	Arduino, ESP32, Raspberry Pi Pico, MicroPython
Control / PLC	OpenPLC, OpenHAB rules engine, IEC 61131-3 editor
Edge / SCADA	Node-RED, Grafana, InfluxDB, Mosquitto (MQTT broker)
Analytics / Cloud	Python (NumPy, Pandas), TensorFlow Lite, Scikit-learn
Communication	OPC UA (open62541), Modbus TCP, MQTT
Modeling / Simulation	OpenModelica, PyPSA, SimPy, PyBaMM

Open-Source Automation: Tools and Benefits

Benefits for Students

- **Hands-on learning:** Build and control systems using low-cost hardware and public repositories.
- **Reproducibility:** Fully inspect, modify, and share experimental setups.
- **Skill development:** Exposure to real industrial protocols (OPC UA, MQTT) and modern data pipelines.
- **Community & Collaboration:** Global developer support and continuous improvement.

Reproducibility in Automation Systems

Why it matters

- Modern automation projects combine hardware, software, and data—making reproducibility essential for verification and scalability.
- Ensures that experiments, control logic, and system behaviour can be **replicated across teams, sites, or devices**.
- Enables traceability and transparency in industrial R&D and academic workflows.

Reproducibility in Automation Systems

Core principles of reproducibility

- **Version control:** Git/GitHub or GitLab to track PLC code, Node-RED flows, and Python scripts.
- **Environment definition:** Docker or Conda environments to reproduce runtime dependencies.
- **Documentation:** Markdown, Jupyter Notebooks, and READMEs for configuration and setup.
- **Automated testing:** Simulate logic and signal behaviour to ensure consistent results.

Example:
A student replicates a temperature-control experiment from another group by cloning the GitHub repo and launching an identical OpenPLC + Node-RED + InfluxDB Docker stack.

Table of contents

Sensor Technologies

Dr Bikash Sah

Universität Siegen

IAS Interconnected Automation Systems

Introduction to Sensor Systems

- Sensors are essential to observe and influence physical, chemical, or biological processes.
- They enable control of systems by providing information about the **current state**.
- Without sensors, automated or intelligent systems cannot perceive their environment.
- Sensor systems combine physical measurement principles with electronic evaluation and control.

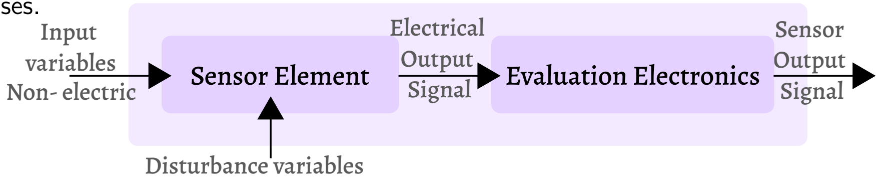
Key idea: To control a process, one must first *measure*, then *evaluate*, and finally *act*.

Definition and Mode of Operation

Definition

The term **sensor** originates from the Latin *sensus* (sense, feeler). A sensor is a device that converts a physical, chemical, or biological quantity into an **electrical output signal**.

- Sensors are used for quantitative and qualitative measurements of various parameters (e.g., temperature, pressure, acceleration, chemical concentration).
- Each sensor system consists of two essential parts:
  1. **Sensor element:** converts non-electrical input variables (e.g., pressure, light) into an electrical signal.
  2. **Evaluation electronics:** amplifies, filters, or digitizes the signal for control or monitoring purposes.



Operating principle of a sensor system

Operating Principle of Sensors

- The sensor receives **input variables** (physical or non-electrical quantities).
- It produces an **electrical output signal** proportional to the input.
- External **disturbance variables** (temperature, vibration, noise) can affect accuracy.
- Modern sensors compensate disturbances using embedded microprocessors and calibration algorithms.
- The combination of sensing element and processing electronics is known as a **smart sensor**.

Smart Sensor

A smart sensor integrates the sensing element and electronic evaluation in a single compact unit, capable of self-calibration and digital communication.

Classification of Sensors

1. Based on Power Source

- **Active sensors:** Generate an electrical signal directly from the measured variable (e.g., piezoelectric, thermoelectric).
- **Passive sensors:** Require external auxiliary power to produce an output (e.g., resistive, capacitive, inductive, ultrasonic).

2. Based on Measurement Principle

- Mechanical, thermal, magnetic, optical, chemical, biological, etc.

3. Based on Nature of Output

- **Analog sensors:** Continuous output (e.g., thermistor).
- **Digital sensors:** Include A/D conversion and provide discrete signals or data streams.

Digital and Virtual Sensors

Digital Sensors

- Integrate **analog-to-digital converters (ADCs)** within the evaluation electronics.
- Output is a **digital signal** directly usable by controllers or microprocessors.
- Example: Digital accelerometer (I^2C/SPI interface).

Virtual Sensors

- Derive the desired variable **computationally** using software models.
- Based on mathematical models or empirical correlations.
- Used when:
  - Real sensors are expensive or prone to wear (e.g., nuclear plants, harsh environments).
  - Quantities can be inferred indirectly (e.g., estimated torque from current and speed).

Example

In automotive systems, a virtual tire pressure sensor can estimate pressure from wheel-speed variations and temperature.

Summary and Discussion

- Sensors are the foundation of automation and control.
- Each sensor has a sensing element and signal-conditioning electronics.
- Classification helps select the right sensor for a given task:
  - Active vs Passive
  - Analog vs Digital
  - Real vs Virtual
- Emerging trends: miniaturization, MEMS, IoT-ready smart sensors.

Classroom Question:

Why might a virtual sensor be preferred over a physical sensor in a battery management or air suspension system?

Piezoelectric Effect

What is the Piezoelectric Effect?

Direct effect: When certain solids are *deformed* by force/pressure, internal charge centers shift $\Rightarrow$ an **electrical voltage** appears at the surfaces (electric polarization P).

Inverse effect: Applied voltage causes a *mechanical strain/oscillation*.

Key ideas from the text

- Polarization P depends on direction of force $\mathbf{F}$ and crystal axes.
- Only *changes* in force generate charge (static force alone produces no output).
- Effect is temperature dependent and can be nonlinear at high fields.

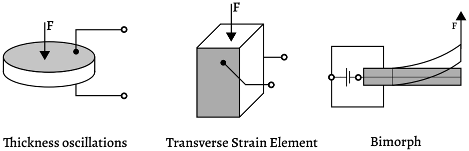
Principle of Operation (Direct) and Modes

Operating chain

Force/pressure → charge displacement → surface charge → measurable voltage.

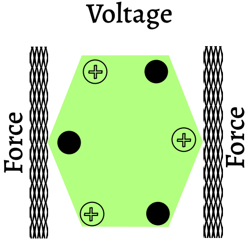
Common modes:

- **Longitudinal** / thickness oscillation (stress ∥ pol. dir.).
- **Transverse** strain element (stress ⊥ pol. dir.).
- **Bimorph** (two layers bend under field/force).



Thickness oscillations Transverse Strain Element Bimorph

Typical modes of the piezoelectric effect.



Principle: force → voltage (longitudinal).

Coordinate Description and Basic Relations

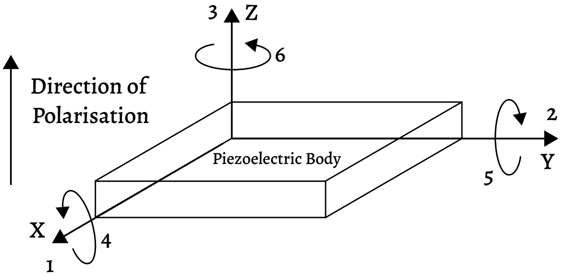
Coordinate system

Indices $i \in \{1, 2, 3\}$ denote force/field directions, $j \in \{1, 2, 3\}$ surface normals.

Constitutive relations

$$Q = d_{kl}\, F, \qquad P = e\, \sigma, \qquad \sigma_{ij} = \frac{F_i}{A_j}$$

d_{kl}: *piezoelectric charge coefficient;* e: *piezoelectric elongation coefficient.*



Axes and polarization direction.

Measurement note: Because $Q \propto \dot{F}$, output is often *current-like*; integrating current yields charge proportional to force change.

Materials

Requirement: Nonconductive, non-centrosymmetric structure (no inversion symmetry).

Groups

- **Natural crystals:** quartz, tourmaline, $LiNbO_3$, $LiTaO_3$ (stable, small d).
- **Ceramics (PZT, $BaTiO_3$)**: *perovskite* structure, high d, tailored by doping.
- **Thin films/polymers:** PVDF, ZnO, AlN on Si (MEMS compatible).

Material	d_{ij}	Value (pC/N)
$LiNbO_3$	d_{22}	0.67
Tourmaline	d_{33}	1.83
Quartz	d_{11}	2.3
$LiTaO_3$	d_{33}	9.2
PVDF film	d_{13}	23
PZT ceramics	d_{33}	593

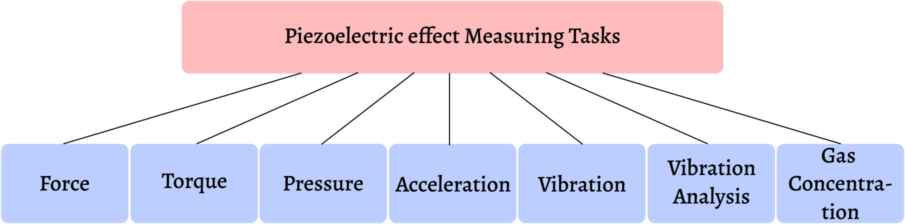
Applications

Sensing (direct effect)

- Force, torque, pressure, acceleration, vibration and modal analysis.
- Acoustic/ultrasonic transducers; gas/flow via acoustic coupling.

Actuation (inverse effect)

- Micro/nano-positioning stages, injection nozzles, inkjet print heads.
- AFM/STM scanners; fast valves and precision metering.



Typical measurement tasks enabled by the piezoelectric effect.

Design and Measurement Tips

- **Dynamic quantity:** best for varying loads; use charge or IEPE amplifiers for low-frequency work.
- **Calibration:** temperature and preload affect sensitivity; account for nonlinearity at high fields.
- **Axis matters:** choose mode (longitudinal/transverse/bimorph) to match signal level and mechanical layout.
- **Integration:** ceramics (PZT) for high output; PVDF/AlN/ZnO for MEMS and flexible substrates.

Resistive & Piezoresistive Effect

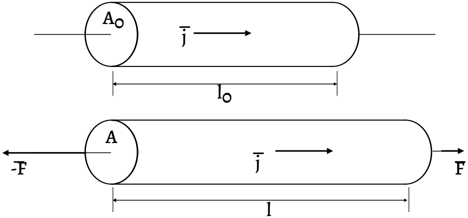
Operating Principle (Resistive Effect)

Specific resistivity ρ of a metal is (to first order) *independent of voltage*.
Electrical resistance of a conductor of length l_0 and cross-section A_0:

$$R = \rho \frac{l_0}{A_0}$$

Implications:

- $R \propto l$ (stretched $\Rightarrow$ longer) $R \propto \frac{1}{A}$ (stretched $\Rightarrow$ thinner).
- In the elastic range, **tension** increases R; **compression** decreases R.



Conductor under mechanical stress

Mechanical stress $\sigma = \frac{F}{A}$, strain $\varepsilon = \frac{\Delta l}{l}$, Hooke: $\sigma = E\varepsilon$.

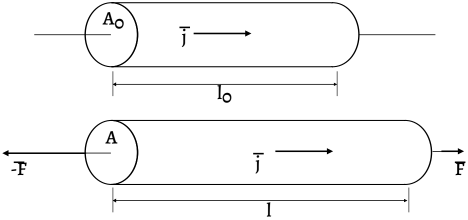
From Strain to Resistance Change

Under axial load, the effective resistance becomes

$$R_F = \rho \, \frac{l_F}{A_F}$$

The relative change is (geometric effect):

$$\frac{\Delta R}{R} = k \, \varepsilon$$



Conductor under mechanical stress

k **(gauge factor)** ≈ 2 for metals (lengthening + Poisson contraction).
For **semiconductors**, ρ also changes with stress (carrier mobility $\Rightarrow$ **piezoresistive effect**), giving much larger k.

Foil Strain Gauges (SGs)

Construction:

- Meandered metallic grid (e.g., Constantan, Karma) on polyimide carrier; protective film.
- Bonded to the test surface by spot welding (rare) or adhesives (cyanoacrylates, epoxies).

Typical numbers:

- Commercial SG: $R \approx 120\,\Omega$; for $\varepsilon = 10^{-3}$ and $k \approx 2$, $\Delta R \approx 0.24\,\Omega$.
- Small $\Delta R \Rightarrow$ use **Wheatstone bridge** and low-noise instrumentation.

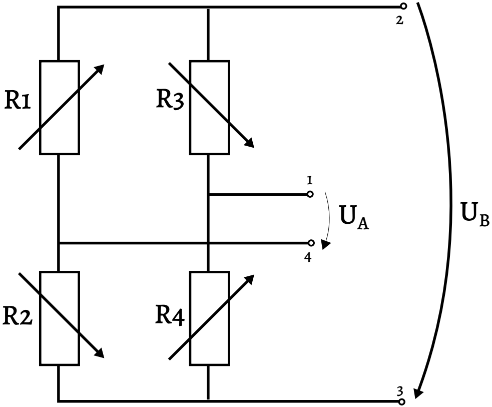
Wheatstone Bridge Readout

General bridge: Four arms $R_1..R_4$, supply U_B, output U_A.

$$\frac{U_A}{U_B} = \frac{R_1}{R_1 + R_2} - \frac{R_4}{R_3 + R_4}$$

For small changes $\Delta R_i \ll R_i$:

$$\frac{U_A}{U_B} = \frac{1}{4}\left(\frac{\Delta R_1}{R_1} - \frac{\Delta R_2}{R_2} + \frac{\Delta R_3}{R_3} - \frac{\Delta R_4}{R_4}\right)$$
$$= \frac{k}{4}\left(\varepsilon_1 - \varepsilon_2 + \varepsilon_3 - \varepsilon_4\right)$$



Quarter, half or full bridges: combine gauges to boost signal, cancel temperature, and measure bending, axial, torsion selectively.

Gauge Factor (k) by Material

Material	**Composition**	k
Constantan	54Cu-45Ni-1Mn	2.05
Karma	73Ni-20Cr-Fe/Al	2.1
Nichrome V	80Ni-20Cr	2.2
Chromel C	65Ni-20Fe-15Cr	2.5
Pt-W	92Pt-8W	4.0
Platinum	100 Pt	6.0

Notes

- Metals: stable, linear, low k; good for structural tests.
- Semiconductors (Si): very high k (tens-hundreds) via *piezoresistivity*; higher temp. dependence and nonlinearity.

Applications and Practical Tips

Applications

- Transducers for **force, mass, torque, pressure**; accelerometers.
- Experimental stress analysis: validation of FE simulations; civil, rail, aerospace, geodynamics.

Good practice

- Use half/full bridges for temperature compensation and higher signal.
- Calibrate with known loads; consider creep and adhesive properties.
- Choose grid orientation to capture principal strains; mind Poisson ratio.

Magnetoresistive Effect: Principle of Operation

Concept: When a magnetic field is applied to a conducting material through which current flows, the electrical resistance changes. This is known as the **magnetoresistive effect (MR)**.

Mathematical Definition:

$$\frac{\Delta R}{R} = \frac{R(H) - R(0)}{R(0)}$$

where $R(H)$ is the resistance in magnetic field H and $R(0)$ is resistance without magnetic field.

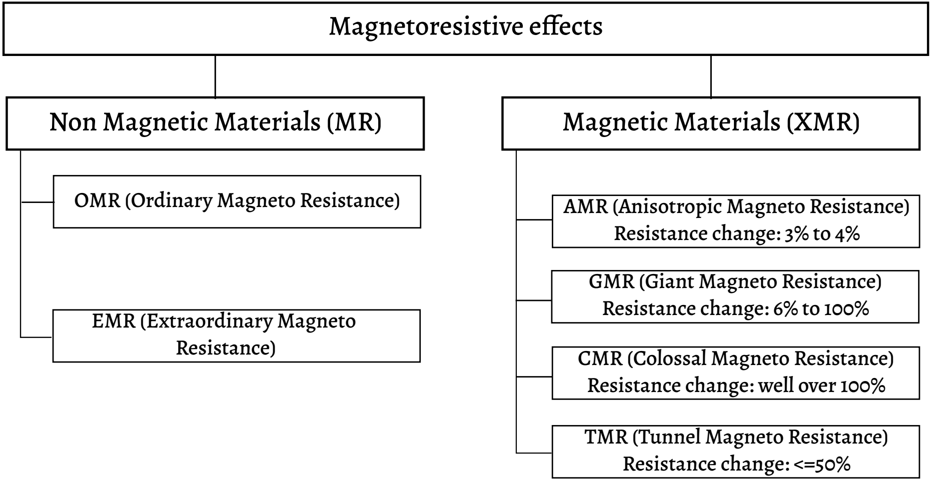
Physical Principle:

- In presence of H, charge carriers experience a Lorentz force:

$$F_L = Q(\mathbf{v} \times \mathbf{B})$$

- This deflects electrons, increasing their path length and hence resistance.
- Effect depends on material type (magnetic or non-magnetic) and electron mobility.

Classification of Magnetoresistive Effects


Magnetoresistive effects
Non Magnetic Materials (MR)
OMR (Ordinary Magneto Resistance)
EMR (Extraordinary Magneto Resistance)
Magnetic Materials (XMR)
AMR (Anisotropic Magneto Resistance)
Resistance change: 3% to 4%
GMR (Giant Magneto Resistance)
Resistance change: 6% to 100%
CMR (Colossal Magneto Resistance)
Resistance change: well over 100%
TMR (Tunnel Magneto Resistance)
Resistance change: <=50%

Overview of Magnetoresistive Effects

Classification of Magnetoresistive Effects

Overview of Magnetoresistive (MR) Effects:

Non-magnetic materials (MR):

- **OMR - Ordinary Magnetoresistance:** Caused by Lorentz-force deflection of charge carriers in a magnetic field.
  Effect is small ($< 1\%$) in most metals, but large in semimetals like Bi (up to 100%).
- **EMR - Extraordinary Magnetoresistance:** Observed in hybrid metal/semiconductor structures (e.g., InSb).
  Large resistance change due to field-induced current redistribution.

Magnetic materials (XMR):

- **AMR - Anisotropic MR:** Change of 3-4 % depending on angle between current and magnetization. Caused by spin-orbit interaction in ferromagnets (e.g., NiFe alloys).
- **GMR - Giant MR:** 6-100 % change in multilayer structures (FM/NM/FM). Due to spin-dependent electron scattering at interfaces.
- **CMR - Colossal MR:** >100 % change in perovskite manganites (e.g., $La_{1-x}Sr_xMnO_3$).
  Linked to double-exchange interaction and phase transitions.
- **TMR - Tunnel MR:** Up to 50 % change in magnetic tunnel junctions (FM/insulator/FM).
  Arises from spin-polarized tunneling; depends on relative magnetization angle.

Applications and Modern Extensions of Magnetoresistive Sensors

From Physical Effect to Engineering Application

- **Anisotropic Magnetoresistance (AMR):**
  - Used in **position, speed, and angle sensors**.
  - Common in automotive wheel speed and camshaft sensors.
  - Offers linear output (under certain conditions) and low cost.
- **Giant Magnetoresistance (GMR):**
  - Utilizes ferromagnetic/non-magnetic multilayers.
  - Found in **read heads of HDDs**, and **current and field sensors**.
  - Change in resistance up to 100%.

Applications and Modern Extensions of Magnetoresistive Sensors contd.

From Physical Effect to Engineering Application

- **Tunnel Magnetoresistance (TMR):**
  - Based on electron tunneling through thin insulator layers.
  - High sensitivity and temperature stability.
  - Used in **precision angle, torque, and robotics sensors**.

Emerging Trends:

- Integration with CMOS for miniaturized smart sensors.
- Use in magnetic memory (MRAM) and biomedical field mapping.
- Enhanced designs with spintronic materials for ultra-low field detection.

Magnetostrictive Effect

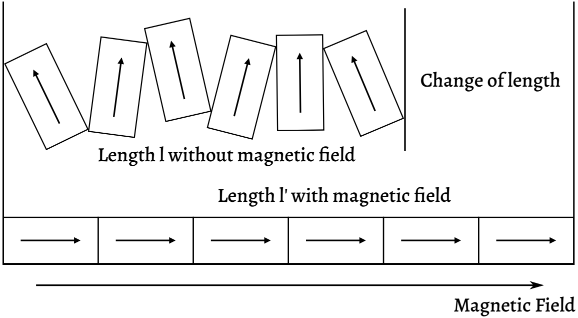
Magnetostrictive Effect: Principle of Operation

Concept: When a magnetic field is applied to a ferromagnetic material, its length changes by Δl. This phenomenon is called **magnetostriction**.

Key Relations:

$$\varepsilon = \frac{\Delta l}{l} = \kappa^* \frac{H}{E}$$

where κ^* = magnetostriction constant, H = magnetic field strength, and E = modulus of elasticity.

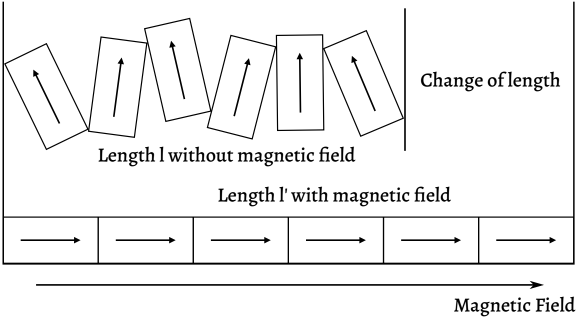

Change of length
Length l without magnetic field
Length l' with magnetic field
Magnetic Field

Magnetostrictive Effect: Length Change under Magnetic Field

Magnetostrictive Effect: Principle of Operation

Physical Description:

- The effect arises from the rotation of **Weiss domains** aligning with the external magnetic field.
- The material volume remains constant — only shape changes.
- Increase in length → *positive magnetostriction*; decrease → *negative magnetostriction*.
- Typical change: -3×10^{-5} m (Ni) to $+5 \times 10^{-5}$ m (Fe).


Change of length
Length l without magnetic field
Length l' with magnetic field
Magnetic Field

Magnetostrictive Effect: Length Change under Magnetic Field

Types of Magnetostrictive Effects

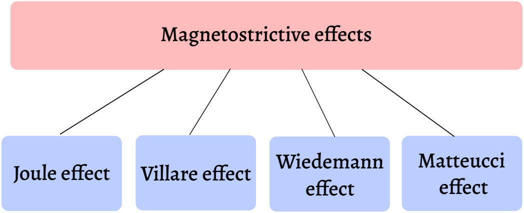
Classification of Magnetostrictive Effects:

Joule Effect

- Change in length or volume under applied magnetic field.
- Discovered first; forms the basis of magnetostriction.
- Generates longitudinal strain — used in ultrasonic transducers.

Villari Effect

- Inverse of Joule effect.
- Magnetic properties vary with mechanical stress.
- Enables stress and force measurement.



Types of Magnetostrictive Effect

Types of Magnetostrictive Effects

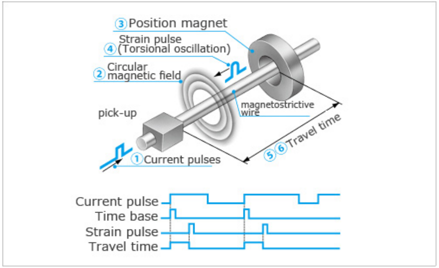
Classification of Magnetostrictive Effects:

Wiedemann Effect

- Torsional motion due to spiral magnetic field.
- Used in magnetostrictive position sensors.

Matteucci Effect

- Inverse of Wiedemann effect.
- Torsional force induces a spiral magnetic field.
- Used for torque measurement.



Wiedemann Effect in a level sensor, source: Nohken Automation, Products

Magnetostriction in Different Materials

Magnitude of Magnetostriction depends on alloy composition and domain mobility.

Typical Materials:

Material	**Max. Length Change** $\Delta l/l \times 10^{-6}$	**Effect Type**
$\text{Fe}_{90}\text{Al}_{4}$	24	Joule
$\text{Fe}_{84}\text{Al}_{16}$	86	Joule
$\text{Fe}_{60}\text{Co}_{40}$	147	Joule
$\text{Fe}_{83}\text{Ga}_{17}$	207	Joule
$\text{FeTb}_{0.3}\text{Dy}_{0.7}\text{Fe}_{2}$ (Terfenol-D)	1600	Joule
$\text{Fe}_{55}\text{Pd}_{45}$	148	Wiedemann
$\text{Fe}_{70}\text{Pd}_{30}$	12,000	Wiedemann
NiMnGa	50,000	Wiedemann

Observation:

- **Terfenol-D** exhibits the highest practical magnetostriction.
- Shape-memory alloys (e.g., NiMnGa) show giant magnetostriction suitable for actuation.

Advantages of Magnetostrictive Sensor Technology

Key Features:

- **Contactless operation** - wear- and maintenance-free.
- **Absolute measurement** - no need for reference points.
- High accuracy (up to 1 µm) and excellent repeatability (up to 0.001% or 4 µm).
- Real-time position measurement across multiple points (up to 20 simultaneously).
- High resistance to mechanical shock, temperature variation, and vibrations.
- Insensitive to dirt, moisture, and environmental interference.
- Quick installation; no calibration required.

Typical Use Cases:

- Machine tools, robotics, and mobile machinery.
- Medical and process automation equipment.

Applications of Magnetostrictive Sensors

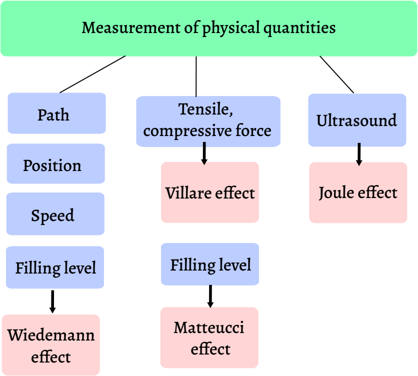
Physical Quantities Measured:

- **Displacement, position, and level:** via Wiedemann effect.
- **Tensile/compressive forces:** via Villari effect.
- **Torque:** via Matteucci effect.
- **Ultrasound generation:** via Joule effect.

Principle of Measurement:

- A magnetic field pulse generates a **torsional wave** in a magnetostrictive waveguide.
- The wave travels at ≈ 2800 m/s.
- The time difference between excitation and reception gives position:

$$\text{Distance} = \text{Speed} \times \text{Time}.$$



Measurable quantities with magnetostrictive effect

Inductive Effect

Inductive Effects: Principle of Operation

Key Inductive Phenomena in Sensors:

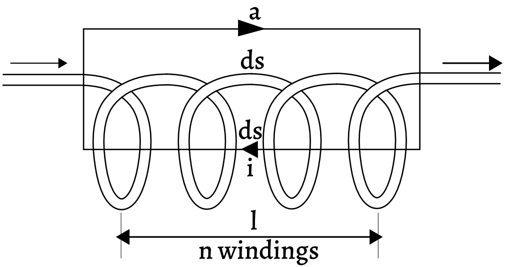
- **Law of Electromagnetic Induction:** Changing magnetic flux induces a voltage.
- **Inductance Behavior:** Variation of inductance in AC circuits provides a sensing mechanism.

Law of Induction:

$$U_{\text{ind}} = -\frac{d\Phi}{dt}$$

where Φ is the magnetic flux through area A:

$$\Phi = \int B\, dA$$

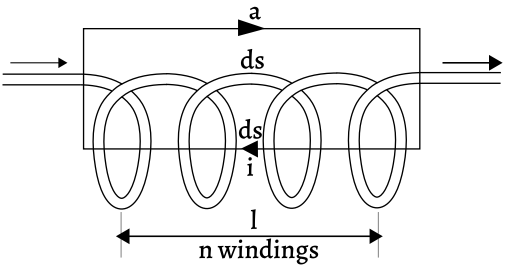


Inductive Effect: Changing Magnetic Flux Induces Voltage

Inductive Effects: Principle of Operation

Interpretation:

- Induced voltage depends on rate of flux change.
- Φ varies if:
  - Magnetic field B changes in magnitude or direction, or
  - Area A permeated by field changes with time.



Inductive Effect: Changing Magnetic Flux Induces Voltage

Magnetic Field and Coil Inductance

Magnetic Field in a Coil:

$$B = \mu_0 \mu_r \frac{NI}{\ell}$$

where μ_0 = magnetic constant, μ_r = relative permeability, N = number of turns, I = current, and ℓ = length of coil.

Flux Linkage and Inductance:

$$\Phi = BA = \mu_0 \mu_r \frac{NIA}{\ell}, \quad L = \frac{N\Phi}{I} = \mu_0 \mu_r \frac{N^2 A}{\ell}$$

Magnetic Resistance:

$$R_{\text{magn}} = \frac{\ell_m}{\mu_0 \mu_r A}$$

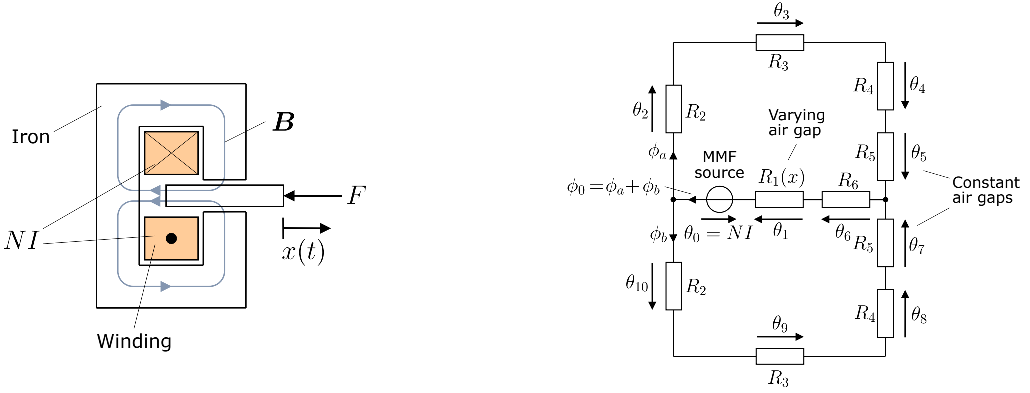
Inverse of permeability — analogous to resistance in an electrical circuit.

Comparison: electric and magnetic network quantities

Electric network			**Magnetic network**		
Voltage	$u = \int \boldsymbol{E} \cdot \mathrm{d}\boldsymbol{s}$	V	Magnetomotive force	$\theta = \int \boldsymbol{H} \cdot \mathrm{d}\boldsymbol{s}$	A
Electric field	$\boldsymbol{E}$	$\frac{\mathrm{V}}{\mathrm{m}}$	Magnetic field	$\boldsymbol{H}$	$\frac{\mathrm{A}}{\mathrm{m}}$
Current	i	A	Magnetic flux	ϕ	Vs
Resistance	R	Ω	Reluctance	R	$\frac{1}{\mathrm{H}}$
Conductance	G	S	Permeance	Λ	H
Conductivity	σ	$\frac{\mathrm{S}}{\mathrm{m}}$	Permeability	μ	$\frac{\mathrm{H}}{\mathrm{m}}$
Ohm's law	$u = Ri$		Hopkinson's law	$\theta = R\phi$	
Kirchoff's first law	$\sum i_k = 0$		Equivalent first law	$\sum \phi_k = 0$	
Kirchoff's second law	$\sum u_k = 0$		Equivalent second law	$\sum \theta_k - \theta_0 = 0$	

Electric and magnetic network quantities and their analogies

Magnetic network example: simple magnetic actuator



(a) Simple magnetic actuator

(b) Magnetic network representation of the actuator

Example for a simple magnetic actuator and its magnetic network representation (adapted from J. Böcker, Mechatronics and Electrical Drives, CC BY-NC-ND)

Eddy currents

- A changing magnetic field induces a voltage.
- In bulky conductive materials (e.g., electromagnetic steel) this voltage drives currents called eddy currents.
- Eddy currents lead to energy losses and heat dissipation.
- To reduce eddy currents, laminated cores are used as they decrease the effective current path width and, therefore, increase the effective resistance per sheet.

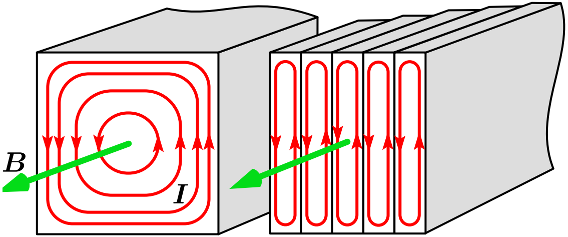


Fig. 2.2: Eddy current formations in solid and laminated steel cores (source: Wikimedia Commons, Chetvorno, CC0)

Inductive Speed and Rotation Measurement

Principle: A rotating ferromagnetic toothed wheel modulates magnetic flux through a stationary coil or permanent magnet. The changing flux induces a voltage.

Flux Variation: B_0 – DC component + $\hat{B}$– AC component, Z: number of teeth

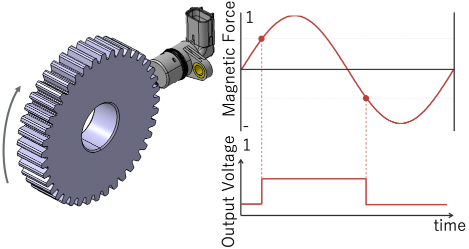
$$B = B_0 + \hat{B}\sin(Z\omega t)$$

Induced Voltage: for effective cross sectional area A

$$u_{\text{ind}} = N\hat{B}AZ\omega\cos(Z\omega t)$$

Key Relationships: K_G – sensor/geometric constant

$$u_S = K_G(B\sin(\omega t))\frac{d\varphi}{dt}, \quad \hat{u}_S = 2\pi K_G B n$$



Inductive speed sensor with toothed wheel modulating magnetic flux (source: Nippon-Seiki Co., Ltd.- Revolution Sensor)

- Output voltage amplitude $\propto$ rotational speed.
- Used in speed, frequency, and position detection.

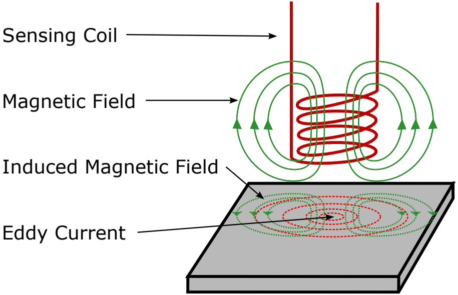
Eddy Current Based Inductive Sensors

Generation of Eddy Currents:

- Alternating magnetic field induces circulating eddy currents in conductive materials.
- Eddy currents produce a secondary magnetic field opposing the original.
- The interaction depends on material position and conductivity.

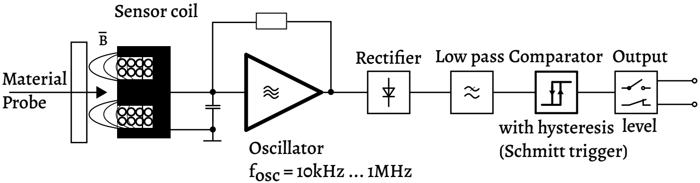
Application:

- Used in contactless position sensors.
- Typical construction:
  - Transmitter coil
  - Receiver coil
  - Conductive rotor or PCB structure



Eddy current sensors (source: Wikimedia Commons, Evan Mason, CC BY-SA 4.0)

Electromagnetic Oscillating Circuits



Inductive sensor

Principle:

- Based on high-frequency **LC resonant circuits**.
- Coil generates an alternating magnetic field at the sensor's **active surface**.
- Presence of metallic objects induces eddy currents that **dampen oscillations**.

Electromagnetic Oscillating Circuits

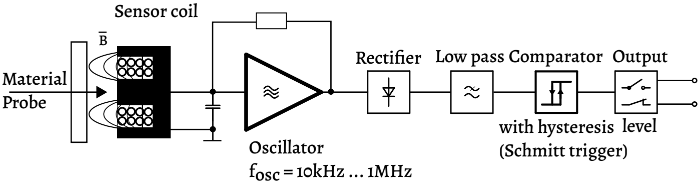


Fig. 2.3: Inductive sensor

Sensor Behavior:

- Two states:
  - Undamped oscillation → high amplitude.
  - Damped oscillation → low amplitude.
- Used as proximity or limit switches.

Note: DIN EN 50010 standardizes sensor characteristics and switching distances.

Advantages of Inductive Sensor Technology

Main Advantages:

- Fully **contactless**, wear- and maintenance-free.
- Operates reliably in harsh industrial environments (dust, oil, moisture).
- High switching frequencies (up to 18 kHz) and repeat accuracy ($< 1\%$).
- Low temperature drift ($< 0.06\%/^\circ$C).
- High EMC and thermal resistance (-60°C to +200°C).
- Large measurement range (0.5 mm – 1.1 m) with linearity up to 0.3%.
- Compact, low power consumption, and long service life.

Standards: Compliant with DIN EN 50010 and widely used in proximity, position, and metal detection systems.

Applications of Inductive Sensor Technology

Application Areas:

- **Forming Technology:** Hammer stroke and forging press monitoring.
- **Process Engineering:** Piston dosing and path measurement.
- **Handling and Welding:** Clamping stroke and welding thickness detection.
- **Mobile Hydraulics:** Excavator boom positioning, electrohydraulic switching.
- **Medical Technology:** Stroke measurement in injection units for microsurgery.

Practical Implementations:

- Screwed proximity switches.
- Embedded limit sensors for automation.

Capacitive Effect

Capacitor and Capacitance: Basic Principle

Definition: A **capacitor** consists of two mutually insulated, oppositely charged conductor surfaces. When a voltage U is applied, charge Q accumulates on the plates, defined by:

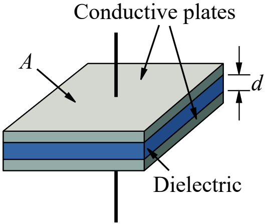
$$C = \frac{Q}{U}$$

The unit of capacitance is the **Farad (F)**.
For a parallel plate capacitor:

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

where:

- ε_0 - electric field constant $(8.854 \times 10^{-12}$ As/Vm)
- ε_r - relative permittivity (dielectric constant)
- A - plate area
- d - distance between plates

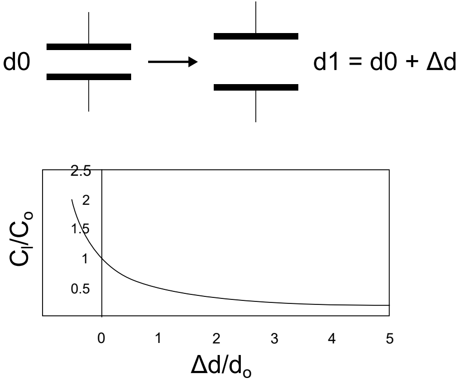


Parallel plate capacitor schematic, source: Wikimedia Commons, Public domain

Parameters Influencing Capacitance

The capacitance of a plate capacitor can be altered through the following parameters:

1. **Plate Distance (d):**
   - Capacitance $\propto 1/d$
   - Nonlinear relation, can be linearized using **differential capacitor circuits.**
2. **Active Area (A):**
   - Capacitance $\propto A$
   - Used in displacement or position sensors where overlap area changes.
3. **Dielectric Permittivity (ε_r):**
   - Capacitance $\propto \varepsilon_r$
   - Depends on the medium between the plates.
   - Used for material, moisture, and level measurement.



Parameters influencing capacitance: (a) plate distance, (b) active area, (c) dielectric permittivity

Permittivity and Dielectric Materials

Permittivity (ε_r) determines how well a material can store electric field energy.

Typical values:

Material	ε_r	**Material**	ε_r
Vacuum, Air	1	Glass, Polyamide	5
Wood (humid)	2-7	Marble	8
Paraffin, Oil	2.2	Al_2O_3	12
Polyethylene	2.3	Alcohol	26
Plexiglass	3.3	Water	81
Quartz Glass	3.7	Ta_2O_5	27
Ceramics (LDC)	10-200	Ceramics (HDC)	10^3-10^4

Observation:

- High $\varepsilon_r \rightarrow$ greater capacitance sensitivity.
- Choice of dielectric depends on required measurement range and sensitivity.

Differential Capacitor Principle

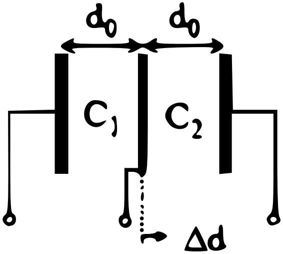
Concept: Two capacitors (C_1, C_2) share a common movable electrode.

Working:

- When the center plate moves by Δd, one capacitance increases (C_1), the other decreases (C_2).
- The differential output $(C_1 - C_2)$ varies linearly with displacement.
- Used for **acceleration and position sensing.**

Acceleration Sensing:

- Movable middle electrode = seismic mass.
- Acceleration changes distance → Δd → measurable ΔC.



Differential capacitor for acceleration measurement

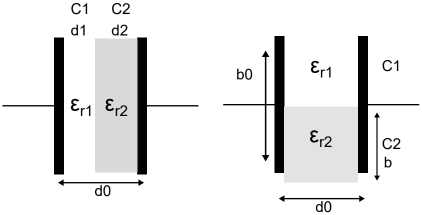
Sensors Based on Dielectric Change

Sensors that measure variations in **permittivity** ε_r are used for:

- Identifying material type (metal, plastic, paper, etc.)
- Counting and sorting materials with different ε_r
- Determining moisture content (air humidity absorbed by polymer dielectric)
- Measuring layer thickness or coatings
- **Level measurement** of solids, liquids, or powders

Working:

- A dielectric with higher ε_r changes the effective capacitance.
- In mixtures (air + material), the capacitance reflects the proportion of each.



Capacitive sensor detecting material presence via dielectric change

Capacitive Level Measurement

Principle:

- Capacitance changes with the **filling level** inside a vessel.
- Higher filling $\rightarrow$ greater ε_r $\rightarrow$ higher capacitance.

Configuration (Nonconductive Material):

- **Active electrode** measures capacitance.
- **Ground electrode** surrounds the active electrode.
- Electric field lines extend into the dielectric (liquid/air mix).

Behavior:

- As filling rises, the effective dielectric constant increases.
- When threshold value is reached, capacitance crosses preset limit $\rightarrow$ output signal.

Conductive and Nonconductive Media Comparison

Nonconductive filling materials:

- Container wall acts as dielectric.
- Electric field partly in wall, partly in filling medium.
- Typical capacitance range: 10^{-14}–10^{-13} F.

Conductive filling materials:

- The filling itself acts as the second capacitor plate.
- When empty $\rightarrow$ low capacitance (large effective d).
- When full $\rightarrow$ high capacitance due to direct conductive coupling.

Applications:

- Immersion sensors in chemical and food industries.
- Level sensing of conductive liquids with high sensitivity.

Capacitance in the AC Circuit

In an AC circuit, a capacitor introduces an **imaginary reactance** X_C given by:

$$X_C = -j\frac{1}{\omega C}$$

Substituting $C = \varepsilon_0 \varepsilon_r \frac{A}{d}$ gives:

$$X_C = -j\frac{d}{\omega \varepsilon_0 \varepsilon_r A}$$

Interpretation:

- Reactance X_C changes proportionally with plate distance d.
- Thus, capacitance sensors can detect distance or approach of objects.

Capacitance in the AC Circuit- Capacitive Sensor

Sensor structure:

- RC oscillator circuit with sensor electrodes.
- Comparator and amplifier generate a switching output.

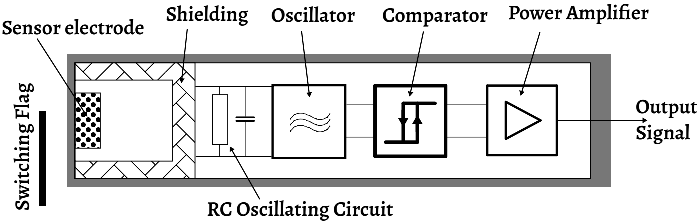


Fig. 2.4: Capacitive proximity sensor circuit

Principle of Capacitive Proximity Sensing

When a metallic or non-metallic object (switching flag) enters the **active electric field zone**, it alters the effective capacitance, changing the RC oscillator amplitude.

Operation modes:

- **Open capacitor (no object)** - large air gap, low capacitance, no oscillation.
- **Object close (switching lug)** - gap decreases, capacitance rises, oscillator starts oscillating.
- Comparator detects oscillation amplitude $\rightarrow$ switching signal.

Switching distance:

- Rated distance: s_n - distance at which the sensor switches.
- Safe distance: $s_a =$ correction factor $\times\, s_n$.

Advantages and Limitations of Capacitive Sensors

Advantages:

- Contactless, wear-free, non-reactive operation.
- Robust, reliable, sweat-proof, bounce-free signal.
- High repeatability (2–5% of real distance).
- Excellent thermal stability (5×10^{-6}/K).
- Nanometer-scale resolution (0.01 nm).
- High EMC resistance, low power, long service life.
- Adaptive and programmable, cost-effective.

Limitations:

- Dust and moisture may affect performance (compensated by design).
- Limited switching distance (15–50 mm typical).

Applications of Capacitive Sensor Technology

Mechanical / Automation:

- Detection through glass/plastic walls.
- Presence check, position, path, and distance measurement.
- Monitoring material thickness and roundness.
- Measurement of acceleration, vibration, and tool gaps.

Food / Pharmaceutical / Chemical / Medical:

- Filling level of solids, powders, or liquids (PVC, water, oils, acids).
- Detects viscous media: glues, disinfectants, blood.

Semiconductor industry:

- Acid level control in wafer processing.
- Wafer presence and alignment checks.

Special Applications:

- PC touchpads, smartphones, portable media, humidity detection.

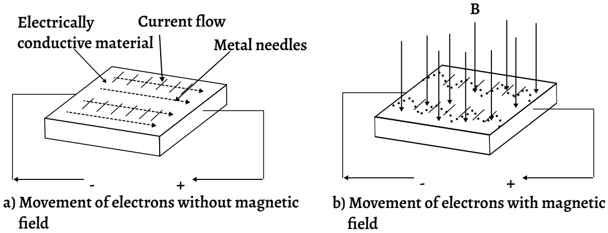
Gaussian Effect

Gaussian Effect: Principle of Operation

Definition: The **Gaussian Effect** is a *galvanomagnetic effect* that occurs when a current-carrying conductor or semiconductor is placed in a magnetic field $\mathbf{B}$ that is **not parallel** to the current flow.

Physical basis:

- A **Lorentz force** $\mathbf{F_L}$ acts on moving charge carriers.
- This deflects electrons, causing them to follow a **zigzag path**.
- The effective path length of electrons increases → **increased resistance**.


Electrically conductive material
Current flow
Metal needles
-
+
B
-
+
a) Movement of electrons without magnetic field
b) Movement of electrons with magnetic field

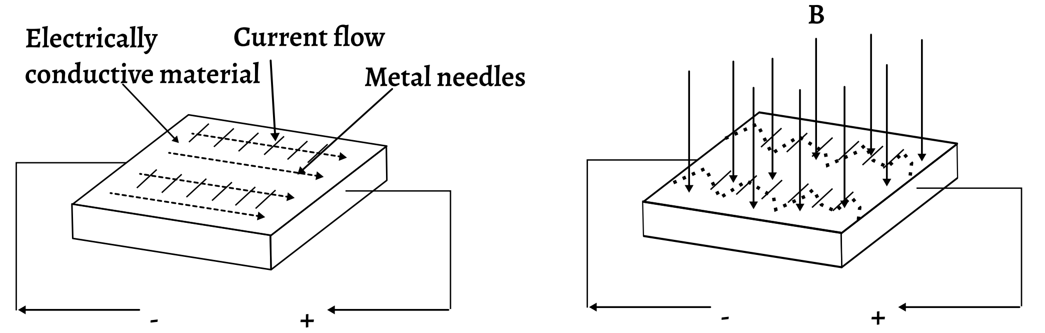
Origin of Gaussian effect: electron deflection in magnetic field

Gaussian Effect: Principle of Operation

Lorentz Force:

$$\mathbf{F_L} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B})$$

Ohm's Law: $\mathbf{j} = \sigma\mathbf{E}$, where $\mathbf{j}$ is current density and σ is electrical conductivity.



a) Movement of electrons without magnetic field

b) Movement of electrons with magnetic field

Origin of Gaussian effect: electron deflection in magnetic field

Microscopic Description of the Gaussian Effect

Microscopic relations:

$$\sigma = qn\mu$$

where:

- q - charge of carrier, n - carrier density, μ - mobility of charge carriers.

Current density:

$$\mathbf{j} = \rho\mathbf{v}, \quad \rho = qn$$

Drift velocity:

$$\mathbf{v} = \mu\mathbf{E}$$

Force balance:

$$\mathbf{F} = q\mathbf{E} \quad \Rightarrow \quad \mathbf{v} = \frac{\mu\mathbf{F}}{q}$$

Interpretation: Electron drift depends on both $\mathbf{E}$ and $\mathbf{B}$, leading to altered trajectories and measurable resistance variation.

Derivation of Conductivity Change in Magnetic Field

When $\mathbf{B}$ is perpendicular to the electron motion:

$$\mathbf{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \mu \begin{pmatrix} E + v_y B \\ -v_x B \end{pmatrix}$$

Solving for v_x:

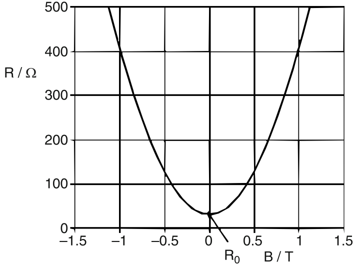
$$v_x = \frac{\mu E}{1 + \mu^2 B^2}$$

Implication:

- Drift velocity **decreases** as B increases.
- Current density $j = \rho v_x$ also decreases.
- Therefore, conductivity σ decreases, and resistance increases.

Resulting resistance relation:

$$R(B) = R_0 \left(1 + \mu^2 B^2\right)$$



Variation of resistance with magnetic flux density due to Gaussian effect

Conclusion: Resistance increases quadratically with magnetic flux density.

Application of the Gaussian Effect

1. Magnetic Field Measurement:

- Used in **Magneto-Resistive Sensors (MDRs)** and field plates.
- Resistance R varies with $\mathbf{B} \rightarrow$ can be calibrated for field strength.

2. Field Plate Construction:

- A ceramic substrate with a thin semiconductor layer (e.g., InSb).
- Embedded metallic needles (e.g., NiSb) guide current paths.
- Magnetic flux density B perpendicular to plate alters resistance.

3. Typical properties of field plates (Table 2.13):

- Output resistance R_0: 10 Ω - 10 kΩ.
- Temperature coefficient $\alpha \approx -0.004\,\mathrm{K}^{-1}$.
- Max operating temperature $\approx 95^\circ$C.
- Max power $\approx$ 0.5 W.

Working Mechanism Summary

Electron motion:

- Without magnetic field → straight-line conduction.
- With magnetic field → Lorentz force causes zigzag trajectory.

Effect on resistance:

- Longer electron paths increase collision frequency.
- Effective resistivity rises proportionally to B^2.

Materials used:

- High-mobility semiconductors: InSb, InAs, GaAs, Si.
- Often doped with Te to adjust resistive response.
- Typically exhibit **negative temperature coefficients.**

Summary: Gaussian Effect in Sensors

Key outcomes:

- Magnetic field $\mathbf{B}$ increases resistance quadratically.
- Based on Lorentz force and carrier deflection.
- Enables non-contact magnetic field and current measurement.

Applications:

- Magnetic field and flux density measurement.
- Current, angle, and rotational speed sensing.
- Field plates and magneto-resistive devices.

Advantages:

- Compact and solid-state design.
- Fast response and high sensitivity.
- Suitable for DC and low-frequency magnetic field detection.

Hall Effect

Hall Effect: Principle of Operation

The **Hall Effect**, like the Gaussian Effect, belongs to the *galvanomagnetic effects*.

Concept:

- When a current-carrying conductor or semiconductor is placed in a **magnetic field** $\mathbf{B}$ with a perpendicular component, charge carriers experience a **Lorentz force**.
- This deflects the electrons, creating a potential difference across the conductor — the **Hall voltage** U_H.

Lorentz force:

$$F_L = -qv_xB_z$$

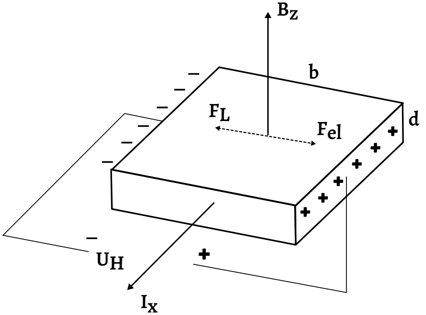
Electric counterforce at equilibrium:

$$F_{el} = -qE_y \quad \text{and} \quad -qE_y = -qv_xB_z$$

Therefore:

$$E_y = v_xB_z$$

$$U_H = E_yb = v_xB_zb$$



Hall effect: charge carrier deflection and Hall voltage generation

Mathematical Description of Hall Voltage

Current density relation:

$$j_x = nev_x$$

where:

- n – number of charge carriers per unit volume,
- e – elementary charge,
- v_x – carrier velocity in x-direction.

Hall voltage:

$$U_H = \frac{1}{ne} j_x B_z b$$

Defining the **Hall coefficient:**

$$A_H = \frac{1}{ne}$$

Then:

$$U_H = A_H j_x B_z b$$

Hall resistance:

$$R_H = \frac{A_H B_z}{d} \quad \Rightarrow \quad U_H = R_H I_x$$

Physical Interpretation

Key points:

- A magnetic field deflects charge carriers (electrons or holes) perpendicular to the current.
- The resulting charge separation builds up a Hall voltage U_H.
- At equilibrium, the Lorentz force equals the electric force.

Dependence:

- $U_H \propto I_x B_z$
- $U_H \propto 1/(ne)$ — inversely proportional to carrier density.
- Independent of the specific resistivity of the material.

Implications:

- Semiconductors produce a much larger U_H than metals.
- Sign of U_H indicates whether conduction is via electrons or holes.

Hall Coefficients for Different Materials

Typical Hall coefficients (A_H) in 10^{-11} m^3/C:

Material	A_H	Material	A_H
Electron conductors:		*Hole conductors:*	
Copper (Cu)	-5.5	Cadmium (Cd)	$+6$
Gold (Au)	-7.5	Tin (Sn)	$+14$
Silver (Ag)	-8.4	Beryllium (Be)	$+24.4$
Sodium (Na)	-25		
Cesium (Cs)	-28		
Semiconductors:			
Bismuth (Bi)	-5×10^4	Silicon (Si), Germanium (Ge)	10^8–10^{10}
Indium antimonide (InSb)	-2.4×10^7	Indium arsenide (InAs)	-10^7

Observation: Semiconductors have high A_H due to higher carrier mobility μ and lower carrier density n.

Applications of the Hall Effect

Basic sensing principle:

- A magnetic field perpendicular to the Hall sensor produces a transverse voltage U_H.
- The Hall voltage is proportional to $I_x B_z$.

Applications:

- Measurement of **magnetic fields** and **currents**.
- **Position and speed** detection (rotational or linear motion).
- Contactless signal transmission and switching.
- Detection of coating thickness or conductive layers.

Advantages:

- Compact, robust, and can be **integrated into semiconductor chips**.
- High sensitivity, low cost, suitable for DC and low-frequency magnetic fields.

Automotive and Industrial Applications

In automotive systems:

- Pedal position sensing (e.g., brake or accelerator).
- Ignition timing control.
- Belt buckle or handbrake detection.
- Rotational speed measurement (wheel or motor speed).

Advantages over inductive sensors:

- Operates even at zero or very low rotational speeds.
- Output signal independent of rotational speed.
- Provides digital square-wave output — easily processed by control electronics.

Common shapes:

- Rectangular, cross, and butterfly designs.

Design optimization:

- 3D field simulations (FEM) used for optimal sensor layout and magnetic coupling.

Summary: Hall Effect Sensors

Operating principle:

- Based on Lorentz-force-induced charge separation.
- Generates measurable Hall voltage U_H proportional to magnetic field B_z.

Key relations:

$$U_H = \frac{1}{ne} j_x B_z b = A_H j_x B_z b$$

$$A_H = \frac{1}{ne}, \quad R_H = \frac{A_H B_z}{d}$$

Applications:

- Magnetic field, current, and position sensing.
- Motion and speed detection.
- Integrated electronics for feedback and control systems.

Advantages:

- Linear response, compact design, compatible with CMOS integration.
- High sensitivity and robust operation in harsh environments.

Thermoelectric Effect

Thermoelectric Effect – Principle

Definition: When two dissimilar conductors are joined at both ends and exposed to different temperatures (T_{AB}, T_{BA}), a voltage U_{th} is generated due to charge carrier diffusion.

$$\Delta T = T_{BA} - T_{AB}$$

This phenomenon is known as the **Seebeck Effect**.
Physical explanation:

- Thermal energy of a metal is stored partly as atomic oscillation and partly as electron kinetic energy.
- Electrons at the hotter end gain higher kinetic energy and diffuse toward the colder end.
- This diffusion creates a potential difference — the **thermoelectric voltage**.

Seebeck Coefficient and Its Origin

Theoretical Seebeck coefficient of material A:

$$S'_A = -\frac{1}{3e}\frac{d}{dT}\left\langle \frac{1}{2}mv^2 \right\rangle$$

where:

- e – electron charge,
- T – absolute temperature,
- $\langle \frac{1}{2}mv^2 \rangle$ – average kinetic energy of conduction electrons.

Simplified expression (for metals):

$$S'_A = -\frac{\pi^2 k_B^2 T}{2eE_F}$$

where k_B is the Boltzmann constant and E_F the Fermi energy.

Interpretation:

- S'_A is material-dependent.
- It is positive for *p-type* materials (hole conduction) and negative for *n-type*.

Thermoelectric Voltage Relation

When conductors A and B are connected at two points with different temperatures T_{AB} and T_{BA}:

$$U_{th} = \int_{T_{AB}}^{T_{BA}} S_{AB}(T)\, dT$$

where:

$$S_{AB}(T) = S_A(T) - S_B(T)$$

is the **differential Seebeck coefficient** between materials A and B.
Key features:

- U_{th} is proportional to the temperature difference ΔT.
- S_{AB} depends on material pair and temperature.
- Platinum is often used as the reference material ($S = 0$).

Material Dependence of the Seebeck Effect

Typical thermoelectric voltages (relative to platinum, 0–100 °C):

- Platinum: 0.0 mV
- Constantan: –3.2 mV
- Nickel: –1.9 mV
- Tungsten: $+0.7$ mV
- Copper: $+0.7$ mV
- Iron: $+1.9$ mV
- Nickel–chrome: $+2.2$ mV

Interpretation:

- The sign indicates current direction (electron or hole conduction).
- Combining two materials (e.g., Cu–Constantan) forms a thermocouple with measurable U_{th}.
- For Cu–Constantan: $0.7 + 3.2 = 3.9$ mV for $\Delta T = 100$ °C.

Standard Thermocouple Types (DIN EN 60584-1)

Type	Materials	Cont. Range (°C)	Short-time (°C)	Remarks
K	Ni-Cr / Ni-Al (“Chromel/Alumel”)	0–1100	–180–1300	Most common; oxidizing atmospheres
J	Fe / Cu-Ni (“Iron/Constantan”)	0–700	–180–800	Dry and reducing atmospheres
N	Ni-Si / Ni-Si-Mg (“Nisil/Nicrosil”)	0–1100	–270–1300	High stability; newer type
R	Pt-13 Rh / Pt	0–1600	–50–1700	High-T; ceramic protection
S	Pt-10 Rh / Pt	0–1600	–50–1750	High-T; ceramic protection
B	Pt-30 Rh / Pt-6 Rh	200–1700	0–1820	Very high T; high-purity ceramic
T	Cu / Cu-Ni (“Copper/Constantan”)	–185–300	250–400	Low to cryogenic T; stable in moisture
E	Ni-Cr / Cu-Ni (“Chromel/Constantan”)	0–800	–40–900	High signal output

Applications of the Thermoelectric Effect

Temperature measurement:

- Measurement of **temperature differences** or absolute temperature (with a known reference).
- Basis of **thermocouples** and **thermopiles**.

Material property measurement:

- Determines conduction type (electron vs. hole).
- Detects chemical or gaseous impurities through **Fermi-level shift**.

Special case - Gas detection:

- Chemisorption modifies charge carrier concentration.
- A fixed temperature gradient produces a voltage change proportional to impurity concentration.

Amplification:

- Thermoelectric voltages are very small (µV–mV range).
- Amplified using low-offset or **chopper-stabilized amplifiers**.

Thermopile Sensors and Heat Radiation Detection

Thermopile:

- Formed by connecting multiple thermocouples in series.
- Increases sensitivity for small temperature differences.

Working principle:

- A thin membrane in the chip center absorbs heat radiation.
- Temperature difference between the center and periphery generates voltage.

Applications:

- Infrared heat detection and non-contact temperature sensing.
- Used in MEMS-based thermopile sensors and pyroelectric detectors.

Summary: Thermoelectric (Seebeck) Effect

Fundamental relation:

$$U_{th} = \int_{T_{AB}}^{T_{BA}} S_{AB}(T)\, dT$$

Key properties:

- S_{AB} depends on materials and temperature.
- U_{th} proportional to temperature difference ΔT.
- Positive for p-type, negative for n-type materials.

Practical relevance:

- Basis for thermocouples (K, J, T, E, etc.).
- Enables temperature and material analysis.
- Can be used in thermopiles for heat radiation sensing.

Advantages:

- Simple, robust, no external power supply required.
- Wide temperature range, fast response, miniaturizable.

Thermoresistance Effect

Thermoresistance Effect — Principle

Concept: The electrical resistance of materials depends on:

- Composition and crystal structure,
- Degree of purity and homogeneity,
- Temperature T.

For metals and metal alloys, resistance increases with temperature:

$$R(T) = R_0 \left[1 + a(T - T_0) + b(T - T_0)^2\right]$$

where:

- R_0 - resistance at reference temperature T_0 (0 °C or 20 °C),
- a - temperature coefficient (linear term),
- b - higher-order coefficient (non-linear term).

Mechanism: Electron scattering increases with atomic vibrations and impurities, raising resistance.

Temperature Coefficient and Platinum Sensors

Temperature coefficient:

$$a = \frac{1}{R_0} \frac{dR(T)}{dT}$$

Typical values:

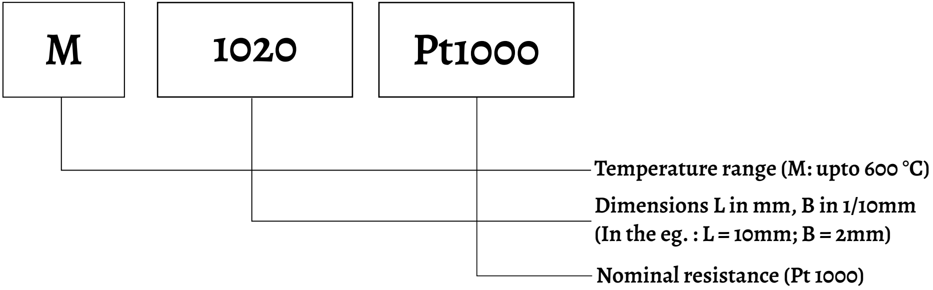
- Platinum (Pt): $3.89 \times 10^{-3}\,\mathrm{K}^{-1}$
- Nickel-Chromium: $(0.02 - 0.05) \times 10^{-3}\,\mathrm{K}^{-1}$

Platinum sensors (Pt 100, Pt 1000):

- Pt 100 → 100 Ω at 0 °C; Pt 1000 → 1000 Ω at 0 °C.
- Highly stable, chemically resistant, and accurate.
- Range: –220 °C to +1000 °C (depending on type).

Equation fit: Linear up to 100 °C; slight deviation above.

Designs and Characteristics of Pt Resistors



M

1020

Pt1000

Temperature range (M: upto 600 °C)

Dimensions L in mm, B in 1/10mm
(In the eg. : L = 10mm; B = 2mm)

Nominal resistance (Pt 1000)

Coding of Pt sensors (source: Heraeus Sensor Technology GmbH)

Properties of Pt thin-film resistors:

- Precise, long-term stable, vibration and shock resistant.
- High resistance to corrosion, humidity, and aggressive chemicals.
- Excellent linearity and standard calibration curves (DIN EN 60751).

Thermal Response and Self-Heating

Thermal response time:

- Time for sensor resistance to change by 50 % ($t_{0.5}$) or 90 % ($t_{0.9}$) of a temperature step.
- Depends on medium (air, water) and sensor design.

Self-heating effect:

$$\Delta T = P \cdot S, \quad P = I^2 R$$

where S is the self-heating coefficient (K/mW). $\rightarrow$ Higher measuring current increases sensor temperature and error.

Measuring currents (Heraeus recommendation):

- Pt 100: 0.3-1.0 mA, Pt 1000: 0.1-0.3 mA.

Applications:

- Automotive, HVAC, industrial process, and life-science temperature sensing.
- High-accuracy measurement (–196 °C to +1000 °C).

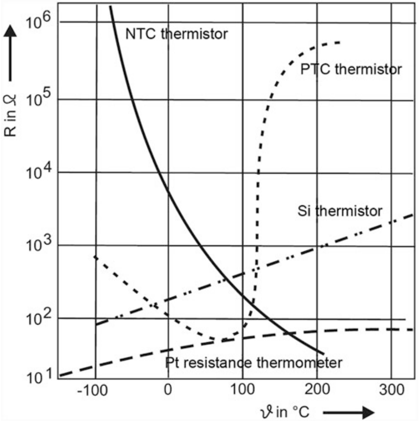
Temperature Effects in Semiconductors

Semiconducting ceramics (thermistors):

- Show much higher temperature dependence of resistance.
- Two types:
  - **PTC** – Positive Temperature Coefficient: $R \uparrow$ with $T \uparrow$
  - **NTC** – Negative Temperature Coefficient: $R \downarrow$ with $T \uparrow$

Behavior:

- Metals (Pt, Ni) → linear positive coefficient.
- Semiconductors → nonlinear high temperature sensitivity.



Resistance vs. Temperature for Pt, PTC, and NTC sensors (source: Hesse, Schnell: Sensors for process and factory automation, Vieweg+Teubner, 2009)

PTC Thermistors (Positive Temperature Coef

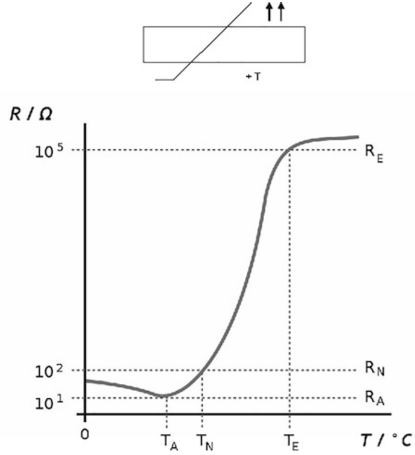
Material: Titanate ceramics - $BaTiO_3$, $SrTiO_3$.
Behavior:

- Below Curie temperature → R decreases (slightly NTC-like).
- Above Curie temperature → R increases exponentially.

$$R(T) = R_N \exp[A(T - T_N)]$$

Typical $A = 0.16\,\mathrm{K}^{-1}$ (≈ 40 × higher than metals).
Applications:

- Temperature compensation and monitoring,
- Over-current and thermal overload protection.



PTC symbol and R-T curve (PTC resistors to DIN 44080)- (source: Ekbert, H., et al.: Sensoren und Messsysteme, 4th ed., Springer Vieweg, 2016)

NTC Thermistors (Negative Temperature Coeffi

Material: Mixed oxide semiconductors (e.g., Fe_2O_3, $MgCr_2O_4$, $Ba(NO_3)_2$).

Behavior:

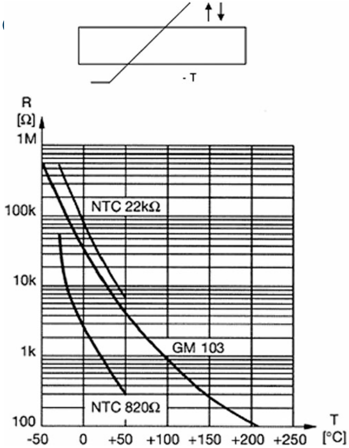
$$R(T) = R_N \exp\left[B\left(\frac{1}{T} - \frac{1}{T_N}\right)\right]$$

$$A = -\frac{B}{T^2}$$

→ Resistance decreases exponentially with temperature.

Applications:

- Temperature measurement, compensation, and voltage regulation.
- Rapid, sensitive response – widely used over PTC types.



NTC symbol and R-T curve (NTC resistors to DIN 44070) - (source: Ekbert, H., et al.: Sensoren und Messsysteme, 4th ed., Springer Vieweg, 2016)

Comparison Summary: Pt, PTC, and NTC Sensors

Property	**Pt Resistor (Metal)**	**PTC Thermistor**	**NTC Thermistor**
Material	Metal / Alloy (Pt, Ni, Cu)	Titanate ceramics ($BaTiO_3$)	Mixed oxides (Fe, Mg, Cr)
Temp. coeff.	Positive, linear	Positive, nonlinear (above Curie T)	Negative, exponential
Accuracy	Very high	Moderate	High (in limited range)
Response time	Moderate	Fast	Very fast
Range (°C)	–220 to +1000	0 to 200	–50 to +250
Applications	Precision thermometers	Over-current, compensation	Temperature sensing, control

Summary: Pt sensors → precision measurement. PTC → protection and switching. NTC → fast thermal sensing.

Photoelectric Effect

Photoelectric Effect - Overview

Definition: The photoelectric effect occurs when a **photon (quantum of light)** is absorbed by an **electron**, releasing it from its bond.

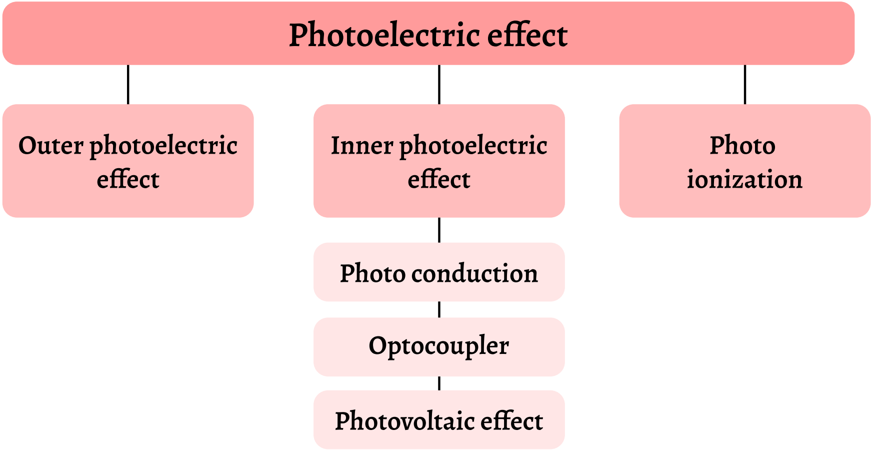
Key Principle:

- A photon transfers its energy to an electron.
- The photon's energy must exceed the electron's **binding energy**.

Types of Photoelectric Effect:

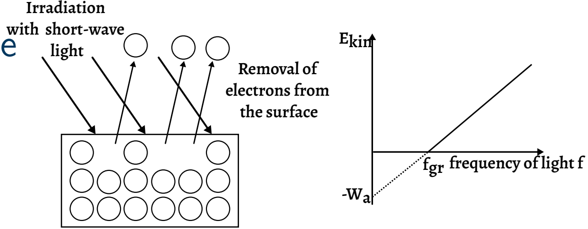
- **Outer photoelectric effect:** Electron emission from metal surfaces.
- **Inner photoelectric effect:** Electron excitation within semiconductors.
- **Photoionization:** Complete removal of an electron from atoms or molecules.

Photoelectric Effect – Overview



Types of photoelectric effect: outer, inner, and photoionization

Outer Photoelectric Effect – Principle


Irradiation with short-wave light
Removal of electrons from the surface
Ekin
fgr
frequency of light f
-Wa

Outer photoelectric effect: photon absorption and electron emission

Mechanism:

- Short-wave electromagnetic radiation (light) strikes a metal surface.
- Electrons are ejected — creating a measurable **photocurrent**.

Energy relationship:

$$E_{\text{kin}} = hf - W_A$$

where

- h – Planck's constant ($6.626 \times 10^{-34}\,\text{Js}$)
- f – frequency of incident light
- W_A – work function (binding energy)

Key Implications:

- No electrons emitted if $hf < W_A$
- **Cut-off frequency:** $f_{\text{gr}} = W_A/h$
- Increasing light intensity increases photocurrent but not electron energy.

Outer Photoelectric Effect – Characteristics

Observations:

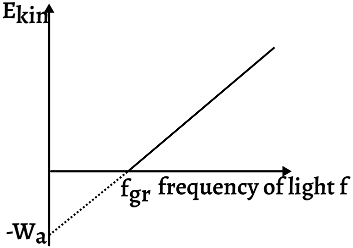
- The **kinetic energy** of photoelectrons depends only on light **frequency**, not intensity.
- The **photocurrent** increases with light intensity.
- There exists a **threshold frequency** below which no emission occurs.

Photon Energy Relation:

$$E_{\text{ph}} = hf = \frac{hc}{\lambda}$$

Thus:

$$E_{\text{ph}} \propto \frac{1}{\lambda}$$



Outer photoelectric effect: photon absorption and electron emission

Inner Photoelectric Effect – Photoconductivity

Mechanism:

- In semiconductors, photons lift electrons from the **valence band** to the **conduction band**.
- This increases the number of free charge carriers → higher conductivity.

Energy Condition:

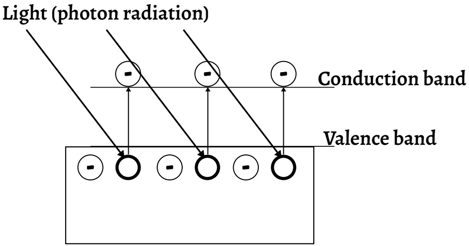
$$E_{\mathrm{ph}} = hf = \frac{hc}{\lambda} \geq E_g$$

or equivalently:

$$\lambda \leq 1.24\,\mu\mathrm{m{\cdot}eV}/E_g$$

Band Model Explanation:

- Photons with $E_{\mathrm{ph}} > E_g$ generate electron-hole pairs.
- More photons → higher conductivity (photoconduction).



Inner photoelectric effect: photon absorption and electron excitation in semiconductors

Applications of Photoelectric Effect

Practical Uses:

- **Photon detectors** (photomultipliers, photodiodes)
- **Optocouplers** for galvanic isolation
- **Light barriers and light curtains** for automation
- **Light and color sensors** for measurement and control
- **Photovoltaic cells** (solar energy conversion)

Summary:

- Outer effect → metal emission and photocurrent.
- Inner effect → semiconductor excitation and increased conductivity.

Optical Effect

Optical Effects – Physical Basis

Light as an Electromagnetic Phenomenon:

- Light exhibits both **wave** and **particle** characteristics.
- As a wave → described by wavelength λ, frequency f, and velocity c:

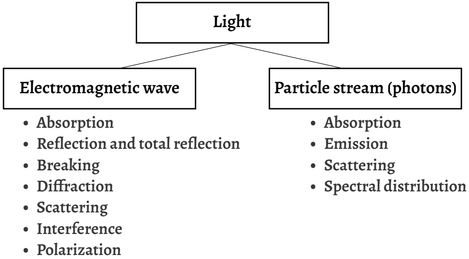
$$c = \lambda \cdot f$$

- As a particle → light consists of **photons** carrying quantized energy:

$$E_{\text{ph}} = h \cdot f$$

Visible Light Spectrum:

- λ range: 380-780 nm, f range: 3.84×10^{14}-7.89×10^{14} Hz.
- Light can be absorbed, reflected, or refracted.



Light

Electromagnetic wave

- Absorption
- Reflection and total reflection
- Breaking
- Diffraction
- Scattering
- Interference
- Polarization

Particle stream (photons)

- Absorption
- Emission
- Scattering
- Spectral distribution

Wave and particle characteristics of light

Wave and Particle Characteristics of Light

Wave Nature:

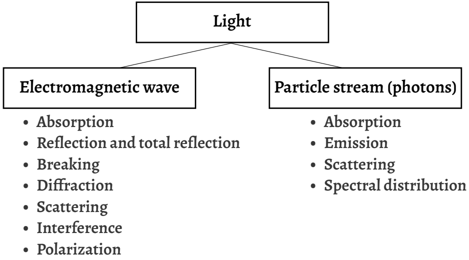
- Involves reflection, refraction, diffraction, interference, and polarization.
- Described by the oscillation of electric (E) and magnetic (H) fields.

Particle Nature:

- Light behaves as a stream of photons with discrete energy packets.
- Responsible for photoelectric and photoconductive effects.

Applications:

- Wave properties → optics, lenses, fiber optics.
- Photon properties → photodiodes, photoelectric sensors, lasers.



Wave and particle characteristics of light

Refraction and Reflection of Light

Refraction: When light passes from one medium to another, its speed and direction change.

$$\frac{\sin \varepsilon}{\sin \beta} = \frac{c}{c'} = \frac{n'}{n}$$

where $n =$ refractive index, $c =$ speed of light in vacuum, $c' =$ in medium.

Snell's Law:

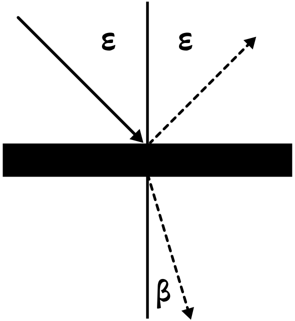
$$n = \frac{c_0}{c}$$

Reflection:

- Angle of incidence = angle of reflection.
- Total internal reflection occurs if angle of incidence > critical angle.

Applications:

- Optical fibers utilize total internal reflection to guide light efficiently.



Refraction and reflection of light at an interface

Optical Sensors – Principle of Operation

Basic Components:

- **Transmitter:** light source (e.g., LED, laser).
- **Transmission medium:** air, glass, or fiber.
- **Receiver:** converts light signal into electrical signal (e.g., photodiode).

Working Concept:

- Based on absorption, reflection, or refraction of light.
- Used for object detection, distance sensing, and imaging.

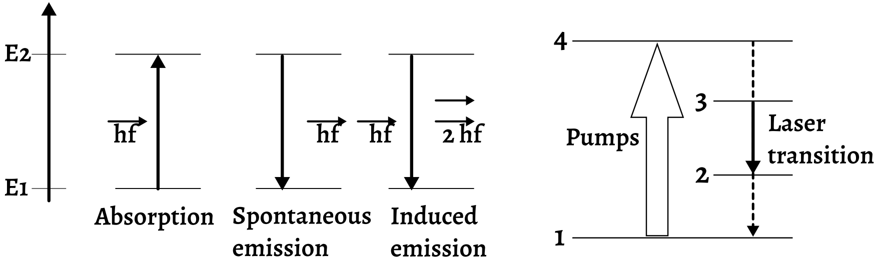
Equation of photon energy:

$$E_{\mathsf{ph}} = h \cdot f = \frac{h \cdot c}{\lambda}$$

Examples:

- Photoelectric sensors, optical encoders, light barriers, and camera modules.

Laser Principle and Optical Emission



Absorption, spontaneous emission, and laser principle

Photon-Energy Interaction:

- **Absorption:** photon excites electron from $E_1 \rightarrow E_2$.
- **Spontaneous emission:** electron falls from $E_2 \rightarrow E_1$, emitting one photon.
- **Stimulated emission:** incident photon induces another photon of same phase $\rightarrow$ amplification.

Laser Principle and Optical Emission

Laser (Light Amplification by Stimulated Emission of Radiation):

- Coherent and monochromatic light output.
- Used in measurement, identification, and communication.

Energy relationship:

$$E_{\mathsf{ph}} = h \cdot f$$

Summary of physical effects used in sensors

- Lits of all the physical effects used in sensors were covered in this lecture series.
- Each effect has unique principles, equations, and applications.
- Understanding these effects is crucial for sensor design and application.
- This knowledge enables the development of advanced sensing technologies across various fields.
- Further reading and exploration of each effect can lead to innovative sensor solutions.
- Try to visualize how these effects can be combined in multi-functional sensors. Maybe you come up with new ideas!

Table of contents

Signal Processing and Conditioning

Dr Bikash Sah

Universität Siegen / IAS Interconnected Automation Systems

Measurement and Signal Processing

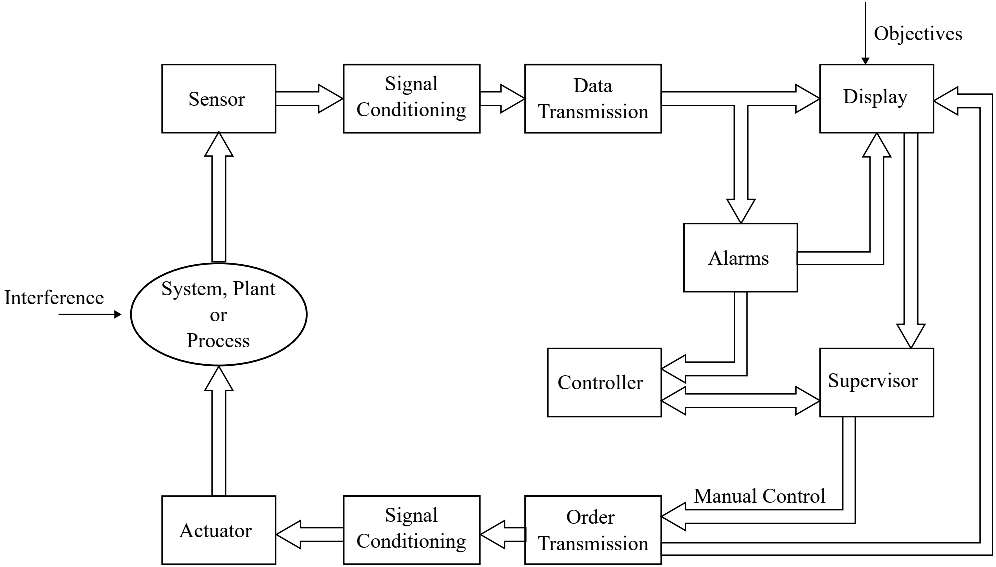
Goal: Transform a physical phenomenon into a measurable and interpretable electrical signal.

- Measurement = objective assignment of numerical value to a property.
- Signal processing ensures that sensor outputs are usable by controllers or digital systems.
- Conditioning bridges the physical world and the information world.

Core Idea

Sensors ⇒ Signal Conditioning ⇒ Data Conversion ⇒ Processing ⇒ Display/Control

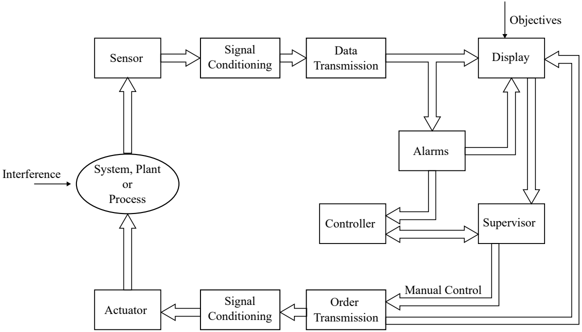
Elements of a Measurement System



Elements of a Measurement System

Subsystems:

1. **Sensor / Transducer** - converts physical to electrical signals.
2. **Signal Conditioning** - amplifies, filters, linearizes.
3. **Data Transmission** - carries signals to control or display unit.
4. **Controller / Processor** - interprets and acts upon data.
5. **Actuator** - applies control action to the system or process.



Transducers, Sensors, and Actuators

Transducer: Converts energy from one form to another.
Sensor: Input transducer–detects a physical variable and converts it to an electrical signal.
Actuator: Output transducer–converts electrical signals into mechanical motion or other actions.

Key Concepts

- Always some energy flow from measured system to sensor $\Rightarrow$ avoid loading effect.
- Six domains: mechanical, thermal, magnetic, electrical, chemical, radiation.
- Sensors with electric outputs dominate modern systems.

Advantages of Electrical Measurement Systems

1. Electric parameters vary with almost any physical quantity (temperature, strain, pressure).
2. Signal energy can be amplified – minimal disturbance to measured system.
3. Wide availability of integrated circuits for conditioning and processing.
4. Data can be easily recorded, transmitted, and displayed electronically.
5. Electrical signals are versatile and compatible with digital systems.

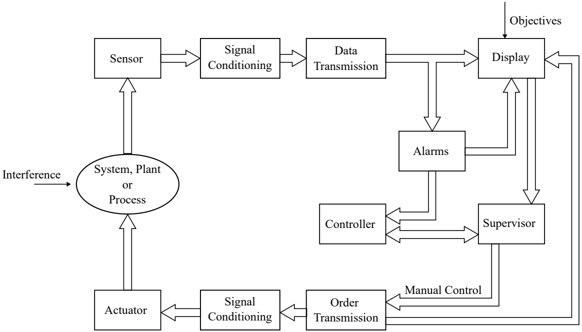
Primary sensor: interacts with the measurand (e.g., diaphragm senses pressure).
Secondary sensor: converts the effect into an electrical signal (e.g., strain gauge).

- Example: Pressure → diaphragm deformation → resistance change → voltage signal.
- Full sensor assembly includes sensing element + package + leads.
- The raw signal typically has low amplitude and needs conditioning.

Signal Conditioning

Signal conditioning: prepares sensor output for transmission, digitization, or display.

- Functions: amplification, filtering, impedance matching, level shifting, modulation/demodulation.
- Converts weak analog signals (mV range) to standard ranges (± 10 V or 4–20 mA).
- Analog-to-digital converters (ADCs) need well-conditioned inputs.
- Final display may be analog (pointer, gauge) or digital (numeric, bar graph).



Interfaces and Data Domains

Measurement systems involve multiple data domains:

- Domains describe how information is represented or transmitted.
- Interface stages (ADC, DAC) convert between domains.

Common domains:

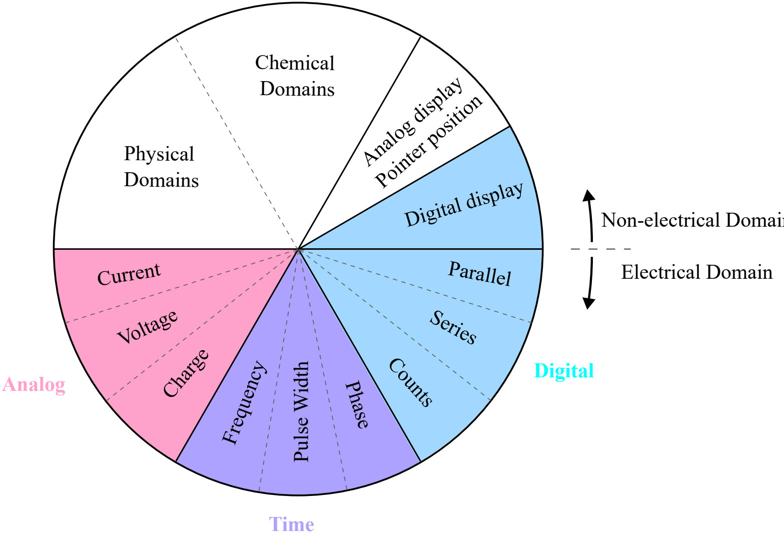
- **Analog:** signal amplitude represents information (voltage, current).
- **Time:** information encoded in timing (frequency, phase, pulse width).
- **Digital:** information represented numerically (binary codes, counts).
- **Chemical/Physical:** domains prior to conversion to electrical form.

Data Domain Representation

Information carriers:

- **Analog domain:** amplitude (voltage/current).
- **Time domain:** period, frequency, or phase.
- **Digital domain:** pulse count, word code.

Each stage in the measurement chain may operate in a different domain, requiring proper interfacing and conversion.



Direct Adjustment Indirect Measurements

Direct measurement:

- Comparison with a reference standard (e.g., weighing scale, voltmeter).

Indirect measurement:

- Computed from other quantities using physical laws.
- Examples:
  - Power $=$ Voltage $\times$ Current
  - Distance $= \int$ speed dt

Most modern measurements are indirect and require signal processing.

Summary and Context for Next Topics

You have learned:

- The structure of a measurement and control system.
- Roles of transducers, sensors, actuators, and conditioners.
- The flow of information through different domains.

Next lectures:

- Resistive and bridge circuits for signal generation.
- Operational amplifiers in conditioning.
- ADCs, DACs, sampling, and quantization.

Overview of Signal Processing and Calibration

Goal: Convert a physical quantity/phenomenon into a usable electrical/digital signal decision

- Sensors generate low-level, often non-linear, analog outputs.
- Signal processing and conditioning amplify, filter, linearize, and standardize them.
- Calibration ensures accuracy and removes offset, drift, and disturbances.

Key challenges:

1. Very small signal levels → require amplification.
2. Influence of disturbance variables (e.g. temperature).
3. Nonlinear characteristic curves → need linearization and filtering.

Output types:

- Binary (on/off)
- Pulse / frequency encoded
- Analog continuous signals

Signal Conditioning Functions

- **Amplification:** Boosts weak sensor signals to usable levels.
- **Filtering:** Removes noise and unwanted frequency components.
- **Linearization:** Corrects nonlinear sensor outputs.
- **Impedance Matching:** Ensures proper signal transfer between stages.
- **Level Shifting:** Adjusts signal baseline for compatibility.
- **Modulation/Demodulation:** Prepares signals for transmission or extraction.
- **Isolation:** Prevents ground loops and protects against high voltages.
- **Calibration:** Adjusts system to ensure accurate measurements.
- **Conversion:** Transforms signals between analog and digital domains.
- **Data Transmission:** Sends signals over distances without degradation.

Common Signal Conditioning Techniques

- **Bridge Circuits:** Convert resistance changes (e.g., strain gauges) into voltage signals.
- **Operational Amplifiers:** Used for amplification, filtering, and buffering.
- **Filters:** Low-pass, high-pass, band-pass, and notch filters to manage frequency content.
- **Analog-to-Digital Converters (ADCs):** Convert analog signals to digital form for processing.
- **Digital-to-Analog Converters (DACs):** Convert digital signals back to analog form.
- **Multiplexers:** Allow multiple sensor signals to be processed by a single conditioning circuit.
- **Isolation Amplifiers:** Provide galvanic isolation between input and output.
- **Temperature Compensation Circuits:** Mitigate effects of temperature variations on sensor outputs.
- **Signal Modulators/Demodulators:** Prepare signals for transmission over long distances.
- **Data Transmission Protocols:** Ensure reliable communication between sensors and processing units.

Introduction to D.C. and A.C. Bridges

Objective: Bridge circuits are used to measure unknown resistance, capacitance, or inductance precisely by comparing with known components.
Conversion of tiny sensor variations (strain gauge, RTD, capacitive sensor) into measurable voltages and principles of null deflection help to curtain noise and improve accuracy.

- Based on the principle of **null deflection** — when the bridge is balanced, no current flows through the detector.
- Used extensively for **signal conditioning** in sensors and transducers.
- D.C. bridges → resistive transducers (strain gauges, resistance temperature detectors (RTDs)).
  A.C. bridges → reactive transducers (inductive, capacitive).

Bridge Balance Principle

$$\frac{Z_1}{Z_2} = \frac{Z_3}{Z_4}$$

At balance: $V_X = V_Y$ and detector current $= 0$.

Introductory Example: Half-Bridge Measurement

Half-bridge concept:

$$e_o = e_i \frac{R_2}{R_1 + R_2}, \qquad [e_o, e_i \text{ in V}]$$

where:

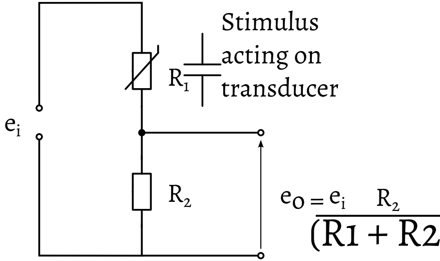
$$R_1 = R + \Delta R, \quad R_2 = R.$$

Issues:

- Always produces non-zero offset: $e_o(e_{\text{stimulus}=0}) \neq 0$.
- Temperature drift affects balance.

Use cases:

- Basic resistance sensing demonstration.
- Precursor to full Wheatstone bridge.


Stimulus acting on transducer
R_1
e_i
R_2
$e_O = e_i \frac{R_2}{(R1 + R2}$

Half-Bridge Circuit

Wheatstone DC Bridge

Balance condition:

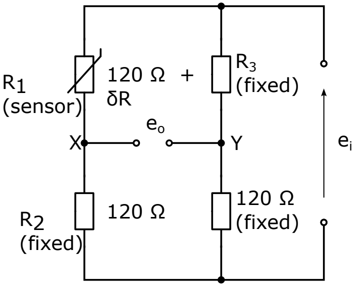
$$\frac{R_1}{R_2} = \frac{R_3}{R_4}, \qquad [R\ \Omega]$$

Output voltage (unbalanced):

$$e_o = e_i \left(\frac{R_2}{R_1 + R_2} - \frac{R_4}{R_3 + R_4} \right)$$

Characteristics:

- High accuracy for small ΔR.
- Sensitive to temperature unless compensated.



Wheatstone Bridge Circuit

Use cases: Strain gauges, RTDs, precision resistance measurement. **Advantages:** simple, accurate, high SNR. **Disadvantages:** 4-resistor matching needed, temp-drift.

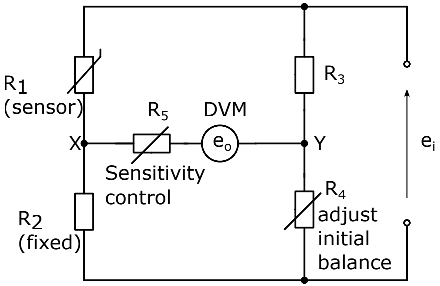
Practical Wheatstone Bridge with Zero & Sensitivity Adjustments

- Variable resistor R_4 used for **zero adjustment**.
- R_S introduces **sensitivity control** for the null detector.

Output near balance:

$$e_o \approx \frac{e_i}{4R}(\Delta R_1 - \Delta R_3)$$

Use cases: Industrial load cells, pressure transducers, torque sensors.



Practical Wheatstone Bridge Circuit

Double Active Arm Bridge (Two-Element Bridge)

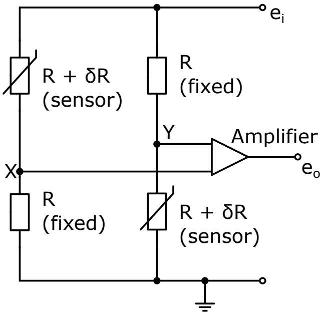
Purpose: Increase sensitivity + temperature compensation.

$$e_o \approx \frac{e_i}{2R}(\Delta R)$$

Advantages:

- Doubles output sensitivity.
- Self-temperature-compensating.

Use cases: Mechanical systems with tension/compression pairs (e.g., 4-arm strain gauge load cells).



Double Active Arm Bridge Circuit

Full Bridge with Four Active Sensors

Maximum sensitivity:

$$e_o = \frac{e_i}{4}\left(\frac{\Delta R}{R}\right) \times 4$$

Advantages:

- Highest output level.
- Excellent common-mode rejection.
- Full temperature compensation.

Use cases: high-accuracy weighing systems, multi-axis force sensors.

General AC Bridge Theory

Impedance representation:

$$Z = R + jX, \qquad [Z\ \Omega]$$

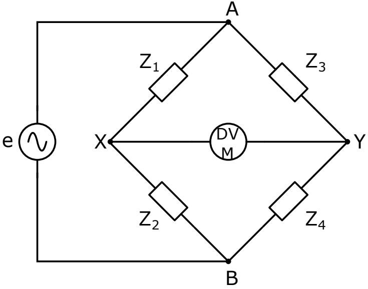
Balance condition:

$$Z_1 Z_4 = Z_2 Z_3$$

For balance:

- Real parts must match.
- Imaginary parts must match.

Use cases: Inductance, capacitance, dielectric loss, dissipation factor.



General AC Bridge Circuit

Maxwell Bridge (Inductance with Resistance)

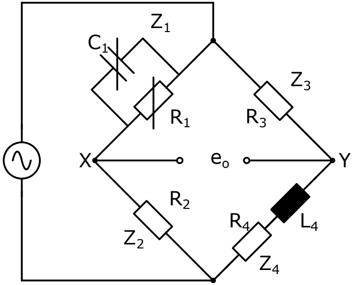
Used for: measuring inductance L in series with resistance R.

$$Z_1 = \frac{R_1}{1 + j\omega R_1 C_1} \quad Z_4 = R_4 + j\omega L_4$$

From balance:

$$R_4 = \frac{R_2 R_3}{R_1}, \qquad [\Omega]$$

$$L_4 = R_2 R_3 C_1, \qquad [\mathrm{H}]$$



Maxwell Bridge Circuit

Advantages:

- Frequency-independent solution.
- Easy to balance physically.

Disadvantages: not suitable for high-Q coils. **Use cases:** medium inductors (10 mH–100 H).

Hay Bridge (High-Q Inductance Measurement)

Used for: inductors with small series resistance (high Q).

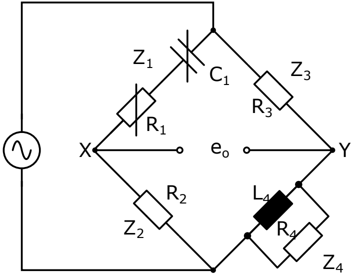
$$R_1 R_4 = R_2 R_3, \qquad [\Omega]$$

$$L_4 = R_2 R_3 C_1, \qquad [\mathrm{H}]$$

Advantages:

- Works well for high-Q inductors.
- Frequency-independent result.

Disadvantages: Not accurate for lossy inductors.



Hay Bridge Circuit

Schering Bridge (Capacitance + Loss Measurement)

Unknown: C_4 with series loss R_4.

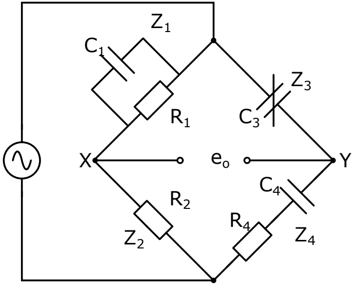
$$R_1 C_3 = R_2 C_4 \qquad [\mathrm{s}]$$

$$R_4 C_4 = R_1 C_1$$

Measures:

- Capacitance C (in farads)
- Dissipation factor, dielectric loss

Use cases: insulation testing, high-voltage apparatus, dielectric materials.



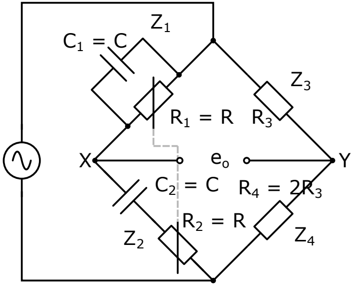
Schering Bridge Circuit

The Wien Frequency Bridge: Principle and Circuit

Purpose: The Wien frequency bridge is designed so that the bridge balances (i.e., $e_o = 0$) **only at one specific frequency** of the applied AC signal. Thus, it acts as a **notch filter** or a selective frequency measurement circuit.

Circuit characteristics:

- Arms Z_3 and Z_4 are pure resistances, with $Z_4 = 2Z_3$.
- Variable resistors R_1 and R_2 are mechanically ganged so both adjust simultaneously to the same value R.
- Capacitors in opposite arms C_1 and C_2 have identical values C.



Wien Frequency Bridge Circuit

Impedances:

$$Z_1 = \frac{R}{1 + j\omega CR}, \qquad Z_2 = R + \frac{1}{j\omega C}$$

$$Z_3 = R_3 \qquad Z_4 = 2R_3$$

Wien Frequency Bridge: Balance Condition and Frequency

Balance condition for AC bridges:

$$Z_1 Z_4 = Z_2 Z_3$$

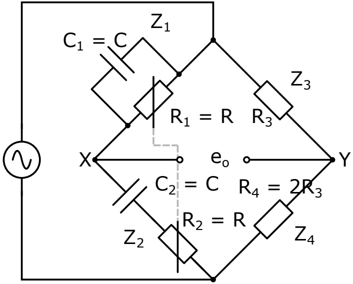
Substitute impedances:

$$\frac{R}{1 + j\omega CR} \cdot 2R_3 = \left(R + \frac{1}{j\omega C} \right) R_3$$

After simplifying and separating real and imaginary parts, the balance occurs when:

$$\omega = \frac{1}{CR} \quad [\text{rad/s}]$$

$$f = \frac{1}{2\pi CR} \quad [\text{Hz}]$$



Wien Frequency Bridge Circuit

Wien Frequency Bridge: Balance Condition and Frequency

Interpretation:

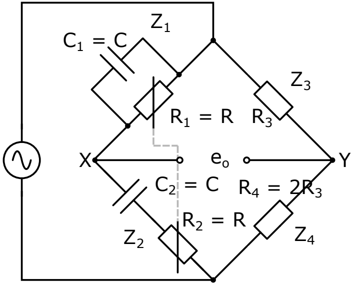
- Bridge gives **zero output** only at a single frequency.
- Acts as a **precise frequency indicator** or **notch filter**.

Advantages:

- High frequency selectivity.
- Components adjust symmetrically → good stability.

Limitations:

- Sensitive to tolerance of R and C.
- Not suitable for broadband measurements.



Wien Frequency Bridge Circuit

Summary Table: DC and AC Bridges

Bridge	Measures	Use Cases
Wheatstone	Resistance, ΔR	Strain, RTD
Practical Wheatstone	Precise ΔR	Load cells
Double-arm bridge	High sensitivity	Force/torque
Maxwell	L with R	Medium inductors
Hay	High-Q inductors	RF coils
Schering	C and dielectric loss	HV equipment
Maxwell–Wien	Precision L	Standards
De Sauty	Ideal C	Cap. comparison
Owen	Low-Q inductors	Lossy coils
Wien frequency	Frequency f	Oscillators

Operational Amplifiers for Signal Conditioning

Introduction to Operational Amplifiers

Why Op-Amps?

- Widely used in analog signal conditioning, filtering, instrumentation, and measurement systems.
- Originally used in analog computers for mathematical operations: addition, subtraction, differentiation, integration.
- Modern op-amps are compact ICs with high performance, low noise, and wide bandwidth.

Role in Signal Conditioning

- Amplification of weak sensor signals.
- Conversion between signal domains (e.g., current–voltage).
- Implementation of filters and mathematical operations.
- Isolation between high/low impedance stages.

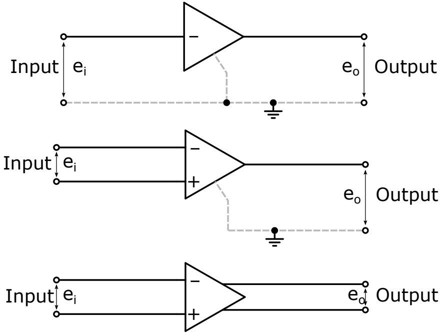
Operational Amplifier Terminals

Terminals of an Op-Amp

- Two input terminals:
  - Non-inverting input $(+)$
  - Inverting input $(-)$
- One output terminal.

Input Configurations

- Single-ended input and output.
- Differential input, single-ended output.
- Differential input and differential output.



Op-amp configurations: single-ended and differential

Characteristics of the Ideal Operational Amplifier

An ideal op-amp is defined by the following properties:

1. **Infinite open-loop gain**, $A_{\mathrm{VOL}} \to \infty$
2. **Infinite input impedance**, $Z_{\mathrm{in}} \to \infty$ → No input current: $i_+ = i_- = 0$
3. **Zero output impedance**, $Z_{\mathrm{out}} = 0$
4. **Infinite bandwidth** → constant gain from DC to ∞
5. **Zero input offset voltage and current**

Implications:

- No voltage difference between inputs under negative feedback.
- Output adjusts to maintain $v_+ = v_-$.

Golden Rules of Ideal Op-Amps

Golden Rule 1: Under negative feedback, the op-amp forces

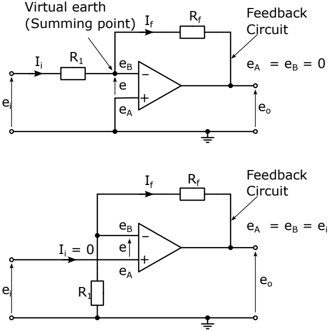
$$v_+ = v_-$$

Golden Rule 2:

$$i_+ = i_- = 0$$

No current flows into either input terminal.
Consequences:

- The inverting input can act as a **virtual ground** if the non-inverting terminal is grounded.
- The inverting node becomes a **summing point** for currents.



Inverting and non-inverting feedback configurations

Inverting Operational Amplifier

Circuit Principle

- Input applied to the inverting terminal via resistor R_i.
- Feedback resistor R_f returns output to inverting input.
- Non-inverting terminal grounded ($v_+ = 0$).

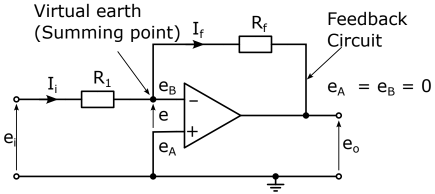
Derivation Using Golden Rules:

$$i_i = \frac{e_i}{R_i}, \quad i_f = \frac{-e_o}{R_f}$$

Since $i_i = i_f$:

$$\frac{e_i}{R_i} = -\frac{e_o}{R_f}$$

$$\Rightarrow \frac{e_o}{e_i} = -\frac{R_f}{R_i}$$



Closed-loop gain:

$$A_{V,CL} = -\frac{R_f}{R_i}$$

Non-Inverting Operational Amplifier

Key Feature: Input is applied to the $+$ terminal, so output is in-phase.

$$e_B = e_i$$

Using feedback current:

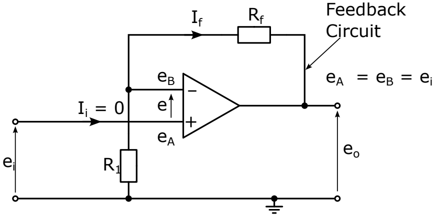
$$i_f = \frac{0 - e_o}{R_1 + R_f}$$

From nodal condition:

$$i_f = \frac{e_i - e_o}{R_f}$$

Solving the equations:

$$A_{V,CL} = \frac{e_o}{e_i} = 1 + \frac{R_f}{R_i}$$



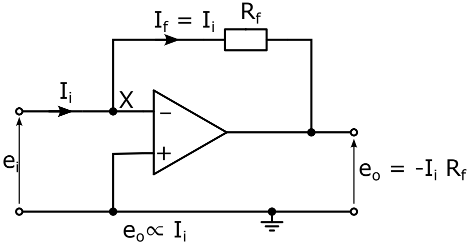
Properties:

- No inversion of signal.
- High input impedance.

Op-Amp as a Current-to-Voltage Converter

Used in:

- Photodiodes
- Sensors generating current output
- Precision current measurement



Derivation: Current flows through R_f:

$$e_o = -I_i R_f \quad \text{(V)}$$

Units: $[I_i] = \text{A}$, $[R_f] = \Omega$, $[e_o] = \text{V}$

Summing Amplifier (Current Summation)

Currents summed at virtual ground:

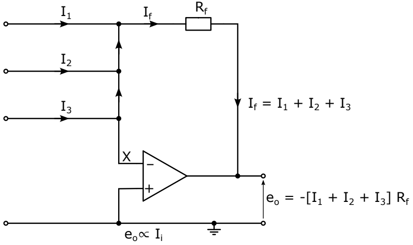
$$I_f = I_1 + I_2 + I_3$$

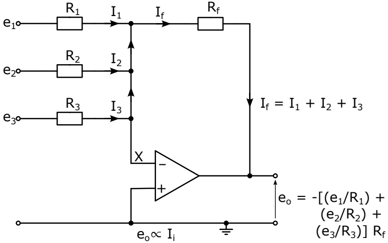
Output:

$$e_o = -R_f(I_1 + I_2 + I_3)$$

Voltage Summation Form:

$$e_o = -R_f\left(\frac{e_1}{R_1} + \frac{e_2}{R_2} + \frac{e_3}{R_3}\right)$$



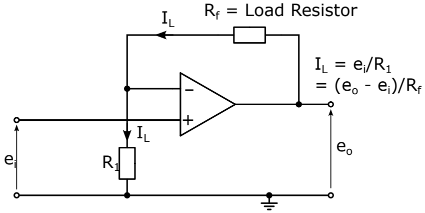


Voltage-to-Current Converter (V-I Converter)

$$I_L = \frac{e_i}{R_1}$$

Load current independent of load resistor: Set by R_1 only.
Used in:

- Actuators
- Transducer excitation
- Analog driving circuits



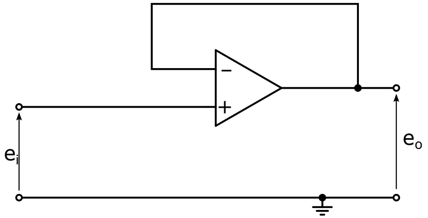
Op-Amp as a Buffer (Voltage Follower)

Unity gain:

$$e_o = e_i$$

Purpose:

- High input impedance.
- Low output impedance.
- Prevents loading of previous stage.



Ideal Op-Amp Differential Subtractor

Deriving the Output Voltage:

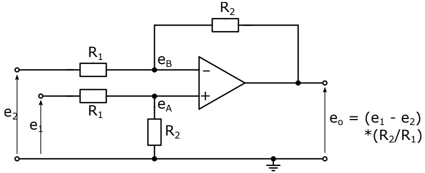
$$e_1 \frac{R_2}{R_T} = e_2 \frac{R_2}{R_T} + e_o \frac{R_1}{R_T}$$

Cancel R_T $(=R_1 + R_2)$ and rearrange:

$$e_o = (e_1 - e_2)\frac{R_2}{R_1}$$

Final Transfer Function:

$$\boxed{e_o = (e_1 - e_2)\frac{R_2}{R_1}}$$



Output expression:

$$e_o = (e_1 - e_2)\frac{R_2}{R_1}$$

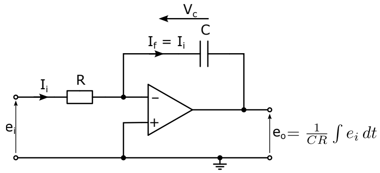
Use Cases:

- Sensor differential measurement.
- Rejecting common-mode interference.

Ideal Op-Amp Integrator

Input current through R:

$$I_i = \frac{e_i(t) - 0}{R} = \frac{e_i(t)}{R} \quad \text{[A]}$$



Feedback current through capacitor C (voltage across C is $v_C = -e_o(t)$):

$$I_f = C\,\frac{\mathrm{d}v_C}{\mathrm{d}t} = C\,\frac{\mathrm{d}(-e_o(t))}{\mathrm{d}t} = -C\,\frac{\mathrm{d}e_o(t)}{\mathrm{d}t}$$

Equating $I_i = I_f$:

$$\frac{e_i(t)}{R} = -C\,\frac{\mathrm{d}e_o(t)}{\mathrm{d}t}$$

Output–input relationship (integrator equation):

$$\frac{\mathrm{d}e_o(t)}{\mathrm{d}t} = -\frac{1}{RC}e_i(t) \quad \Longrightarrow \quad e_o(t) = -\frac{1}{RC}\int e_i(t)\,\mathrm{d}t + e_o(0)$$

For a properly reset integrator with $e_o(0) = 0$:

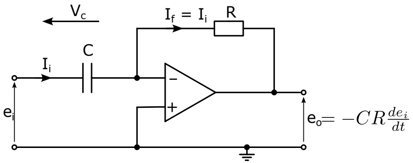
$$\boxed{e_o(t) = -\frac{1}{RC}\int e_i(t)\,\mathrm{d}t}$$

with R in Ω, C in F, so RC has units of s.

Ideal Operational Amplifier as a Differentiator

Concept:

- A differentiator produces an output proportional to the **time derivative** of the input voltage.
- Achieved by interchanging the feedback resistor and input capacitor of the integrator circuit.
- Uses **inverting** op-amp configuration.



Derivation:

$$i_i = C\frac{de_i}{dt}$$

$$i_f = \frac{-e_o}{R}$$

$$i_i = i_f$$

$$C\frac{de_i}{dt} = -\frac{e_o}{R}$$

Output Voltage:

$$\boxed{e_o = -CR\frac{de_i}{dt}}$$

Ideal Operational Amplifier as a Differentiator

Characteristics:

- **Gain factor:** CR (units: s)
- Output leads input by 90° in phase for sinusoidal signals.

Advantages:

- Ideal for edge detection and high-frequency transient measurement.
- Produces sharp spikes for fast changes in input.

Limitations (Practical):

- Highly sensitive to high-frequency noise (since noise has large derivatives).
- Real circuits require a small series resistor with C and RC stabilisation.

Applications:

- Signal processing: differentiation of waveforms.
- Edge/transition detection (square $\rightarrow$ impulse).
- Analog computing, PID controllers (derivative term).

Summary of Op-Amp Configurations

Configuration	Transfer Function	Use Cases
Inverting Amplifier	$e_o = -\frac{R_f}{R_i} e_i$	Signal inversion, amplification
Non-Inverting Amplifier	$e_o = \left(1 + \frac{R_f}{R_i}\right) e_i$	Buffering, high input impedance
Current-to-Voltage Converter	$e_o = -I_i R_f$	Photodiodes, current sensors
Summing Amplifier	$e_o = -R_f \left(\frac{e_1}{R_1} + \frac{e_2}{R_2} + \dots\right)$	Audio mixing, sensor fusion
Voltage-to-Current Converter	$I_L = \frac{e_i}{R_1}$	Actuator driving, transducer excitation
Buffer (Voltage Follower)	$e_o = e_i$	Impedance matching, isolation
Differential Subtractor	$e_o = (e_1 - e_2)\frac{R_2}{R_1}$	Differential measurement, noise rejection
Integrator	$e_o(t) = -\frac{1}{RC} \int e_i(t)\, dt$	Signal integration, analog computing
Differentiator	$e_o = -CR\frac{de_i}{dt}$	Edge detection, transient analysis

Operational Amplifier Frequency Response

Key idea: The practical operational amplifier does *not* respond instantaneously to an input signal. Finite internal compensation introduces a **delay** and **frequency-dependent gain**.

- At **low frequencies (DC – few Hz)**, delay is negligible.
- At **higher frequencies**, finite gain-bandwidth causes:
  - magnitude drop (gain roll-off),
  - phase lag between input and output,
  - distortion for large-signal or fast-changing inputs.
- Therefore, analysis uses only **small-signal sinusoidal** inputs.

Model: For many op-amps, the open-loop transfer function is:

$$A_{OL}(jf) = \frac{A_{OL}}{1 + j\frac{f}{f_c}}$$

where:

$$A_{OL} \text{ (DC gain)}, \quad f_c \text{ (break frequency in Hz)}$$

Voltage Gain in Decibels

For sinusoidal steady-state, the gain is expressed in **decibels (dB)** as:

$$G_{\mathrm{dB}} = 20\log_{10}\left(\frac{e_o}{e_i}\right)$$

Interpretation:

- Negative dB → attenuation.
- Positive dB → amplification.
- Note: *Power* is proportional to voltage squared → doubling power corresponds to $+3$ dB.

Example table values:

- $e_o/e_i = 1 \rightarrow 0$ dB → unity gain.
- $e_o/e_i = 10 \rightarrow 20$ dB.
- $e_o/e_i = 0.5 \rightarrow -6$ dB → quarter power.

Tab. 3.1: Voltage ratio and decibel relationship

$\frac{e_o}{e_i}$	dB	Power output
0.1	-20	
0.5	-6	Quarter
0.707	-3	Half
1	0	Unity
1.41	3	Double
2	6	
10	20	
100	40	

Complex Gain Representation and Phase Lag

Amplifier gain is a **complex quantity**:

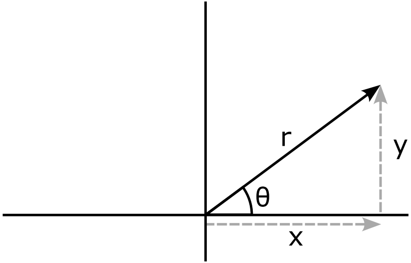
$$A = |A|e^{j\theta}$$

A sinusoidal signal can be expressed as:

$$r = x + jy, \qquad |r| = \sqrt{x^2 + y^2}, \qquad \theta = \tan^{-1}\left(\frac{y}{x}\right)$$

Importance:

- Phase lag increases with frequency.
- Causes reduction in stability (important for feedback design).



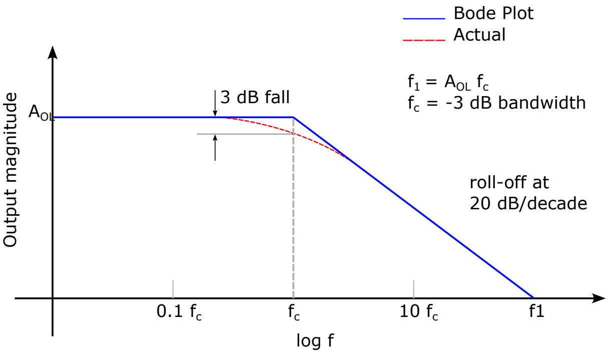
Bode Plot for Op-Amp Magnitude Response

Open-loop magnitude response:

$$|A_{OL}(jf)| = \frac{A_{OL}}{\sqrt{1 + \left(\frac{f}{f_c}\right)^2}}$$

Key points:

- For $f \ll f_c$: gain $\approx A_{OL}$
- For $f = f_c$: gain drops by **3 dB**
- For $f \gg f_c$: gain falls with slope **20 dB/decade**
- At unity-gain frequency f_1: $|A| = 1$ or 0 dB



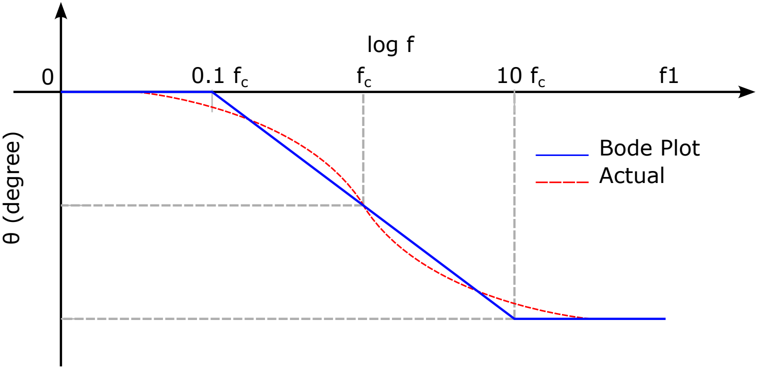
Bode Plot for Op-Amp Phase Response

Phase shift for single-pole op-amp:

$$\theta(f) = -\tan^{-1}\left(\frac{f}{f_c}\right)$$

Important values:

- At $f = 0$: $\theta = 0^\circ$
- At $f = f_c$: $\theta = -45^\circ$
- At $f \gg f_c$: $\theta \to -90^\circ$



Break Frequency and Unity-Gain Frequency

Break frequency f_c: Point where open-loop gain drops by $3\,$dB.

$$f_c = \frac{1}{2\pi RC} \quad \text{(for internally compensated op-amp)}$$

Unity-gain frequency f_1:

$$A_{OL}(f_1) = 1$$

$$f_1 \approx A_{OL} \cdot f_c$$

Implications:

- Sets maximum usable closed-loop bandwidth.
- Determines speed/slew performance.

Why Frequency Response Matters in Practice

1. **Stability considerations**

- Large phase lags → risk of oscillations in feedback systems.
- Phase margin must be $> 45^\circ$ for robustness.

2. **Closed-loop bandwidth limitations**

$$\text{Closed-loop bandwidth} \approx \frac{f_1}{\text{Closed-loop gain}}$$

3. **Accuracy and distortion**

- High frequencies → increased error.
- Important for instrumentation and fast control loops.

4. **Noise performance**

- Noise increases at high frequency.
- Filtering and layout become critical.

Linear Scaling Circuits

- Linear scaling circuits produce an output e_o that is a linear function of input e_i:

  $$e_o = f(e_i) = me_i + c$$

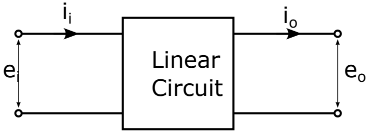
  where:

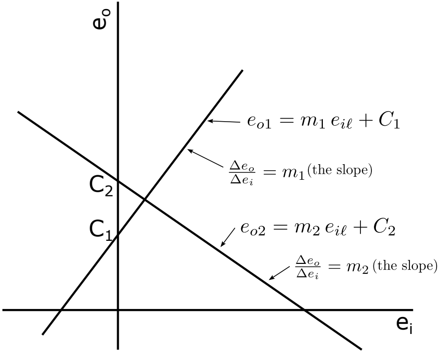
  - m — slope or gain (dimensionless)
  - c — output offset at zero input (V)
- Linear behaviour implies:
  - Proportional change in output for change in input.
  - No distortion or frequency dependence (requires purely resistive components).
- Applications:
  - Voltage scaling (amplification/attenuation)
  - Impedance conversion
  - Signal addition/subtraction
  - Voltage-to-current or current-to-voltage conversion

Graphical Interpretation of Linearity

- Linear circuits maintain a straight-line e_o–e_i relationship.
- Examples with positive and negative slopes.

$$\frac{\Delta e_o}{\Delta e_i} = m$$





Linear e_o vs. e_i relationships

Design Requirements

- Circuit must behave identically over relevant frequency range → resistive components essential.
- Selection of resistor values:
  - Must avoid loading the sensor/source.
  - Should not be extremely large (bias current → offset errors).
  - Should not be extremely small (excess current loading).
- General-purpose op-amps require minimum output load:

$$R_L \geq 2\ \text{k}\Omega \text{(check datasheet!)}$$

- In inverting amplifiers, input resistor R_1 determines input impedance:

$$Z_{\text{in}} = R_1$$

- High-precision resistors (0.1% tolerance or better) recommended for accuracy.
- Common examples are: inverting/non-inverting amplifiers, summing amplifiers, voltage/current converters, buffers that we learnt in previous slides.

Summary

- In summary, linear scaling circuits using op-amps provide precise control over signal amplitude and offset.
- Proper design ensures minimal loading, high accuracy, and stable operation across the desired frequency range.
- These circuits form the foundation for more complex signal conditioning and processing tasks in instrumentation systems.
- There are topics related to non linear circuits and other advanced op-amp configurations that can be explored at your end based on interest.
- This concludes our discussion on introduction of using operational amplifiers.

Transducer Amplifiers

Transducer Amplifiers: Overview

Transducers convert one form of energy into another. In instrumentation systems, we often use transducers that convert a physical stimulus into an electrical quantity.

Common resistive transducers:

- Strain gauges
- Resistance thermometers (RTDs)
- Thermistors
- Potentiometric transducers
- Light-dependent resistors (LDRs)

Most resistive transducers are used inside a **Wheatstone bridge**. When the transducer resistance changes by a small fraction, the bridge becomes unbalanced and generates a measurable voltage.

Goal: Amplify this small differential voltage to a useful level using an operational amplifier.

Bridge Amplifier

- The transducer element has resistance $R_X = R(1+a)$, where a is a small fractional change.
- Bridge is biased by supply E, with opposite corners labelled X and Y.
- The op-amp forces its input nodes to the same potential: $V_X = V_Y = V$.
- Small imbalance is converted into output voltage e_o.

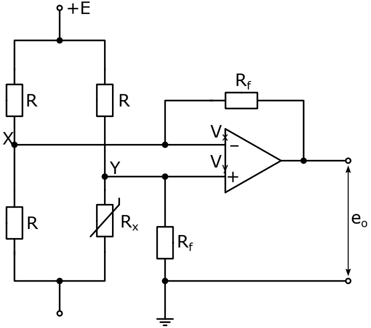
At node X:

$$\frac{E - V_X}{R} = \frac{V_X - e_o}{R_f} + \frac{V_X}{R}$$

At node Y:

$$\frac{E - V_Y}{R} = \frac{V_Y}{R(1+a)} + \frac{V_Y}{R_f}$$

Because the op-amp forces $V_X = V_Y = V$, the above can be combined to yield the bridge output relation:



Bridge amplifier with resistive transducer

$$e_o = \left[\frac{V}{R} - \frac{V}{R(1+a)}\right] R_f$$

Bridge Amplifier: Solving for Output

From analysis of the right-hand side of the bridge:

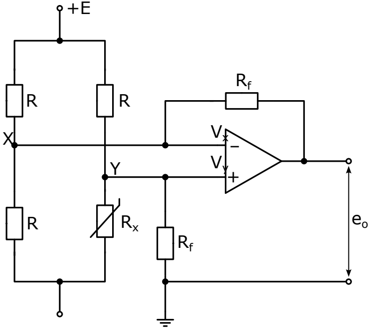
$$V = V_Y = E\left[\frac{R(1+a) \parallel R_f}{R + (R(1+a) \parallel R_f)}\right]$$

This simplifies to:

$$V = \frac{ER_f(1+a)}{R(1+a) + R_f + R_f(1+a)}$$

Substituting back into the earlier expression for e_o gives:

$$e_o = \frac{ER_f a}{R} \cdot \frac{1}{(1+a)\left(\frac{R+R_f}{R_f}\right) + 1}$$



Bridge amplifier with resistive transducer

If $a \ll 1$, then:

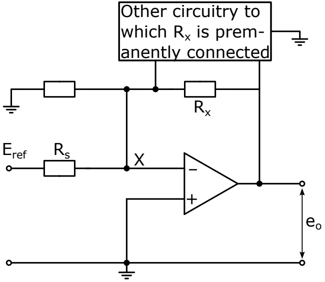
$$e_o \approx \frac{ER_f}{R} a.$$

Thus the circuit behaves linearly for small transducer changes.

Resistance Measurement

Goal: Measure an unknown resistance R_X in situ, without disturbing its electrical environment.

- A known reference resistor R_s is fed from a reference source E_{ref}.
- Current through R_s is forced through the unknown resistor R_X.
- The op-amp holds point X at virtual earth (0 V), so all current must flow through R_X.



Resistance measurement circuit

Key relationship:

$$e_o = \frac{E_{ref} R_X}{R_s}$$

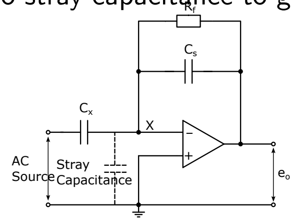
Thus, the amplifier output voltage is **directly proportional** to the unknown

Advantages:

- Does not disturb existing circuitry connected to R_X.
- Virtual-earth node eliminates errors due to stray capacitances

Capacitance Measurement

Goal: Measure small capacitances while eliminating errors due to stray capacitance to ground.



Capacitance measurement circuit

Concept:

- Connect unknown capacitor C_X to the inverting terminal.
- Op-amp maintains virtual earth at X ⇒ stray capacitances to ground carry negligible voltage.
- Known capacitor C_s is used in the feedback path.

AC gain expression:

$$\frac{e_o}{e_i} = \frac{j\omega C_X R_f}{1 + j\omega C_s R_f}$$

Assuming:

$$\omega \gg \frac{1}{2\pi C_s R_f}$$

The output magnitude becomes approximately:

$$C_X \approx \frac{e_o/e_i}{C_s}$$

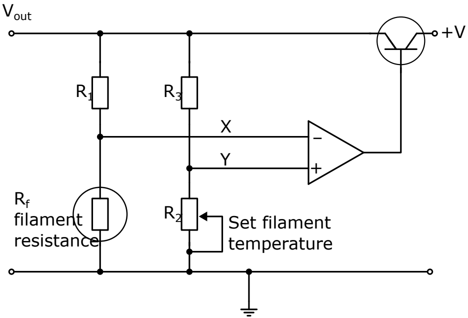
Important: Choose C_s of the same order as C_X for accuracy.

Air Speed Measurement

Principle: Air flowing over a heated filament changes its resistance. This imbalance is sensed via a bridge and amplified.

Process:

- A platinum filament (positive temperature coefficient) is heated by a bias current.
- Cooling airflow reduces filament temperature, causing resistance R_f to decrease.
- The op-amp adjusts output voltage V_{out} to restore bridge balance.
- Thus: **output voltage is proportional to airflow speed**.



Air speed measurement using heated filament and bridge

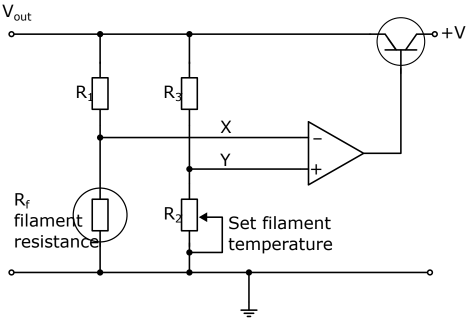
Air Speed Measurement

The bridge balance condition:

$$R_f = \frac{R_1 R_2}{R_3}$$

Advantages of constant-temperature anemometry:

- Rapid response (no thermal lag).
- High sensitivity.
- Good linearity around the operating point.



Air speed measurement using heated filament and bridge

Additional Analogue Processing Circuits

Beyond the circuits studied, many other analogue processing techniques exist:

- **Capacitance multiplication**: simulate large capacitors using small ones + op-amp.
- **Arithmetic averaging amplifiers**: compute mean values.
- **Time-averaging filters**: perform low-pass smoothing.
- **Precision rectifiers and diode compensators**.
- **Ratiometric converters**: maintain accuracy against supply variations.

These can be explored further depending on course depth and student interest.

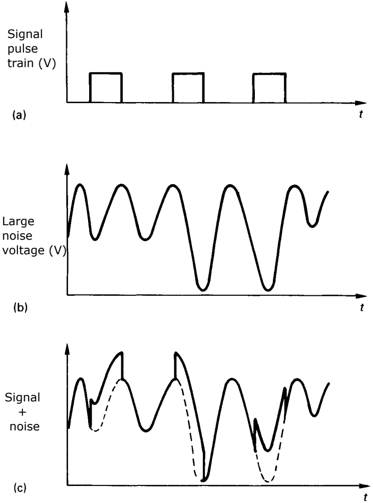
Noise in Measurement Systems

What is Noise?

Definition

Noise is any **unwanted electrical signal** (voltage or current) that becomes intermixed with the wanted signal and degrades its readability.

- In measurement systems, noise may originate from the sensor, wiring, environment, or electronics.
- If noise amplitude is comparable to or larger than the signal, **information can be lost**.
- Noise is typically **random** and described statistically (RMS, power spectral density).



Signal corrupted by noise

Internal vs External Noise

Internal Noise

- Generated **within** electronic components.
- Present even with zero input signal.
- Examples: thermal noise, shot noise, 1/f noise.

External Noise

- Picked up **from surroundings**.
- Couples through stray capacitance, inductance, or radiation.
- Examples: mains interference, EMI from machines, lightning, radio transmitters.

Key idea: External noise can often be reduced by design; internal noise sets the ultimate sensitivity limit.

Signal-to-Noise Ratio (Linear Form)

Definition at a given frequency / bandwidth

$$\mathrm{SNR} = \frac{S}{N}$$

where

- S = wanted signal **power** in the band of interest [W],
- N = unwanted noise **power** in the same band [W].

- SNR is always defined for a **specified bandwidth** B.
- If the system bandwidth increases, noise power generally increases (often proportional to B).

SNR in Decibels (dB)

Power ratio to decibels

$$\boxed{\mathrm{SNR_{dB}} = 10\log_{10}\left(\frac{S}{N}\right)\ \text{dB}}$$

Voltage or current ratios:

- When signal and noise are measured as **RMS voltages** across the same resistance R:

$$\frac{S}{N} = \left(\frac{V_{S,\mathrm{rms}}}{V_{N,\mathrm{rms}}}\right)^2 \quad \Rightarrow \quad \mathrm{SNR_{dB}} = 20\log_{10}\left(\frac{V_{S,\mathrm{rms}}}{V_{N,\mathrm{rms}}}\right)$$

- Similarly for currents: $\mathrm{SNR_{dB}} = 20\log_{10}(I_{S,\mathrm{rms}}/I_{N,\mathrm{rms}})$.

Note: Amplifier gain does *not* improve input SNR; it amplifies signal and noise alike, and adds its own noise.

Example of SNR Calculation

Suppose an amplifier input has:

- wanted signal power: $S = 3\text{mW} = 3 \times 10^{-3}\text{W}$,
- noise power: $N = 30\mu\text{W} = 30 \times 10^{-6}\text{W}$.

$$\text{SNR} = \frac{S}{N} = \frac{3 \times 10^{-3}}{30 \times 10^{-6}} = 100$$

$$\Rightarrow \text{SNR}_{\text{dB}} = 10\log_{10}(100) = 10 \times 2 = 20\text{dB}$$

- 20dB may be acceptable for voice links, but high-quality audio typically needs > 60dB.

Overview of External Noise Sources

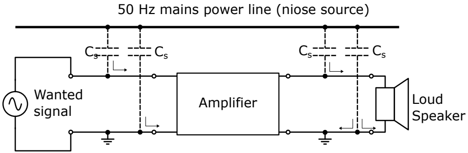
- External noise includes all unwanted signals **picked up** from the environment.
- Coupling paths:
  - **Capacitive coupling** via stray capacitances C_s.
  - **Inductive coupling** via changing magnetic fields and loop areas.
  - **Radiated EMI** through space at RF.
  - **Conducted interference** through shared supplies/grounds.
- We focus on three representative groups:
  1. mains interference (50/60Hz),
  2. other man-made interference,
  3. natural noise sources.

Mains Interference (50/60 Hz Pick-up)

- A common external noise is coupling from the mains power line.
- Stray capacitances C_s between power lines and signal wiring inject a small 50Hz component.
- Not only the fundamental (50Hz) but also harmonics (100, 150Hz, . . .) may appear.

Mitigation preview:

- increase physical separation of power and signal wiring,
- reduce loop area,
- use shielding and single-point grounding (next section).



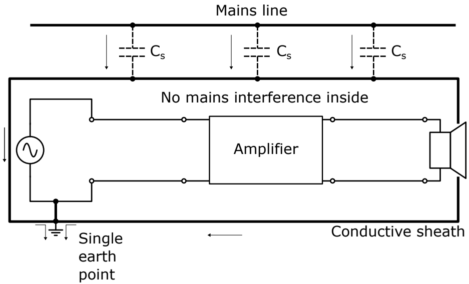
Capacitive coupling of mains interference

Conductive Sheath / Shield Against Mains Pick-up

- Enclosing sensitive wiring/equipment in a conductive sheath tied to a **single earth point** can block capacitive coupling.
- The sheath provides a low-impedance return for displacement currents induced via C_s.

Key takeaway

Good shielding requires a clear reference node; multiple earth points may create earth-loop noise.



Shielded cable to reduce mains pick-up

Other Man-made Noise Sources

- Generated by nearby electrical/electronic equipment.
- Examples:
  - sparking commutators and brushes in DC/universal motors,
  - fluorescent lamps and switching power supplies,
  - switchgear producing rapid di/dt and dv/dt transients,
  - radio/telemetry equipment.
- Often broadband and impulsive; can fall inside measurement band.

Three important mechanisms:

1. adjacent channel interference,
2. cross-talk between cables,
3. intermodulation products in non-linear devices.

Other Man-made Noise Sources

Adjacent Channel Interference

- In radio/data systems, channels operate at carrier frequencies $f_1, f_2, \ldots$.
- A strong interferer on a nearby channel may leak into the wanted channel if filtering is insufficient.
- Mitigation:
  - better frequency-selective front-end filtering,
  - adequate geographical/physical separation,
  - proper antenna orientation/polarization.

Cross-talk Between Cables

- When two or more cables run close together, changing currents create magnetic fields that induce voltages in adjacent conductors.
- Electrostatic coupling between conductors adds capacitive cross-talk.
- Mitigation:
  - route sensitive and noisy cables separately,
  - use twisted pairs and screened (shielded) cables,
  - minimize parallel run length.

Intermodulation Noise

- Two signals at f_1 and f_2 passing through a non-linear device produce new frequencies:

$$f_1,\ f_2,\ f_1 + f_2,\ |f_1 - f_2|,\ 2f_1,\ 2f_2,\ \ldots$$

- Some products may lie within the measurement band and degrade SNR.
- Mitigation:
  - avoid device saturation/non-linear regions,
  - band-limit before non-linear stages,
  - maintain adequate headroom.

Natural Noise Sources

- Predominantly affects radio-frequency and long-distance sensing.
- Two common types:
  1. **Atmospheric / static noise** from lightning and electric storms.
  2. **Galactic / cosmic noise** from stars and solar activity.

Frequency ranges

- Atmospheric noise is significant up to about 25MHz.
- Cosmic noise dominates beyond ~ 1GHz.

Suppression Strategies: Big Picture

- External noise suppression depends on:
  - frequency content of the interference,
  - power level of the noise source,
  - sensitivity/bandwidth of the measurement chain.
- Three practical levers:
  1. **Shielding**,
  2. **Good circuit layout**,
  3. **Eliminating earth loops**.

Suppression Strategies

- **Electrostatic (capacitive) shielding**:
  - Use a conductive enclosure tied to earth.
  - Provides low-impedance path for displacement currents.
- **Magnetic shielding at low frequency**:
  - Use high-permeability materials (e.g., mu-metal- nickel-iron soft alloy).
  - Offers a low-reluctance path to divert interfering flux.
- **RF shielding (skin effect)**:
  - A low-resistance metal shield (Al, Cu, brass).
  - Incident RF flux induces eddy currents; their opposing field cancels penetration.

Practical Shielding Techniques

- Braided shields or coaxial cables for signal lines.
- Mesh shields can reduce electrostatic noise while staying lightweight.
- Always maintain **continuous shield connection** to reference (avoid floating ends unless designed as such).

Pitfall

Multiple shield earth points can create circulating currents (earth loops), re-introducing noise.

Good Circuit Layout

- Keep mains / high-current tracks **far from** low-level signal paths.
- Avoid shared routing of power and signal through the same trunking or conduit.
- Minimize loop areas in sensitive circuits to reduce inductive pick-up.
- Use star-ground or single-point reference where possible.

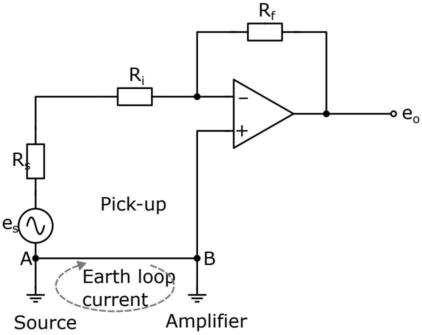
Rule of thumb

Treat layout as part of the circuit: parasitic C_s and mutual inductance are layout-controlled components.

Earth Loops (Ground Loops) Concept and Mitigation

- An earth loop occurs when a system has **more than one earth bond**.
- Different earth potentials ($e_A \neq e_B$) drive a loop current that couples into the signal path.
- The loop acts like an aerial, picking up magnetic/electric interference.

- Use a **single-point earth** (star grounding).
- Avoid ground connections through multiple equipment chassis.
- If unavoidable, break loops with:
  - differential inputs / instrumentation amplifiers,
  - isolation (transformer, opto-isolator),
  - careful cable shielding strategy.



Example of an earth loop

Why Internal Noise Matters

- Even with perfect shielding, every resistor, semiconductor, and amplifier generates noise.
- Internal noise sets the ultimate **minimum detectable signal**.
- We focus on three core types:
  1. thermal (Johnson) noise,
  2. shot noise,
  3. flicker (1/f) noise.

Thermal (Johnson) Noise in Resistors

- Caused by random thermal motion of charge carriers.
- Produces a noise voltage with zero mean but non-zero RMS.
- Spread uniformly over frequency → **white noise**.

RMS noise voltage over bandwidth B

$$\boxed{E_n = \sqrt{k_\mathrm{B} T B R}}$$

where

- k_B = Boltzmann constant $\approx 1.38 \times 10^{-23}$J/K,
- T = absolute temperature,
- B = bandwidth,
- R = resistance.

Implication: doubling B increases E_n by $\sqrt{2}$ and increases noise *power* by 2.

Thermal Noise: Key Design Insights

- Noise scales with **temperature**: cooling reduces E_n.
- Noise scales with **resistance**: larger R gives larger noise voltage.
- Noise scales with **bandwidth**: keep B only as wide as needed.
- Equivalent noise power in R:

$$P_n = \frac{E_n^2}{R} = 4k_{\mathrm{B}}TB \text{ W}$$

  (note independence from R for a matched load).

Shot Noise in Semiconductors

- Due to discrete nature or random variations in the flow of charge / carriers crossing junctions.
- Dominant when DC current passes through a diode/transistor junction.
- Also noise power uniform across frequency spectrum – white over frequency (within typical bands).

RMS shot-noise current

$$\boxed{i_n = \sqrt{2eIB}\ \text{A}}$$

where

- $e =$ electronic charge $\approx 1.60 \times 10^{-19}$C,
- $I =$ DC junction current [A],
- $B =$ bandwidth [Hz].

Takeaway: higher DC current I increases shot noise proportionally to $\sqrt{I}$.

Flicker Noise (1/f or Pink Noise)

- Arises from random trapping/recombination in semiconductors.
- Magnitude increases as frequency decreases.
- Often significant below about 10Hz.

Spectral form (qualitative)

Noise power spectral density typically follows

$$S_{1/f}(f) \propto \frac{1}{f^{\alpha}}, \quad \alpha \approx 1.$$

Practical meaning: low-frequency measurements (slow sensors, DC offsets) are most affected.

Partition Noise

- Occurs in transistors because emitter current divides randomly between base and collector.
- Can be viewed as an additional internal current-noise source.
- Usually smaller than thermal/shot noise but relevant in precision low-level circuits.

- Additional noise-reduction and signal-conditioning methods exist:
  - capacitance multiplication, arithmetic/time averaging,
  - chopper stabilization, modulation/demodulation,
  - digital filtering and oversampling.
- We will explore these in advanced sessions or projects.

Summary

- Noise is the unavoidable unwanted part of any measurement.
- **SNR** quantifies quality; in dB: $\mathrm{SNR}_{\mathrm{dB}} = 10\log_{10}(S/N)$.
- External noise can be reduced through shielding, layout, and ground-loop elimination.
- Internal noise types:
  - Thermal: $E_n = \sqrt{4k_{\mathrm{B}}TBR}$.
  - Shot: $i_n = \sqrt{2eIB}$.
  - Flicker: $S_{1/f}(f) \propto 1/f$.
- Design for low noise means: minimize bandwidth, resistance where possible, and avoid non-linear operation.

Fiters in Measurement Systems- Passive Filters and Active Filters

Passive Filters: Purpose and Types

- A **filter** is a network that passes AC signals only over a defined frequency range and attenuates others.
- Widely used in:
  - telecommunications and audio,
  - instrumentation and sensor interfaces,
  - power electronics and EMI/noise suppression.
- Two broad classes:

  Passive filters: only **R**, **L**, **C** elements, no active gain devices.

  Active filters: use op-amps/transistors in addition to RLC.
- Four basic responses (for both passive and active filters):
  1. Low–pass (LPF),
  2. High–pass (HPF),
  3. Band–pass (BPF),
  4. Band–stop / notch (BSF).

First–Order Passive Low–Pass CR Filter

- Circuit: series resistor R followed by shunt capacitor C.
- Input voltage e_i applied across series (R–C); output e_o taken across C.

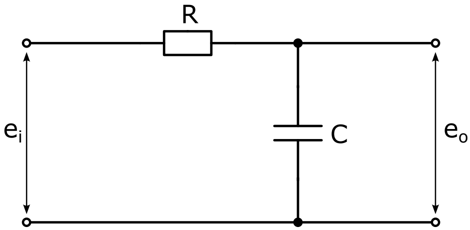
Complex transfer relation

Capacitive reactance:

$$X_C = \frac{1}{2\pi f C} \quad [\Omega]$$

Using voltage division in complex form:

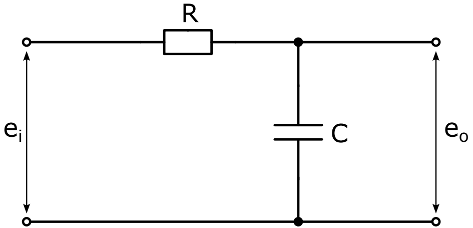
$$e_o = -j\,\frac{X_C}{R - jX_C}\,e_i$$



First–order passive low–pass CR filter

First–Order Passive Low–Pass CR Filter

- At low frequency ($f \to 0$): $X_C \to \infty$, so $e_o \approx e_i$ (pass).
- At high frequency ($f \to \infty$): $X_C \to 0$, so $e_o \approx 0$ (blocked).



First–order passive low–pass CR filter

Low–Pass Magnitude Response and Cut–off Frequency

From the complex relation:

$$\left|\frac{e_o}{e_i}\right| = \frac{X_C}{\sqrt{R^2 + X_C^2}} = \frac{1}{\sqrt{1 + (2\pi f RC)^2}}$$

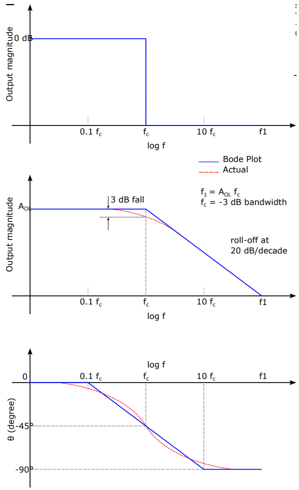
Cut–off (corner) frequency

Defined at the -3dB point:

$$\left|\frac{e_o}{e_i}\right| = \frac{1}{\sqrt{2}} \approx 0.707$$

which occurs at

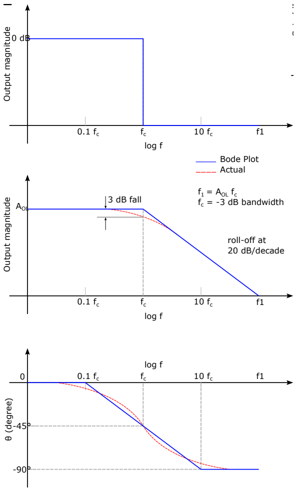
$$f_c = \frac{1}{2\pi RC} \quad \text{[Hz]}$$



First–order passive low–pass CR filter

Low–Pass Magnitude Response and Cut–off Frequency

- For $f \ll f_c$: $\left|\frac{e_o}{e_i}\right| \approx 1$ (flat pass–band).
- For $f \gg f_c$: gain falls with slope $\approx -20\text{dB}/\text{decade}$.
- Phase shifts from 0° (low f) to -90° (high f), with -45° at f_c.



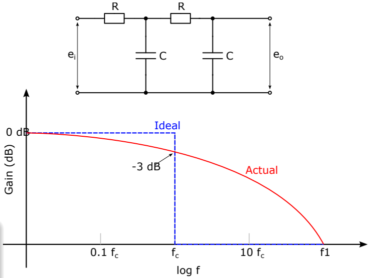
First–order passive low–pass CR filter

Second–Order Passive Low–Pass CR Filter (Cascade)

- A steeper roll–off can be obtained by cascading two first–order sections.
- Example: two identical CR low–pass sections.

Qualitative behaviour

- Each section contributes -20dB$/$decade above its cut–off.
- If both have the same f_c, the asymptotic high–frequency slope is -40dB$/$decade.
- In practice, the gain in the pass–band is reduced and the transition region is more rounded than the ideal second–order approximation.



Cascaded CR low–pass filter and Bode response

- Cascaded RC sections are simple but give limited control over pass–band flatness (e.g. Butterworth vs. Chebyshev characteristics).

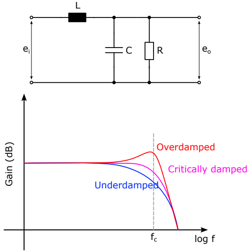
Second–Order Low–Pass LCR Filter (RLC Network)

- Use a series inductor L feeding a parallel C–R network (Figure 9.4).
- L and C chosen so that their natural resonant frequency is near the desired cut–off.

Resonant frequency and damping

$$f_0 \approx \frac{1}{2\pi\sqrt{LC}} \quad \text{[Hz]}$$

- R controls damping:
  - Low R: **underdamped** – peaking and ringing.
  - Critical R: **critically damped** – fast response without overshoot.
  - Very high R: **overdamped** – slow, no overshoot.



LCR low–pass filter and resonance curve

- Advantages: steeper roll–off and adjustable “knee” region.
- Drawbacks: needs an inductor (bulky, lossy, sensitive to magnetic coupling).

First–Order Passive High–Pass CR Filter

- High–pass filter obtained by simply **interchanging** R and C of the low–pass filter.
- Circuit: series capacitor C followed by resistor R to ground; output across R.

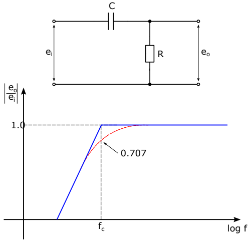
Magnitude response

With $X_C = 1/(2\pi f C)$,

$$\left|\frac{e_o}{e_i}\right| = \frac{R}{\sqrt{R^2 + X_C^2}} = \frac{2\pi f RC}{\sqrt{1 + (2\pi f RC)^2}}$$

Cut–off frequency:

$$f_c = \frac{1}{2\pi RC} \quad \text{[Hz]}$$



CR high–pass filter circuit

- For $f \ll f_c$: $|e_o/e_i| \approx 0$ (low frequencies blocked).
- For $f \gg f_c$: $|e_o/e_i| \approx 1$ (high frequencies passed).
- Slope is $+20\text{dB/decade}$ below f_c.

Second–Order Band–Pass LCR Filter

- Use a series C–L–R network.
- At resonance, the series LC combination has minimal impedance, so the output across R peaks.

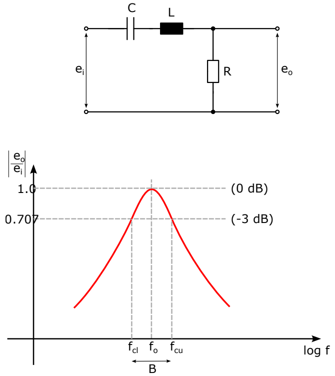
Key parameters

- Resonant (centre) frequency:

$$f_0 = \frac{1}{2\pi\sqrt{LC}} \quad \text{[Hz]}$$

- Lower and upper -3dB cut–off frequencies: f_{cl} and f_{cu}.
- Bandwidth:

$$B = f_{cu} - f_{cl} \quad \text{[Hz]}$$



LCR band–pass filter and frequency response

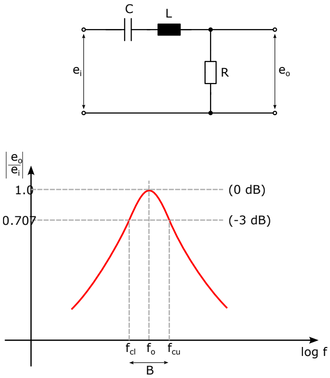
Second–Order Band–Pass LCR Filter

Key parameters

- Quality factor:

$$Q = \frac{f_0}{B}$$

- Higher $Q \Rightarrow$ narrower pass–band and more selective filter.
- Used for channel selection, tuned amplifiers, and sensor resonance shaping.



LCR band–pass filter and frequency response

Second–Order Band–Stop (Notch) LCR Filter

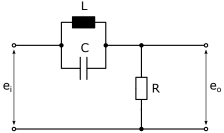
- Parallel L–C network in series with R.
- At resonance, the parallel LC presents a very high impedance, so almost no signal reaches the output.

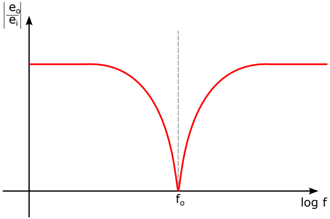
Notch behaviour

- Resonant (notch) frequency:

$$f_0 = \frac{1}{2\pi\sqrt{LC}} \quad [\text{Hz}]$$

- For $f \approx f_0$: output is strongly attenuated (deep notch).
- For $f \ll f_0$ or $f \gg f_0$: the circuit behaves approximately like a simple low–pass or high–pass path, so the signal is transmitted.

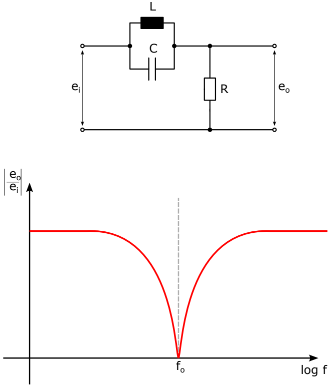




LCR band–stop (notch) filter and frequency response

Second–Order Band–Stop (Notch) LCR Filter

- Typical use: rejection of a single interfering tone (e.g. $50\,$Hz mains hum).



LCR band–stop (notch) filter and frequency response

Summary: Passive Filters and Their Limitations

Four basic passive filter types

- Low–pass, high–pass, band–pass, band–stop.
- Built from combinations of **R**, **L**, **C** elements.
- First–order and second–order sections can be cascaded to obtain desired shapes.

Higher–order and named responses

- By choosing component values appropriately in multi–section networks, one can approximate standard responses: Butterworth, Bessel, Chebyshev, Cauer, etc.
- These shapes trade off pass–band flatness, transition sharpness, and phase/step response.

Summary: Passive Filters and Their Limitations

Practical limitations of passive filters

- Inductors are bulky, heavy, lossy, and radiate magnetic fields (coupling/EMI issues).
- Component tolerances and loading can shift the cut–off and detune the filter.
- Passive networks cannot provide voltage or current gain; they only attenuate.

Motivation for next topic: Active filters using operational amplifiers overcome many of these limitations while implementing the same transfer functions.

Active Filters: Motivation and Basic Idea

- **Filter goal:** shape the frequency content of an analogue signal (noise rejection, anti–aliasing, channel selection, etc.).
- **Passive filters** use only R, L, C:
  - No power gain; they always *attenuate*.
  - Input and output impedances depend on R, L, $C \Rightarrow$ strong **loading** effects.
  - Inductors can be bulky, expensive, and magnetically noisy.
- **Active filters** use R, C and an **active device** (transistor or operational amplifier):
  - Provide *voltage or current gain* while filtering.
  - High input impedance, low output impedance $\Rightarrow$ minimal loading of signal source and load.
  - Inductor–like behaviour can be synthesised using R and C (*gyrator* (impedance converter) concept).
- In practice, modern active filters are almost always op-amp based; early examples used transistor emitter followers for buffering.

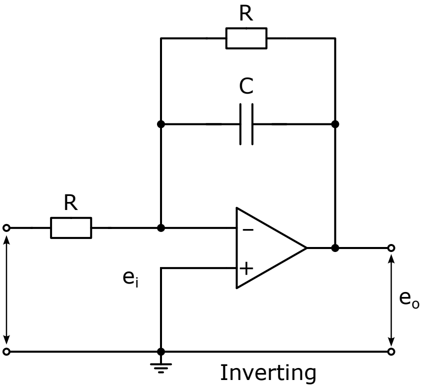
First-Order Active Low-Pass Filter (Op-Amp)

- Realised by placing a **single** CR **network** in the op-amp feedback path.
- For the inverting configuration:
  - Input resistor R from source e_i to inverting node.
  - Feedback branch: parallel connection of R and C.
  - Non-inverting input at reference (typically 0 V).
- Closed-loop transfer function (magnitude response):

$$\frac{e_o}{e_i(jf)} = \frac{1}{1 + j\dfrac{f}{f_c}}$$

with

$$f_c = \frac{1}{2\pi CR} \quad [\text{Hz}].$$



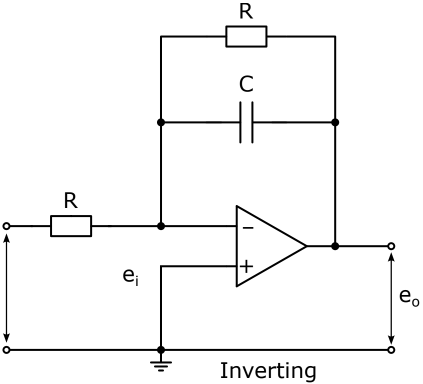
First–order active low–pass filter using op-amp

First-Order Active Low-Pass Filter (Op-Amp)

- **Frequency behaviour:**
  - $f \ll f_c$: $|e_o/e_i| \approx 1$ (pass–band, ≈ 0 dB).
  - $f = f_c$: $|e_o/e_i| = 1/\sqrt{2} \approx 0.707$ (-3 dB point).
  - $f \gg f_c$: $|e_o/e_i| \propto 1/f$ (slope -20 dB/decade).
- Phase shift:

$$\angle \frac{e_o}{e_i} = -\tan^{-1}\left(\frac{f}{f_c}\right)$$

→ 0° at low f, -45° at f_c, -90° at high f.

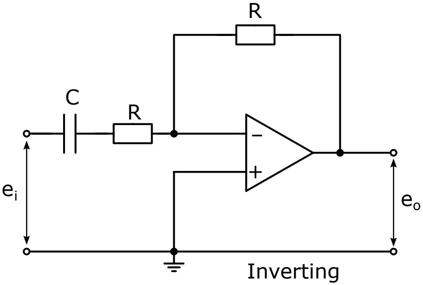


First–order active low–pass filter using op-amp

First-Order Active High-Pass Filter (Op-Amp)

- Swap the positions of R and C in the input branch:
  - Capacitor C in series with the input signal.
  - Resistor R to the virtual earth node.
  - Feedback is resistive so that high frequencies pass.
- Idealised transfer function:

$$\frac{e_o}{e_i(jf)} = \frac{j\,f/f_c}{1 + j\,f/f_c}, \qquad f_c = \frac{1}{2\pi CR}.$$



First–order active high–pass filter using op-amp

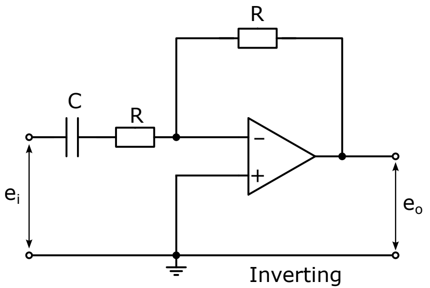
First-Order Active High-Pass Filter (Op-Amp)

- **Frequency behaviour:**
  - $f \ll f_c$: $|e_o/e_i| \approx f/f_c$ (strong attenuation of low frequencies; stop–band).
  - $f = f_c$: $|e_o/e_i| = 1/\sqrt{2}$ (-3 dB point).
  - $f \gg f_c$: $|e_o/e_i| \approx 1$ (flat pass–band, 0 dB).
- Phase shift:

$$\angle \frac{e_o}{e_i} = +\tan^{-1}\left(\frac{f_c}{f}\right)$$

$\rightarrow +90^\circ$ at very low f, $+45^\circ$ at f_c, 0° at high f.

- Practical note: at very low frequencies the input capacitor blocks DC and slow drifts, useful for removing offset or hum.

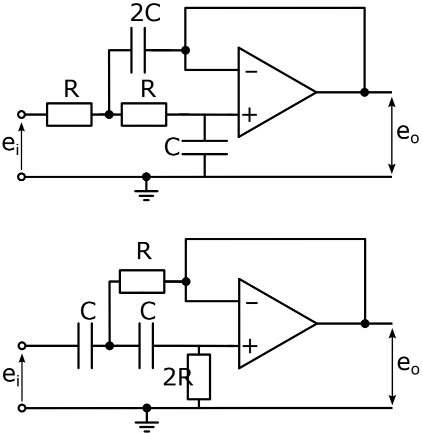


First–order active high–pass filters using op-amp

Second-Order Active Low- and High-Pass Filters

- By **adding a second** CR **branch** and using appropriate feedback, we obtain a **second–order** response:
  - Low–pass: two frequency–selective elements in the feedback.
  - High–pass: dual structure with series C and parallel R.
- Normalised low–pass transfer function (Butterworth choice):

$$\frac{e_o}{e_i(jf)} = \frac{1}{1 - \left(\frac{f}{f_c}\right)^2 + \sqrt{2}\,j\left(\frac{f}{f_c}\right)}.$$

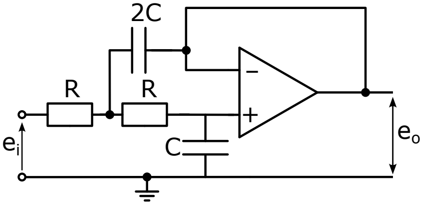


Second–order active low–pass filters using op-amp

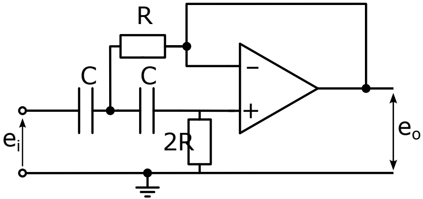
Second-Order Active Low- and High-Pass Filters

- Magnitude:

$$\left|\frac{e_o}{e_i}\right| = \frac{1}{\sqrt{\left[1-\left(\frac{f}{f_c}\right)^2\right]^2 + 2\left(\frac{f}{f_c}\right)^2}}.$$



- **Properties:**
  - Flat pass–band (Butterworth) with no ripple.
  - Slope in stop–band: -40 dB$/$decade (twice first–order).
  - "Knee" sharpness and damping determined by ratio of capacitor and resistor values.
- High–pass version has a dual transfer function with (f/f_c) in the numerator; its stop–band and pass–band are interchanged.



Second–order active low–pass filter using op-amp

Normalisation and Scaling of Active Filters

- Start from a **single–pole** active low–pass filter:

$$f_c = \frac{1}{2\pi CR} \quad \text{or} \quad \omega_c = \frac{1}{CR} \quad \text{[rad/s]}.$$

- **Normalised design:**
  - Choose $R = 1\ \Omega$, $C = 1$ F $\Rightarrow \omega_c = 1$ rad/s.
  - Easy to tabulate/filter responses as functions of ω/ω_c.
- **Impedance scaling** (e.g. to $500\ \Omega$ level):
  - Multiply *all* resistances by scaling factor k_Z.
  - Divide *all* capacitances by k_Z to keep CR constant.
  - Example: $k_Z = 500$ gives $R = 500\ \Omega$, $C = 1/500$ F $= 2$ mF, still $\omega_c = 1$ rad/s.
- **Frequency scaling** (change ω_c without changing impedance level):
  - To increase ω_c by factor k_f, reduce the product CR by k_f.
  - Often achieved by dividing either R or C by k_f (or both appropriately).
  - Example: from $\omega_c = 1$ rad/s to 50 rad/s at $R = 500\ \Omega$ requires $C = 1/(50 \times 500) = 40\ \mu$F.

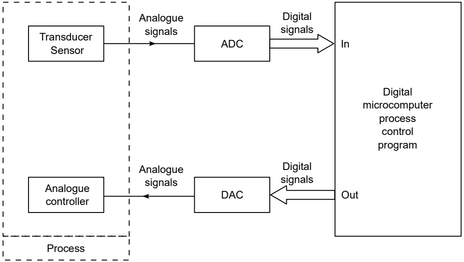
Active Filter Design: Summary for Practice

- **Step 1:** Choose required filter *type* and *order* (low–pass, high–pass, band–pass, band–stop; first or second order, etc.).
- **Step 2:** Select an appropriate *response family* (Butterworth for flat pass–band, Chebyshev for sharper roll–off, etc.).
- **Step 3:** Start from a **normalised** prototype (e.g. $R = 1\ \Omega$, C values from tables, $\omega_c = 1$ rad/s).
- **Step 4:** Apply *impedance scaling* and *frequency scaling* to obtain practical R, C and desired cut–off f_c.
- **Step 5:** Choose an op–amp:
  - Bandwidth $\gg$ required filter bandwidth.
  - Low input bias current and offset for low–frequency filters.
  - Adequate output swing and slew rate.
- **Step 6:** Verify with Bode plots (simulation $+$ hand sketch) that:
  - Pass–band gain and flatness meet requirements.
  - Cut–off frequency and slope are correct (20 dB/decade per pole).
  - Phase behaviour is acceptable for the intended application.

Analogue and Digital Conversions

Analogue and Digital Conversions: Motivation

- Modern control and instrumentation systems increasingly use **digital microcomputers** for:
  - process control,
  - data acquisition,
  - communication and supervision.
- However, the **process itself** and most sensors/actuators remain **analogue**:
  - transducers generate analogue voltages/currents,
  - actuators (valves, motors, heaters) are driven by analogue signals or PWM.
- Therefore we need:
  - **ADC** (Analogue-to-Digital Conversion) to bring sensor information into the digital domain.
  - **DAC** (Digital-to-Analogue Conversion) to send control actions back to the analogue process.



Typical analogue–digital control loop with ADC and DAC

Typical Analogue–Digital Control Loop

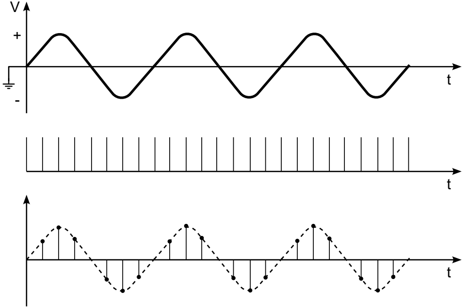
- **Transducer sensor** converts physical quantity (force, . . .) to analogue voltage or current.
- **ADC** periodically samples the analogue signal and produces a digital word for the microcomputer.
- **Digital microcomputer**:
  - executes control algorithm,
  - compares measured value with stored setpoint,
  - computes a new control output.
- **DAC** converts the digital control word to an analogue signal.
- **Analogue controller / actuator** uses this signal to adjust process variables (fuel valve, pump speed, heater power, . . .).

Key Design Requirements

- Conversion must be **fast enough** to follow process dynamics.
- Conversion must be **accurate enough** (resolution, noise, linearity).
- ADC and DAC must integrate with signal conditioning (filters, amplifiers, isolation).

Sampling of the Analogue Input

- Goal of sampling: represent a **continuous-time analogue waveform** by a sequence of discrete-time values.
- In figure:
  - (a) original analogue waveform,
  - (b) regularly spaced sampling instants,
  - (c) sampled amplitudes forming a pulse train.
- The **height** of each sample pulse equals the instantaneous value of the analogue signal at the sampling instant.
- The more samples per period, the better the sampled data follow the original waveform.



Sampling an analogue waveform

Nyquist Sampling Theorem and Practical Rule

Nyquist Sampling Theorem

If the highest frequency present in the analogue signal is $f_{\max}$, then:

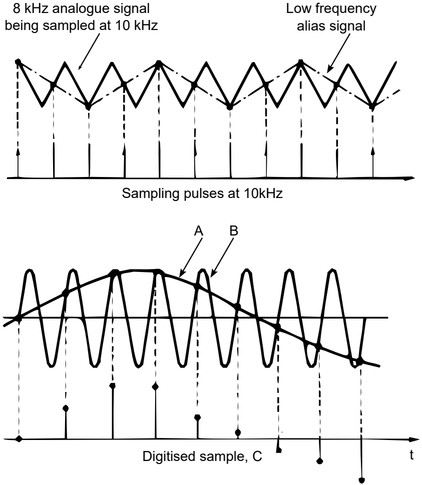
$$f_s \geq 2f_{\max}$$

is sufficient to reconstruct the signal exactly (in the absence of noise and with ideal filtering).

- $f_s =$ sampling frequency, period $T_s = 1/f_s$.
- At least **two samples per cycle** of the highest frequency component are required.
- In practice:
  - often choose $f_s \gg 2f_{\max}$ to simplify anti-alias filtering and to allow for non-idealities,
  - additional margin is useful when the signal bandwidth is not precisely known or may vary.
- If the sampling rate is too low, **aliasing** occurs.

Aliasing: When Sampling is Too Slow

- Aliasing occurs when $f_s < 2f_{\max}$.
- High-frequency components are **misinterpreted** as lower frequency signals in the sampled data.
- Figure shows:
  - (a) triangular wave of $8\,$kHz sampled at only $10\,$kHz: apparent digital waveform has a lower frequency (alias).
  - (b) two different analogue sinusoids A and B generating the same digitised sample sequence C.
- Result: the discrete-time representation contains **false frequency content** and cannot uniquely represent the original waveform.



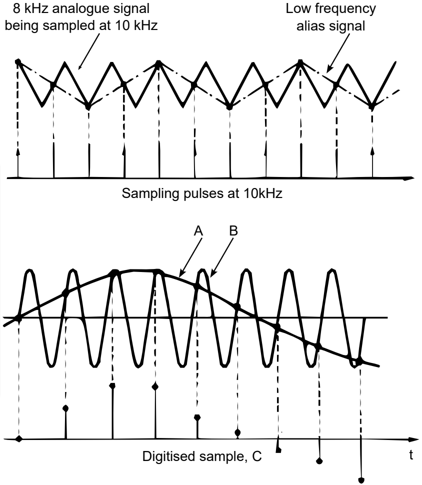
8 kHz analogue signal being sampled at 10 kHz
Low frequency alias signal
Sampling pulses at 10kHz
A
B
Digitised sample, C
t

Aliasing of a wave sampled too slowly

Aliasing: When Sampling is Too Slow

Anti-aliasing

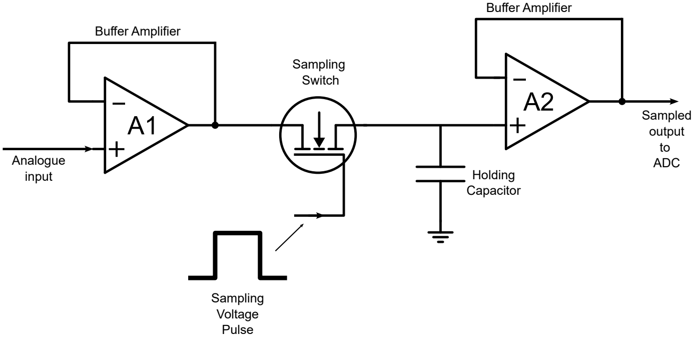
- Use an **analogue low-pass filter** before the ADC input.
- Limit input spectrum to below $f_s/2$ to prevent high-frequency components from folding into the band of interest.


8 kHz analogue signal being sampled at 10 kHz
Low frequency alias signal
Sampling pulses at 10kHz
A
B
Digitised sample, C
t

Aliasing of a wave sampled too slowly

Sample-and-Hold Circuit: Why It Is Needed

- Practical ADCs require a **constant input voltage** during the conversion time.
- A **sample-and-hold** (S/H) circuit:
  - samples the analogue signal during a short *aperture time*,
  - stores it on a capacitor,
  - holds it essentially constant while the ADC performs conversion.

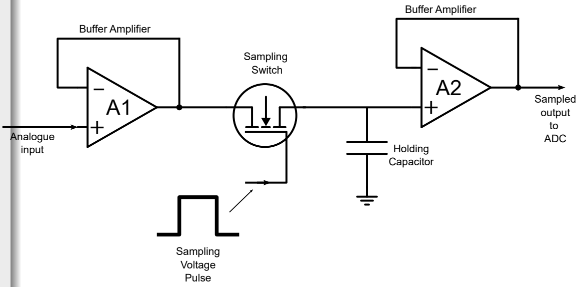


Sample-and-hold timing diagram

Sample-and-Hold Circuit: Why It Is Needed

Typical Sample-and-Hold Arrangement

- Buffer amplifier A1 isolates the signal source and charges the hold capacitor C rapidly.
- MOSFET sampling switch connects C briefly to the analogue input when a sampling pulse arrives.
- During *hold*, A2 buffers the capacitor, presenting a stable voltage to the ADC.
- Non-idealities: **droop** due to leakage currents, finite acquisition time, aperture uncertainty (timing jitter).



Sample-and-hold timing diagram

Quantisation in an N-bit ADC

- After sampling, each analogue value must be mapped to one of 2^N discrete **quantisation levels**.
- Example:
  - Analogue range: 0 to 3.5 mV.
  - 3-bit ADC $\Rightarrow 2^3 = 8$ levels, spacing $\Delta V = 0.5$ mV.
  - Each level assigned a 3-bit binary code (000 to 111).
- Any analogue value is replaced by the **nearest** level: *quantising*.

Resolution (Quantum Size)

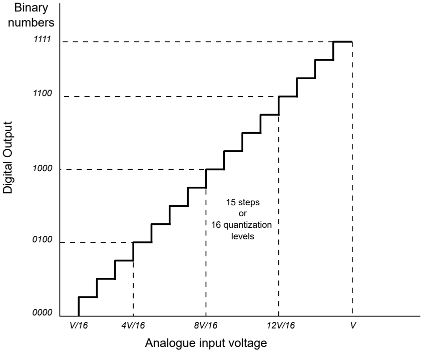
For a full-scale range V_{FS} and N bits:

$$\Delta V = \frac{V_{\mathsf{FS}}}{2^N - 1}$$

- ΔV is both the smallest step in output code and the smallest detectable change (ideal case).
- Increasing N reduces ΔV and thus improves measurement resolution.

Quantisation Error and Resolution Examples

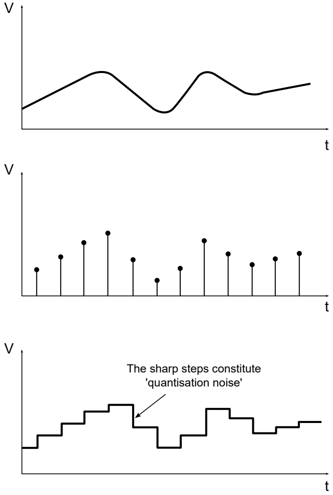
- Quantisation error is typically taken as $\pm\frac{1}{2}\Delta V$.
- For the 3-bit example (range 0–3.5 mV, $\Delta V = 0.5$ mV):
  - quantisation error $\approx \pm 0.25$ mV.
  - binary code 010 represents all inputs between 0.75 mV and 1.25 mV.
- Increasing bit length:
  - 4 bits: error reduces to about ± 0.117 mV.
  - 8 or 12 bits: error becomes very small (sub-microvolt range in the example).



Quantisation levels and "staircase" trasfer characteristics for 4-bit ADC

Quantisation Noise

- The analogue signal is continuous; the quantised version has **discrete steps**.
- Figure illustrates:
  - (a) original smooth analogue waveform,
  - (b) discrete sampled points,
  - (c) reconstructed waveform showing a jagged, castellated shape.
- The difference between the original and quantised/reconstructed waveforms appears as **quantisation noise**.
- It is often modelled as an *approximately white* noise source uniformly distributed between $\pm \Delta V/2$ (for many codes and no overload).

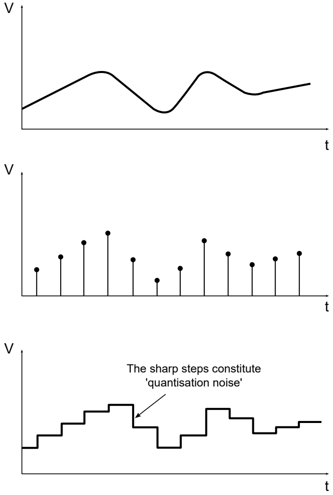


Quantisation noise due to finite resolution

Quantisation Noise

Design Trade-off

- More bits ⇒ smaller steps ⇒ lower quantisation noise but higher converter complexity, cost, and conversion time.
- Choice of N depends on required accuracy, noise floor, and bandwidth.



Quantisation noise due to finite resolution

Digital-to-Analogue Conversion (DAC): Overview

- DAC performs the **reverse** of ADC: converts digital word to analogue output voltage or current.
- Widely used for:
  - generating analogue control signals from a microcontroller,
  - reconstructing waveforms from digital data (audio, arbitrary waveform generation),
  - implementing many ADC architectures internally.
- Key performance parameters:
  - resolution (number of bits),
  - settling time (time to reach final value within a specified error after a code change),
  - linearity and monotonicity,
  - output range and drive capability.

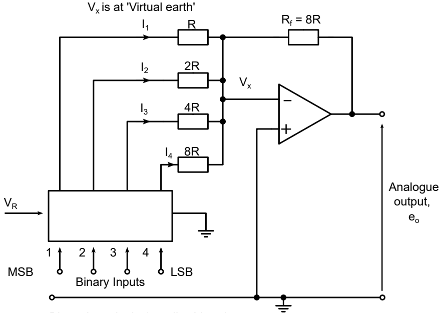
Digital Word

- Input to DAC is a binary number (word).
- Each bit contributes a weighted current or voltage.
- Analogue output represents the sum of all weighted contributions.

Types of DAC: Weighted Resistor DAC: Principle

- Consider the 4-bit DAC of figure using an op-amp summing amplifier.
- Each digital input bit controls a switch connecting either:
  - reference voltage V_R (logic 1), or
  - ground (logic 0)

  to a resistor branch.
- Resistors are weighted: R, $2R$, $4R$, $8R$.
- Currents I_1 to I_4 from each branch sum at the op-amp input node V_X (virtual earth).



Binary input logic 1 applies V_R to input

Binary input logic 0 applies 0 V to input

Binary Input	e$_o$
1111	15 V_R
1010	10 V_R
0011	3 V_R
⋮	⋮
0000	0

4-bit weighted resistor DAC using op-amp summing amplifier

Types of DAC: Weighted Resistor DAC: Principle

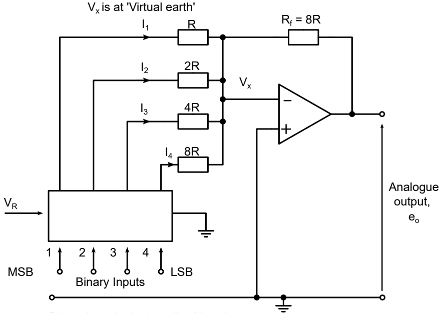
Current and Output Voltage

For a given digital input pattern:

$$I_f = I_1 + I_2 + I_3 + I_4$$

$$e_o = I_f R_f$$

where R_f is the feedback resistor (often chosen as $8R$ in the example to give convenient scaling).



Binary input logic 1 applies V_R to input

Binary input logic 0 applies 0 V to input

Binary Input	e_o
1111	15 V_R
1010	10 V_R
0011	3 V_R
⋮	⋮
0000	0

4-bit weighted resistor DAC using op-amp summing amplifier

Worked Example: 4-bit Weighted Resistor DAC

Example:

- Resistor values: R, $2R$, $4R$, $8R$.
- Feedback resistor: $R_f = 8R$.
- Reference voltage: V_R.
- Digital input word: 1010 (MSB to LSB).

Step 1: Determine Branch Currents

- Bits 1 and 3 are 1; bits 2 and 4 are 0.
- Therefore:

$$I_1 = \frac{V_R}{R}, \quad I_3 = \frac{V_R}{4R}, \quad I_2 = I_4 = 0$$

- Total current:

$$I_f = I_1 + I_2 + I_3 + I_4 = \frac{V_R}{R} + \frac{V_R}{4R} = \frac{5V_R}{4R}$$

Worked Example: 4-bit Weighted Resistor DAC

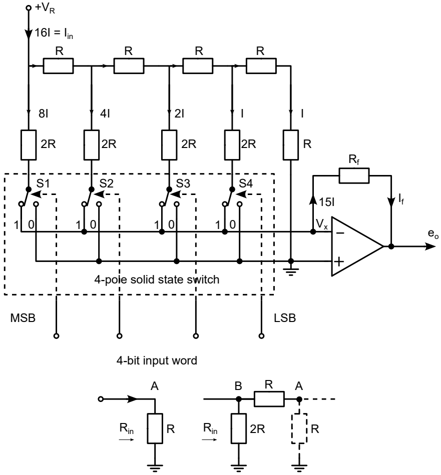
Step 2: Output Voltage

$$e_o = I_f R_f = \frac{5V_R}{4R} \times 8R = 10V_R$$

Comment: extending this approach to 8 or 12 bits requires many precise resistor values, which is why practical DACs prefer the R–2R ladder implementation.

Types of DAC: R–2R Ladder DAC

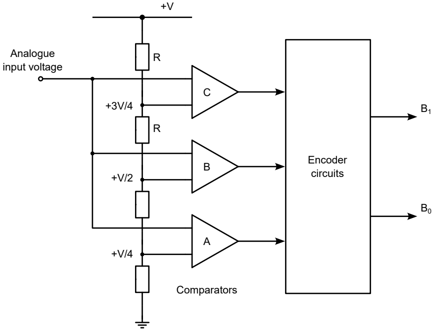
- The R–2R ladder DAC uses only two resistor values, R and $2R$, arranged in a repeated ladder network.
- Digital switches connect each ladder node either to ground (bit 0) or to a reference voltage (bit 1).
- The ladder presents a constant input resistance to the reference source, independent of word length.
- For a 4-bit ladder with input word 1111, the combined current through the feedback resistor R_f produces the maximum output voltage.



4-bit R–2R ladder DAC using op-amp summing amplifier

Types of ADC: Flash (Parallel) ADC

- A flash ADC uses a bank of comparators in parallel, each set to a different reference threshold.
- The analogue input is compared simultaneously with all thresholds; the pattern of comparator outputs is encoded into a binary word.
- Provides extremely fast conversion (sub-20 ns) at the cost of large hardware: $2^N - 1$ comparators for N bits.
- Common for very high-speed, low-to-medium resolution applications (oscilloscopes, RF receivers).

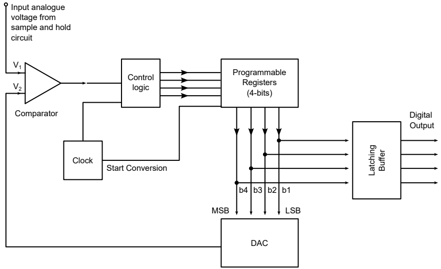


Analogue Input Voltage	Comparator Outputs			Binary Outputs	
	A	B	C	B_0	B_1
0 to V/4	0	0	0	0	0
V/4 to V/2	1	0	0	0	1
V/2 to 3V/4	1	1	0	1	0
3V/4 to V	1	1	1	1	1

Flash ADC block diagram and example with 2-bit output

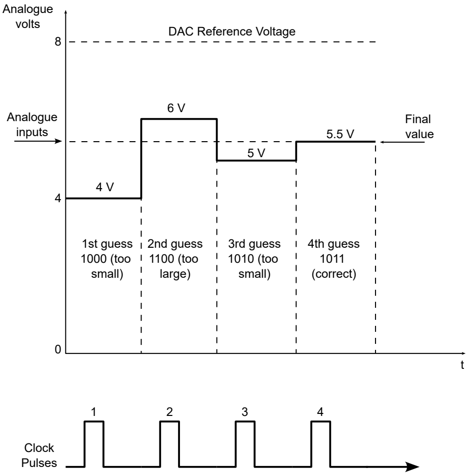
Types of ADC: Successive Approximation ADC

- Uses a binary search to converge on the input voltage.
- A DAC generates trial voltages from a programmable register; these are compared with the input using a single comparator.
- Starting from MSB, each trial bit is set or cleared depending on whether the DAC output is too high or too low.
- Requires exactly N comparison steps for an N-bit result; offers good speed and resolution with moderate hardware.



Successive Approximation ADC block diagram

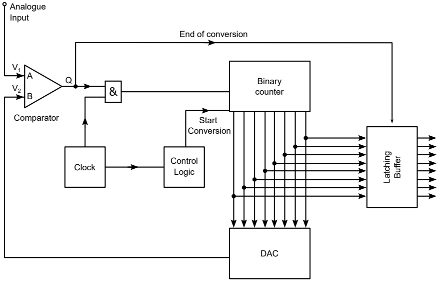
Types of ADC: Successive Approximation ADC



Successive Approximation ADC block diagram

Types of ADC: Counter-Ramp (Integrating) ADC

- A DAC generates a staircase or ramp voltage V_2 controlled by a binary counter.
- The ramp is compared with the analogue input V_1; while $V_2 < V_1$ the counter is allowed to increment.
- When V_2 exceeds V_1, the comparator disables the clock and the counter value is latched as the digital result.
- Conversion time is proportional to input level and limited by the maximum count; simpler but usually slower than SAR or flash ADCs.



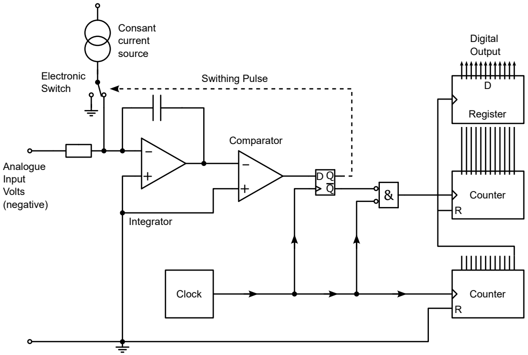
Counter-ramp ADC block diagram

Other Integrating ADC Types

- **Single-slope ramp ADC**: uses a linear ramp generated by an integrator. The time taken for the ramp to reach the input voltage is measured with a counter. Simple but sensitive to ramp slope and clock errors.
- **Dual-slope (dual-ramp) ADC**: first integrates the unknown input for a fixed time, then integrates a reference voltage of opposite polarity until the integrator output returns to zero. The second integration time gives the digital result.
- Dual-slope conversion averages the input over many clock cycles and strongly rejects noise and component drift, at the expense of slower conversion.
- These concepts motivate more advanced integrating schemes such as **delta–sigma** converters.

Types of ADC: Delta–Sigma Charge-Balance ADC: Principle

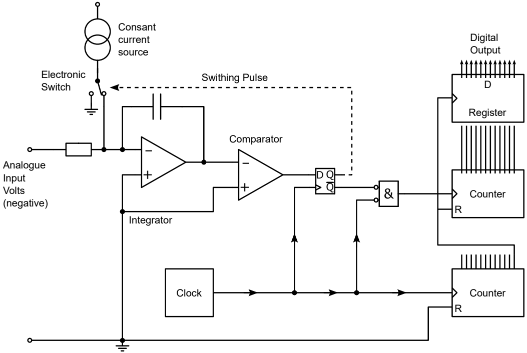
- Aim: obtain high resolution by converting the *average* analogue input over many clock cycles into a digital count.
- The analogue input current charges the integrator capacitor; the integrator output is compared with zero.
- Depending on comparator output, fixed-size current pulses (derived from a reference source) are switched into or out of the summing junction.
- The feedback pulses balance the input on average, keeping the integrator output bounded around zero.



Delta–sigma charge-balance ADC block diagram

Types of ADC: Delta–Sigma Charge-Balance ADC: Operation

- Over a window of N clock periods, a total of n charge pulses is applied to balance the input.
- The condition of zero net charge at the summing node implies that n is proportional to the average analogue input current.
- The digital output is simply the pulse count n (or its binary representation), giving very fine effective resolution.
- Key advantages: excellent noise averaging, high linearity and suitability for low-frequency, high-accuracy measurement (e.g. digital multimeters).



Delta–sigma charge-balance ADC block diagram

Table of contents

Processor Technologies in Automation Systems

Dr Bikash Sah

Types of Processors used in Automation Systems

What is a Processor?

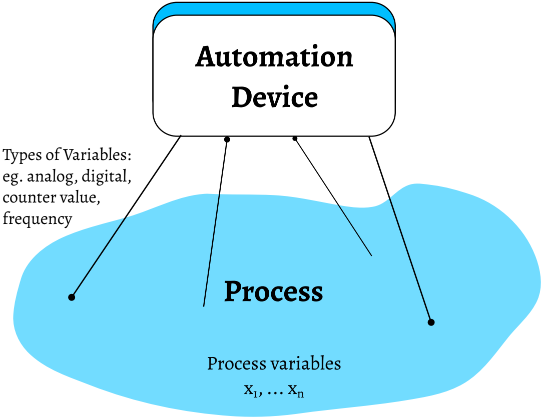
- A processor is an electronic component that executes instructions to perform tasks.
- It serves as the "brain" of an automation system, processing data and controlling operations.
- Processors can vary in complexity, speed, and functionality depending on the application.
- They are essential for executing control algorithms, managing communication, and interfacing with sensors and actuators.

Types of Processors:

- Microcontrollers (MCUs)
- Digital Signal Processors (DSPs)
- Field-Programmable Gate Arrays (FPGAs)
- Application-Specific Integrated Circuits (ASICs)
- Programmable Logic Controllers (PLCs)

Programmable Logic Controllers (PLCs)

Interface of Automation Device - Process



Automation Device

Types of Variables: eg. analog, digital, counter value, frequency

Process

Process variables

$x_1, \ldots x_n$

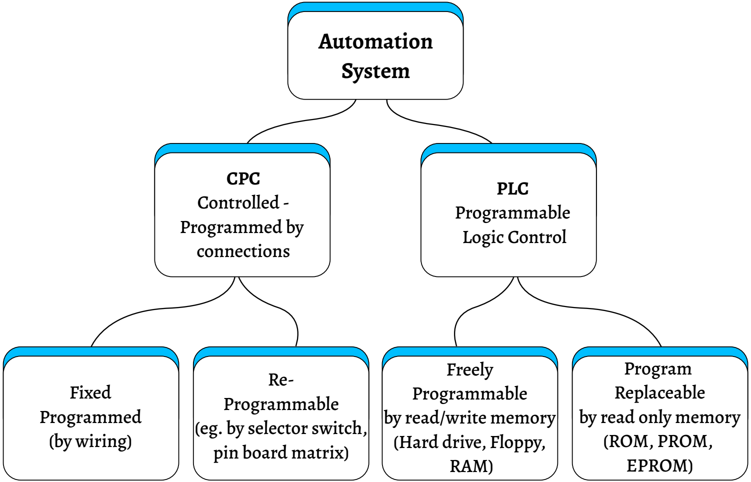
Introduction to basics of Industrial Automation

Terms

1. **Logic Control** - assigns certain logic states of the output signals to certain states of the input signals with the help of boolean operations.
2. **Sequence Control** - a control method with an inevitably stepwise sequence, in which the transition from one step to the following step(s) according to the program takes place depending of transition conditions.
3. **Program of a Control** - Complete set of instructions and declarations to process signals and control a system (or plant) so it behaves as required.

Remark: This definition is valid independently of the signal processing being connection oriented or memory programmed.

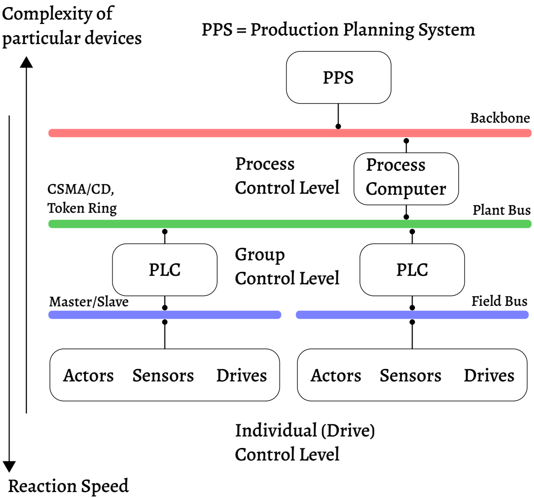
Different Possibilities of Program Realization


Automation System
CPC
Controlled - Programmed by connections
PLC
Programmable Logic Control
Fixed Programmed (by wiring)
Re-Programmable (eg. by selector switch, pin board matrix)
Freely Programmable by read/write memory (Hard drive, Floppy, RAM)
Program Replaceable by read only memory (ROM, PROM, EPROM)

Definitions regarding program implementation

1. **Connection oriented control** - control program is defined by the type of the functional units and by their connections.
   - **Control with fixed program** - a connection oriented control, where program modifications are not planned.
   - **Reprogrammable control** - a connection oriented control, where program modifications are provided for and are easy to carry out.
2. **Programmable logic control** - control program is stored in a program memory.
   - **Freely programmable control** - a programmable logic control with read/write memory (RAM) as programm memory, whose entire content can be modified even in arbitrarily small extent without mechanical intervention into the control device.
   - **Control with replaceable program** - a programmable logic control with read-only memory (ROM) as program memory, whose content can be modified after programming being once completed only by a mechanical intervention into the control device.

Hierarchical arrangement of an Automation System



Definitions regarding control hierarchy

1. **Individual control, drive control** – a functional unit for the control of a single control element or a drive.
    - **Individual control level, drive control level** – summarizes all parts of a control system that act directly on the process by means of final control elements or drives.
2. **Group control** – a functional unit for the control of a coherent process part, which is above associated individual or drive controls.
    - **Group control level** – summarizes all parts of the control unit that act on certain partitions of the individual control level.
3. **Process control** – a functional unit above the the group controls responsible for the control of the entire process.
    - **Process control level** – summarizes all parts of the control unit that act on the group control level.

Software of PLC devices

Organization of operational software

For structuring of the software in PLC devices so called blocks or program organization entities (POEs) are employed. The block types are separated into

- Organizational blocks (OB)
- Function blocks (FB) and functions (FC)
- Data blocks (DB)
- System functions (SFC) and system function blocks (SFB)

A block is a part of the application program grouped by function, structure, or purpose. There are:

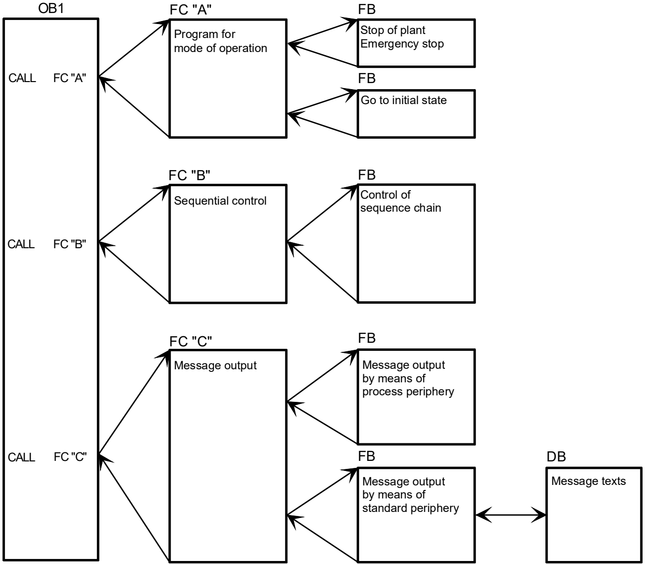
- Code blocks (OBs, FCs, FBs) that contain instructions for data processing
- Data blocks (DBs) that store data

Some organizational blocks (OBs) form the interface between the application program and the system program. They are triggered by the system and contain user-written code. By placing code in specific OBs, the user determines whether it runs cyclically, on a timer, or in response to an alarm.

Program Blocks Overview

1. **Organizational Blocks (OBs)** Interface between system program and application program. They are called by the operating system and determine: cyclical and alarm-controlled execution, startup behavior, error handling. OBs are divided into:
   - **System-called OBs**: control program flow, startup, and error response; user programs these.
   - **Integrated special functions**: built-in system functions called when needed by the application.
2. **Functions (FCs)** Code blocks *without memory*. They are parameterizable and return a value to the calling block.
3. **Function Blocks (FBs)** Code blocks *with memory*. Used for recurring or complex tasks (e.g., digital logic, sequences, closed-loop control). They can be called multiple times with different parameters.
4. **Data Blocks (DBs)** Contain fixed or variable data used by the program. They hold no executable code. Data may include: bit patterns, numeric values, text, various data types.
5. **System Functions (SFCs) and System Function Blocks (SFBs)** Prefabricated operating-system blocks for tasks such as communication or built-in system operations.

Raw structuring of programs


OB1
CALL FC "A"
CALL FC "B"
CALL FC "C"
FC "A"
Program for mode of operation
FB
Stop of plant Emergency stop
FB
Go to initial state
FC "B"
Sequential control
FB
Control of sequence chain
FC "C"
Message output
FB
Message output by means of process periphery
FB
Message output by means of standard periphery
DB
Message texts

Process Image Table

Process Image of Inputs (PII)

- At the start of every cycle, all input states are scanned.
- These values are stored in the Process Image Input table.
- During program execution, the PLC uses the stored values.
- Ensures consistent input data throughout the cycle.

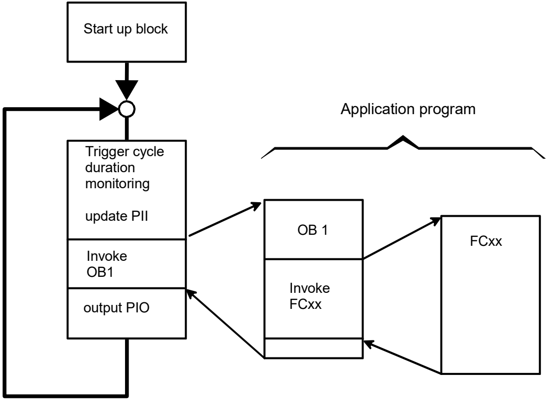
Process Image of Outputs (PIO)

- Output values written by the program are stored in the PIO table.
- At the end of the cycle, the table is copied to actual outputs.
- Then a new cycle begins.

Cyclic Processing in PLC

In a programmable logic controller (PLC), the program usually runs in a cycle (loop).

- PLC programs normally run in a **cyclic loop**.
- After switching on, the PLC executes a **start-up block** once.



Cyclic Program Scan

1. Start at the **top of the main program** (e.g., Siemens: OB1).
2. Execute each instruction **sequentially** to the end.
3. Run internal **system routines**.
4. Restart from the **top of the program**.

Cyclic Processing in PLC

Program Blocks

- Functions and function blocks can be called inside the main block.
- Nested calls are possible (depth depends on the PLC manufacturer).

Processing Time (Scan Time)

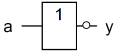
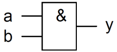
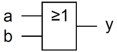
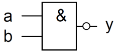
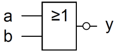
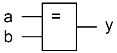
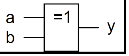
- Total cycle time = sum of processing times of all called blocks.
  $T_{cyc} = T_{read\_PII} + T_{exec\_prog} + T_{write\_PIO} + T_{overhead}$
  where, T_{exec_prog} = sum of execution times of all blocks called in the cycle, T_{read_PII} = time to read process image of inputs, T_{write_PIO} = time to write process image of outputs, $T_{overhead}$ = additional overhead time.
- If a block is called n times, its time is counted n times.

Interrupts in Cyclical Processing

Cyclical program processing can be interrupted by:

- **Alarm processing**
  - The normal cycle is interrupted by an external or internal interrupt signal.
  - The PLC executes a special block (Interrupt Service Routine).
  - Allows immediate reaction to important process events.
  - After finishing, the PLC returns to the point where it was interrupted.
- **Time-controlled processing**
  - An internal clock triggers an interrupt at defined time intervals.
  - The PLC runs a specific time-controlled program block.
  - After execution, the PLC returns to the normal cycle.
  - Often used in closed-loop control tasks requiring precise timing.
- **Programming failures and device failures**
- **Signals from the hardware**
- **Change of mode of operation**
  - Caused by the device's mode switch
  - Or by the programming tool

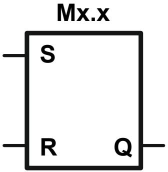
Building blocks of PLC: Boolean Operations

Math. Symbol	Name	Definition					Operation	Graphical Symbol
		a = b =	0 0	1 0	0 1	1 1		
‾ ¬	Negation	a = 1 a = 0	y = 0 y = 1				$y = \bar{a} = \neg a$; (a’, \a, /a) not a	
$\wedge$	Conjunction AND	y =	0	0	0	1	$y = a \wedge b$; a and b	
$\vee$	Disjunction OR	y =	0	1	1	1	$y = a \vee b$; a or b	
$\overline{\wedge}$	NAND	y =	1	1	1	0	$y = \overline{a \wedge b}$; a nand b	
$\overline{\vee}$	NOR	y =	1	0	0	0	$y = \overline{a \vee b}$; a nor b	
≡	Equivalence (E)XNOR	y =	1	0	0	1	$y = a \equiv b$; a equivalent b a EXNOR b	
≠ ⊗	Antivalence (E)XOR	y =	0	1	1	0	$y = a \otimes b$; a antivalent b a EXOR b	

Building blocks of PLC: RS Flip-Flop

RS Flip-Flop:
A R/S flip flop is a storing function with two input variables R and S and an output variable Q, where the value Q = 1 belongs to the input states R = 0 and S = 1 and Q = 0 belongs to the input states R = 1 and S = 0

R	S	Q_{k+1}
0	0	Q_k
0	1	1
1	0	0
1	1	0 (R has priority)



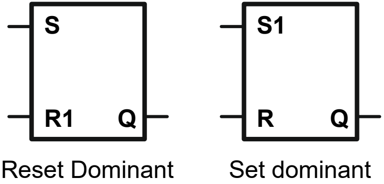
Building blocks of PLC: RS Flip-Flop

Priority rules:

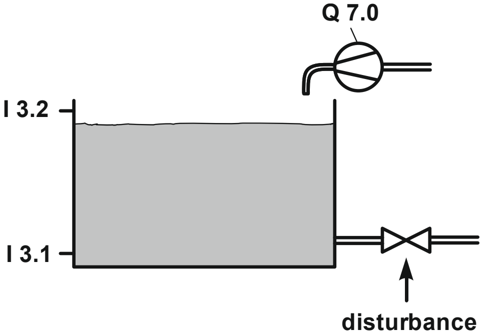
- If **R has priority**: $R = 1$ dominates and forces $Q = 0$.
- If **S has priority**: $S = 1$ dominates and forces $Q = 1$.

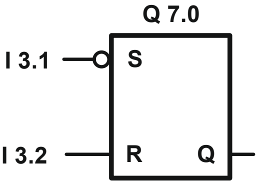
In PLC software:

- Priority is determined by the **order of instructions**.
- The instruction processed **last** determines the final result.



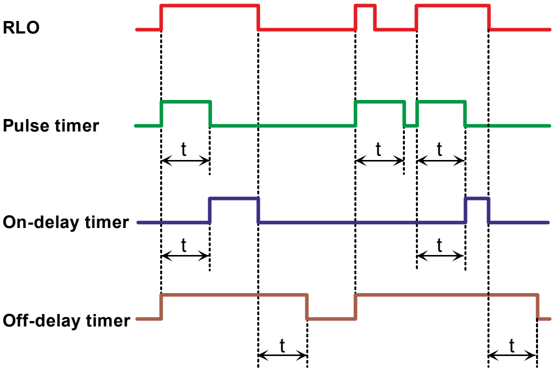
RS Flip-Flop Application Example (Level Control)





Timer Characteristics

Syntax	Meaning
SI T	Start a timer as a pulse timer
SE T	Start a timer as an on-delay timer
SA T	Start a timer as an off-delay timer



Building blocks of PLC: Data Types (elementary e.g. from SIEMENS)

Data Type	Description (Width)	Example
BYTE	Byte, 8-bit hexadecimal number, 8 Bits	B#16#01
CHAR	One character (ASCII), 8 Bits	'A'
WORD	16-bit hexadecimal, 16-bit binary, count value, 3 decades BCD, 2x8-bit unsigned decimal numbers	W#16#F100, 2#1100_0011_1001_0110, C#250, B#(20,103)
DWORD	32-bit hexadecimal, 32-bit binary, 4x8-bit unsigned decimal numbers, 32 Bits	DW#16#E800_0054, 2#00111100_11000011_10010110_11110000, B#(1,55,3,0)
INT	Fixed-point number, 16 Bits, -32768 to +32767	1234
DINT	Fixed-point number, 32 Bits, -2 147 483 648 to +2 147 483 647	12345678
REAL	Floating-point number, 32 Bits, decimal or exponential	+1.234567E-01
S5TIME	Time value in S5-format, 16 Bits	S5T#0ms, S5TIME#2h46m30s
TIME	Time value in IEC format, 32 Bits	T#0ms, TIME#24d20h31m23s647ms
DATE	Date, 16 Bits	D#1990-01-01, DATE#2168-12-31
TIME_OF_DAY	Time of day, 32 Bits	TOD#00:00:00, TIME_OF_DAY#23:59:59.999

Extended Data Types

Data Type	Description / Example
DATE_AND_TIME	Date and time, 64 Bits Examples: DT#1990-01-01-00:00.000, DATE_AND_TIME#2168-12-31:23:59:59.999
STRING	String of ASCII characters, variable length Example: 'String 1'
ARRAY	Array of components with same data type Example: ARRAY[10] OF INT
STRUCT	Structure of components with arbitrary data types Example: STRUCT { INT x; REAL y; }

Building blocks of PLC: Operand Types

Type	Indicator / Description
Constants	*)
Inputs	E, %I *)
Outputs	A, %Q *)
Memory data	M, %M *)
Local data	L
Periphery inputs	PE
Periphery outputs	*)
Timers	T
Counters	C
Data	D *)
System data	SM *)

Note: The indicator '*)' can be supplemented by a second character (e.g., W for WORD or D for DOUBLE WORD) to determine the type exactly.

Building blocks of PLC: Compare and Arithmetic Operations

- **Compare operations:** $<$, $>$, $\leq$, $\geq$, $==$, $\neq$
  - Result: Boolean variable
  - Can be used in binary operations or control program flow
- **Basic arithmetic:** ADD, SUB, MUL, DIV
  - SIEMENS: $+$, $-$, $*$, $/$ with type specifier: I (INT), D (DINT), R (REAL)
  - Fixed-point multiplication: width doubles (INT 16-bit × INT 16-bit → DINT 32-bit)
  - Division: result type $=$ Dividend type
  - Floating point: watch exponent overflow
  - Always handle division by 0

Building blocks of PLC: Load, Assignment and Organizational Functions

- **Load operations:** L or LD
  - Provides input variable for an operation
- **Assignment / Transfer:** T or ST
  - Assign value to a variable, e.g., accumulator
  - Can also be implemented with ‘=‘
- **Organizational functions:** Branching
  - Conditional/unconditional based on RLO or internal status
  - Syntax examples: SPA [label], SPB [label], SPO [label]

Building blocks of PLC: Shift Functions and Type Conversions

- **Shift and rotate operations:**
  - Shift left/right, rotate left/right on accumulator or memory
  - Examples: SLW 3 (shift left word by 3 bits), SRW 5 (shift right words by 5 bits), RLD 6 (rotate left word by 6 bits)
- **Type conversions:**
  - BTI: BCD → INT
  - ITB: INT → BCD
  - DTR: Fixed-point → Floating-point
  - RND: Floating-point → Fixed-point

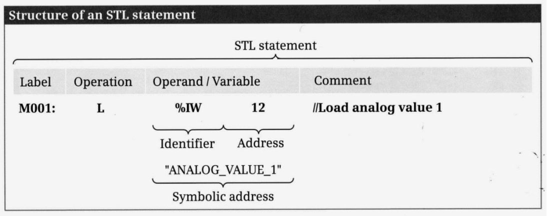
Building blocks of PLC: Statement List (STL) - Control Statements

- A statement list consists of multiple control statements.
- Each control statement:
  - Starts on a new line
  - Contains an **operator** (optionally with a modifier)
  - May contain one or more operands, separated by commas
  - May include a preceding label followed by a colon (:)
  - May include a comment at the end of the line
- Blank lines can be inserted between statements
- For binary operands, the pre-occupation of Result of Logic Operation (RLO) can be done by the first statement of an AND or OR operation
- **Note:** Nested logic (use of ‘(‘ and ‘)‘) requires a stack of RLOs

Structure and Example of STL

Example Statement Fields:

Label	Operator	Operand	Comment
START:	L	I0.1	Pushbutton
	AN	M0.5	Not locked
	S	Q0.2	Fan on



Structure of an STL statement

STL statement

Label	Operation	Operand / Variable		Comment
M001:	**L**	**%IW**	**12**	**//Load analog value 1**
		Identifier	Address	

"ANALOG_VALUE_1"

Symbolic address

Building blocks of PLC: Operations, Modifiers, and Operands

- Default semantics:

  $$\text{RLO} = \text{RLO OP Operand}$$

  where RLO = result of the preceding logic operation.
- Example:
  - Statement: `A "Value"`
  - Interpreted as: `RLO = RLO AND "Value"`
- Modifier `N`: boolean negation of the operand
  - Statement: `AN "INPUT_1"`
  - Interpreted as: `RLO = RLO AND NOT("INPUT_1")`
- Modifier `(`: suspend processing until `)` occurs
  - Statements:

    ```
    A(
    O "INPUT_2"
    O "INPUT_3"
    )
    ```
  - Interpreted as: `RLO = RLO AND ("INPUT_2" OR "INPUT_3")`
- Modifier `C` (conditionally): statement processed only if previous RLO = 1 (or 0 if combined with N-modifier)

Operators - Part 1

Operator	Modifier	Operand	Semantics
A	N, (	Boole	Boolean AND
O	N, (	Boole	Boolean OR
N	(	Boole	Boolean NOT
XOR	N, (	Boole	Boolean EXCLUSIVE OR
=, ST, T	N, see Rem. 3	All variable types	Assignment
S	see Rem. 1	Boole	Sets boolean operand to ”1”
R	see Rem. 1	Boole	Sets boolean operand to ”0”
)			Evaluation of suspended operation
CF		Counter value	Count forward
CB		Counter value	Count backward
ADD, +		Integer, Real	Addition
SUB, -		Integer, Real	Subtraction
MUL, *		Integer, Real	Multiplication
DIV, :		Integer, Real	Division
GT, ¿	see Rem. 2	Integer, Real	Comparison: ¿
GTE, ¿=	see Rem. 2	Integer, Real	Comparison: ¿=
EQ, =	see Rem. 2	Integer, Real	Comparison: =

Operators - Part 2

Operator	Modifier	Operand	Semantics
LT, ¡	see Rem. 2	Integer, Real	Comparison: ¡
LTE, ¡=	see Rem. 2	Integer, Real	Comparison: ¡=
BTI		Integer	Code converter Decimal/Binary
ITB		Integer	Code converter Binary/Decimal
NOP			Null operation
L		All variable types	Load (provides operands for subsequent operations)
SP	C	Label	Jump to label
BA		Name	Invocation of a block
CALL		Name	Invocation of a block
BE		C	Block end
C			These operations are only processed when RLO = ”1”

Rem. 1: Only processed if RLO = ”1”.

Rem. 2: Result is of Type ”Bool”.

Rem. 3: Only with binary operand.

LAD (Ladder Diagram)

- A **LAD program** provides processing, testing, and modification of data using standardized graphical symbols.
- These symbols are arranged in **networks** that resemble the rungs of a relay logic diagram.
- LAD networks are bounded on the left and right by **conductor rails**.

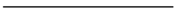
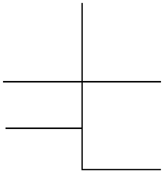
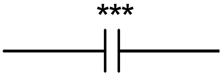
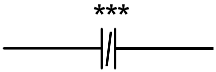
Conductor Rails and Connection Elements

- A LAD network is bounded by vertical lines called **conductor rails**:
  - **Left conductor rail**: mandatory, state is **ON** unless connected to an inactive step
  - **Right conductor rail**: optional, no defined state
- **Connection elements** are horizontal or vertical lines:
  - States: **ON** (1) or **OFF** (0)
  - **Horizontal lines**: transfer state from left to right
  - **Vertical lines**: state = OR of all connected left horizontal lines
    - **OFF**: all left lines OFF
    - **ON**: at least one left line ON
  - Vertical line state determines all horizontal lines on the right
  - States are **not copied** to lines on the left

Scan Operations and Assignment Operations

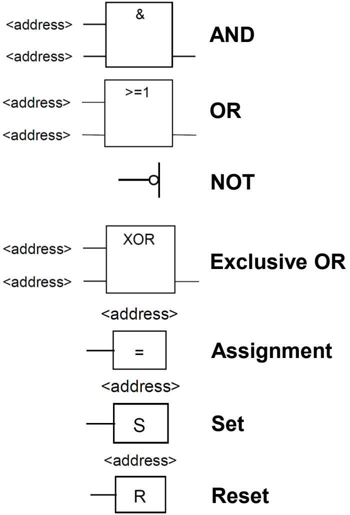
- **Scan Operations (Contacts)**:
  - Pass the state of the horizontal connection on the left to the right.
  - Only passes the state if the associated boolean input, output, or memory variable is **ON**.
  - Does **not** change the state of the boolean variable itself.
- **Assignment Operations (Coils)**:
  - Copy the state of the left-hand connection to the right-hand connection.
  - Update the associated boolean variable with the left-hand connection state.
  - No modification to the signal occurs; the coil reflects the left-hand state.

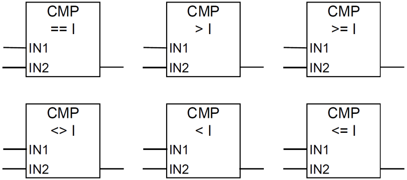
LAD Symbols

Symbol	Description
	Horizontal connection
	Vertical connection in combination with horizontal connections
	"NO Contact" (normally open) The state of the left connection is passed to the right connection, if the state of the corresponding boolean variable (marked with ***) is ON. Otherwise the state of the connection to the right is OFF.
	"NC Contact" (normally closed) The state of the left connection is passed to the right connection, if the state of the corresponding boolean variable (marked with ***) is OFF.

Symbol	Description
*** ——( )——	"Coil" The state of the left hand connection is copied to the boolean variable and to the right hand connection.
*** ——(/)——	"Negative Coil" The state of the left hand connection is copied to the right hand connection. The inverse state of the left hand connection is copied to the associated boolean variable i.e. if the state of the left hand connection is OFF the state of the associated variable is set to ON and vice versa
*** ——(S)——	"SET coil" The associated boolean variable is SET to the ON state, if the left hand connection is in the ON state and remains set until the coil is RESET.
*** ——(R)——	"RESET coil" The associated boolean variable is RESET, if the left hand connection is in the ON state and remains reset until the coil is SET.

Function Block Diagram (FBD): Bit Logic Instructions and Comparison Instructions





IN1 and IN2 are compared according to the type of comparison you choose:

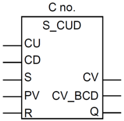
== IN1 is equal to IN2
<> IN1 is not equal to IN2
> IN1 is greater than IN2
< IN1 is less than IN2
>= IN1 is greater than or equal to IN2
<= IN1 is less than or equal to IN2

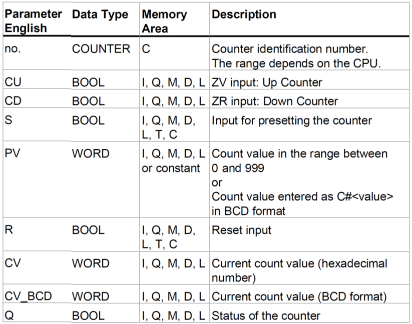
If the comparison is true, the RLO of the function is "1". Otherwise, it is 0. You cannot negate the comparison result itself, but you can achieve the same effect as negation by using the opposite compare function.

The following comparison instructions are available:

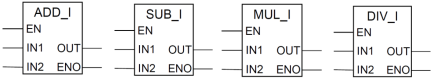
- CMP ? I : Compare Integer
- CMP ? D : Compare Double Integer
- CMP ? R : Compare Real

Counter Instructions and Integer Math Instructions

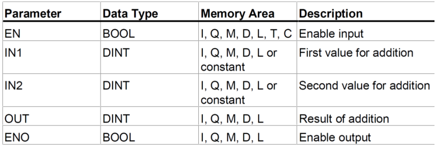




Parameter English	Data Type	Memory Area	Description
no.	COUNTER	C	Counter identification number. The range depends on the CPU.
CU	BOOL	I, Q, M, D, L	ZV input: Up Counter
CD	BOOL	I, Q, M, D, L	ZR input: Down Counter
S	BOOL	I, Q, M, D, L, T, C	Input for presetting the counter
PV	WORD	I, Q, M, D, L or constant	Count value in the range between 0 and 999 or Count value entered as C#<value> in BCD format
R	BOOL	I, Q, M, D, L, T, C	Reset input
CV	WORD	I, Q, M, D, L	Current count value (hexadecimal number)
CV_BCD	WORD	I, Q, M, D, L	Current count value (BCD format)
Q	BOOL	I, Q, M, D, L	Status of the counter

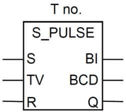


- ADD_I : Add Integer
- SUB_I : Subtract Integer
- MUL_I : Multiply Integer
- DIV_I : Divide Integer

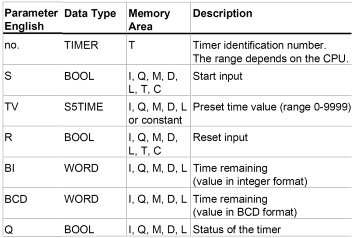


Parameter	Data Type	Memory Area	Description
EN	BOOL	I, Q, M, D, L, T, C	Enable input
IN1	DINT	I, Q, M, D, L or constant	First value for addition
IN2	DINT	I, Q, M, D, L or constant	Second value for addition
OUT	DINT	I, Q, M, D, L	Result of addition
ENO	BOOL	I, Q, M, D, L	Enable output

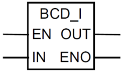
Timer Instructions and Conversion Instructions



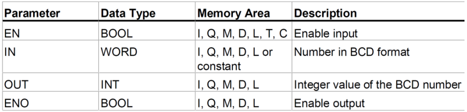
- SP : Start Pulse Timer
- SE : Start Extended Pulse Timer
- SD : Start On-Delay Timer
- SS : Start Retentive On-Delay Timer
- SF : Start Off-Delay Timer



Parameter English	Data Type	Memory Area	Description
no.	TIMER	T	Timer identification number. The range depends on the CPU.
S	BOOL	I, Q, M, D, L, T, C	Start input
TV	S5TIME	I, Q, M, D, L or constant	Preset time value (range 0-9999)
R	BOOL	I, Q, M, D, L, T, C	Reset input
BI	WORD	I, Q, M, D, L	Time remaining (value in integer format)
BCD	WORD	I, Q, M, D, L	Time remaining (value in BCD format)
Q	BOOL	I, Q, M, D, L	Status of the timer

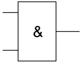
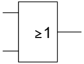
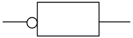
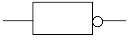
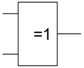


- BCD_I : BCD to Integer
- I_BCD : Integer to BCD
- BCD_DI : BCD to Double Integer
- I_DI : Integer to Double Integer

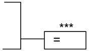
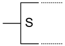
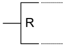
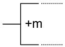
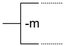
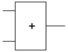
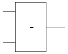


Parameter	Data Type	Memory Area	Description
EN	BOOL	I, Q, M, D, L, T, C	Enable input
IN	WORD	I, Q, M, D, L or constant	Number in BCD format
OUT	INT	I, Q, M, D, L	Integer value of the BCD number
ENO	BOOL	I, Q, M, D, L	Enable output

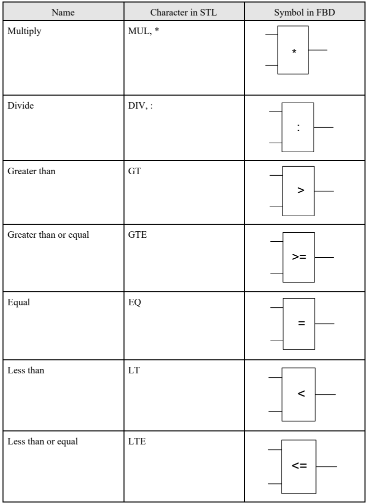
Display of Functions in statement list (STL) and FBD - 1

Name	Character in STL	Symbol in FBD
AND	A	
OR	O	
NOT	N	at the input 
		at the output 
Exclusive-OR	XOR	

Display of Functions in STL and FBD - 2

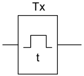
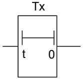
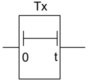
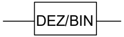
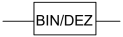
Name	Character in STL	Symbol in FBD
Assignment	=, ST	
Set	S	
Reset	R	
Count up	CU	 Count (+1) when signal changes from "0" to "1"
Count down	CD	 Count (-1) when signal changes from "0" to "1"
Add	ADD, +	
Subtract	SUB, -	

Display of Functions in STL and FBD - 3



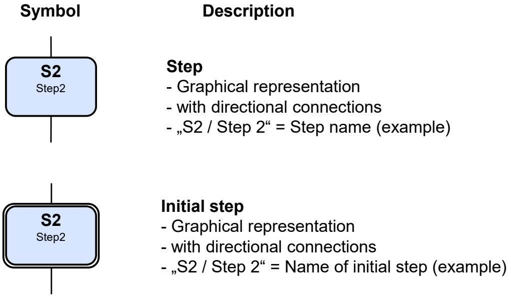
Name	Character in STL	Symbol in FBD
Multiply	MUL, *	*
Divide	DIV, :	:
Greater than	GT	>
Greater than or equal	GTE	>=
Equal	EQ	=
Less than	LT	<
Less than or equal	LTE	<=

Display of Functions in STL and FBD - 4

Name	Character in STL	Symbol in FBD
Pulse timer	SP Tx or SE Tx	
On delay timer	SD Tx or SS Tx	
Off delay timer	SF Tx	
Code converter Decimal/Binary	BTI	
Code converter Binary/Decimal	ITB	

Sequential Function Chart (SFC) Overview

- SFC subdivides a PLC program into **Steps** and **Transitions**.
- Steps and transitions are connected by **directional connections**.
- Each step has a set of **Actions**; each transition has a **Transition Condition**.



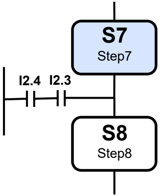
Symbol | **Description**

Step
- Graphical representation
- with directional connections
- „S2 / Step 2“ = Step name (example)

Initial step
- Graphical representation
- with directional connections
- „S2 / Step 2“ = Name of initial step (example)

SFC Step and Initial step in S7 Graph

Different Notations of Transition in S7 Graph

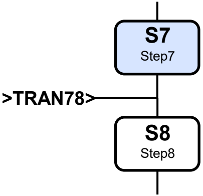


Preceding step

Transition condition

Following step

Transition programmed in LAD

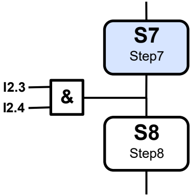


Preceding step

Transition condition

Following step

Sequencer view

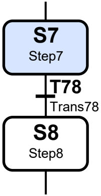


Preceding step

Transition condition

Following step

Transition programmed in FBD



Preceding step

Transition mark

Following step

Sequencer view

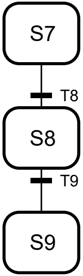
Steps

- A step defines behavior of the program for given inputs/outputs via its **actions**.
- A step is either **active** or **inactive**.
- Step representation:
  - Symbol containing the Step name
  - Incoming directional connection at the top
  - Outgoing directional connections at the bottom
- Step flag: boolean variable (**1 = active**, **0 = inactive**).
- Initial steps determine the program's starting state (may be marked with twin lines).

Transitions

- A transition specifies the condition for moving from preceding steps to following steps.
- Represented by a horizontal line crossing vertical directional connections.
- Associated with a boolean variable or expression (transition condition).
- Control moves from the bottom of predecessor steps to the top of follower steps.
- Transition conditions:
  - Always true: symbol **1**
  - Can be connected via LAD/FBD networks, STL statements, or transition labels.

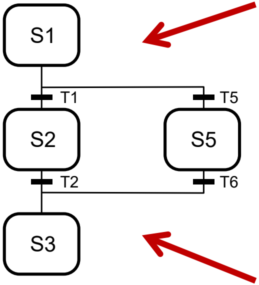
Syntax and Semantics of Possible Combinations of Steps and Transitions in Sequential Function Chart (1)



Simple chain
The change step-transition is always repeated.

Example:
A sequence from step S7 to S8 only takes place, if S7 is in an active state and the transition condition T8 is fullfilled. The next sequence from step S8 to S9 takes place if S8 is in an active state and transition condition T9 is fullfilled.

Syntax and Semantics of Possible Combinations of Steps and Transitions in Sequential Function Chart (2)

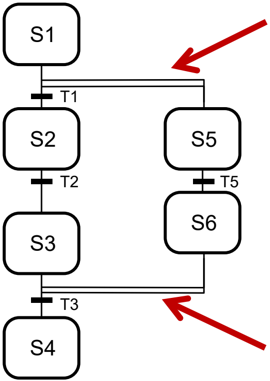


Beginning of sequence selection
A selection between several chains is displayed by as many transitions symbols under the horizontal line, as different possible sequences are present.
Example:
A sequence from S1 to S2 may take place only if S1 is active and transition condition T1 is fullfilled, or from S1 to S5 only if S1 is active and T5 is fullfilled.

End of sequence selection
The end of a chain selection is displayed by as many transition symboles above the horizontal line, as selection paths to be finished exist.
Example:
A sequence from S2 to S3 may take place only, if S2 is active and the Transition condition T2 is fullfilled or from S5 to S3 only if S5 is active and T6 is fullfilled.

Remark
For selecting only one chain it is necessary, that the transition conditions connected to the chains are mutually exclusive, so that they are never true at the same time. It is always possible to include an order of priority into the formulation of the transition conditions.

Syntax and Semantics of Possible Combinations of Steps and Transitions in Sequential Function Chart (3)



Beginning of simultaneous sequence
Only one common transition sympol directly above the twin horizontal synchronization line is possible.
Example:
A sequence from S1 to S2, S5, …may only take place only if S1 is active and transition condition T1, which is belonging to the commontransition, is fullfilled. After the synchronous activation of S2, S5 and so on the chains sequences are independent of one another.

End of simultaneous sequence
Only one common transition symbol directly under the twin horizontal Synchronisation line is possible.
Example:
A sequence from S3, S6, … may only take place, if all steps being above and connected with the twin horizontalline are active and the transition condition T3, which belongs to the common transition, is fullfilled.

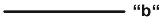
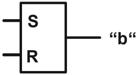
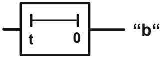
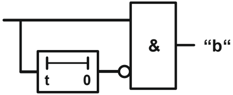
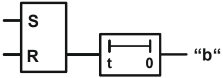
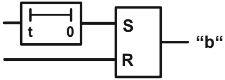
Actions and Action Blocks

- **Actions:** Operations executed while the associated step is active.
- Steps with no actions act as **Waiting** steps.
- Action connections:
  - LAD left conductor rail
  - FBD networks
  - Action blocks
  - STL control statements
- **Action blocks:** Concisely represent complex actions and may provide a boolean feedback variable.
- Multiple action blocks can be connected; one boolean input controls all simultaneously.

Action Block

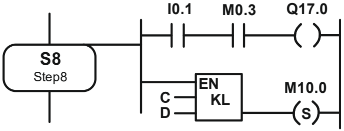
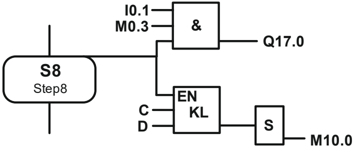
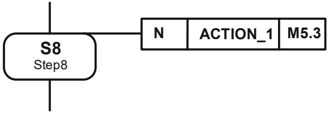
<table>
<tr><th>Symbol</th><th>Statement</th></tr>
<tr><td><table><tr><td>“a“</td><td>“b“</td><td>“c“</td></tr></table></td><td>General symbol
"a": Qualification of the command according to sheet37
"b": Textual description of the command
"c": Reference lable of the corresponding check back variable (optional)</td></tr>
<tr><td><table><tr><td>N</td><td>ACTION_1</td><td>BACK</td></tr><tr><td>SD
t=
10s</td><td>Motor_1 left</td><td></td></tr><tr><td>D</td><td>M7.7</td><td>M0.0</td></tr></table></td><td>Example for action blocks arranged in a row</td></tr>
</table>

Action Types

Timing	Description	Control structure / remarks
N	**not stored**	
S	**set / stored**	
R	**reset**	
D	**not stored, but delayd**	
L	**not stored, but time limited**	
SD	**stored and delayed**	
R	**reset**	
DS	**delayed and stored**	
R	**reset**	

In this table the valid values for the timing characterization of the action (“a“ in sheet 36) are indicated, in combination with the associated control structure in the FBD language

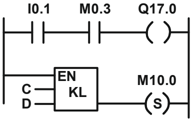
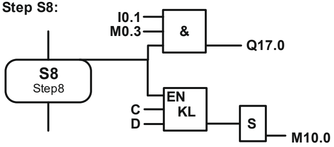
Example of Action (1)

No.	Example	Description
(a)	 S8 Step8 I0.1 M0.3 Q17.0 EN KL C D M10.0 S	**LAD language**
(b)	 S8 Step8 I0.1 M0.3 & Q17.0 EN KL C D S M10.0	**FBD language**
(c)		**Action block**

Note:
If S8 is inactive, the impact is fed to the logic connected;
e.g. Q17.0 = 0 and M10.0 keeps its previous state.

Note:
The block “Comparator with Enable input“ applied above is a user defined Block (so called Function block) and Is not part of the basic operation pool of a PLC

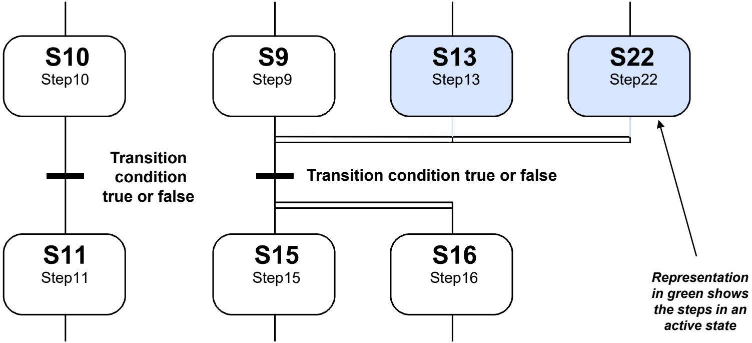
Example of Action (2)

Nr.	Example	Description
	Step S8:  **END_STEP**	**LAD language**
	Step S8:  I0.1 M0.3 & Q17.0 S8 Step8 EN KL C D S M10.0	**FBD language**
	Step S8: **L S8** **U I0.1** **U M0.3** **= Q17.0** **L C** **LT D** **U S8** **S M10.0** **END_STEP**	**Action block**

Sequence Rules

- Initial state defined by active initial steps and initial variable values.
- Transition switching:
  - Enabled if all preceding steps are active
  - Occurs when transition condition is fulfilled
- Switching activates following steps and resets preceding steps.
- Rules:
  - Steps never connected directly; always separated by a transition
  - Transitions never connected directly; always separated by a step
- Simultaneous chains: multiple steps activated at the same time, indicated by twin horizontal lines.

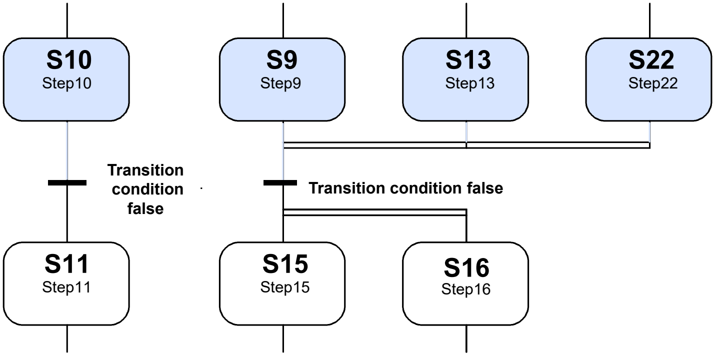
Transition not enabled


S10
Step10
S9
Step9
S13
Step13
S22
Step22
Transition condition true or false
Transition condition true or false
S11
Step11
S15
Step15
S16
Step16
Representation in green shows the steps in an active state

Transition not enabled

The transitions are not enabled, because the steps 9 or 10 resp. are not activ.
The corresponding transition conditions may be true or false.

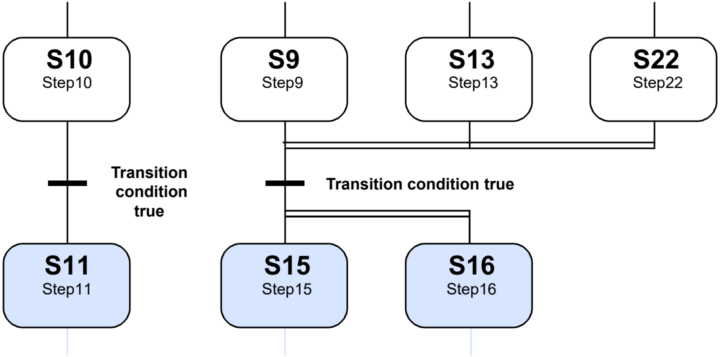
Transition enabled



Transition enabled

The transitions are enabled, but can not switch,
because the corresponding transition conditions are false.

Transition has switched



Transition has switched

The transitions have switched now,
because the corresponding transition conditions are true.

Summary

- PLC programs are structured into various program blocks (OBs, FCs, FBs, DBs, SFCs, SFBs).
- The main program runs in a cyclic loop, with possible interruptions for alarms or timed events.
- The Process Image Table ensures consistent input/output states during program execution.
- Boolean operations, flip-flops, timers, and data types are fundamental building blocks in PLC programming.
- Statement List (STL), Ladder Diagram (LAD), Function Block Diagram (FBD), and Sequential Function Chart (SFC) are key programming languages used in PLCs.

Digital Signal Processing (DSPs)

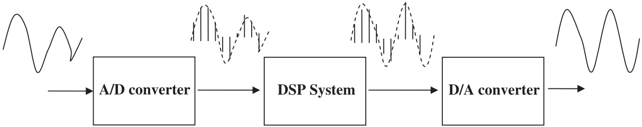
Digital Signal Processing: Overview

- **Goal of DSP:** Process real–world signals (speech, audio, images, biomedical, radar, etc.) after converting them into discrete–time, discrete–amplitude sequences.
- In early electronics, processing was done in the **analogue** domain:
  - Direct amplification, filtering, modulation using RLC circuits, op-amps, etc.
  - Performance limited by component tolerances, ageing, temperature drift and noise.
- **Digital implementation**:
  - Signals are converted to digital form via A/D conversion.
  - Processing is carried out using algorithms on DSP processors, microprocessors or FPGAs.
  - Advantages: programmability, reproducibility, ease of design changes, robustness to ageing and environmental variations.
- Target: introduce DSP concepts that affect hardware implementation and prepare ground for FPGA-based realizations.

DSP Hardware Platforms

- **General-purpose microprocessors**
  - Flexible, but not optimized for real-time multiply–accumulate intensive tasks.
- **Dedicated DSP microprocessors**
  - Special hardware: MAC units, hardware loop controllers, addressing modes for circular buffers.
  - Ideal for audio, communications, control applications.
- **FPGAs**
  - Highly parallel, reconfigurable fabric; can implement custom datapaths and pipelines.
  - Particularly suited for high-throughput, low-latency applications (e.g. baseband processing, image processing).
- For FPGA implementation we must understand:
  - Sampling rates and throughput requirements.
  - Wordlength / number representation (fixed vs floating point).
  - Algorithmic complexity (number of multiplications/additions).

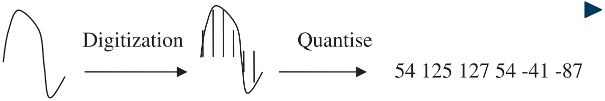
Basic DSP System



- Analogue input is first **sampled** and **quantized** by an A/D converter.
- Result is a discrete-time, discrete-amplitude sequence $x(n)$.
- The **DSP system** applies algorithms:
  - Filtering, spectral analysis.
  - Compression / coding.
  - Detection, estimation, equalisation, etc.
- Output sequence $y(n)$ is converted back to analogue via a D/A converter and reconstruction filter.

- Key design question: what signal processing must occur in the digital domain, and what sampling rate / resolution are required to meet system specifications?

Digitisation of Analogue Signals



Digitization

Quantise

54 125 127 54 -41 -87

- **Sampling**: capture signal amplitude at uniform time instants $t = nT$.
- **Quantisation**: map continuous amplitude to discrete levels (integers).
- Typical digital representation:

$$x(n) \in \mathbb{Z}, \quad n = \dots, -1, 0, 1, \dots$$

- Trade-off:
  - Higher sampling rate $\Rightarrow$ better temporal resolution, higher data rate.
  - More quantisation bits $\Rightarrow$ higher SNR, more hardware cost.

- DSP algorithms operate on these numeric sequences; the quality of digitisation sets a fundamental limit on achievable performance.

Sampling of Continuous-Time Signals

- Let $x_a(t)$ be a continuous-time analogue signal.
- Uniform sampling with sampling period T produces discrete-time sequence

$$x(n) = x_a(nT), \quad -\infty < n < \infty.$$

- The sequence $\{x(n)\}$ can be viewed as:
  - A list of measurements at equally spaced instants.
  - A representation of $x_a(t)$ suitable for digital processing.
- The **sampling rate** (or sampling frequency) is

$$f_s = \frac{1}{T} \quad \text{[samples/second]}.$$

- Key issue: how large must f_s be so that $x_a(t)$ can be reconstructed (approximately) from $x(n)$?

Nyquist–Shannon Sampling Theorem

- Assume $x_a(t)$ is bandlimited: its continuous-time Fourier transform $X_a(f) = 0$ for $|f| > f_m$.
- **Sampling theorem:**

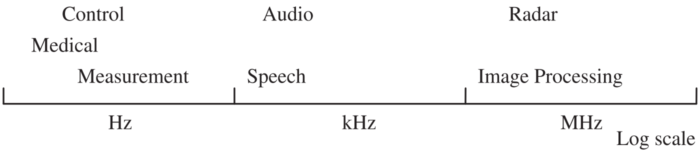
Perfect reconstruction of $x_a(t)$ from uniform samples is possible if

$$f_s \geq 2f_m.$$

The minimum sampling rate $f_s = 2f_m$ is the **Nyquist rate**.

- If $f_s < 2f_m$, high-frequency components fold back into low frequencies (**aliasing**).
- Practical implication:
  - Use analogue *anti-aliasing filters* before the ADC to limit bandwidth to f_m.
  - Select f_s taking into account non-ideal filter roll-off and system margins.

Sampling Rates in Practice



- Sampling rates span orders of magnitude:
  - **Medical measurements**: Hz range (e.g. slow temperature signals, some biosignals).
  - **Audio / speech**: kHz range (e.g. 8 kHz telephony, 44.1 kHz CD audio).
  - **Radar / image processing**: MHz range for high spatial/temporal resolution.
- Horizontal axis is logarithmic; highlights that DSP techniques must be scalable across very different data rates.

- Sampling rate selection is driven by:

Sampling Examples: Speech, Music, EEG

- **Speech**
  - Most energy below ≈ 4 kHz; standardized sampling at 8 kHz.
  - Higher harmonics above ≈ 4 kHz contribute little to intelligibility.
- **Music**
  - Human hearing up to ≈ 20 kHz.
  - CD audio uses 44.1 kHz $\Rightarrow$ Nyquist ≈ 22.05 kHz, covering audible spectrum.
- **Biomedical signals (EEG, ABR)**
  - EEG bandwidth of interest often tens of Hz to a few kHz.
  - Example: auditory brainstem response (ABR) in 100–3000 Hz range.
  - EEG may be sampled at 20 kHz, giving Nyquist of 10 kHz for capturing fine temporal features and allowing digital filtering.
- These examples illustrate how sampling rate is chosen not only from theory but also from the physics and perceptual/diagnostic needs of the application.

DSP Transformations: Time vs Frequency Domain

- Many DSP tasks are easier in the **frequency domain**:
  - Filtering, spectral analysis, modulation, compression, feature extraction.
- Core transforms used:
  - **Discrete Fourier Transform (DFT)** and its efficient implementation, the **FFT**.
  - **Discrete Cosine Transform (DCT)** – real-valued transform heavily used in image/video compression.
  - **Wavelet transforms** – multi-resolution analysis for non-stationary signals.
- Hardware implications:
  - These transforms are computation–intensive (many multiplications/additions).
  - Efficient FPGA or DSP implementations rely on exploiting symmetry, periodicity and pipeline/parallel architectures.

Discrete Fourier Transform (DFT)

- For a finite-length sequence $x(n)$, $n = 0, \ldots, N-1$, the N-point DFT is

  $$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \quad k = 0, \ldots, N-1,$$

  where the **twiddle factor** is

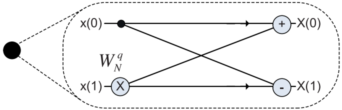
  $$W_N = e^{-j2\pi/N}.$$

- Interpretation:
  - $X(k)$ are samples of the discrete-time Fourier transform on the unit circle.
  - Each $X(k)$ corresponds to a complex sinusoid of frequency $2\pi k/N$.
- Inverse DFT:

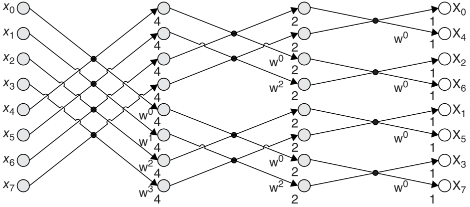
  $$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk}.$$

- Direct computation requires N^2 complex multiplications and a similar number of additions – expensive for large N.

Fast Fourier Transform (FFT)



(a) FFT element



(b) FFT structure

- FFT algorithms exploit **symmetry** and **periodicity** in W_N^{nk}.
- Basic idea (radix-2, $N = 2^m$):
  - Decompose N-point DFT into smaller DFTs of size $N/2$, $N/4$, . . . , 2.
  - Reuse intermediate results via "butterfly" computations.
- Computational complexity reduced from $O(N^2)$ to $O(N \log_2 N)$.
- Essential for real-time spectrum analysis, OFDM communication, and many other high-speed applications.

- In OFDM (e.g. IEEE 802.11a), FFT/IFFT (inverse) are used to map data symbols onto many orthogonal subcarriers and recover them at the receiver.

Radix-2 DIT FFT: Even–Odd Decomposition

- Start from the N-point DFT:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}.$$

- Split the sum into even and odd indices, $x_e(n) = x(2n)$, $x_o(n) = x(2n+1)$:

$$X(k) = \sum_{n=0}^{N/2-1} x_e(n) W_N^{2nk} + W_N^k \sum_{n=0}^{N/2-1} x_o(n) W_N^{2nk}.$$

- Use $W_N^{2nk} = W_{N/2}^{nk}$ to obtain

$$X(k) = E(k) + W_N^k O(k),$$

  where $E(k)$ and $O(k)$ are $N/2$-point DFTs of x_e and x_o.
- For $k + N/2$:

$$X(k+N/2) = E(k) - W_N^k O(k), \text{giving a pair of outputs from one butterfly.}$$

- Recursively applying this decomposition $m = \log_2 N$ times yields the full radix-2 FFT.

FFT Butterfly and Implementation Notes

- A radix-2 **butterfly** computes

$$X_1 = A + WB, \qquad X_2 = A - WB,$$

  where A and B are intermediate values and W is a twiddle factor.
- Data-flow organisation:
  - **Decimation-in-time (DIT)**: split in time domain, outputs appear in bit-reversed order.
  - **Decimation-in-frequency (DIF)**: split in frequency domain, inputs in bit-reversed order.
- Hardware aspects:
  - Each stage has $N/2$ butterflies; total of $(N/2)\log_2 N$ butterflies.
  - Fixed-point implementations require careful scaling to avoid overflow.
  - Pipeline and systolic FFT architectures exploit concurrency for high throughput.
- Practical design choices:
  - Power-of-two sizes for radix-2; mixed-radix FFTs for composite N.
  - Use of in-place algorithms to minimise memory.

Discrete Cosine Transform (DCT)

- The 1-D DCT of a length-N sequence $x(n)$ is

$$X(k) = \sum_{n=0}^{N-1} x(n) \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right) k\right], \quad k = 0, \ldots, N-1.$$

  (Only cosine terms, so $X(k)$ is real-valued.)

- Widely used in image and video compression (JPEG, MPEG):
  - The DCT can be viewed as the DFT of an **even extension** of $x(n)$:
    - Reflect the sequence around the boundaries to avoid artificial discontinuities.
    - This leads to purely real, cosine-only basis functions.
  - Enables effective **quantization** and **compression** by discarding high-frequency coefficients.

Wavelet Transform and DWT

- Wavelets are short-duration oscillatory functions that provide **time–frequency** localization.
- Wavelet decomposition:
  - Decomposes a signal into shifted and scaled versions of a prototype wavelet.
  - Very suitable for non-stationary signals where classical FFT (fixed frequency resolution) is inadequate.
- **Discrete Wavelet Transform (DWT)** via filter banks:
  - Input $x(n)$ is passed through a **low-pass** filter $g(n)$ and a **high-pass** filter $h(n)$.
  - Outputs are down-sampled by 2 to produce approximation and detail coefficients:

$$y_{\text{low}}(n) = \sum_{k=-\infty}^{\infty} x(k)g(2n-k),$$

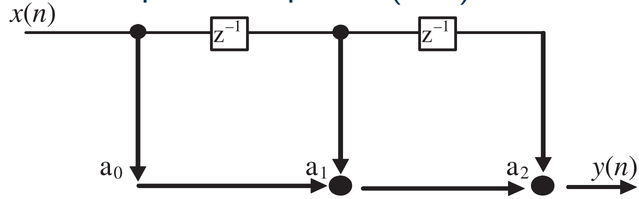
$$y_{\text{high}}(n) = \sum_{k=-\infty}^{\infty} x(k)h(2n-k).$$

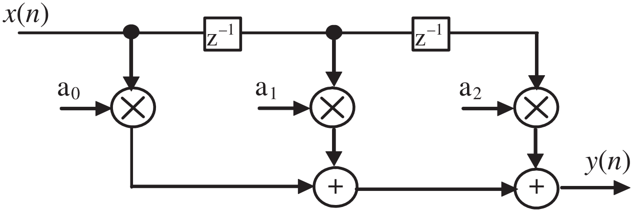
  - Process can be iterated on the low-pass branch to obtain multi-resolution representations.
- Applications: image compression and denoising, biomedical signal analysis, transient detection.

Digital Filters: Overview

- A **filter** shapes the spectral content of a signal:
  - Low-pass, high-pass, band-pass, band-stop, etc.
- **Analogue filters**:
  - Implemented using resistors, capacitors, inductors, op-amps.
  - Continuous-time, continuous-amplitude.
- **Digital filters**:
  - Implemented as algorithms operating on discrete-time sequences.
  - Use arithmetic operations (add, multiply, delay).
  - Very precise and stable; easily reconfigurable in software or hardware (DSP, FPGA).
- Two main classes:
  - Finite Impulse Response (FIR) – impulse response of finite duration.
  - Infinite Impulse Response (IIR) – involves feedback, theoretically infinite impulse response.

Finite Impulse Response (FIR) Filters





- General FIR filter of length N:

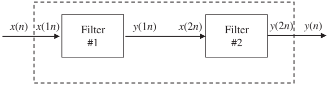
$$y(n) = \sum_{i=0}^{N-1} a_i x(n-i).$$

- Realized using:
  - Delay elements z^{-1} (registers / memory).
  - Multipliers implementing coefficients a_i.
  - Adders accumulating partial sums.

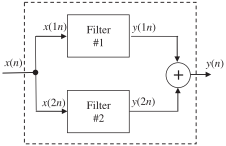
- FIR filters are always **bounded input bounded outout (BIBO) stable** since their impulse response is finite.
- With symmetric coefficients $(a_i = a_{N-1-i})$, FIR filters exhibit **linear phase**: they delay all frequency components equally, avoiding waveform distortion.

LTI Properties and FIR Structures

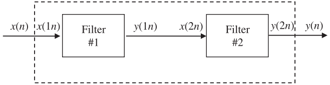
- An FIR filter (or any digital filter) is **linear** and **time-invariant** if it satisfies:
  - **Superposition**: response to $x_1(n) + x_2(n)$ is $y_1(n) + y_2(n)$.
  - **Homogeneity**: response to $c\,x(n)$ is $c\,y(n)$.
  - **Shift invariance**: shift in input produces equal shift in output.
- These properties enable:
  - Convolution description of filtering.
  - Use of frequency response $H(e^{j\omega})$ to characterise behavior.
- FIR filters can be combined:
  - **Cascade** structure: output of one filter is input to the next.
  - **Parallel** structure: inputs processed by multiple filters, then summed.



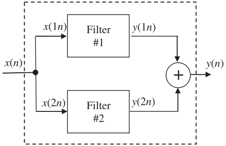
(a) Cascade FIR filter



(b) Parallel FIR filter

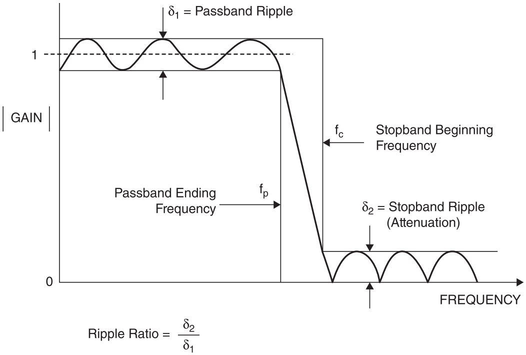


(a) Cascade FIR filter



(b) Parallel FIR filter

Low-Pass FIR Filter Specifications


δ_1 = Passband Ripple
1
GAIN
f_c
Stopband Beginning Frequency
Passband Ending Frequency
f_p
δ_2 = Stopband Ripple (Attenuation)
0
FREQUENCY
Ripple Ratio = $\frac{\delta_2}{\delta_1}$

- Design starts from a **desired magnitude response**:
  - Passband with nearly constant gain.
  - Stopband with strong attenuation.
- Key parameters:
  - Passband edge frequency f_p.
  - Stopband begin frequency f_c.
  - Passband ripple δ_1.
  - Stopband ripple / attenuation δ_2.
  - **Transition band**: region between f_p and f_c.
- Ripple ratio: $\text{Ripple ratio} = \dfrac{\delta_2}{\delta_1}$.
- Example plot shows the ripples in both passband and stopband and the ideal gain level.

- Specifications $(\delta_1, \delta_2, f_p, f_c)$ directly determine filter length and window choice.

From Ideal Response to Practical FIR Filter

- Start with an **ideal** (infinite-length) low-pass response:
  - Sharp transition; zero passband/stopband ripple.
  - Impulse response $h_{\text{ideal}}(n)$ extends to $\pm\infty$.
- **Design idea**:
  1. Specify desired magnitude response $|H_d(\omega)|$ with given $f_p, f_c, \delta_1, \delta_2$.
  2. Compute corresponding *inverse transform* to obtain $h_{\text{ideal}}(n)$.
  3. Truncate to a finite length N and shift to make it causal.
- Simple truncation ⇒ multiply $h_{\text{ideal}}(n)$ by a **rectangular window**.
  - This introduces oscillations (ripples) near discontinuities in the frequency domain ⇒ **Gibbs phenomenon**.
  - Ripples occur in both passband and stopband.
- Practical design uses alternative windows (Hamming, Hann, Kaiser, Dolph–Chebyshev, . . .) to control ripple and transition width.

Why Window and Filter Length Matter (FIR Low-Pass)

The message to remember (industrial relevance)

Two design knobs (window type, filter length N) determine **three outcomes** that directly impact **control/measurement quality** and **embedded implementation cost**.

Two knobs you can tune

- **Window type** (e.g., Hamming vs Dolph–Chebyshev)
  - sets **sidelobes / ripple** (leakage, stopband attenuation)
  - influences **transition sharpness** (mainlobe width)
- **Filter length** N (number of taps)
  - larger $N \Rightarrow$ **narrower transition band**
  - larger $N \Rightarrow$ **lower ripple** (for same window)

Three outcomes to practice

- **Signal quality:** noise rejection vs distortion
  - ripple/leakage ⇒ residual noise, spectral leakage
  - too-wide transition ⇒ poor selectivity
- **System behavior:** delay and closed-loop impact
  - FIR group delay $\approx \frac{N-1}{2}$ samples (linear phase)
- **Compute and memory:** feasibility on MCU/FPGA/PLC
  - cost $\propto N$ multiplies/adds per sample (+ coefficient memory)

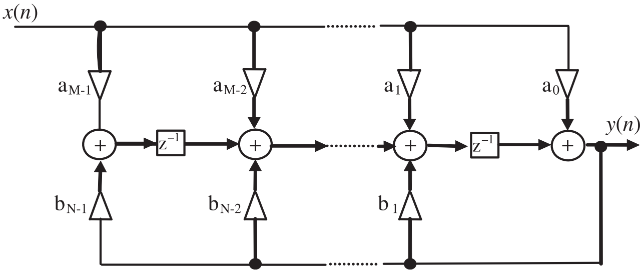
Rule-of-thumb decision

- Need **sharper cutoff**? **Increase** N (most reliable).
- Need **more stopband attenuation**? **Change window** (then adjust N if needed).
- Need **lower cost/latency**? **Reduce** N and accept wider transition / less attenuation.

Motivation for IIR Filters

- FIR filters are simple and robust, but:
  - Require many taps to achieve very sharp cut-off.
  - Computational cost increases linearly with filter length.
- **Infinite Impulse Response (IIR) filters** introduce feedback (recursive structure):
  - Can approximate analogue filter characteristics (Butterworth, Chebyshev, Elliptic) with far fewer coefficients.
  - Sharper transition for a given order compared to FIR.
- Main trade-offs:
  - Possible **non-linear phase** (phase distortion).
  - Susceptible to **stability** and **quantisation** issues due to feedback.
- Design approach:
  - Often start from a well-known analogue prototype $H_A(s)$.
  - Transform into digital $H_D(z)$ using mappings such as impulse invariance, z-transform, or bilinear transform.

Direct-Form IIR Structure



- General IIR difference equation:

$$y(n) = \sum_{i=0}^{N-1} a_i x(n-i) + \sum_{j=1}^{M-1} b_j y(n-j).$$

- Corresponding transfer function:

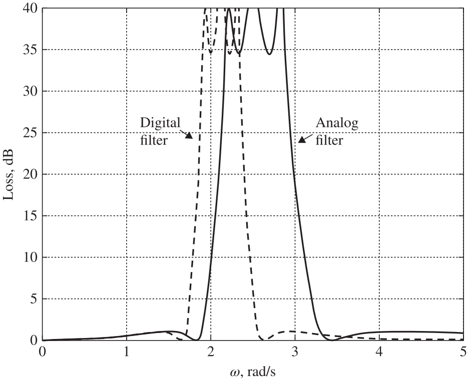
$$H(z) = \frac{\sum_{i=0}^{N-1} a_i z^{-i}}{1 - \sum_{j=1}^{M-1} b_j z^{-j}}.$$

- Realisation:
  - Feedforward path: FIR-like part with coefficients a_i.
  - Feedback path: uses previous outputs scaled by b_j.

Wave Digital Filters (WDFs): Concept

- WDFs form a special class of digital filters derived from classical analogue **lattice/ladder** networks.
- Basic idea:
  - Express analogue circuit in terms of *wave variables* (incident and reflected waves at ports).
  - Map each analogue element (resistor, capacitor, inductor) to a corresponding digital adaptor.
- Advantages:
  - Very **low sensitivity** to coefficient quantisation and round-off.
  - Good control of losses and passband/stopband attenuation.
  - Structures are well suited for fixed-point implementation with limited wordlength.
- Frequency response of WDFs preserves key properties of the original analogue network, while also benefiting from digital flexibility.

Frequency Impact of Warping in WDFs



- Figure compares loss / attenuation vs frequency for:
  - Analogue prototype filter.
  - Corresponding digital WDF implementation.
- Due to frequency warping, some shift between analogue and digital curves occurs, especially near band edges.
- Nevertheless, WDF structure keeps:
  - Low sensitivity to coefficient errors.
  - Close match to desired attenuation characteristics.

- In DSP hardware design, WDFs enable efficient, robust realizations for applications requiring stringent filter specs and fixed-point arithmetic.

DSP in a Nutshell: From Signals to Implementations

- **Start: real-world signals**
  - Continuous-time, continuous-amplitude (speech, audio, images, biomedical, radar).
  - Sampled and quantised $\Rightarrow$ discrete-time sequences $x(n)$ for digital processing.
- **Analyse in the transform domain**
  - DFT/FFT: complex exponential basis, efficient $O(N \log N)$ spectrum computation.
  - DCT: cosine-only basis with strong energy compaction (JPEG/MPEG).
  - Wavelet/DWT: multi-resolution time–frequency analysis for non-stationary signals.
- **Shape spectra using digital filters**
  - FIR filters: convolution with finite $h(n)$, always bounded input, bounded output stability (BIBO) stable, linear phase with symmetric taps.
  - IIR filters: recursive structures with poles/zeros; sharp responses with low order but care needed for stability and quantisation.
  - Design from specifications $(\delta_1, \delta_2, f_p, f_c)$ via windows, analogue prototypes, bilinear transform, etc.

DSP in a Nutshell: From Signals to Implementations

- **Map mathematics to hardware/software**
  - Realisation as structures of adds, multiplies and delays on DSPs, FPGAs or ASICs.
  - FFT pipelines, FIR/IIR direct forms, second-order sections, and wave digital filters for robust fixed-point implementation.
- **Big picture**
  - DSP provides a toolbox: *sample* → *transform* → *filter* / *modify* → *reconstruct*.
  - Design choices (transform, filter type, structure) are driven by application constraints: accuracy, complexity, latency and implementation platform.

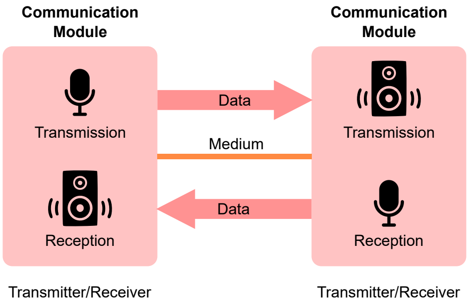
Table of contents

Industrial Communication

Dr Bikash Sah

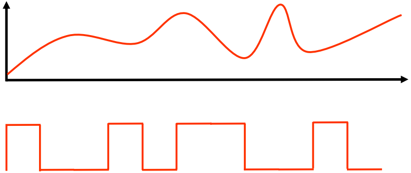
Universität Siegen / IAS Interconnected Automation Systems

Elements of an Industrial Communication System



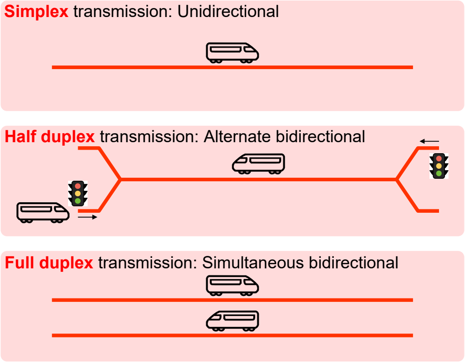
- **End devices (nodes):**
  - Sensors, actuators, HMIs, PLCs, drives.
  - Act as **transmitters**, **receivers** or both.
- **Data:** physical quantities (voltage levels, light, RF power) to which information and direction are assigned.
- **Communication modules:**
  - Convert local electrical/logical signals into a format suitable for the network.
  - Implement protocol stack (framing, addressing, error detection, etc.).
- **Medium:**
  - Copper (twisted pair, RS-485), fibre optics, radio (WLAN, 5G), sometimes power-line.
  - Defines attenuation, noise, EMC robustness, maximum distance and speed.

Analog vs Digital Transmission



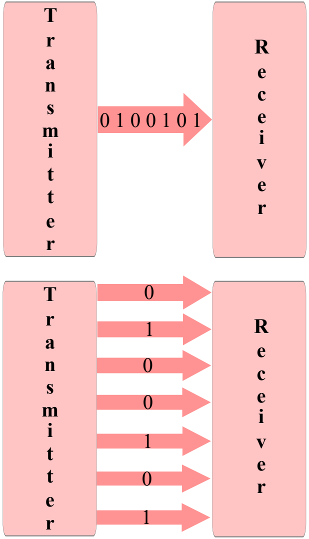
- **Analog transmission**
  - Signal amplitude varies *continuously* with time.
  - Typical in legacy process control: 4–$20\,\mathrm{mA}$ loops, 0–10 V signals.
  - Very intuitive but sensitive to noise, offset and long-term drift.
- **Digital transmission**
  - Information encoded in discrete levels (usually two: “0” and “1”).
  - Achieved by *sampling* and *quantisation* of analog values.
  - Robust against moderate noise, allows error detection, addressing and routing.
- Industrial networks today are almost exclusively **digital**; analog signals are converted at the sensor/actuator interface.

Transmission Directions: Simplex, Half Duplex, Full Duplex


Simplex transmission: Unidirectional
Half duplex transmission: Alternate bidirectional
Full duplex transmission: Simultaneous bidirectional

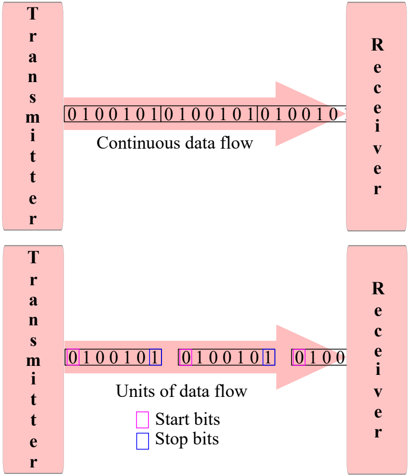
- **Simplex:** unidirectional, e.g. broadcast radio, sensor with one way telemetry.
- **Half duplex:** bidirectional but not simultaneous.
  - Medium is shared in time (master–slave polls, token passing).
  - Requires arbitration or master control but saves wiring.
- **Full duplex:** simultaneous bidirectional transfer.
  - Implemented by separate physical channels or echo cancellation.
  - Used in switched Ethernet, fibre links, etc.
- Choice depends on topology, cost and required throughput; many fieldbuses use **half-duplex** to keep transceivers simple and robust.

Serial vs Parallel Transmission



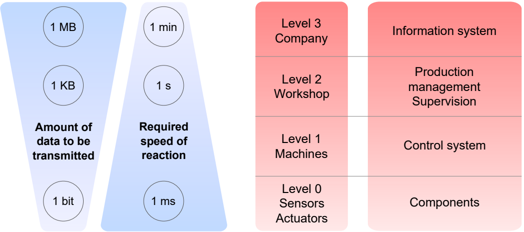
- **Serial transmission**
  - Bits sent sequentially over one pair (or one fibre).
  - Lower pin count, better EMC, easily galvanically isolated.
  - Dominant for industrial networks: RS-232, RS-485, CAN, Ethernet, etc.
- **Parallel transmission**
  - Multiple bits sent simultaneously on separate conductors.
  - High raw data rate, but only feasible over short distances.
  - Susceptible to skew, crosstalk and EMC issues.
- Modern fieldbuses and Industrial Ethernet variants are all **serial** links with advanced encoding and framing.

Serial Transmission: Synchronous vs Asynchronous



- **Synchronous serial**
  - Continuous bit stream with a shared clock (or embedded clock).
  - High efficiency: little overhead per data word.
  - Examples: SPI, synchronous HDLC, physical layers of Ethernet.
- **Asynchronous serial**
  - Data sent in frames (characters) separated by idle periods.
  - Each frame has *start*, *data*, optional *parity*, *stop* bits.
  - Local clocks are allowed to differ slightly; resynchronisation at every start bit.
  - Example: UART/RS-232/RS-485, classic Modbus RTU.
- Many industrial buses use **half-duplex, asynchronous, serial, digital transmission** over RS-485 to minimise cost and provide robust EMC behaviour.

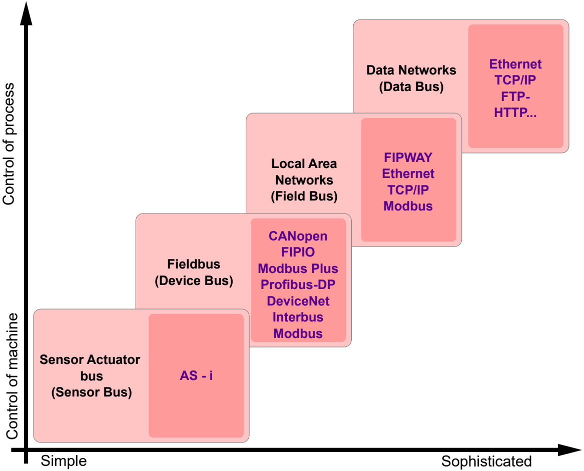
Industrial Communication Requirements



Level 3 Company	Information system
Level 2 Workshop	Production management Supervision
Level 1 Machines	Control system
Level 0 Sensors Actuators	Components

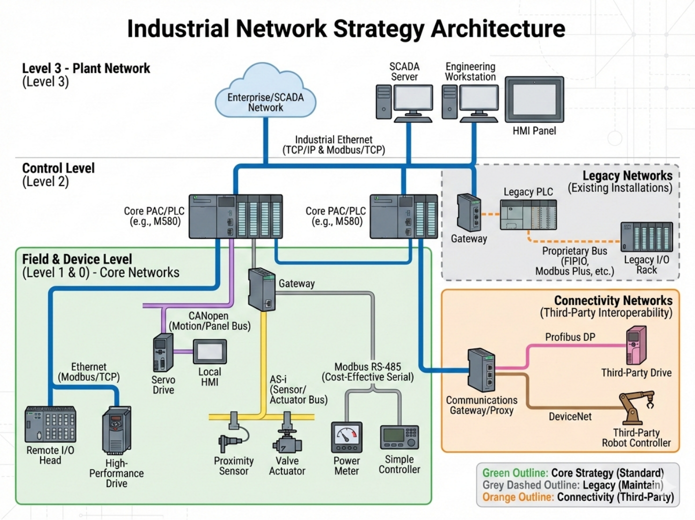
- Plant communication can be grouped into hierarchical levels:
  - **Level 3 – Company:** ERP, MES, enterprise IT.
  - **Level 2 – Workshop / Cell:** production planning, supervision, SCADA.
  - **Level 1 – Machines:** PLCs, CNC, drive controllers.
  - **Level 0 – Field:** sensors, actuators, I/O modules.
- **Data volume vs reaction time**:
  - Upper levels: large data blocks, relaxed timing (seconds–minutes).
  - Lower levels: small data units (bits, words) with tight deadlines (ms).
- No single bus can optimally serve all levels; hence the coexistence of fieldbus and Ethernet-based networks.

Main Industrial Networks and Buses



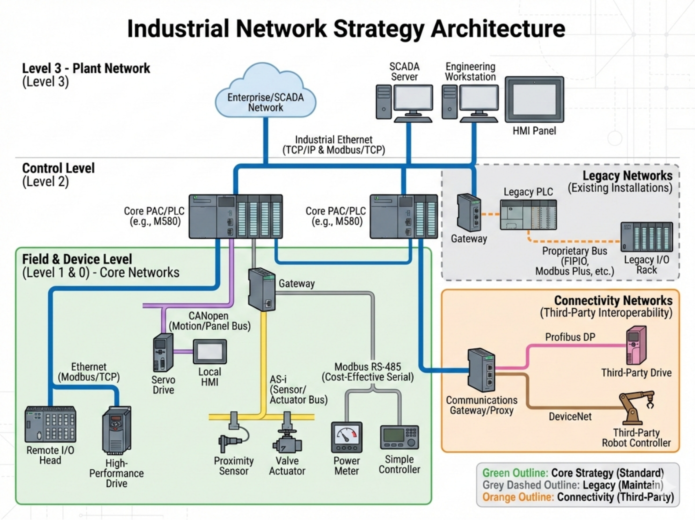
- **Sensor/actuator buses (Level 0):**
  - Simple wiring, low data rates, very robust.
  - Examples: AS-Interface (AS-i), IO-Link.
- **Fieldbuses / device buses (Level 1):**
  - Deterministic cyclic I/O exchange between PLCs and devices.
  - Examples: Modbus (serial), Profibus-DP, CANopen, DeviceNet, Interbus.
- **Local area networks / data buses (Levels 2–3):**
  - Based on Ethernet TCP/IP, support large data volumes and standard IT services (FTP, HTTP, OPC UA).
  - Industrial variants: Profinet, EtherNet/IP, Modbus/TCP, POWERLINK, etc.

Example: Network Strategy in an Industrial Vendor


Industrial Network Strategy Architecture
Level 3 - Plant Network
(Level 3)
Enterprise/SCADA Network
SCADA Server
Engineering Workstation
HMI Panel
Industrial Ethernet (TCP/IP & Modbus/TCP)
Control Level
(Level 2)
Core PAC/PLC (e.g., M580)
Core PAC/PLC (e.g., M580)
Legacy Networks
(Existing Installations)
Legacy PLC
Gateway
Proprietary Bus (FIPIO, Modbus Plus, etc.)
Legacy I/O Rack
Field & Device Level
(Level 1 & 0) - Core Networks
Gateway
CANopen (Motion/Panel Bus)
Servo Drive
Local HMI
Ethernet (Modbus/TCP)
AS-i (Sensor/ Actuator Bus)
Modbus RS-485 (Cost-Effective Serial)
Remote I/O Head
High-Performance Drive
Proximity Sensor
Valve Actuator
Power Meter
Simple Controller
Connectivity Networks
(Third-Party Interoperability)
Profibus DP
Third-Party Drive
Communications Gateway/Proxy
DeviceNet
Third-Party Robot Controller
Green Outline: Core Strategy (Standard)
Grey Dashed Outline: Legacy (Maintain)
Orange Outline: Connectivity (Third-Party)

- **Core networks**
  - **Ethernet TCP/IP & Modbus/TCP** for plant-wide information and control (Levels 2–3), increasingly extended down to controllers (Level 1).
  - **CANopen** for internal device and panel bus structures.
  - **AS-i** for simple binary sensor/actuator connectivity at Level 0.
  - **Modbus RS-485** when Ethernet is not economical or topology is constrained.

Example: Network Strategy in an Industrial Vendor



- **Legacy networks**
  - Older proprietary solutions (e.g. FIPIO, Modbus Plus, Uni-Telway) still found in existing plants.
- **Connectivity networks**
  - Interoperability with third-party buses such as DeviceNet, Profibus, Interbus when required by the installed base.

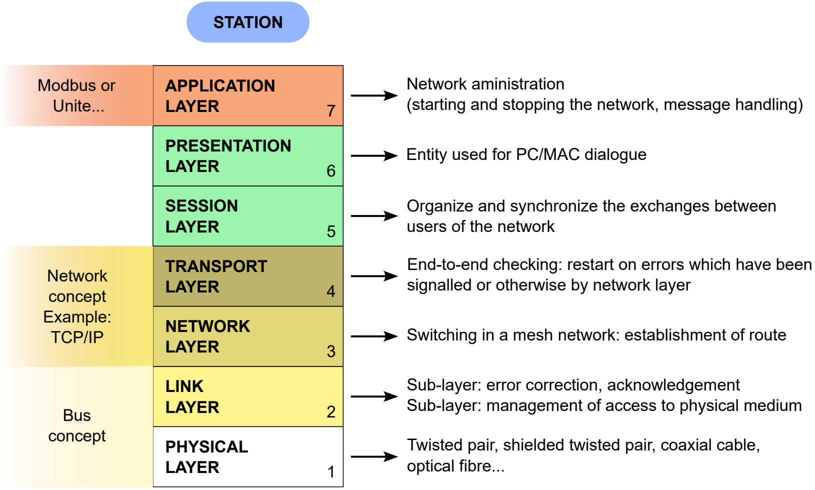
Industrial Communication: Big Picture

- **Physical / link level:**
  - Analog vs digital signalling, serial vs parallel, synchronous vs asynchronous, simplex / half / full duplex.
  - Implementation on copper, fibre or wireless with strong EMC requirements.
- **Protocol level:**
  - Framing, addressing, error detection (CRC), and medium access (master–slave, token, CSMA/CD, TDMA).
  - Real-time behaviour (cyclic vs acyclic traffic, jitter, determinism).

Industrial Communication: Big Picture

- **System level:**
  - Matching network type to plant level:
    - Sensor buses at Level 0.
    - Fieldbuses at Level 1.
    - Industrial Ethernet and IT networks at Levels 2–3.
  - Typical industrial solution: **half-duplex, asynchronous, serial digital** communication at field level, bridged to high-speed Industrial Ethernet above.
- **Design goal for the engineer:**
  - Select and configure networks so that required reaction times, reliability, diagnostics and life-cycle cost are all satisfied for the given application.

Introduction to international organisation of standarisation (ISO) model



STATION

Modbus or Unite...

APPLICATION LAYER 7 → Network aministration (starting and stopping the network, message handling)

PRESENTATION LAYER 6 → Entity used for PC/MAC dialogue

SESSION LAYER 5 → Organize and synchronize the exchanges between users of the network

Network concept Example: TCP/IP

TRANSPORT LAYER 4 → End-to-end checking: restart on errors which have been signalled or otherwise by network layer

NETWORK LAYER 3 → Switching in a mesh network: establishment of route

Bus concept

LINK LAYER 2 → Sub-layer: error correction, acknowledgement
Sub-layer: management of access to physical medium

PHYSICAL LAYER 1 → Twisted pair, shielded twisted pair, coaxial cable, optical fibre...

TCP: Transmission Control Protocol (Layer 4)
IP: Internet Protocol (Layer 3)

Fig. 5.1: ISO/open systems interconnection (OSI) reference model for communication systems

Example Frames and ISO/OSI Model

Modbus RTU request frame

Bytes	1	1	2	2	2
	Slave address = 7	Function code = 3	No. of first word = 5	No. of words to be read = 2	CRC 16

- Compact frame transported over a serial line (typically RS485).
- Fields:
  - **Slave address** – target device on the bus.
  - **Function code** – type of request (read, write, diagnostics, . . .).
  - **First word address** and **word count**.
  - **CRC–16** – error detection at the data link layer.

Ethernet TCP/IP frame

Bytes	8	6	6	2	20	20	46 to 1500	4
	Preamble	Destin. addr.	Source. addr.	LLC	IP	TCP	Application layers FTP, HTTP, SMTP Modbus, etc.	FCS

- Example of a layered frame:
  - **Preamble, MAC addresses, LLC/FCS** – physical/data link.
  - **IP header** – network layer (routing).
  - **TCP header** – transport layer (ports, reliability).
  - **Application payload** – protocols such as HTTP, FTP, Modbus/TCP.
- Shows how an industrial protocol (Modbus) can be carried on top of standard IT stacks.

Physical Media in Industrial Networks

- The **transmission medium** determines:
  - Achievable data rate (bandwidth).
  - Maximum distance without regeneration.
  - Immunity to electromagnetic disturbances.
  - Installation cost and mechanical robustness.
- Common media in industrial environments:
  - **Twisted pair copper** – cheapest, easy to install, widely used for fieldbuses.
  - **Coaxial cable** – better shielding, good for higher frequencies and noise immunity.
  - **Optical fibre** – uses light, immune to EMI, ideal for long distances and harsh areas.

Electrical Standards for Twisted Pair Links

- Industrial protocols are often mapped onto simple **serial line standards**:

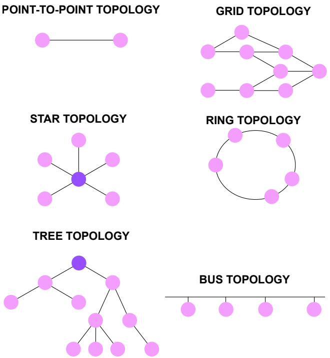
  RS–232 Point-to-point, single-ended signalling.
    - Short distances ($\sim$ 15 m), low speed ($\sim$ 20 kbps typical).
    - Historically used for configuration ports, engineering tools.

  RS–422A Differential signalling, 4-wire, multi-drop.
    - Full-duplex communication (separate TX/RX pairs).
    - Good noise immunity; up to ≈ 1200 m at moderate speeds.

  RS–485 Differential signalling, 2-wire, multi-drop.
    - Half-duplex, several tens of nodes on one bus.
    - Backbone for Modbus RTU and many vendor-specific fieldbuses.

- Differential standards (RS–422/485) are preferred near motors, drives, and switching power stages.

Network Topologies in Industrial Systems



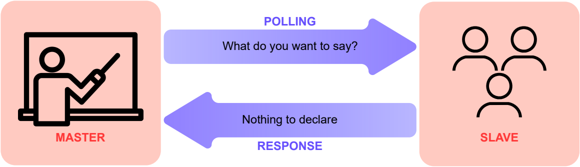
- **Point-to-point**: single link between two devices (e.g. PLC–drive link).
- **Bus**: all nodes share a common line (classic fieldbus, RS485 trunk with drops).
- **Star**: devices connect to a central switch or hub (switched Ethernet).
- **Ring**: nodes form a closed loop; traffic passes via intermediate stations (Token Ring, some proprietary safety rings).
- **Tree / hierarchical**: combination of stars/buses, follows plant structure (cell, line, area).
- **Grid / mesh**: multiple paths; high availability, used in high-end process networks.

- Choice of topology affects redundancy, installation cost, and determinism.

Medium Access Control (MAC) in Industrial Networks

- Several stations share the same physical medium ⇒ need **rules** to decide who may transmit.
- Main MAC strategies in industrial communication:
  - **Master–Slave (polling)** – one central controller grants access.
  - **Token passing** – a logical “token” circulates to grant the right to send.
  - **Random access / CSMA** – nodes listen to the medium and compete for access.
- Key performance aspects:
  - Determinism (worst-case delay known?).
  - Scalability with number of nodes.
  - Behaviour under high load and in presence of errors.

Master–Slave Access (Polling)

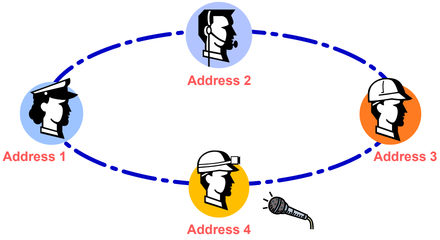


Eg: Profibus-DP

- Advantages:
  - Predictable cycle time, simple implementation in devices.
- Drawbacks:
  - Single point of failure, master can become a bottleneck.

- Implemented at the **data link layer**.
- One node is the **master**:
  - Initiates all exchanges ("polls" the slaves).
  - Decides the sequence and rate of queries.
- Other nodes are **slaves**:
  - Only speak when addressed by the master.
  - Typical response: process data, diagnostics, or "no data".
- Example: **Profibus DP**, many proprietary RS485-based buses.

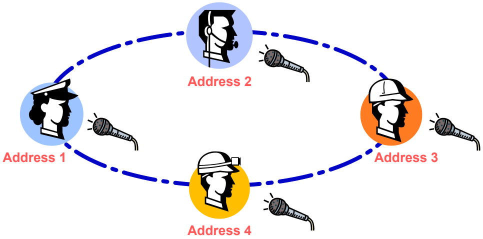
Token Ring Access



- Pros:
  - Highly deterministic, fair sharing of bandwidth.
- Cons:
  - Token management overhead, sensitivity to a node or link failure (often mitigated by fault-tolerant ring variants).

- Also a link-layer MAC scheme.
- Stations form a **logical ring**; a special frame called the **token** circulates.
- A node may transmit only while holding the token, then passes it on.
- Access order is strictly defined ⇒ bounded access delay.
- Industrial example: **Modbus Plus**, some legacy factory networks.

Random Access and CSMA Principles



- In pure CSMA networks, delay and jitter increase strongly with load ⇒ non-deterministic.
- Still attractive in many industrial Ethernet variants when combined with full-duplex switching and traffic engineering.

- **Carrier Sense Multiple Access (CSMA)**:
  - Nodes *listen* to the medium before transmitting.
  - If the channel is free, they start transmission.
  - If a collision occurs, a recovery procedure is invoked.
- Behaviour is similar to "many people around a table": everyone speaks when it becomes quiet again.

CSMA/CD vs CSMA/CA in Practice

- **CSMA/CD – Collision Detection** (classic shared Ethernet)

  1. Nodes monitor the line while transmitting.
  2. If a collision is detected, they abort the frame.
  3. Wait a random back-off time (binary exponential back-off).
  4. Retry transmission.

  - Collisions waste bandwidth; acceptable on short, lightly loaded segments.

- **CSMA/CA – Collision Avoidance** (e.g. CAN bus)

  1. Nodes start transmission after sensing idle bus.
  2. If two nodes start simultaneously, arbitration on the identifier ensures that only the higher-priority frame continues.
  3. Lower-priority node detects loss of arbitration and retries later.

  - Collisions are *non-destructive*; high-priority traffic is guaranteed.

- Design implication: CSMA/CA-type buses (CAN, CANopen, DeviceNet) are very attractive for time-critical control traffic at the device level.

From Physical Layer to Industrial Protocols

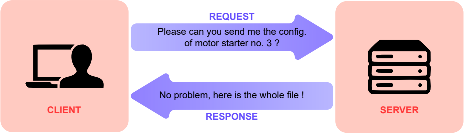
- Physical and link-layer choices (RS232/422/485, Ethernet, fibre, topology, MAC) must support required:
  - Cycle time / reaction time.
  - Distance and environmental robustness.
  - Scalability (number of nodes, bandwidth).
- On top of these, **industrial protocols** define:
  - **Frame formats** (as for Modbus RTU vs Modbus/TCP).
  - **Addressing and diagnostics** (device IDs, status codes).
  - **Application semantics** (I/O image, parameters, alarms).
- Modern plants typically combine:
  - Real-time fieldbuses at Level 0/1.
  - Industrial Ethernet at Level 1/2.
  - TCP/IP-based IT networks at Level 2/3.
- A solid understanding of media, topologies, and MAC schemes is essential to choose the right network for each level.

Application-level communication concepts

Why application-level models?

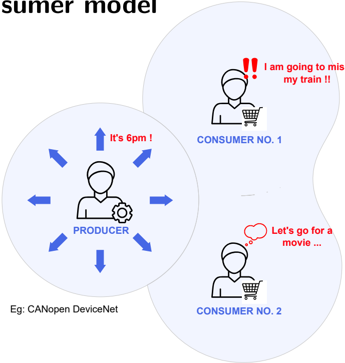
- Separate *what* is exchanged (semantics of variables, parameters, files) from *how* it is transported (fieldbus / Ethernet protocol stack).
- Allow PLC programs, SCADA systems and drives to be engineered independently of the underlying network.

Client–Server model


CLIENT
REQUEST
Please can you send me the config. of motor starter no. 3 ?
No problem, here is the whole file !
RESPONSE
SERVER

Eg: Modbus

Producer–Consumer model



Eg: CANopen DeviceNet

Application-level communication concepts

Client–Server model

- **Client**: actively issues a service request over the network (e.g. "read holding register 40005").
- **Server**: provides the requested service and returns the result (e.g. sensor value, configuration block, file).
- Typical services: read/write variables, download parameters, diagnostics, file transfer.
- Used for *sporadic*, configuration and diagnostic traffic.
- Examples: Modbus (master as client, slave as server), OPC UA services, HTTP/REST APIs in modern controllers.

Producer–Consumer model

- **Producer**: one node periodically or on event *publishes* a data object (e.g. position set-point, status word).
- **Consumers**: one or many nodes *subscribe* and use the same data simultaneously.
- Efficient for distributing the same process data to many devices (drives, I/O modules, HMIs).
- Minimises bus load: one frame, many receivers (hardware filtering).
- Examples: CANopen/DeviceNet PDOs, Ethernet POWERLINK, PROFINET RT multicast.

Traffic types and the concept of profiles

Traffic types in industrial networks

- **Cyclic (periodic) data**
  - Process values refreshed with fixed cycle time (e.g. every 1 ms).
  - Small telegrams, strict jitter and latency requirements.
  - Used for fast control loops: I/O images, drive set-points and feedback.
- **Acyclic (event- or request-based) data**
  - Sent on demand for configuration, parameterisation, diagnostics.
  - Larger payloads, relaxed real-time requirements.
  - Must not disturb deterministic cyclic traffic (priority / time slots).
- **Isochronous / time-synchronised traffic (in some Ethernet systems)**
  - All nodes share a common time base (sub-μs synchronisation).
  - Supports motion control with tight phase relationships between multi-axis drives (e.g. PROFINET IRT, EtherCAT DC).

Traffic types and the concept of profiles

Open systems, interoperability and profiles

- An **open system** is built from interoperable and interchangeable components from different vendors.
- **Interoperability**: devices understand each other's communication services → strict adherence to protocol specifications.
- **Interchangeability**: one device can be replaced by another (possibly from another vendor) without modifying (for example, the PLC program) → adherence to **application profiles**.
- A **profile** is a standardised description of functions and parameters for a device class (e.g. drives, encoders, I/O).
    - Defines mandatory / optional objects, semantics, units, scaling.
    - Typically distributed as electronic description files: EDS, GSD, GSDML, XML, etc.
    - Examples: PROFIdrive, CiA 402 (drive profile for CANopen/EtherCAT), Encoder profiles, IO-Link device description.

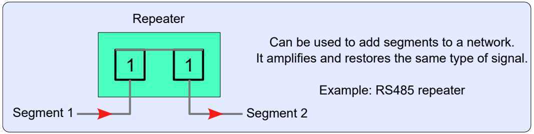
Interconnection products in industrial networks

Why interconnection products?

- Extend cable length, segment networks, and connect heterogeneous media.
- Control collision domains and broadcast domains for deterministic behaviour.
- Map between different physical layers and protocol stacks.

Physical / link-layer devices

- **Repeater**
  - Regenerates and reshapes signals on the same medium / standard.
  - Extends maximum segment length (e.g. RS485 repeater).

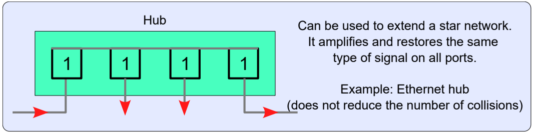


Can be used to add segments to a network.
It amplifies and restores the same type of signal.

Example: RS485 repeater

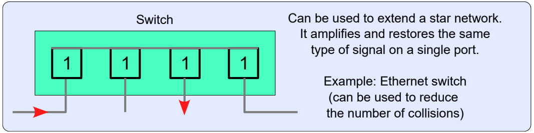
Interconnection products in industrial networks

- **Hub**
  - Multiport repeater, shares one collision domain.
  - Broadcasts each frame on all ports (classic Ethernet hubs).
- **Switch**
  - Forwards frames based on MAC address (layer 2).
  - Segments collision domains; enables full-duplex links.
  - Industrial switches support VLANs, QoS, redundancy (RSTP, MRP).



Can be used to extend a star network.
It amplifies and restores the same
type of signal on all ports.

Example: Ethernet hub
(does not reduce the number of collisions)



Can be used to extend a star network.
It amplifies and restores the same
type of signal on a single port.

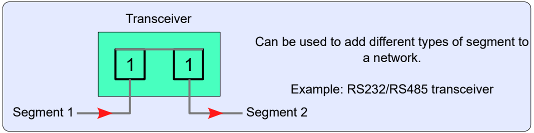
Example: Ethernet switch
(can be used to reduce
the number of collisions)

Interconnection products in industrial networks

- **Transceiver**
  - Converts between different physical layers while preserving the same data link protocol (e.g. RS232–RS485).

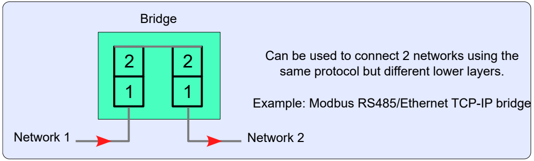
Network-layer and higher devices

- **Bridge**
  - Connects two segments that use the same protocol but may have different physical media (e.g. Modbus/RS485 to Modbus/TCP-IP).
  - Operates mainly at OSI layer 2.



Can be used to add different types of segment to a network.

Example: RS232/RS485 transceiver

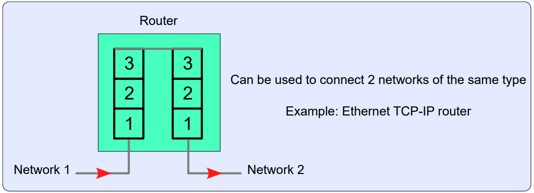


Can be used to connect 2 networks using the same protocol but different lower layers.

Example: Modbus RS485/Ethernet TCP-IP bridge

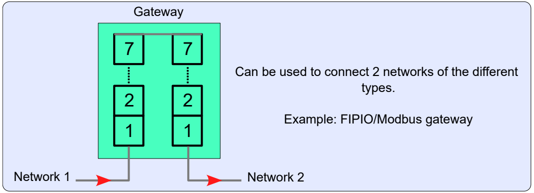
Interconnection products in industrial networks

- **Router**
  - Connects IP networks and chooses routes based on IP addresses.
  - Separates broadcast domains and allows access to higher-level plant networks or corporate IT (e.g. between two PROFINET subnets).
- **Gateway**
  - Translates between *different* protocols and data models (layers 7 down to 3, if necessary).
  - Maps variables, data types and diagnostics (e.g. FIPIO–Modbus gateway, PROFIBUS–PROFINET proxy, CANopen–OPC UA gateway).
  - Essential for brown-field integration and IIoT connectivity.



Can be used to connect 2 networks of the same type

Example: Ethernet TCP-IP router



Can be used to connect 2 networks of the different types.

Example: FIPIO/Modbus gateway

Industrial Communication Protocols – Overview

- **ASi** – Simple sensor/actuator bus; low-cost wiring; deterministic master–slave polling for binary I/O.
- **CANopen** – CAN-based; distributed control and motion; object dictionary and standardized device profiles.
- **DeviceNet** – CAN-based; focus on plug-and-play integration of sensors, actuators, and drives in factory automation.
- **Ethernet TCP/IP Modbus** – Modbus/TCP on standard Ethernet; very common for PLC–HMI and SCADA communication.
- **Profibus-DP** – High-speed cyclic I/O between PLCs and remote I/O/drives; deterministic and widely deployed.
- **FIPIO** – Deterministic fieldbus with producer–consumer model; used in process and factory automation.
- **Interbus** – Ring-oriented remote I/O system; good synchronisation of many distributed modules.
- **Modbus RTU** – Legacy serial master–slave protocol; extremely simple and still popular in drives and instruments.

These protocols span from simple sensor buses (ASi, Modbus) to richer fieldbuses and Ethernet-based solutions for higher-level control.

AS-Interface (ASi) at a Glance

- **Purpose:** Low-cost fieldbus for connecting **discrete sensors and actuators** (binary I/O, motor starters, valve islands) at the lowest level of factory automation.
- **Standardisation:** EN 50295, IEC 62026-2; governed by the AS-International Association.
- **Topology:** *Free topology* (line, star, tree, mixed) without termination; allows easy retrofits in existing installations.
- **Communication and power on the same 2-wire cable:**
  - Yellow, polarised flat cable carries both **data ($167\,$kbps)** and **30 V DC** supply.
  - Separate black cable often used for auxiliary power to actuators.
- **Network size / cycle time:**
  - ASi V1: 1 master + 31 standard slaves, cycle time $\approx 5\,$ms.
  - ASi V2: 1 master + 62 A/B slaves, cycle time $\approx 10\,$ms.
- **Typical position in automation pyramid:** between PLC I/O modules and simple field devices, complementing higher-level networks such as Profibus, Profinet, EtherNet/IP, etc.

ASi – Historical Development

- **1990:** Two universities and eleven (mainly German) companies form the ASi consortium to define a **low-cost interface** for sensors and actuators.
- **1992:**
  - First ASi chips become commercially available.
  - Creation of the international AS-International Association (headquartered in Germany).
- **1995:** Establishment of national promotional bodies (France, Netherlands, UK, . . .) and growing ecosystem of devices and masters.
- **2001:** ASi V2 specification:
  - Up to **62 A/B slaves** per master.
  - Extended support for **analogue I/O** and improved diagnostics.
  - Introduction of **“Safety at Work”** concept for safety-related I/O over standard ASi wiring.
- Since then: integration into many PLC platforms (Schneider, Siemens, Rockwell, etc.); used in conveyors, packaging machines, automotive body shops, intralogistics, and process auxiliaries.

ASi and the OSI Model

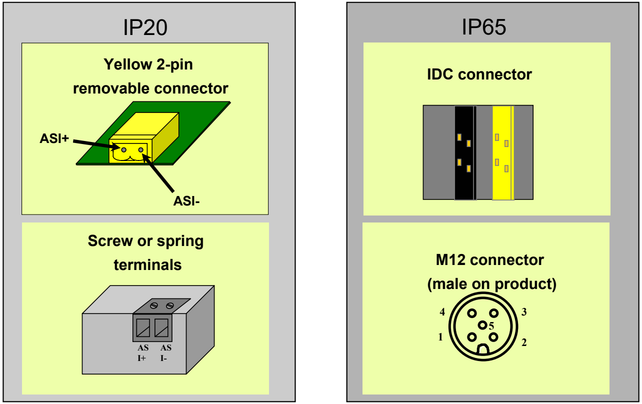
Application	Client/Server requests, profiles, cyclic and acyclic services
Presentation	(not used)
Session	(not used)
Transport	(not used)
Network	(not used; single segment, no routing)
Link (LLC+MAC)	Master–Slave polling, frame format, error detection
Physical	2-wire flat cable, power + data, free topology

- **Only 3 layers are implemented explicitly:** Physical, Data Link, Application, plus the use of **profiles**.
- **Link layer:** deterministic master–slave schedule, giving guaranteed cycle time for I/O refreshing.
- **Application layer:** standardised request set for network management, process data exchange, diagnostics and parameterisation.

Physical Layer Characteristics

- **Medium:**
  - 2-wire yellow **flat ribbon cable with polarisation** (data + power).
  - Alternative: unshielded round cable for special installations.
- **Topology: Free** (line, star, tree, ring-like, mixed) without line terminators.
- **Length:**
  - Up to 100 m without repeaters.
  - Up to 300 m with repeaters and suitable power injection.
- **Bit rate:** 167 kbps; each data exchange transaction ≈ 150 µs.
- **Device count:**
  - ASi V1: 1 master + 31 slaves.
  - ASi V2: 1 master + 62 A/B slaves on one segment.
- **Connectors / protection classes (examples):**
  - **IP20** (cabinet): yellow 2-pin plug, screw or spring terminals.
  - **IP65/IP67** (field): IDC “piercing” connectors and M12 circular connectors.

Types of ASi Connectors



IP20

Yellow 2-pin removable connector

Screw or spring terminals

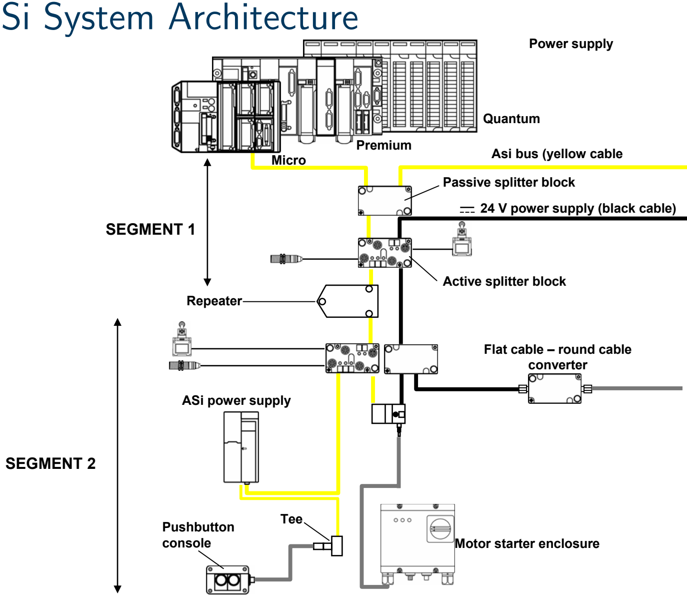
IP65

IDC connector

M12 connector (male on product)

- **2-wire flat ribbon cable with polarisation** (data + power).
- Yellow cable carries both **data (167 kbps)** and **30 V DC** supply.
- Separate black cable often used for auxiliary power to actuators.

Example ASi System Architecture



Example ASi System Architecture

- **Master side:** ASi master integrated in a PLC rack (e.g. Micro, Premium, Quantum) or as a standalone gateway to Profibus, Profinet, EtherNet/IP, Modbus TCP, etc.
- **Segmentation:**
  - **Segment 1:** directly connected to the master; may include passive and active splitter blocks, flat-cable to round-cable converters, and local sensors/actuators.
  - **Segment 2:** extended via repeaters or additional ASi power supplies for long distances or high current demand.
- **Typical field devices:** pushbutton stations, limit switches, inductive sensors, light curtains, valve terminals, motor starters for conveyor drives, etc.
- **Power concept:**
  - Yellow cable: ASi voltage and data.
  - Black cable: 24 V auxiliary power for actuators with higher current demand.

Link Layer: Medium Access and Reliability

- **Medium access: Master–Slave** polling at the link layer.
  - Only the master initiates frames on the bus.
  - Each slave is addressed in turn and may respond within a fixed time slot.
  - Ensures **deterministic cycle time** for all I/O points.
- **Useful data per transaction:**
  - Up to **4 output bits** in the master request (3 bits for ASi V2 A/B slaves in certain profiles).
  - Up to **4 input bits** in the slave response.
- **Transmission security:** "Numerous checks" at bit and frame level, e.g.
  - Start bit delimiter and end bit delimiter.
  - Use of half-wave pulses for robust decoding.
  - Minimum pause between bits and frames.
  - End-of-frame parity and frame length checks.
- Combination of short frames, polling, and error detection gives **high immunity** to industrial noise and predictable reaction times in the millisecond range.

Application Layer Services and Traffic Types

Standardised request classes:

1. **Network administration:** addressing, identification, parameter setting, device reset.
2. **Cyclic I/O exchange (process data):**
   - Max. 4 output bits (3 for A/B slaves), 4 input bits per slave.
   - Cycle time: $5\,\mathrm{ms}$ (31 slaves) to $10\,\mathrm{ms}$ (62 slaves).
3. **Cyclic network monitoring:** read status / diagnostics (I/O errors, slave faults).
4. **Parameter data transmission:** write parameter requests for device configuration.

Traffic types:

- **Cyclical data:**
  - Small amount of process data refreshed periodically according to the polling cycle.
  - Drives the PLC process image (start/stop, sensor states, actuator commands).
- **Acyclic data:**
  - Larger or less time-critical information, exchanged on request or on events: configuration, extended diagnostics, identification, firmware information, etc.
  - Uses reserved time slots or background bandwidth, so it does not disturb cyclic I/O refresh.

Profiles and Device Interchangeability

- **Goal:** ensure that devices from different manufacturers with the same function can be exchanged without engineering changes.
- Every ASi slave carries a **profile** that is **engraved in silicon** (read-only).
- **ASi V1: 2-digit profile**

$$\text{Profile} = \text{IO}_{\text{code}}.\text{ID}_{\text{code}}$$

  - IO_{code} – number and arrangement of inputs/outputs (0–F).
  - ID_{code} – device type (e.g. standard I/O module, valve island, motor starter).
- **ASi V2: 4-digit profile**

$$\text{Profile} = \text{IO}_{\text{code}}.\text{ID}_{\text{code}}.\text{ID1}_{\text{code}}.\text{ID2}_{\text{code}}$$

  - ID1_{code} – customisation options (e.g. safety, diagnostics level).
  - ID2_{code} – product sub-type (e.g. number of channels, special behaviour).
- Profiles are typically delivered as **electronic description files** (EDS, GSD, etc.), which engineering tools import to configure the network offline.

ASi – Strengths and Limitations

Strengths

- Very simple, cost-efficient wiring (two-wire flat cable, piercing connectors).
- Deterministic master–slave polling with short and predictable cycle times.
- Power and communication on the same cable reduce panel I/O wiring.
- Mature ecosystem and strong interoperability via well-defined profiles.
- Safety extension ("ASi Safety at Work") reuses the same infrastructure.

Limitations

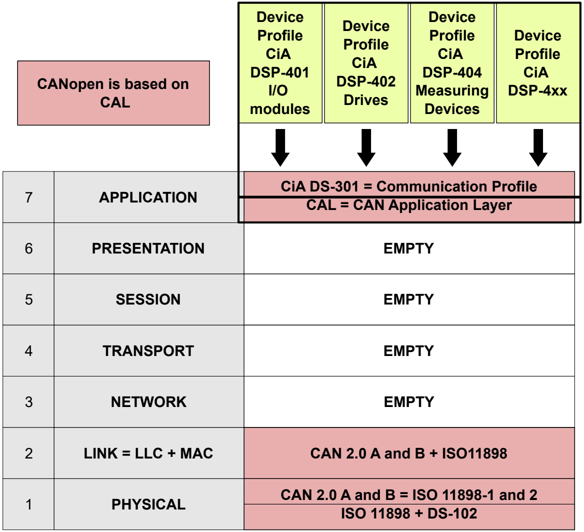
- Limited payload: few bits per slave, mainly suited for discrete I/O.
- Moderate data rate (167 kbps); not appropriate for high-bandwidth motion control.
- No intrinsic routing or multi-segment networking at protocol level (handled via gateways/masters to higher-level buses).
- Analogue and complex devices are supported but less efficiently than in Ethernet-based fieldbuses.

CANopen — History (Key Milestones)

- **1980–1983: CAN** conceived (Bosch) to address robust in-vehicle communication; emphasis on reliability and cost-effective ECUs.
- **1983–1987:** High-volume automotive adoption makes CAN controllers/transceivers economically attractive for industrial devices.
- **1991: CiA (CAN in Automation)** founded to promote and standardize industrial CAN-based communication.
- **1993: CAL (CAN Application Layer)** specifications published: defines mechanisms, but not a complete interoperable device behavior model.
- **1995:** CiA releases **DS-301** communication profile: **CANopen** is established as a structured application layer + object model.
- **2001:** CiA releases **DS-304** (CANopen Safety / "CANsafe") to support safety-related communication on CANopen networks.

CAN specifies mainly *Physical + Data Link behavior*; CANopen completes the ecosystem with *object dictionary, services, and profiles* for interoperability.

CANopen and the ISO/OSI Model (Layer Coverage)



CANopen is based on CAL

Device Profile CiA DSP-401 I/O modules

Device Profile CiA DSP-402 Drives

Device Profile CiA DSP-404 Measuring Devices

Device Profile CiA DSP-4xx

7	APPLICATION	CiA DS-301 = Communication Profile CAL = CAN Application Layer
6	PRESENTATION	EMPTY
5	SESSION	EMPTY
4	TRANSPORT	EMPTY
3	NETWORK	EMPTY
2	LINK = LLC + MAC	CAN 2.0 A and B + ISO11898
1	PHYSICAL	CAN 2.0 A and B = ISO 11898-1 and 2 ISO 11898 + DS-102

CANopen and the ISO/OSI Model (Layer Coverage)

Key message:

- **CAN** standardizes **Layer 1 (Physical)** and **Layer 2 (Data Link)** behavior.
- **CANopen** standardizes the **Application layer services** and an **Object Dictionary**.
- **Profiles** (DSP-4xx family) ensure interoperability (same semantics across vendors).

Profiles concept (as in figure):

- **DS-301:** communication profile (network management + services)
- **DSP-401/402/404/...:** device profiles (I/O, drives, measuring devices, etc.)

CANopen — Physical Layer Essentials

Medium and signalling

- Differential signalling on **shielded twisted pair** (robust EMC behavior).
- 2-wire bus (CAN_H/CAN_L); 4-wire sometimes used if supplying power separately in device-specific systems.

Topology and termination

- **Bus topology** with short stubs; avoid long tap lines.
- **120 Ω termination** at both physical ends of the bus to prevent reflections.

Distance vs. bitrate (engineering trade-off)

- Higher bitrate ⇒ shorter permissible bus length (signal integrity, propagation delay).
- Typical rule-of-thumb: **25 m @ 1 Mbps**, up to **1000 m @ 10 kbps** (installation dependent).

Node count (logical)

- Up to **128 node IDs** (1 master + 127 slaves in classical addressing).

CANopen — Link Layer: Arbitration and Communication Models

Medium access: CSMA/CA with nondestructive arbitration

- Any node may start transmission when bus is idle; collisions are resolved by **bitwise arbitration**.
- Arbitration uses **dominant/recessive** bits; a transmitter that loses arbitration stops immediately.
- **Priority rule:** Lower identifier value ⇒ higher priority (wins arbitration).

Communication models used by CANopen

- **Producer/Consumer:** one producer, multiple consumers (typical for **PDOs**).
- **Client/Server:** request/response (typical for **SDOs** and configuration).
- CANopen combines both to separate *real-time process data* from *parameterization/diagnostics*.

Arbitration gives *deterministic priority*, but not deterministic *latency* unless the network is engineered (busload, message set, priorities).

CANopen — Link Layer: Payload and Transmission Reliability

Max. useful payload

- Classical CAN frame carries up to **8 bytes** of application data.
- Small payload encourages compact cyclic messages (excellent for sensor/actuator process variables).
- Larger transfers are handled at the application layer via **segmentation** (SDO).

Reliability and error handling (why CAN works in harsh environments)

- Strong error detection: CRC, bit monitoring, bit stuffing checks, frame format checks.
- Error confinement: error counters and **error states** (error-active / error-passive / bus-off).
- Faulty nodes are isolated to protect the bus (critical in industrial uptime scenarios).

Practical takeaway

- Robustness comes from *physical layer design* + *link-layer checks* + *network engineering discipline*.

CANopen — Application Layer: Services and Object Types

Core idea: CANopen standardizes a set of services operating on the **Object Dictionary** (OD), so devices behave consistently across vendors.

- **Network administration (NMT):** node state control, start-up, reset, supervision (master/slaves).
- **PDO (Process Data Object): real-time, low-volume** process data (producer/consumer).
  - Trigger modes: event-driven (change-of-state), cyclic, SYNC-driven, or polled by master.
- **SDO (Service Data Object): parameter data** (often > 8 bytes) via segmentation (client/server).
- **SYNC / TIME:** synchronization and time distribution for coordinated control.
- **EMCY:** emergency messages for fault signalling; supports fast diagnostics.
- **Heartbeat / Node guarding:** node liveness monitoring (supervision).

PDOs are engineered for deterministic cyclic control; SDOs are engineered for commissioning/configuration without real-time guarantees.

CANopen — Identifier Allocation (Function Code + Node ID)

Why identifier structure matters

- CAN arbitration priority depends on the identifier; CANopen therefore structures identifiers systematically.
- **11-bit base format (common):** split into a **function code** and a **node ID**.

Operational interpretation

- **Function code** selects the message class (PDO/SDO/EMCY/SYNC, etc.).
- **Node ID** addresses the specific device (often set via DIP switch or configuration).
- Engineering implication: critical control PDOs can be assigned higher priority via lower COB-IDs.

Examples of message classes (typical)

- NMT (network control), SYNC, TIME, EMCY
- TPDO/RPDO (transmit/receive PDO)
- SDO client↔server
- Heartbeat

CANopen — Connectors and Practical Installation Notes

CiA connector guidance (installation consistency)

- CiA recommendations (e.g., DR-303-1) document typical connector options and pin conventions.
- Industrial design goal: predictable wiring, shielding continuity, and serviceability.

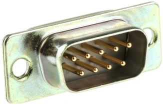
Common connector families you will encounter

- **DB9 (SUB-D, DIN 41652):** historically common on devices and gateways.
- **M12 (5-pin):** robust, IP-rated, vibration-resistant; typical for harsh industrial environments.
- **RJ45:** appears on some devices/gateways; convenient but not always ideal for rugged field wiring.
- **Open style terminal blocks:** used in cabinets; ensure strain relief and shield termination discipline.

Installation discipline

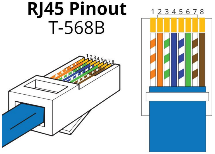
- Terminate only at bus ends; avoid star wiring for classical CAN.
- Keep stubs short; route away from high dv/dt power cables; maintain shield strategy.

CANopen Connectors

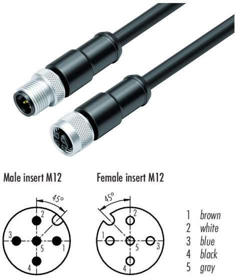


9-Pin SUB D
DIN 41652

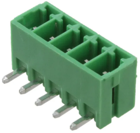
RJ45 Pinout
T-568B



1. White Orange
2. Orange
3. White Green
4. Blue
5. White Blue
6. Green
7. White Brown
8. Brown



5-Pin Microstyle
M12



Open Style
Connector

CANopen Profiles: Interoperability Beyond “It Communicates”

Problem: Two devices can speak CANopen yet still be incompatible if they expose different semantics.
Solution: Communication + Device profiles

- **DS-301 (Communication profile):** defines network management, services, and OD access mechanisms.
- **DSP-4xx (Device profiles):** define standardized OD entries and behavior per device class.
  - **DSP-401:** I/O modules **DSP-402:** drives and motion control **DSP-404:** measuring devices

Engineering impact

- Commissioning becomes scalable: PLC/SCADA can configure devices using known OD structures.
- Multi-vendor interchangeability improves, reducing lock-in and simplifying spare-part strategy.

CANopen — Strengths and Weaknesses (Engineering View)

Strengths

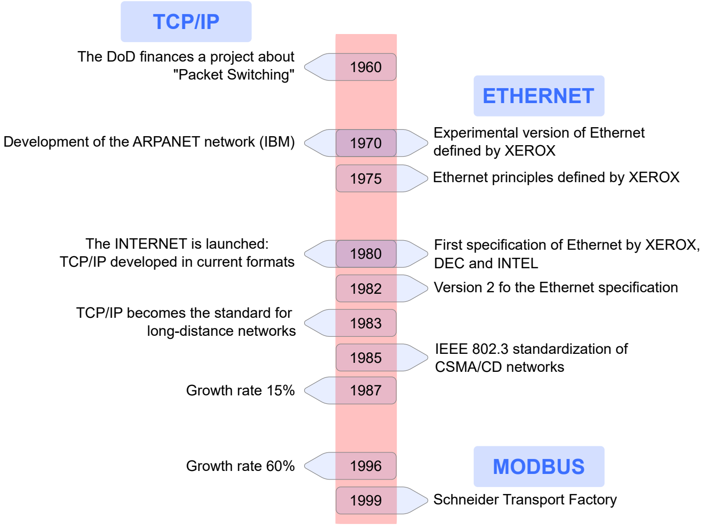
- Robust physical/link layer (EMC resilience + strong error handling).
- Priority arbitration enables deterministic *message precedence* for critical traffic.
- Clean separation: **PDO** for real-time process data, **SDO** for configuration/diagnostics.
- Mature ecosystem (profiles, tools, gateways, widespread device availability).

Weaknesses / constraints

- Limited payload per frame (8 bytes in classical CAN) $\Rightarrow$ segmentation overhead for parameters.
- Rate–distance trade-off; wiring/termination mistakes immediately degrade reliability.
- Shared bus (not switched) $\Rightarrow$ scalability requires disciplined busload engineering.
- Deterministic behavior depends on correct identifier planning and traffic design.

CANopen excels in robust, low-level control applications; careful engineering is needed to fully leverage its strengths.

Modbus Ethernet TCP/IP – Historical Development



TCP/IP

1960 – The DoD finances a project about "Packet Switching"

ETHERNET

1970 – Development of the ARPANET network (IBM)

1970 – Experimental version of Ethernet defined by XEROX

1975 – Ethernet principles defined by XEROX

1980 – The INTERNET is launched: TCP/IP developed in current formats

1980 – First specification of Ethernet by XEROX, DEC and INTEL

1982 – Version 2 fo the Ethernet specification

1983 – TCP/IP becomes the standard for long-distance networks

1985 – IEEE 802.3 standardization of CSMA/CD networks

1987 – Growth rate 15%

MODBUS

1996 – Growth rate 60%

1999 – Schneider Transport Factory

Modbus Ethernet TCP/IP – Layer Coverage

7	APPLICATION	Modbus	HTTP	FTP	BootP DHCP	---
6	PRESENTATION	EMPTY				
5	SESSION	EMPTY				
4	TRANSPORT	TCP				
3	NETWORK	IP				
2	LINK = LLC + MAC	CSMA/CD				
1	PHYSICAL	Ethernet V2 or 802.3				

Physical layer (Ethernet): topology, distance, speed, node count

Topology: Ethernet supports multiple topologies. Historically: shared bus (coax). Today: **star/tree** via switches is dominant.

Distance and speed trade-off:

- Copper twisted pair is commonly limited to **100 m per link** (typical 100/1000BASE-T).
- Fibre extends distance (hundreds of meters to **kilometers**) and improves EMC immunity.
- Higher speeds tighten link budgets and require better cabling, connectors, and installation quality.

Node count:

- In practice, the number of devices is limited by **switch ports**, **addressing**, and (most importantly) **traffic engineering** (broadcast domains, scanning rates, congestion).
- "Determinism" is improved by **segmentation** (VLANs), **managed switches**, and controlling background services.

Transmission media: coax, twisted pair, fibre

Why media selection matters in plants:

- **EMC/EMI:** drives, inverters, contactors, and long cable runs create harsh environments.
- **Installation:** bending radius, connector robustness, grounding/shield termination.
- **Maintenance:** fault isolation, segment replacement, and standardization of spare parts.

Typical modern practice:

- **Shielded twisted pair** for most machine-level Ethernet.
- **Fibre backbones** between cabinets, buildings, or noisy areas.
- Coax is mostly legacy (rare in new industrial deployments).

Design tip for Modbus/TCP: Even though Modbus payloads are small, overall performance is often dominated by **polling strategy**, **scan rate**, and **switch queuing** rather than raw bitrate.

Transmission media: classic Ethernet variants

From the provided table: Ethernet exists on three media families:

- **Coaxial cable** (legacy bus Ethernet)
- **Shielded twisted pair** (dominant at machine level)
- **Optical fibre** (dominant for backbones / long distance / EMI immunity)

Medium	Name	Description	Speed	Max. length
Coax	10BASE-5	Thick Ethernet	10 Mbps	500 m
Coax	10BASE-2	Thin Ethernet	10 Mbps	185 m
Twisted pair	10BASE-T	Twisted pair	10 Mbps	100 m
Twisted pair	100BASE-TX	Twisted pair (Cat 5)	100 Mbps	100 m
Fibre	10BASE-F	2 fibres	10 Mbps	2000 m
Fibre	100BASE-FX	2 fibres	100 Mbps	2000 m

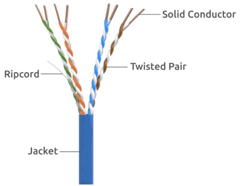
Twisted pair (UTP/STP), categories, connectors

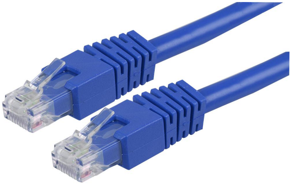
Twisted pair principles:

- Twisting reduces loop area → lowers susceptibility and emissions.
- Shielding (STP) improves noise immunity but requires correct **360° shield termination** and a clear grounding concept.

Category and speed:

- Higher categories support higher frequency content with lower attenuation/crosstalk.
- In industrial settings, prefer cables/connectors rated for vibration, oils, temperature.





Connector note: RJ45 is common in IT; many industrial installations use ruggedized connectors (e.g., M12 variants) while keeping Ethernet electrical compatibility.

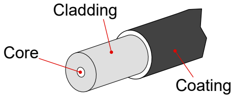
Optical fibre: why and where it is used

Why fibre in industrial networks:

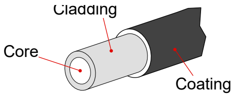
- **EMI immunity:** no electrical current in the transmission medium.
- **Galvanic isolation:** breaks ground loops between distant cabinets/buildings.
- **Distance:** supports long links (often used as **backbones**).
- **Security/resilience:** harder to tap with simple electrical methods; less sensitive to lightning-induced surges.

Practical engineering:

- Multimode is common for shorter in-plant runs; single-mode for longer distances.
- Fibre requires attention to connectors, cleanliness, bend radius, and patch panel management.



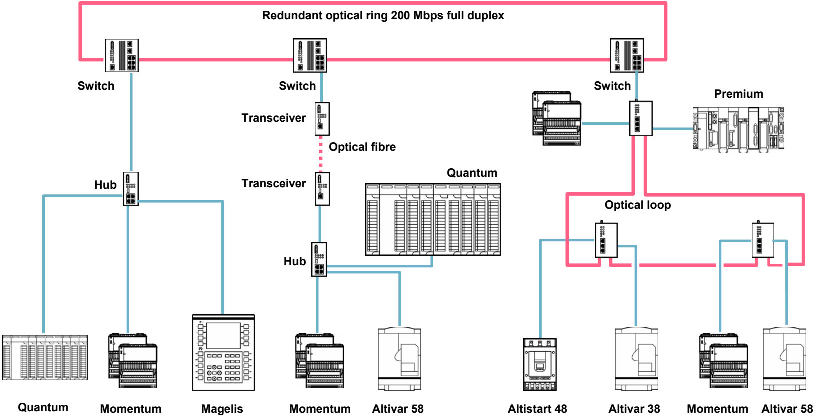
Single-Mode Fiber (SM)



Multimode Fiber (MM)

For Modbus/TCP: fibre does not change the protocol, but can drastically improve robustness and availability in high-noise environments.

Modbus Ethernet TCP/IP fibre example



Transport / network / link layers: CSMA/CD, packets, and reliability

Key characteristics of Modbus Ethernet TCP/IP:

- **Medium access method: CSMA/CD** (Carrier Sense Multiple Access with Collision Detection).
- **Determinism:** "resolved using segmentation" with **load factor** $< 10\%$.
- **Transmission method:** in **packets** / IP datagrams, **64 to 1500 bytes**.
- **Max useful data: 1442 bytes per packet (application protocol data unit-APDU)**.
- **Transmission security:**
  - **CRC32** at link layer level
  - **Acknowledgement** at TCP link level
  - **Response** at application level (UNITE/Modbus)

Industrial interpretation:

- TCP reliability can increase latency variability under loss/retransmissions.
- Modbus/TCP performance is governed by the **polling schedule**, timeouts, and the background traffic.
- Segmentation (VLANs, separate cells/lines, managed switching) is the primary lever for predictability.

Coexisting application protocols on TCP/IP networks

Other application protocols commonly found alongside Modbus/TCP:

- **HTTP:** web transfer (HTML)
- **FTP:** file transfer (client/server)
- **SNMP:** network management (configuration/monitoring/administration)
- **DNS:** name-to-IP translation
- **BOOTP:** IP address assignment by server
- **TELNET:** terminal interfacing (ASCII)
- **UNITE & MODBUS:** client/server protocols (UNITE by Telemecanique; Modbus by Modicon)
- **I/O scanning:** periodic I/O update via automatic Modbus requests

Why you must care: These "other" protocols generate background traffic and maintenance activity. Therefore, for stable Modbus/TCP timing, separate: **control traffic** (scan cycles) from **engineering/IT traffic** (web/FTP/management), using segmentation and policies (QoS, ACLs, managed switching).

Transparent Ready: Implementation classes overview (definition + device families + identifier structure)

Definition: Implementation classes define a **list of services** to implement in order to ensure interoperability of **Transparent Ready** products.
Device families listed:

- **Controllers:** PLC, numerical controllers, etc.
- **Devices:** drives, motor starters, remote I/O
- **Gateways**
- **HMI/SCADA**

Identifier structure:

- a **letter A to Z** relating to **WEB services**
- followed by a **number 00 to 99** relating to **user services and communication**
- and an **ASCII suffix** relating to the **physical layer**

Implementation classes: combining Web level + Modbus access level

Implementation class structure:

- **Web services level** on server/client:
  - Server: A (without web) → E (distributed)
  - Client: A (without web) → W (distributed)
- **User/communication levels** include:
  - 00: without Modbus
  - 01: Modbus basic access
  - 05: Modbus regular access
  - 10/20/30/40: Modbus on TCP-IP (basic/management/added-value/distributed control)

Implementation classes: combining Web level + Modbus access level

Use this as a practical rubric to classify devices based on their **capabilities at both the web service level and the Modbus access level**

- **Commissioning**: requires configuration + diagnostics services
- **Operations**: requires monitoring + alarm services
- **Maintenance**: requires documentation + upgrade paths

Web services levels: what is mandatory vs. optional

Extracted structure: Web services are organized across typical lifecycle functions:

- Maintenance, Monitoring, Diagnostics, Documentation, Configuration

and characterized by whether they are **mandatory** or **optional** at given levels.

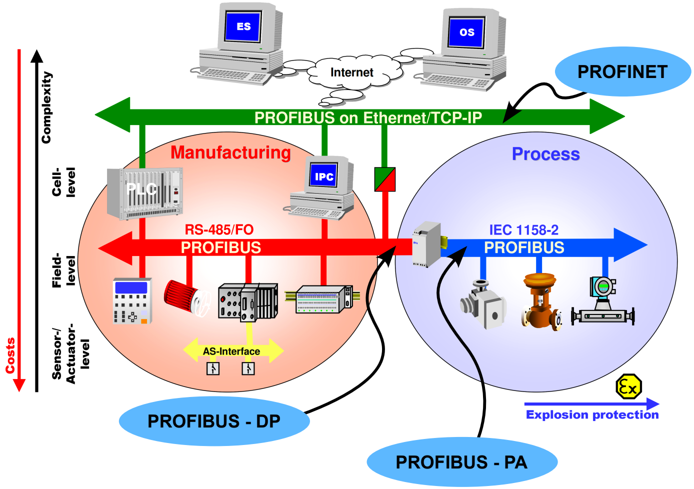
Industrial system thinking:

- Web services improve usability, but they add traffic and potential exposure.
- In control networks, isolate web traffic (separate VLAN/segment) and restrict access.
- For reliable Modbus/TCP operation, enforce:
  - bounded scan traffic (known cycle time),
  - controlled background services,
  - monitoring via SNMP/management but without flooding the control plane.

PROFIBUS-DP – History and ecosystem

- **1987:** a German federal initiative creates a **“Fieldbus” working group** (13 organizations incl. **SIEMENS** and research institutes).
- **PROFIBUS** (**PRO**cess **FI**eld**BUS**) is born as a standardized approach for industrial networks.
- Governance and community:
  - Managed via a user group: **PROFIBUS CLUB**.
  - International user clubs in major industrial countries provide support in local languages.
  - Coordinated by **PROFIBUS International (PI)** (the material indicates > 750 members).
- Key idea: a broad ecosystem enables **multi-vendor interoperability** and long-term maintenance.

PROFIBUS- types overview



PROFIBUS-DP – Layer coverage in the ISO/OSI model

		FMS Profiles	DP Profiles
			DP Functions
7	APPLICATION	FMS = Fieldbus Message Specification	
6	PRESENTATION	EMPTY	
5	SESSION	EMPTY	
4	TRANSPORT	EMPTY	
3	NETWORK	EMPTY	
2	LINK = LLC + MAC	FDL = Fieldbus Data Link	
1	PHYSICAL	RS485 or optical fibre	

Where PROFIBUS-DP sits in the communication stack

A practical, engineering-oriented view

- **Physical layer:** typically RS-485 electrical signaling over shielded twisted pair, with **bus topology** and **termination** to control reflections.
- **Link layer (FDL):** defines **medium access**, frame control, addressing, and timing.
  - Determinism via **token passing** among active stations (masters).
  - **Master–slave** polling for cyclic exchange with field devices.
- **Application (DP):** cyclic I/O (process data) and acyclic services (parameters, diagnostics).
- **Interoperability:** enforced by **profiles** (e.g., drives, encoders) and **GSD files**.

Key message

PROFIBUS-DP is designed for **predictable control-loop traffic**:

- cyclic exchange with bounded latency,
- explicit configuration,
- structured diagnostics and parameterization.

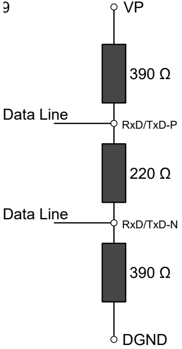
Physical layer

Topology and wiring philosophy

- **Topology: bus** with **active line terminators**.
- Termination is not cosmetic: it shapes signal integrity, reduces reflections, and stabilizes bit error rate under EMI.
- Repeaters extend reach and allow segmentation; each segment must respect termination rules.

Distance, speed, and node count

- **Speed range: 9.6 kbps to 12 Mbps**.
- **Maximum distance:** depends on medium and speed.
  - **Min:** 100 m at 12 Mbps (without repeaters).
  - **Max:** 4800 m at 9.6 kbps (with 3 repeaters).
- **Max. stations:** 32 without repeaters; 124 with 3 repeaters.



Engineering implications of the physical layer

Why “distance depends on medium and speed” is the key sentence

- PROFIBUS-DP is commonly deployed on RS-485: cable attenuation + reflections + EMI determine the feasible baud rate for a given trunk length.
- **Trade-off:** higher bit rate ⇒ shorter segment length (lower link margin), and stricter installation quality requirements.
- **Active termination:** ensures a defined bias/impedance; incorrect termination is a frequent cause of intermittent faults.
- **Repeaters:** extend distance and node count but introduce additional devices and require disciplined segment-level design (end-of-segment termination, grounding, shielding strategy).

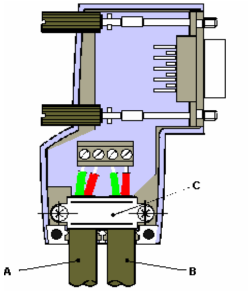
Engineering implications of the physical layer

Practical checklist for labs and plants

- Bus topology: avoid stubs.
- End nodes terminated; mid nodes not.
- Shielding continuity and 360° bonding.
- Cable type and characteristic impedance.
- Ground reference and equipotential bonding.
- Baud rate selection tied to length.
- Repeater placement and segmenting.
- Node count planning (32/124 guidance from material).
- EMC: separation from power cables.
- Commissioning: validate with diagnostics.

Types of connectors used in PROFIBUS-DP

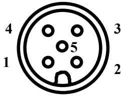
IP20



9-pin SUB D
Female, product side
with or without line
terminator

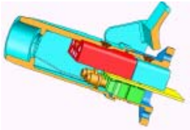
IP65

M12 connector



Female, Product side

Han-Brid



DESINA recommendation

Link layer: hybrid access method

Hybrid access method in PROFIBUS

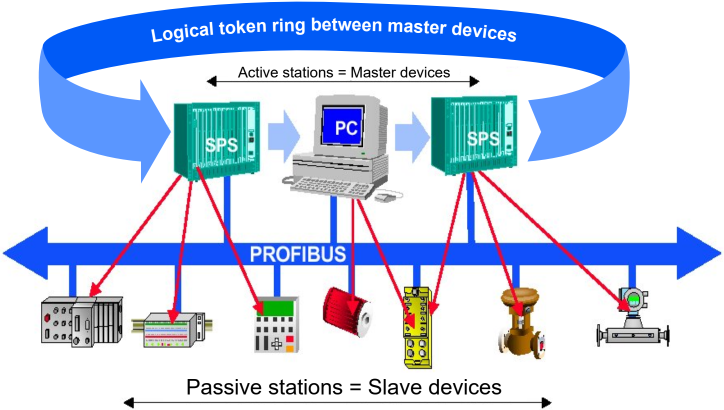
PROFIBUS uses a **hybrid access method**:

- Communication between **active stations** is based on the **token ring** concept.
- **Passive stations (slaves)** use the **master–slave** concept.

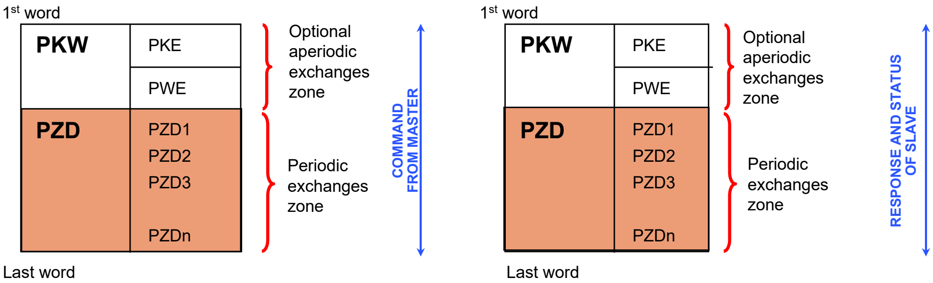
Why this matters for control

- Token passing bounds contention: only the token holder initiates traffic in its time window.
- Master–slave polling yields predictable cyclic I/O updates (crucial for drives, remote I/O, process automation).
- Diagnostics and parameter access can be integrated without breaking the cyclic exchange structure.

Example architecture PROFIBUS-DP


Logical token ring between master devices
Active stations = Master devices
SPS
PC
SPS
PROFIBUS
Passive stations = Slave devices

PPO concept- PROFIBUS-DP



PPO concept: cyclic + (optional) acyclic exchanges

Core message from the PPO diagram

- The master sends a **cyclic request** to the slave; the master receives a **cyclic response**.
- Data layout separates:
  - **PZD** (*Prozessdaten*): **process data** for periodic exchange (e.g., control word, setpoints, status, actual values).
  - **PKW** channel: parameters/diagnostics (aperiodic, when needed).
- The figure explicitly distinguishes:
  - **Periodic exchanges zone** (real-time I/O),
  - **Optional aperiodic exchanges zone** (parameters, service).

Field interpretation

- In drives, keep PZD minimal and stable for control performance.
- Use PKW for commissioning, diagnostics, and parameter tuning with controlled timing impact.

PKW fields and read/write semantics

Terms defined

- **PKW** = Parameter–Kennung–Wert (Parameter–Address–Value)
- **PKE** = Parameter–Kennung (parameter address / identifier)
- **PWE** = Parameter–Wert (parameter value corresponding to the address)
- **PZD** = Prozessdaten (process data)

PKW usage: command/response patterns

- **PKE output:**
  - Bits 0 to E: address of variable
  - Bit F: 0 = single read/write; 1 = continuous read/write
- **R/W output codes:** 16#0052 = Read; 16#0057 = Write
- **R/W/N input codes:** 16#0052 Read correct; 16#0057 Write correct; 16#004E Read/Write error
- **PWE input:** value / copy / error (0 address incorrect, 1 write refused)

DP communication profiles: who talks to whom

Three station types defined

- **DP master class 1 (DPM1):** programmable controllers such as **PLCs, PCs**, etc.
- **DP master class 2 (DPM2):** development or diagnostic tool.
- **DP slave:** peripheral device performing **cyclic exchanges** with its active station.

Why two master classes?

- DPM1 is the operational master for cyclic control (runtime determinism).
- DPM2 is commonly used for commissioning, diagnostics, parameterization, and service.
- Separation helps avoid uncontrolled service traffic disrupting the control loop.

DP application profiles: standardization by application domain

What an application profile does

- Completes the standard for a given area of application.
- Defines consistent semantics for:
  - process data mapping,
  - device states and commands,
  - diagnostics behavior,
  - parameter sets and scaling rules.

Examples listed in the material

- Numerical controllers and robots (sequential diagrams, movement/command viewpoint).
- Encoders (rotary/angle/linear encoders; functions such as scaling and diagnostics).
- **PROFIDRIVE** variable speed drives (drive commands and states described).
- Process control and supervision (HMI) linking higher-level systems and using extended communication functions.

GSD files: the engineering contract between device and system

Definition

- The characteristics of a PROFIBUS device are described as an **electronic device data sheet** (**GSD**) in a predefined format.
- **GSD files must be provided** by all PROFIBUS device manufacturers.

Sections named in the material

- **General specifications:** manufacturer, product name, HW/SW versions, supported speeds, etc.
- **Specifications relating to masters:** e.g., max. number of slaves, upload/download options (not for slave devices).
- **Specifications relating to slaves:** I/O variable types and counts, diagnostic texts, modular product module information.

GSD files: the engineering contract between device and system

Practical interpretation

- GSD enables **tool-based commissioning**: configuration, address assignment, module selection, and diagnostics integration.
- Missing/incorrect GSD is a common root cause of “works on paper” integration failures.

Strengths and limitations of PROFIBUS-DP (engineering view)

Strengths

- **Deterministic cyclic communication:** token ring + master–slave polling supports bounded timing for industrial control.
- **Structured payload and services:** cyclic PZD for process data, PKW for parameterization/diagnostics when needed.
- **Interoperability at scale:** certification + communication/application profiles + GSD-based engineering.
- **Proven industrial ecosystem:** large installed base, mature troubleshooting practices.

Limitations

- **Physical constraints:** distance–baud trade-off, strict termination and installation rules, sensitivity to wiring errors.
- **Scaling and evolution:** node count and segmenting require careful planning (repeaters, diagnostics, maintainability).

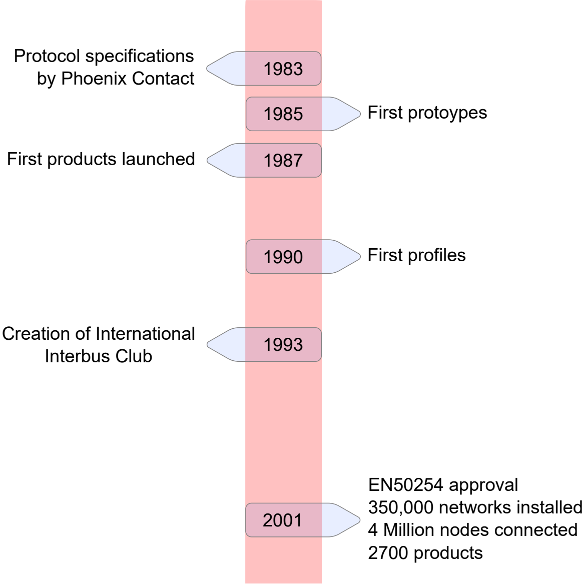
Strengths and limitations of PROFIBUS-DP (engineering view)

- **Service traffic discipline:** uncontrolled acyclic access (parameters/diagnostics) can affect cycle times if poorly engineered.

Suggested classroom exercise

Given a required cycle time and a plant layout, choose (i) baud rate, (ii) segmenting strategy with repeaters, and (iii) which data belongs in PZD vs PKW, then justify determinism and diagnosability.

INTERBUS- history and ecosystem



1983 Protocol specifications by Phoenix Contact

1985 First protoypes

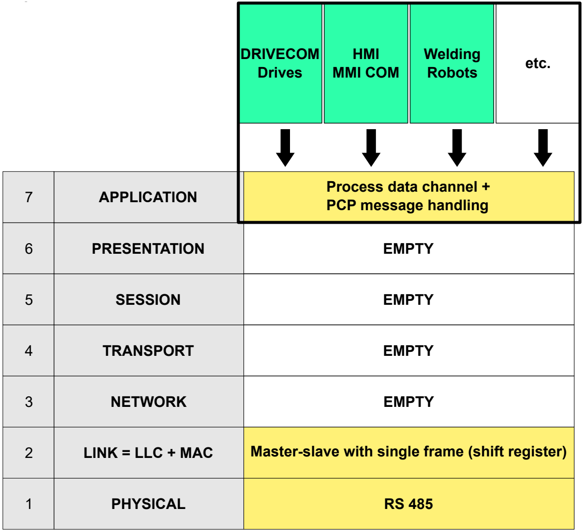
1987 First products launched

1990 First profiles

1993 Creation of International Interbus Club

2001
EN50254 approval
350,000 networks installed
4 Million nodes connected
2700 products

INTERBUS- ISO model



7	APPLICATION	Process data channel + PCP message handling
6	PRESENTATION	EMPTY
5	SESSION	EMPTY
4	TRANSPORT	EMPTY
3	NETWORK	EMPTY
2	LINK = LLC + MAC	Master-slave with single frame (shift register)
1	PHYSICAL	RS 485

Physical Layer : Medium, Topology, Limits

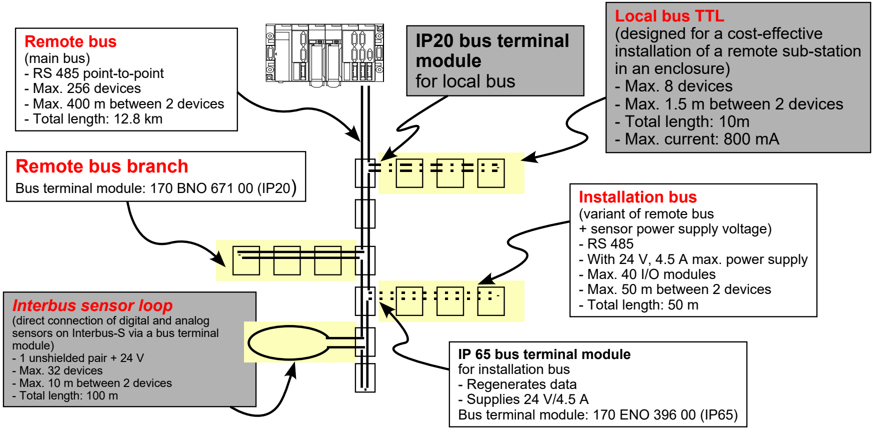
Key points

- **Medium:** Shielded twisted double pair
  - 1 pair for receiving, 1 pair for sending
- **Topology:** Ring type
  - Externally resembles a bus; connecting cable contains the signal loop-back
- **Maximum distance:** 400 m between 2 devices, 12.8 km total
- **Speed:** 500 kbps
- **Max. number of devices:** 512

Engineering interpretation (teaching notes)

- **EMC rationale**: shielding improves noise immunity in industrial environments (drives, contactors, welding).
- **Signal integrity**: segment length and node count constrain attenuation, reflections, and timing margins.
- **Topology constraint**: ring simplifies deterministic scanning but impacts fault handling (ring break, bypassing).

INTERBUS- types



Remote bus
(main bus)
- RS 485 point-to-point
- Max. 256 devices
- Max. 400 m between 2 devices
- Total length: 12.8 km

Remote bus branch
Bus terminal module: 170 BNO 671 00 (IP20)

Interbus sensor loop
(direct connection of digital and analog sensors on Interbus-S via a bus terminal module)
- 1 unshielded pair + 24 V
- Max. 32 devices
- Max. 10 m between 2 devices
- Total length: 100 m

IP20 bus terminal module
for local bus

Local bus TTL
(designed for a cost-effective installation of a remote sub-station in an enclosure)
- Max. 8 devices
- Max. 1.5 m between 2 devices
- Total length: 10m
- Max. current: 800 mA

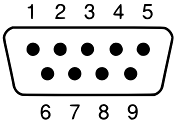
Installation bus
(variant of remote bus
+ sensor power supply voltage)
- RS 485
- With 24 V, 4.5 A max. power supply
- Max. 40 I/O modules
- Max. 50 m between 2 devices
- Total length: 50 m

IP 65 bus terminal module
for installation bus
- Regenerates data
- Supplies 24 V/4.5 A
Bus terminal module: 170 ENO 396 00 (IP65)

INTERBUS- connectors

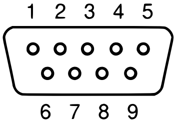
IP20

9 pin SUB D IN



Male, product side

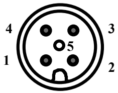
9 pin SUB D OUT



Female, product side

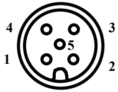
IP65

M12 IN connector



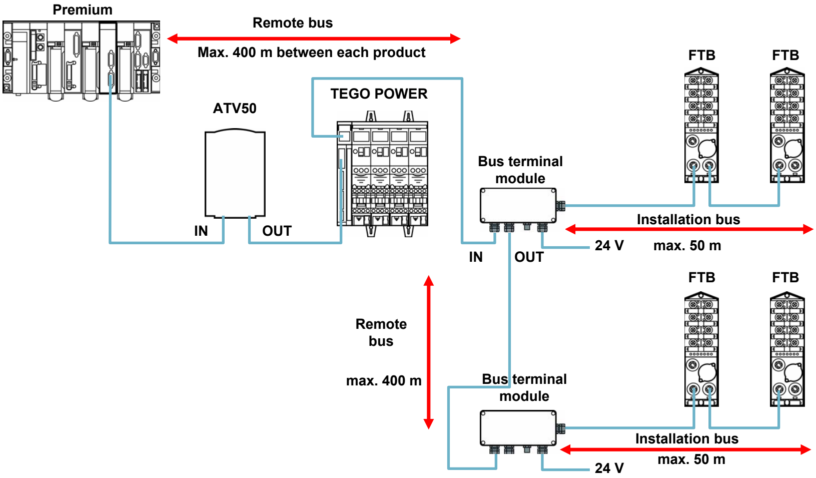
Male, product side

M12 OUT connector



Female, product side

INTERBUS- example architecture



Link Layer: Access Method and Frame Concept

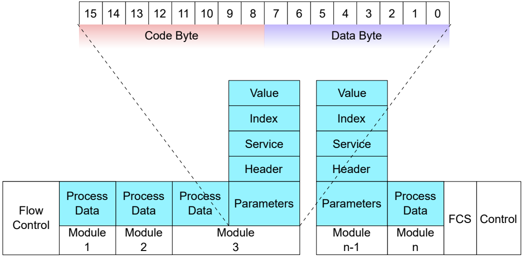
Key points

- **Medium access method:** Master/Slaves
- **Single frame** contains both sensor (input) and actuator (output) data.
- Frame managed like a **shift register** with **max. 256 words**.
- Each slave (station) is a component of the register.
- Hybrid frame supports two data classes (max 32 words per device):
  - Cyclic process data (periodic slave I/O words)
  - Acyclic parameter data (fixed memory space)

Why “shift-register framing” matters

- **Determinism**: the master controls scan order and timing; no collisions.
- **Cycle time scales** with total frame length (words) and bit rate.
- **Practical PLC view**: cyclic data maps into a *process image* (inputs/outputs updated each scan).

Acyclic Exchanges: PCP for Parameter Data



Extracted key points

- Acyclic data is transmitted using **PCP**
- PCP = Peripherals Communication Protocol
- PCP **fragments parameter data**

PCP in practice

- **Motivation**: parameter/diagnostic payloads can exceed what fits conveniently in a cyclic I/O word allocation.
- **Fragmentation**: large parameter objects are split across multiple transfers; reassembled at the endpoint.
- **Real-time discipline**: PCP transactions should be bounded and scheduled so they do not destabilize cyclic scan timing.

Profiles: Interoperability and System Integration

Key points

Interbus profiles define for a product family:

- Recognition of a device by means of its **ID code**
- Format of **command data** (outputs) and **status words** (inputs) exchanged
- The **status chart**

What “profiles” buy you (industrial reasoning)

- **Interoperability**: consistent semantic mapping of bits/words across vendors within a device class.
- **Reduced engineering risk**: fewer vendor-specific ad-hoc conventions for command/status.
- **Commissioning speed**: configuration tooling can auto-recognize devices via ID code and apply expected mappings.

Strengths and Limitations of INTERBUS (engineering view)

Strengths

- **Deterministic cyclic communication**: master/slave polling with shift-register framing supports bounded timing for industrial control.
- **Structured payload and services**: cyclic process data for real-time I/O, acyclic parameterization/diagnostics via PCP.
- **Interoperability at scale**: certification + communication/application profiles + ID-based engineering.
- **Proven industrial ecosystem**: large installed base, mature troubleshooting practices.
- **Scalable distributed wiring**: ring topology with simple cabling and device addition/removal.

Strengths and Limitations of INTERBUS (engineering view)

Limitations

- **Physical constraints**: distance–baud trade-off, strict installation rules, sensitivity to wiring errors.
- **Scaling and evolution**: node count and segmenting require careful planning (repeaters, diagnostics, maintainability).
- **Service traffic discipline**: uncontrolled acyclic access (parameters/diagnostics) can affect cycle times if poorly engineered.
- **Proprietary technology**: limits vendor diversity and long-term support compared to open standards.

MODBUS: History and Core Idea (Application-Layer Protocol)

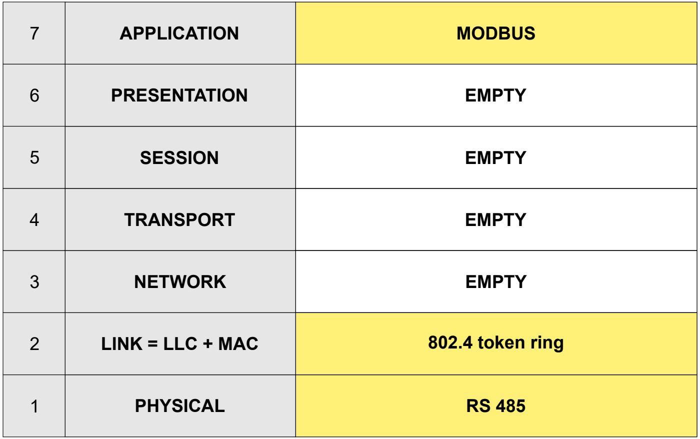
- **Origin (1979):** Created by **MODICON** to connect PLCs to programming tools.
- **Today:** Widely used for **master/client** communications with **slaves/servers** between intelligent field devices (drives, meters, RTUs, remote I/O, gateways).
- **Key concept: Modbus is independent of the physical layer.**
  - Same application protocol can run on **RS232/RS422/RS485** and other media (e.g., fibre, radio) via suitable transport/bridging.
- **Engineering relevance:**
  - Simple request/response model, deterministic polling by the master.
  - Large installed base in industry ⇒ interoperability and maintainability.

MODBUS: MODBUS serial link and the ISO model

7	**APPLICATION**	**MODBUS**
6	**PRESENTATION**	**EMPTY**
5	**SESSION**	**EMPTY**
4	**TRANSPORT**	**EMPTY**
3	**NETWORK**	**EMPTY**
2	**LINK = LLC + MAC**	**Master/Slave**
1	**PHYSICAL**	**RS 485**

MODBUS on a serial link operating at 1200 to 56 Kbps with a master/slave access method.

MODBUS: MODBUS plus and the ISO model



7	**APPLICATION**	**MODBUS**
6	**PRESENTATION**	**EMPTY**
5	**SESSION**	**EMPTY**
4	**TRANSPORT**	**EMPTY**
3	**NETWORK**	**EMPTY**
2	**LINK = LLC + MAC**	**802.4 token ring**
1	**PHYSICAL**	**RS 485**

MODBUS PLUS is a bus operating at 1 Mbps based on a token ring access method which uses the MODBUS message handling structure.

MODBUS: Ethernet TCP/IP

7	APPLICATION		MODBUS
6	PRESENTATION		EMPTY
5	SESSION		EMPTY
4	TRANSPORT		TCP
3	NETWORK		IP
2	LINK = LLC + MAC	CSMA/CD	
1	PHYSICAL	ETHERNET V2 or 802.3	

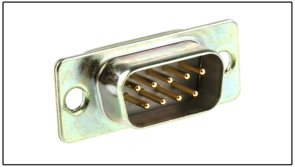
MODBUS Ethernet TCP/IP uses TCP/IP and Ethernet 10 Mbps or 100 Mbps to carry the MODBUS message handling structure.

Modbus on RS485: Physical Layer Essentials

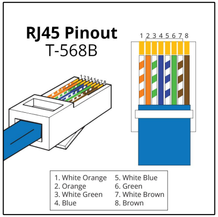
- **Medium: Shielded twisted pair** (differential signalling ⇒ noise immunity).
- **Topology: Bus type** with tap links and **line terminators**
  - Terminate at both ends (typ. 120 Ω) to suppress reflections.
  - Use **bias/failsafe** (pull-up/pull-down) so the bus has a defined idle state.
- **Distance / speed:**
  - Up to **1300 m without repeaters** (practically: depends strongly on baud rate/cable).
  - Example speed: **19,200 bps** (some products up to **56 kbps**).
- **Device count: 32 devices** (1 master + 31 slaves) per segment
  - Repeaters increase segment count and improve loading margin.
- **Practical caution:** half-duplex bus ⇒ only **one talker at a time** (master schedules who talks).

Connectors and Wiring for Modbus RS485

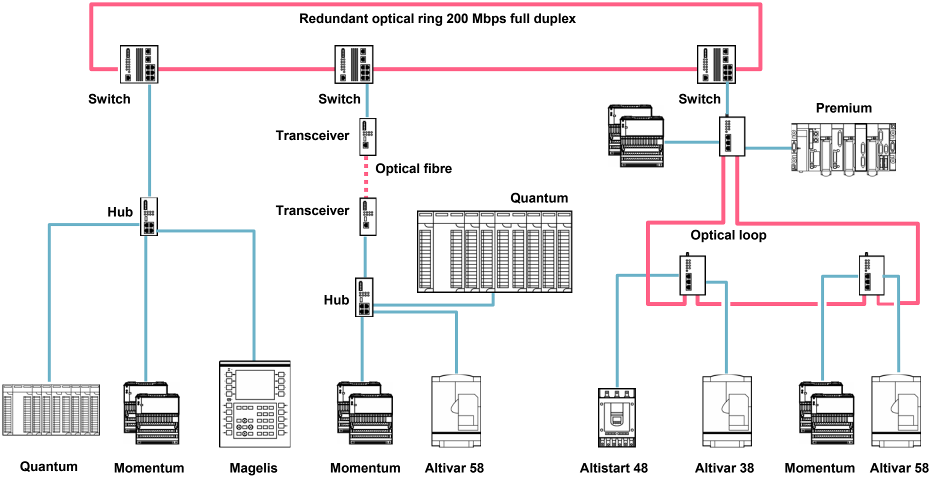
TIA/EIA-485/9-pin SUB-D



TIA/EIA-485/RJ45



MODBUS example



Modbus Serial Link Layer View: Who Talks When, and How Integrity is Ensured

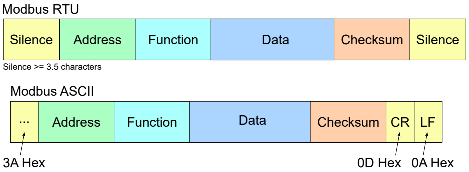
- **Medium access method: Master/Slave**
  - Master initiates all transactions; slaves respond only when addressed.
  - This avoids collisions on a shared serial bus.
- **Transmission method: Client/Server**
  - Client request → server executes function → server response.
  - Typical mapping: **PLC (client)** reads/writes **device registers (server)**.
- **Max useful data: 120 PLC words** (implementation-dependent constraint)
  - In practice, Modbus PDU limits and device tables often drive smaller limits.
- **Transmission security:**
  - **LRC or CRC** for error detection (CRC16 is common in RTU).
  - **Start/stop delimiters** for framing (ASCII) and **silent intervals** (RTU).
  - **Parity bit** (optional) + **continuous flow** constraints at byte level.

Modbus ASCII vs Modbus RTU: Same Functions, Different Framing/Encoding

- **Two serial encodings:**
  - **ASCII mode:** each byte is sent as **two ASCII characters** (hex representation).
  - **RTU mode:** bytes are sent as **binary** (compact), with strict timing.
- **Throughput implication:**
  - **RTU is faster** for the same baud rate (less overhead per payload byte).
  - ASCII is more verbose but human-readable and easier to debug with simple serial tools.
- **Timing robustness:**
  - ASCII allows inserting up to **1 second** time interval between two characters without declaring a framing error (useful on slow/variable links).
  - RTU relies on **inter-frame silence** to detect boundaries (tight real-time framing).
- **Engineering takeaway:** Choose **RTU** for performance and industry default; choose **ASCII** when readability / tolerant timing is valuable.

Structure of a Modbus Frame: Fields and Framing Rules (ASCII vs RTU)

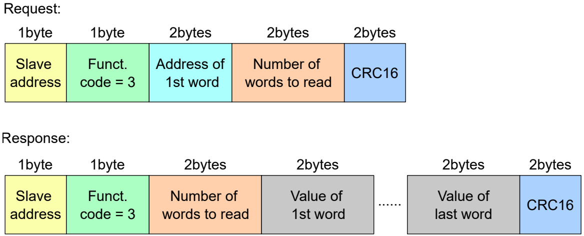
- **Same structure for request and response.**
- **Core fields (conceptual):**
  - **Address:** which slave (unit id) should respond.
  - **Function:** operation (read/write registers/coils, diagnostics, etc.).
  - **Data:** addresses, counts, values (payload).
  - **Checksum:** LRC (ASCII) or CRC16 (RTU) to detect bit errors.
- **Framing differences:**
  - **RTU:** frame bounded by **silence** (e.g., ≥ 3.5 character times).
  - **ASCII:** frame starts with **:** (0x3A) and ends with **CRLF** (0x0D 0x0A).



- **Common debugging workflow:**
  1. Identify frame boundary (silence or :/CRLF).
  2. Decode address + function.
  3. Parse data length/count.
  4. Verify checksum.

Example (RTU Mode): Function Code 03 “Read n Words” (How to Read the Telegram)

- **Function code 03: Read n words** (commonly: **Holding Registers**).
- **Request fields (interpretation):**
  - Slave address (1 byte)
  - Function code = 03 (1 byte)
  - Start register address (2 bytes)
  - Number of registers to read (2 bytes)
  - CRC16 (2 bytes)
- **Response fields (interpretation):**
  - Slave address + function code
  - Byte count (how many data bytes follow)
  - Data bytes: register values (first word . . . last word)
  - CRC16



Request:

1byte	1byte	2bytes	2bytes	2bytes
Slave address	Funct. code = 3	Address of 1st word	Number of words to read	CRC16

Response:

1byte	1byte	2bytes	2bytes		2bytes	2bytes
Slave address	Funct. code = 3	Number of words to read	Value of 1st word	......	Value of last word	CRC16

Modbus Implementation Classes (Transparent Ready): Interoperability by Service Subsets

- **Idea:** Modbus message handling **implementation classes** define a **list of services** to ensure interoperability of Schneider **Transparent Ready** products.
- **Scope note:** For the **server device family** (drives, motor starters, remote I/O, etc.), **three classes** are defined.
- **Meaning in engineering terms:**
  - A device claiming a class must support a **minimum set of Modbus requests** (functions/objects) ⇒ predictable integration effort.
  - Helps system integrators avoid "it connects but does not support the needed function".
- **Classes:**
  - **Basic:** access to words and identification.
  - **Regular:** Basic + bit access + network diagnostics.
  - **Extended:** Regular + other types of access (broader service coverage).
- **Teaching hook:** Present this as a **compatibility contract** between vendor and user.

Strengths and Limitations of Modbus (engineering view)

Strengths

- **Simplicity**: easy to understand and implement; minimal protocol overhead.
- **Wide adoption**: large installed base in industrial automation; many devices support Modbus.
- **Flexibility**: can run over various physical layers (RS485, Ethernet, etc.).
- **Deterministic master/slave model**: predictable communication timing for control applications.
- **Interoperability via implementation classes**: defined service subsets facilitate integration.
- **Open standard**: freely available specification encourages widespread use.

Strengths and Limitations of Modbus (engineering view)

Limitations

- **Limited data types**: primarily supports simple data types (coils, registers); complex types require custom handling.
- **Scalability constraints**: limited to 247 devices per network segment; performance degrades with many devices.
- **Lack of built-in security**: no native encryption or authentication mechanisms; vulnerable to cyber threats.
- **Basic error handling**: relies on simple checksums; lacks advanced error correction features.
- **No formal timing guarantees**: while master/slave is deterministic, Modbus does not specify timing constraints.
- **Vendor-specific extensions**: variations in implementation can lead to interoperability issues.