DEMOCRATIC AND POPULAR REPUBLIC OF ALGERIA

MINISTER FOR HIGHER EDUCATIONAND SCIENTIFIC RESEARCH

IBN-KHALDOUN UNIVERSITY OF TIARET

FACULTY OF APPLIED SCIENCES

DEPARTMENT OF ELECTRICAL ENGINEERING

END-OF-CYCLE PROJECT

For the Bachelor's degree

Field: Science and technology

Major: Automatic

Specialty:Automatic

Theme

Internet of Things (IoT) based system for measuring mechanical constraints

Prepared by:

1-HATI MOHAMED

2-BOUGHALIM ABDELLATIF khaireddine

Jury

R. OTMANI	**President & Examiner**	University of Tiaret
A.ADDA BENATIA	**Examiner**	University of Tiaret
Y. BELHADJI	**Supervisor**	University of Tiaret

Promotion: 2024/2025

Acknowledgements:

- I, Bougalim Abdellatif, would like to begin by praising Allah, who granted me the strength and determination to complete this work. Peace and blessings be upon our Prophet Muhammad, his family, and companions.

- I am deeply grateful to our supervisor, Mr. Belhadji Youcef, for his dedicated efforts, valuable guidance, and continuous support throughout the course of our graduation project.

-My sincere thanks also go to the respected jury members, Mr. Othmani and Mr. Adda Ben Attia, for their valuable time and constructive feedback. A special and heartfelt thank you goes to my father Bougalim Mohamed, my dear mother Braki Khaira, and my beloved grandmother El-Aidi Zineb, who has always been like a second mother to me. Their unwavering support through the most challenging moments of my life has meant everything to me.

- I would also like to express my appreciation to all the esteemed professors in the Department of Electrical Engineering at Ibn Khaldoun University – Tiaret, especially those in the Automation Division, for their dedication, professionalism, and kind-hearted treatment—as if we were their own children or siblings.

- Finally, I extend my warm thanks to my close friends for their moral support and encouragement during this important chapter of my life.

- I, Hati Mohamed, would like to begin by praising Allah, who granted me the strength and determination to complete this work. Peace and blessings be upon our Prophet Muhammad, his family, and companions.
- I am deeply grateful to our supervisor, Mr. Belhadji Youcef, for his dedicated efforts, valuable guidance, and continuous support throughout the course of our graduation project. and thakns to my partner in the project Boughalim Abd Elatif
-My sincere thanks also go to the respected jury members, Mr. Othmani and Mr. Adda Ben attia and Mr.safa
, for their valuable time and constructive feedback. A special and heartfelt thank you goes to my father Wadah Hati, my dear mother Naceur Naima, and my brother Nor El isslam, who has always been like a to me and My brother Ishak and my sister . Their unwavering support through the most challenging moments of my life has meant everything to me.
- I would also like to express my appreciation to all the esteemed professors in the Department of Electrical Engineering at Ibn Khaldoun University – Tiaret, especially those in the Automation Division, for their dedication, professionalism, and kind-hearted treatment—as if we were their own children or siblings.
- Finally, I extend my warm thanks to my coach Bonhare Sahraoui and French Teacher Ms. S.Hadjer to my close friends Mustapha Benanane, Ait djebara Rachid, Yassine Balag, Haidour Mounir, Benalia Abd El Djalil, Aous Djalal,G. Zakaria ,F. Khaled for their moral support and encouragement during this important chapter of my life.
Greetings and thanks to my promo licence Automatique 2025 ...
Thank you all

Table of content

General Introduction

In numerous engineering disciplines and industrial applications, the accurate and real-time monitoring of mechanical constraints is paramount for ensuring structural integrity, optimizing performance, and preventing potential failures. Traditional methods for measuring these constraints often involve localized sensors and manual data collection, which can be time-consuming, costly, and may not provide continuous or remote monitoring capabilities.

The advent of the Internet of Things (IoT) has revolutionized various fields by enabling interconnected devices to collect, exchange, and analyze data. Integrating IoT technologies with mechanical constraint measurement systems presents a significant opportunity to develop intelligent, efficient, and remotely accessible monitoring solutions. By leveraging low-cost sensors, wireless communication protocols, and cloud-based platforms, it becomes possible to create a distributed network capable of providing continuous and comprehensive data on the mechanical stresses and strains experienced by structures and components.

This project explores the development of an IoT-based system for measuring mechanical constraints. This approach aims to provide a cost-effective and scalable solution for real-time monitoring, data logging, and analysis of mechanical stresses in various applications. The integration of IoT principles allows for remote access to critical data, enabling proactive maintenance, improved safety, and enhanced operational efficiency. This work will delve into the key components of such a system, including sensor selection, data acquisition, communication protocols, data processing, and potential applications, highlighting the benefits and challenges of utilizing IoT for mechanical constraint measurement.

Chapter 1: Introduction to sensors – types and applications

1.1 Introduction

Sensors are devices that detect and respond to physical inputs from the environment such as light, heat, motion, pressure, or sound. These inputs are converted into readable signals, which can be displayed or transmitted electronically for further analysis or control. In the context of the Internet of Things (IoT), sensors act as the interface between the physical world and the digital system, enabling intelligent environments where systems can monitor, process, and act based on real-time data. By collecting and analyzing this data, sensors support smarter decision-making in various domains such as smart homes, industrial automation, aerospace, agriculture, and healthcare. Essentially, sensors serve as the 'eyes' and 'ears' of IoT networks, allowing physical phenomena to be translated into actionable information.

1.2 Classification of Sensors and transducers

A transducer primarily focuses on converting one form of energy into another, whereas a sensor concentrates on detecting and measuring a specific input. Additionally, transducers often require an external power source, while sensors can function using the energy from the input they detect. Sensors can be classified using different criteria. This section provides an overview of classifications based on power requirements, output type, physical interaction, operating principle, and application.

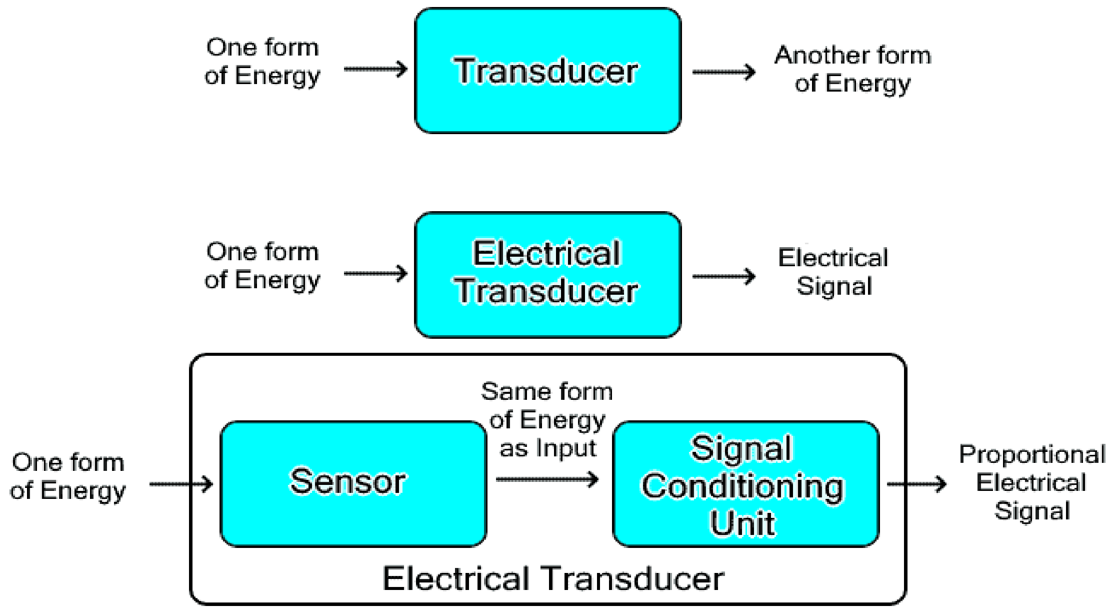


Figure 1: Comparison between sensor and transducer

Classification Basis	**Categories**	**Example Types**
Power Requirement	Active, Passive	Weather satellite sensors, Infrared sensors
Output Signal	Analog, Digital	Thermocouples, Digital pressure sensors
Physical Interaction	Contact, Non-contact	Tactile sensors, Proximity sensors
Operating Principle	Resistive, Capacitive, Inductive, Optical, etc	Thermistors, LDRs, Hall sensors
Application-Specific	Based on environment monitored	Speed, Pressure, Rain, Chemical sensors

Table 1: Overview of Sensor Classifications

1.3 Force Sensors

Force sensors are instruments designed to convert mechanical force such as tension, compression, torque, or pressure into measurable electrical signals. These sensors are integral to numerous industries, ranging from medical devices and robotics to automotive systems and industrial automation. The importance of force sensors lies in their ability to provide accurate, real-time feedback on physical interactions, enabling better control and monitoring.

1.3.1 Operating Principles

Force sensors operate based on several physical principles, including resistive, piezoelectric, capacitive, and optical mechanisms. Each type uses a distinct method to detect force and convert it into an electrical output:

- **Resistive:** Utilizes strain gauges that change resistance under mechanical stress.
- **Piezoelectric:** Generates electric charge in response to mechanical deformation.
- **Capacitive:** Detects changes in capacitance resulting from force-induced displacement.
- **Optical:** Measures force based on changes in light transmission or reflection.

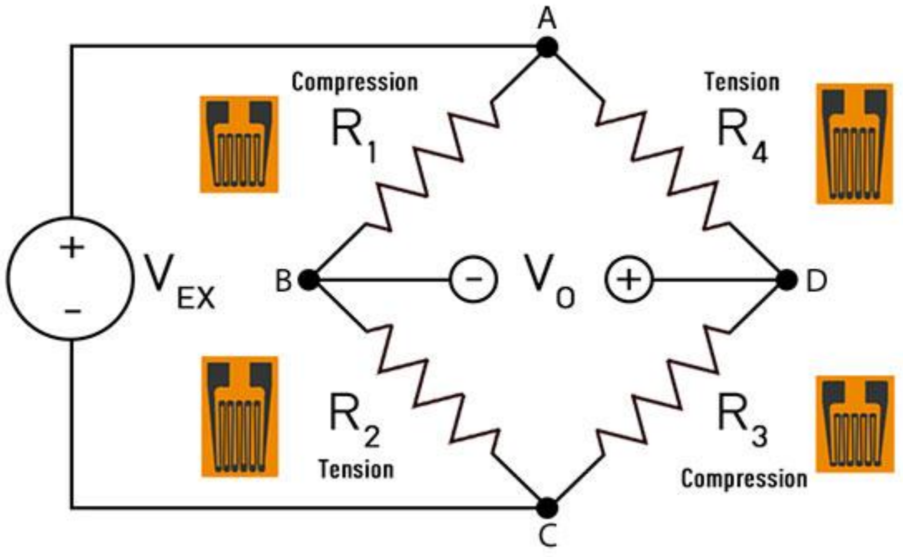
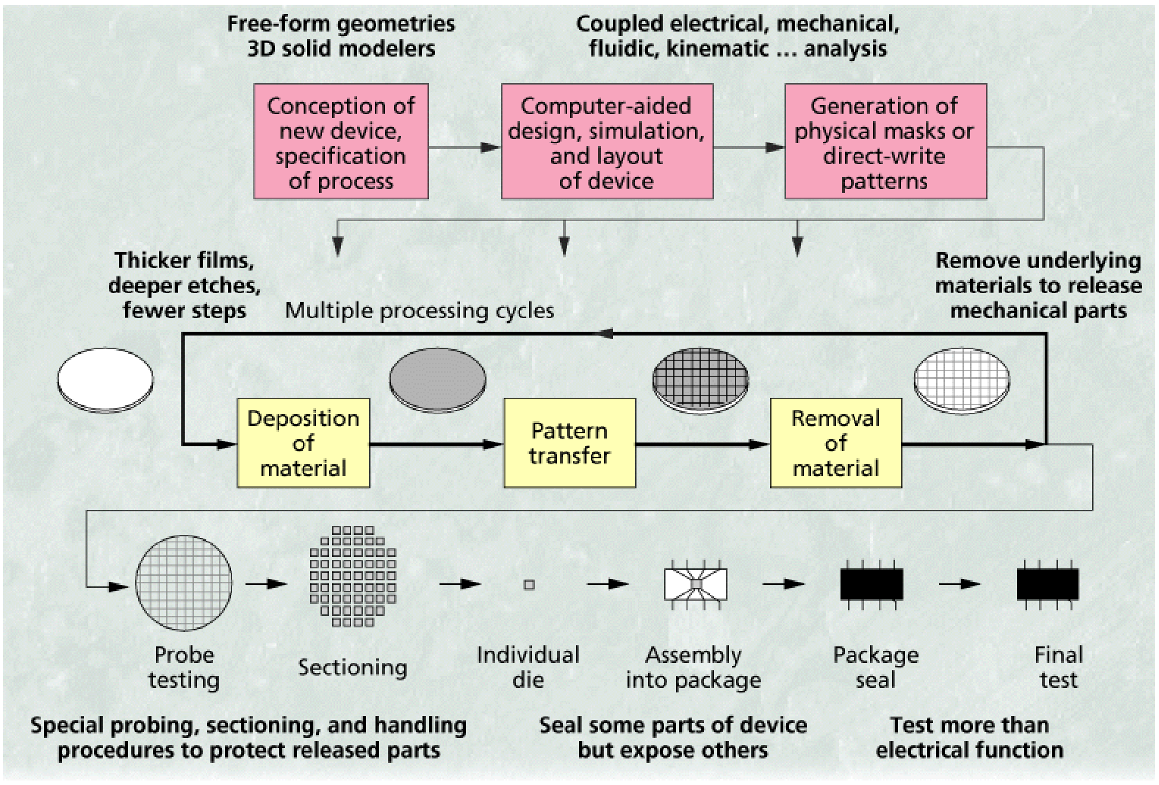


Figure 2: Diagram of Wheatstone bridge circuit for strain gauge force sensor

1.3.2 MEMS Fabrication and Design

Micro-Electro-Mechanical Systems (MEMS) enable the miniaturization of force sensors through photolithographic fabrication processes. These processes involve etching and bonding silicon wafers to create tiny, integrated structures capable of measuring pressure and force. Key steps in MEMS fabrication include the deposition of protective oxide layers, KOH etching for diaphragm formation, and anodic bonding with Pyrex wafers. Resistors are patterned using thermal evaporation and protected with Parylene coatings.


Free-form geometries
3D solid modelers
Coupled electrical, mechanical, fluidic, kinematic ... analysis
Conception of new device, specification of process
Computer-aided design, simulation, and layout of device
Generation of physical masks or direct-write patterns
Thicker films, deeper etches, fewer steps
Multiple processing cycles
Remove underlying materials to release mechanical parts
Deposition of material
Pattern transfer
Removal of material
Probe testing
Sectioning
Individual die
Assembly into package
Package seal
Final test
Special probing, sectioning, and handling procedures to protect released parts
Seal some parts of device but expose others
Test more than electrical function

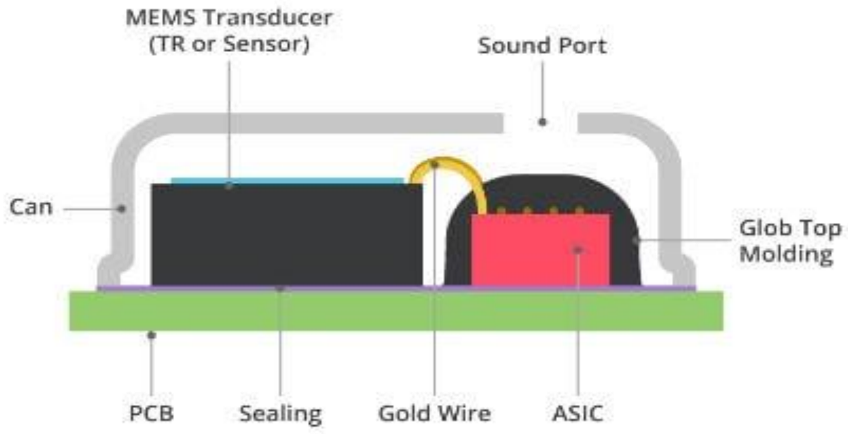


Figure 3: a) MEMS fabrication process flow b) Typical architecture of a MEMS microphone

1.3.3 Application areas

Force sensors are used across a variety of sectors, as detailed in the table below:

Sector	Example applications
Industrial	Assembly lines, Load monitoring in presses
Medical	Dialysis machines, Endoscopic tools, Physical therapy equipment
Automotive	Seat occupancy detection, Brake sensors, Axle load monitoring
Consumer Electronics	Weighing scales, Mobile pressure sensing
Aerospace	Structural feedback, Flight data logging, Autopilot support

Table 2: Application Areas of Force Sensors

1.3.4 Advantages and disadvantages of force sensors

Each type of force sensor has specific strengths and limitations. Choosing the appropriate sensor depends on factors such as application requirements, environmental conditions, accuracy, response time, and durability. For instance, strain gauge sensors are cost-effective and provide excellent linearity but require frequent calibration. Piezoelectric sensors are highly sensitive and ideal for dynamic measurements but not suitable for static force detection. Capacitive sensors are precise but can be affected by environmental factors like humidity and temperature.

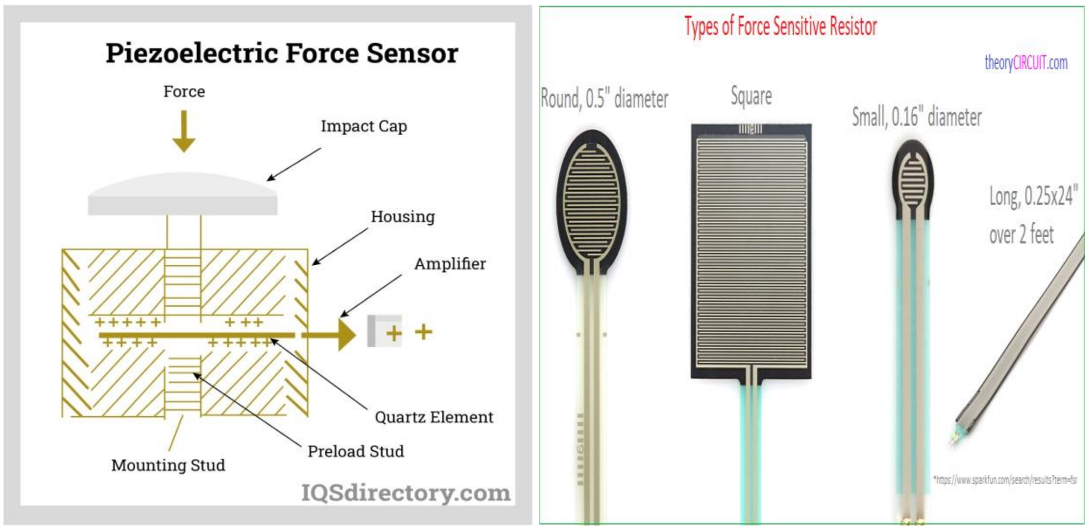


Figure 3: Force sensor types

Sensor Type	**Operating Principle**	**Advantages**	**Limitations**	**Common Applications**
Strain Gauge	Change in resistance due to deformation *(Fraden, 2016)*	High accuracy, cost-effective, mature technology	Temperature sensitive, requires amplification	Weighing scales, structural monitoring
Piezoelectric	Generates voltage under dynamic mechanical stress *(Webster, 2010)*	High-frequency response, no external power required	Not suitable for static force measurement	Vibration monitoring, dynamic pressure sensing
Capacitive	Change in capacitance with deformation *(Fraden, 2016)*	High sensitivity, good resolution, small size	Sensitive to temperature and humidity	MEMS sensors, touch sensing, robotics
Inductive (LVDT)	Variation in inductance due to displacement *(Beckwith et al., 2007)*	Robust, non-contact, excellent for harsh environments	Large size, complex signal conditioning	Automotive, aerospace actuators
Optical	Measures changes in light intensity or phase *(Fraden, 2016)*	Immune to EMI, highly precise, can be multiplexed	High cost, complex to align and maintain	Biomedical devices, fiber-optic pressure monitoring
Magnetic (Hall effect)	Detects magnetic field variations *(Webster, 2010)*	Contactless, durable, works in dirty environments	Limited to magnetic materials, lower sensitivity	Proximity sensors, industrial automation
MEMS-based	Micro-fabricated sensors using diverse principles *(Dally & Riley, 1993)*	Miniaturized, integrable, low power	Complex fabrication, non-linear behavior at extremes	Wearables, smartphones, biomedical implants

Table 3: Comparison of force sensor technologies

1.3.5 Mathematical Modeling of Force and Strain Sensors

Mathematical modeling of force and strain sensors is fundamental for understanding how mechanical inputs are converted into measurable electrical signals. It also allows for accurate interpretation of physical phenomena and the design of calibration systems. The primary model is derived from Hooke's Law, uniaxial strain**,** and the electro-mechanical behavior of strain gauges**,** which define the relationship between applied force and material deformation.

1.3.5 Mathematical Modeling of Force and Strain Sensors

Mathematical modeling of force and strain sensors is fundamental for understanding how mechanical inputs are converted into measurable electrical signals. It also allows for accurate interpretation of physical phenomena and the design of calibration systems. The primary model is derived from Hooke's Law, uniaxial strain, and the electro-mechanical behavior of strain gauges, which define the relationship between applied force and material deformation.

a. Basic Principle – Hooke's Law

Hooke's Law describes the relationship between the force applied to an elastic material and its resulting deformation:

$$F = k \times x \qquad (1.1)$$

- F: Applied force (Newtons)
- k: Stiffness or spring constant (N/m)
- x: Displacement or elongation (m)

b. Stress and Strain Relationships

Uniaxial strain is defined as:

$$\varepsilon = \Delta L / L_0 \qquad (1.2)$$

Where:

- ε: Strain (dimensionless)
- ΔL: Change in length
- L_0: Original length

Stress is given by:

$$\sigma = F / A \qquad (1.3)$$

σ: represents the stress (Pa or N/m^2) and A is the cross-sectional area (m^2)

The linear stress-strain relationship (Hooke's Law in materials) is:

$$\sigma = E \times \varepsilon \quad \Leftrightarrow \quad \varepsilon = \sigma / E \qquad (1.4)$$

E represents the Young’s modulus or elastic modulus (Pa)

c. Electrical Model of a Strain Gauge Sensor

The resistance change in a strain gauge is modeled as:

$$\Delta R/R = K \times \varepsilon \qquad (1.5)$$

- ΔR: Change in resistance
- R: Initial resistance
- K: Gauge factor (typically ≈ 2.0 for metal gauges)

The Wheatstone bridge converts this resistance change to voltage:

$$\Delta V = (V_{in} /4) \times K \times \varepsilon \qquad (1.6)$$

- ΔV: Output voltage
- V_{in}: Input voltage to the bridge

d. Final Measurement Equation

Combining all expressions gives the full model for force measurement:

$$\Delta V = (V_{in}/4) \times K \times (F / (E \times A)) \times (1 / L_0) \qquad (1.7)$$

This equation is used to calibrate the sensor and derive force from the electrical signal.

Chapter 2:The Internet of Things (IoT)

Introduction

Over the past few decades, the Internet has evolved far beyond its original purpose of merely connecting computers. It now encompasses an extensive network of physical objects ranging from everyday household appliances to complex industrial machinery capable of collecting, transmitting, and processing data. This paradigm shift, often referred to as the Internet of Things (IoT), has led to an unprecedented surge in the number of connected devices worldwide. These devices generate vast amounts of data, offering new opportunities for monitoring, analysis, and automation across multiple sectors.

The interdisciplinary nature of IoT is one of its defining strengths. It brings together key elements from electronics, computer science, telecommunications, and automation, creating systems that are not only interconnected but also intelligent and responsive.

In the context of mechanical stress measurement, the integration of IoT technologies presents a transformative opportunity. IoT-based systems enable remote monitoring, real-time data acquisition, and greater measurement accuracy, which are crucial for maintaining the integrity and performance of mechanical structures. Furthermore, by continuously collecting and analyzing stress data, these systems pave the way for predictive maintenance and structural health monitoring, reducing the risk of unexpected failures and extending the lifespan of critical components.

2.1 Definition and Core Concepts of IoT

2.1.1 Defining the Internet of Things

The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) provides a standard definition of the Internet of Things (IoT) in its Recommendation Y.2060 (June 2012): "The Internet of Things is a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) Things based on existing and evolving interoperable information and communication technologies."

Breaking down this definition:

- **Global infrastructure for the information society:** This emphasizes that IoT is a worldwide networked system supporting the growing digital society, connecting people, organizations, and devices.

- **Enabling advanced services:** IoT goes beyond simple connectivity; it facilitates sophisticated applications such as smart cities, healthcare monitoring, industrial automation, and more.
- **Interconnecting (physical and virtual) Things:** 'Things' include both tangible physical objects and virtual entities (software-based items), which are connected and communicate within the IoT ecosystem.
- **Based on existing and evolving interoperable information and communication technologies:** IoT leverages current and future technologies that can work together seamlessly, ensuring compatibility and integration across diverse systems.

Beyond the ITU-T formalism, IoT can be described as a vast network of physical objects ("things") embedded with sensors, software, and network connectivity. These capabilities allow the things to collect data from their environment, exchange information with other devices or systems, and potentially act on the data. This integration enables automation, monitoring, and enhanced decision-making in numerous domains.

Key Characteristics of IoT

The Internet of Things exhibits several defining characteristics:

- **Connectivity:** Devices can connect to networks and communicate with each other, forming a web of interconnected things.
- **Sensing:** Devices employ sensors to detect and measure environmental conditions such as temperature, pressure, humidity, motion, and more.
- **Intelligence:** Data collected by devices is processed either locally (on the device), at the edge (close to the data source), or in the cloud to extract useful insights and make autonomous decisions.
- **Scalability:** IoT systems can expand to accommodate vast numbers of devices and handle large volumes of data efficiently.
- **Dynamic Nature:** Devices and the system adapt to changing conditions, such as new devices joining or leaving the network, or varying environmental factors.

2.1.2 The "Things" in IoT

The term "things" in IoT encompasses a broad spectrum of devices with diverse complexity and functions. These range from simple sensors that detect physical phenomena to complex machines and systems capable of sophisticated tasks.

Examples include:

- **Sensors:** Devices like temperature sensors, pressure sensors, humidity sensors, and strain gauges that collect physical data.
- **Actuators:** Components such as motors, valves, and relays that can act upon the environment, often in response to sensor data or commands.
- **Wearable devices:** Gadgets like smartwatches and fitness trackers that monitor personal health and activity.
- **Industrial equipment:** Machines and robots used in manufacturing and other industrial processes, often equipped with embedded IoT capabilities for automation and monitoring.
- **Vehicles:** Cars, drones, and other transportation devices that utilize IoT for navigation, safety, and efficiency.

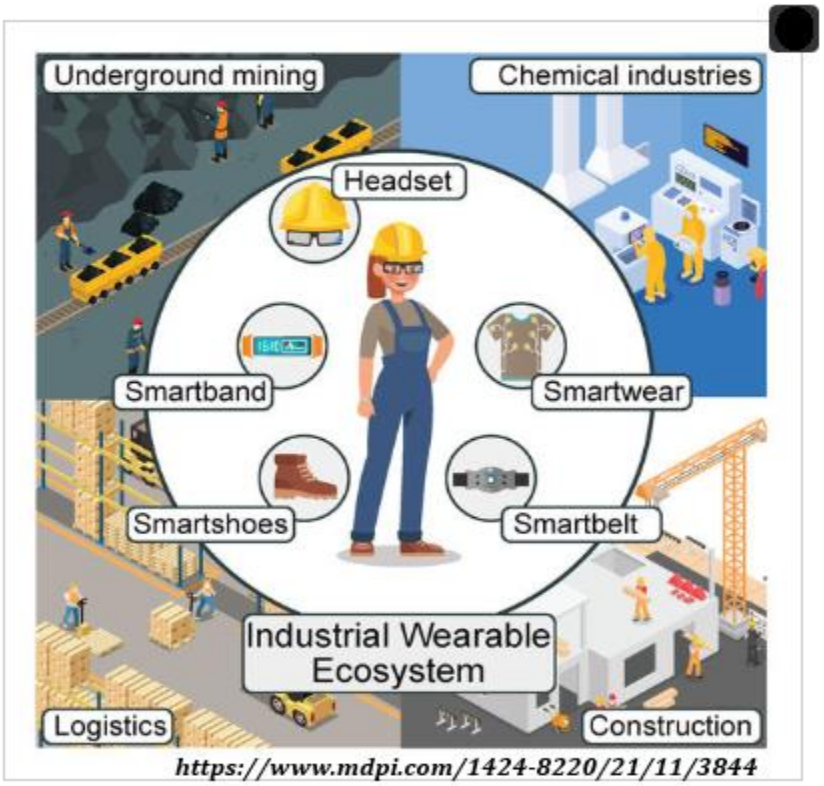


Figure 1. The concept of using wearable devices in industries to maintain work safety.

2.1.3 Embedded Systems and IoT

Embedded systems are specialized computer systems designed to perform dedicated functions or tasks within a larger mechanical or electrical system. Unlike general-purpose computers, embedded systems are optimized for specific control applications and are often constrained in resources such as processing power, memory, and energy consumption. These systems are typically integrated into devices to provide real-time, reliable, and efficient operation tailored to their application environment.

Embedded systems form the core building blocks of IoT devices. They provide essential capabilities including processing power to execute algorithms, memory for storing code and data, and interfaces to communicate with sensors, actuators, and networks. Through embedded systems, IoT devices can gather data from the physical environment, process it locally or send it over communication channels, and actuate responses or control mechanisms accordingly. This integration allows IoT devices to perform dedicated tasks efficiently within a connected ecosystem. Microcontrollers (MCUs) are compact integrated circuits that serve as the "brains" of many IoT devices. They combine a processor core, memory (both volatile and non-volatile), and programmable input/output peripherals on a single chip, enabling efficient control of sensors, actuators, and communication modules. MCUs are selected based on their power consumption, processing capabilities, cost, and compatibility with required interfaces.

Popular MCUs in IoT applications include:

- **ARM Cortex-M series:** Widely used in low-power and real-time applications with rich ecosystem support.
- **ESP32:** A versatile MCU with built-in Wi-Fi and Bluetooth connectivity, favored for IoT prototypes and commercial devices.

Some IoT devices require managing time-critical operations such as sensor data acquisition, control loops, or communication protocols. Real-time Operating Systems (RTOS) provide deterministic scheduling and resource management to ensure tasks meet strict timing constraints. RTOS enable IoT devices to run multiple tasks concurrently with predictable timing, enhancing reliability and responsiveness.

2.1.4 Data in IoT

IoT devices generate data from a wide variety of sources including:

- **Sensor readings:** Measurements from environmental sensors like temperature, pressure, humidity, motion, or strain.
- **Device status:** Operational parameters such as battery level, connectivity status, error logs.
- **User interactions:** Inputs from human interfaces or commands received remotely.

These diverse data sources feed into IoT systems for monitoring, analysis, and decision-making.

IoT data exhibits distinctive characteristics often summarized as the "5 Vs":

- **Volume:** IoT deployments can produce massive amounts of data continuously over time.
- **Velocity:** Data is generated at high speed, often requiring real-time or near-real-time processing.
- **Variety:** Data types include structured (numeric sensor values), semi-structured (JSON or XML messages), and unstructured data (images, audio).
- **Veracity:** Ensuring data accuracy and reliability is critical for meaningful analysis and trustworthy outcomes.

Effective handling of IoT data involves several stages:

- **Data acquisition:** Capturing data from sensors and devices.
- **Preprocessing:** Cleaning, filtering, and transforming raw data for consistency and quality.
- **Storage:** Managing data in databases or cloud repositories optimized for large-scale IoT data.
- **Analysis:** Extracting actionable insights using analytics, machine learning, or rule-based processing to enable intelligent decision-making.

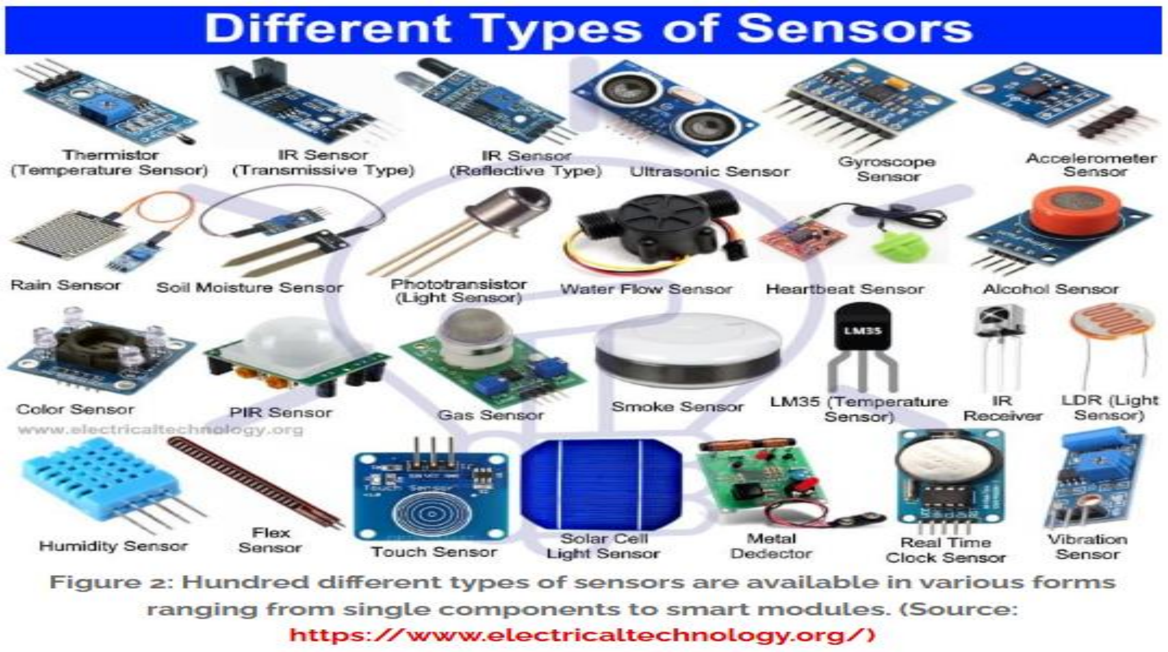

Different Types of Sensors
Thermistor (Temperature Sensor)
IR Sensor (Transmissive Type)
IR Sensor (Reflective Type)
Ultrasonic Sensor
Gyroscope Sensor
Accelerometer Sensor
Rain Sensor
Soil Moisture Sensor
Phototransistor (Light Sensor)
Water Flow Sensor
Heartbeat Sensor
Alcohol Sensor
Color Sensor
www.electricaltechnology.org
PIR Sensor
Gas Sensor
Smoke Sensor
LM35 (Temperature Sensor)
IR Receiver
LDR (Light Sensor)
Humidity Sensor
Flex Sensor
Touch Sensor
Solar Cell Light Sensor
Metal Dedector
Real Time Clock Sensor
Vibration Sensor
Figure 2: Hundred different types of sensors are available in various forms ranging from single components to smart modules. (Source: https://www.electricaltechnology.org/)

Figure 02: The different types of sensors used in IoT

2.2 IoT System Architecture

2.2.1 Layered Architecture

Adopting a layered architecture provides a systematic and modular framework to design and implement IoT systems. This approach helps in separating concerns, enabling developers to focus on specific functional areas, simplifying system maintenance, updates, and scalability. It also promotes interoperability by defining clear interfaces between layers.

A typical IoT architecture is often organized into three or four layers:

- **Perception Layer (Device Layer):** This layer includes sensors and actuators responsible for detecting physical phenomena and interacting with the environment. For applications related to mechanical stress measurement, sensors such as strain gauges and accelerometers are critical. They convert physical parameters into electrical signals for further processing.

The network layer is responsible for connecting IoT devices to each other and to external systems through various communication technologies. This layer manages the transmission of data collected from the perception layer to processing and application layers.

Communication Technologies

- **Short-Range Communication:**
  - **Bluetooth/BLE (Bluetooth Low Energy):** Offers low power consumption with various versions supporting different topologies such as point-to-point and mesh networks. Commonly used for wearable devices and proximity sensing with robust security features.
  - **Zigbee:** Based on IEEE 802.15.4 standard, Zigbee supports mesh networking ideal for home automation and industrial control with low power consumption.
  - **Wi-Fi:** Offers higher data rates and broader coverage, with standards evolving from 802.11 a/b/g/n/ac to ax (Wi-Fi 6). Security protocols include WPA2 and WPA3.
- **Long-Range Communication:**
  - **Cellular (4G/LTE, 5G):** Provides wide-area connectivity with high data rates suitable for mobile and remote IoT devices.
  - **Low Power Wide Area Networks (LPWAN):** Technologies like LoRaWAN, NB-IoT, and Sigfox provide long-range, low-power communications for battery-operated devices in smart cities, agriculture, and asset tracking.
- **Gateways:** Act as intermediaries performing protocol conversion, data aggregation, edge computing, and enforcing security before forwarding data to cloud or processing layers.

The application layer interprets IoT data to provide meaningful services and user applications. It includes software platforms that offer dashboards, control systems, and analytics tools to end users or automated processes.

- Examples include smart home management apps, industrial monitoring systems, predictive maintenance platforms, and data analytics dashboards.
- Application protocols such as HTTP, MQTT, and CoAP facilitate communication between devices and cloud services.

Middleware acts as an intermediary layer providing essential services such as device management, data handling, security enforcement, and service orchestration. It abstracts complexity and enables interoperability among heterogeneous devices and platforms.

Perception Layer

- **Sensors:** Include strain gauges, accelerometers, temperature sensors, and more. Key parameters include accuracy, sensitivity, range, and response time. Sensor interfacing involves converting physical measurements into electrical signals.
- **Actuators:** Devices such as motors, valves, and relays perform actions based on commands from IoT systems.
- **Data Acquisition Systems:** Collect and digitize sensor data for processing.

Network Layer

- **Short-Range Protocols:**
    - Bluetooth/BLE: Versions, GATT profiles, mesh networking, and security.
    - Zigbee: Architecture, protocol stack, and application areas.
    - Wi-Fi: Standards evolution and security protocols.
- **Long-Range Protocols:**
    - Cellular (4G/5G): Architectures and capabilities.
    - LPWAN: LoRaWAN, NB-IoT, Sigfox with their characteristics and trade-offs.
- **Gateways:** Their roles in protocol translation, data aggregation, and security enforcement.

Application Layer

- Protocols: HTTP, MQTT, CoAP.
- Data visualization and integration with other IT systems.

2.2.2 Edge Computing in IoT

Concept: Edge computing processes data closer to where it is generated, reducing dependency on centralized cloud resources.

Advantages: Reduced latency, bandwidth savings, enhanced privacy, and improved system reliability.

Edge Devices: Can include gateways, powerful sensors, and dedicated edge servers capable of running complex analytics locally.

Source: Research papers on edge computing; industry reports; hardware specifications.

2.2.3 Cloud Computing and IoT

Role of the Cloud: Provides scalable infrastructure for storage, processing, and management of IoT data.

Cloud Services for IoT: Device management, data analytics, machine learning integration, and application hosting.

Cloud Architectures: Use of microservices and serverless computing to optimize scalability and cost.

2.3 Communication Protocols in IoT

2.3.1 Importance of Protocols

Protocols enable interoperable, reliable communication between diverse IoT devices, handling varying constraints and data types.

2.3.2 Low-Power, Short-Range Protocols

- **Bluetooth/BLE:** GATT profile, mesh capabilities.
- **Zigbee:** Layered architecture and PRO enhancements.
- **IEEE 802.15.4:** Foundation for Zigbee and others.

2.3.3 Low-Power, Long-Range Protocols (LPWAN)

- **LoRaWAN:** Architecture, device classes, MAC layer, security.
- **NB-IoT:** Cellular integration, features.

- **Sigfox:** Technology overview.
- **Comparison:** Range, data rate, power, latency, cost.

2.3.4 Application-Layer Protocols

- **MQTT:** Publish/subscribe, QoS, suited for constrained devices.
- **CoAP:** RESTful client/server over UDP with DTLS.
- **HTTP/HTTPS:** Common web protocols with security.

2.4 Security in IoT

2.4.1 Security Challenges

- Vulnerabilities at device, network, and application layers.
- Resource constraints, heterogeneity, and scalability issues.

2.4.2 Security Solutions and Best Practices

- **Device:** Secure boot, firmware updates, hardware security modules.
- **Network:** Encryption (TLS/DTLS), authentication, segmentation.
- **Data:** Encryption, access control, privacy technologies.
- **Platform:** Secure cloud infrastructure, API security.

2.4.3 Privacy Considerations

- Importance of user privacy.
- Techniques: Data minimization, anonymization, consent.

Source: Data privacy regulations and articles.

2.5 Applications of IoT

2.5.1 Smart Homes

IoT enables home automation, enhancing security, energy efficiency, and user convenience through interconnected devices such as smart thermostats, lighting, and security systems.

Figure 03 : https://www.iotevolutionworld.com/smart-home/articles/438532-how-secure-smart-home-devices-5-steps.htm

Figure 03 : Smart home with connected devices

2.5.2 Industrial IoT (IIoT)

Industrial Internet of Things (IIoT) refers to the application of IoT technologies in industrial environments such as manufacturing, energy, transportation, and supply chain management. IIoT integrates sensors, actuators, embedded systems, and advanced communication networks to connect industrial assets, enabling real-time monitoring, automation, and data-driven decision-making.

IIoT enhances operational efficiency by providing continuous insights into machine health, production processes, and environmental conditions. Predictive maintenance is a key benefit, where sensor data is analyzed to predict equipment failures before they occur, reducing downtime and maintenance costs. Additionally, IIoT facilitates process optimization through automation and adaptive control systems, improving product quality and throughput.

Security and reliability are paramount in IIoT due to the critical nature of industrial operations. Thus, IIoT systems incorporate robust security protocols, redundant communication paths, and real-time control capabilities, often leveraging edge computing to ensure low-latency responses and resilience against network disruptions.

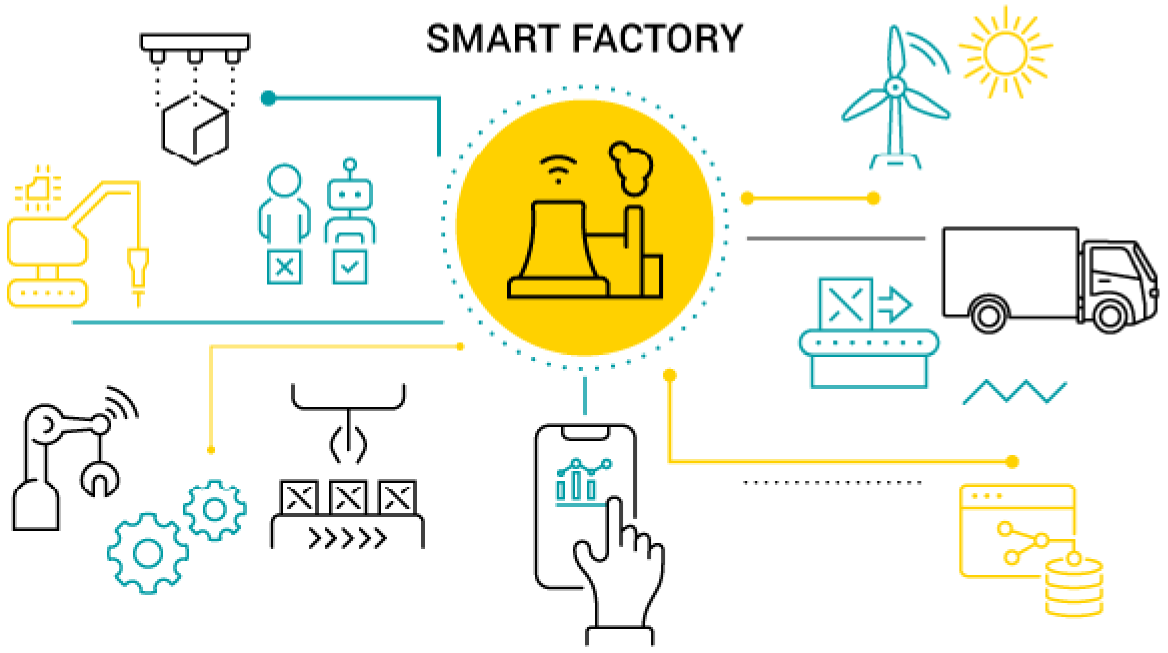


Figure 4 : ***https://avsystem.com/blog/iot/smart-factory***

2.5.3 Smart Cities

Smart Cities utilize Internet of Things (IoT) technologies to enhance the efficiency, sustainability, and livability of urban environments. By integrating sensors, connected devices, and data analytics across infrastructure systems, Smart Cities aim to improve public services, reduce environmental impact, and enhance the quality of life for citizens.

Key applications of IoT in Smart Cities include intelligent transportation systems, which optimize traffic flow through real-time data from road sensors, GPS devices, and connected vehicles. Smart street lighting adjusts brightness based on pedestrian and vehicle presence, conserving energy. Waste management systems use smart bins equipped with sensors to monitor fill levels and optimize collection routes, increasing operational efficiency.

Public safety is also bolstered through connected surveillance cameras, emergency response systems, and environmental monitoring for air quality, noise levels, and weather conditions. Smart utilities manage water and electricity usage more effectively through automated metering and predictive maintenance of infrastructure. As cities face growing populations and resource constraints, IoT-driven Smart City initiatives are crucial for creating sustainable, resilient, and responsive urban ecosystems

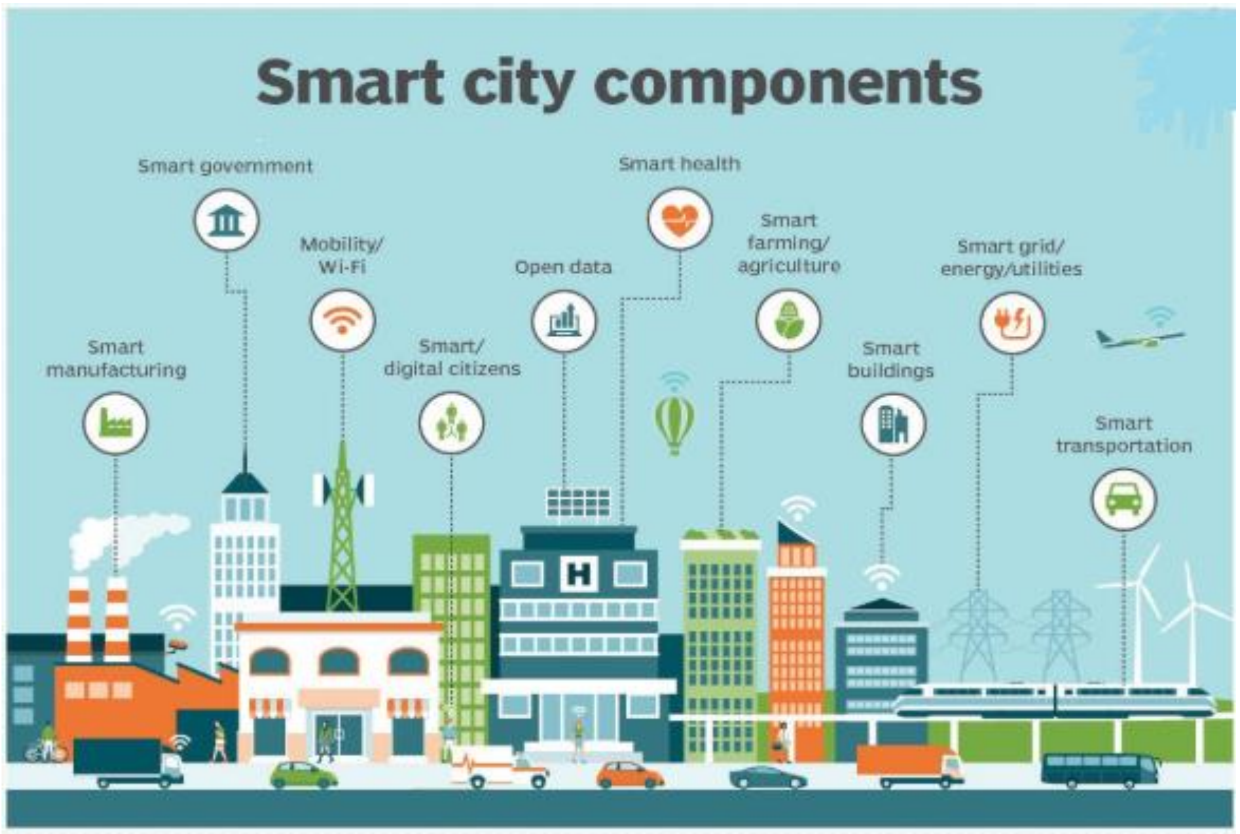


Figure 5:*The components of a smart city are diverse and encompass the government, private sector and citizens.* **https://www.techtarget.com/iotagenda/definition/smart-city**

2.5.4 Healthcare IoT (IoMT – Internet of Medical Things)

Healthcare IoT, often referred to as the Internet of Medical Things (IoMT), represents the integration of IoT technologies into medical and healthcare systems. It encompasses a wide range of connected medical devices, health monitoring systems, and supporting software applications designed to improve patient outcomes, enhance operational efficiency, and enable real-time healthcare delivery. IoMT devices include wearable fitness trackers, remote patient monitoring (RPM) systems, smart implants, and connected diagnostic tools. These devices continuously collect health-related data such as heart rate, blood pressure, glucose levels, and oxygen saturation and transmit it to healthcare providers for remote analysis and intervention. This real-time monitoring supports chronic disease management, early diagnosis, and personalized treatment. Hospitals and clinics benefit from IoT through asset tracking (e.g., for wheelchairs and medical equipment), environmental monitoring (temperature, humidity in storage rooms), and intelligent scheduling systems that optimize patient flow and resource usage. Smart medication dispensers and connected drug delivery systems also ensure accurate dosage and adherence to treatment plans.

One of the most transformative aspects of Healthcare IoT is telemedicine, which allows doctors to consult, diagnose, and treat patients remotely, reducing the need for physical visits and expanding access to care in underserved regions. The collected health data is often stored and analyzed in the cloud, enabling predictive analytics and AI-driven decision support systems.

https://mobisoftinfotech.com/resources/blog/wearable-technology-in-healthcare

Figure 06: Image de wearable devices in healthcare

Chapter 03: Implementation of IoT application for strain and force monitoring

In the world of the Internet of Things, collecting and managing data in real time is crucial to monitoring physical systems or environments. For example, measuring the force exerted on a surface can be useful in many applications, such as monitoring pressure on structures, moving objects, or automated systems. In this project, we will use an FSR (Force Sensitive Resistor) force sensor and an ESP32 or ESP8266 microcontroller to monitor these forces in real-time and display the results via a web interface. This system is simple to set up and offers great flexibility for future improvements.

In this chapter we will discuss the steps of implementing a monitoring system, from reading the sensor to displaying the data in a web browser. The core of the system is a web server that collects and shares sensor data asynchronously, providing a smooth, real-time experience.

This system involves three main steps:

1. Data reading: The FSR sensor is used to measure the applied force.
2. Data processing: Analog data is converted into interpretable digital values (force and voltage).
3. Data Display: The data is sent to the web server and displayed on an HTML page.

3.1. System Overview

The system we proposed is based on the use of an FSR force sensor, which measures the pressure or force applied to an object. This sensor works by changing its resistance depending on the force applied, which allows this variation to be converted into an analog voltage. This voltage is then read by a microcontroller (ESP32 or ESP8266), which transforms it into digital data that can be easily exploited.

The ESP32 or ESP8266, thanks to its built-in Wi-Fi capability, will be configured to host a web server. This server will receive requests from users via a browser and return information about the measured strength. The advantage of this system is its ability to provide a remotely accessible interface, which makes it possible to monitor the forces applied to an object without the need for a physical connection.

In summary, the project involves three main stages:

1. Data reading: The FSR sensor is used to measure the applied force.
2. Data processing: Analog data is converted into interpretable digital values (force and voltage).
3. Data Display: The data is sent to the web server and displayed on an HTML page.

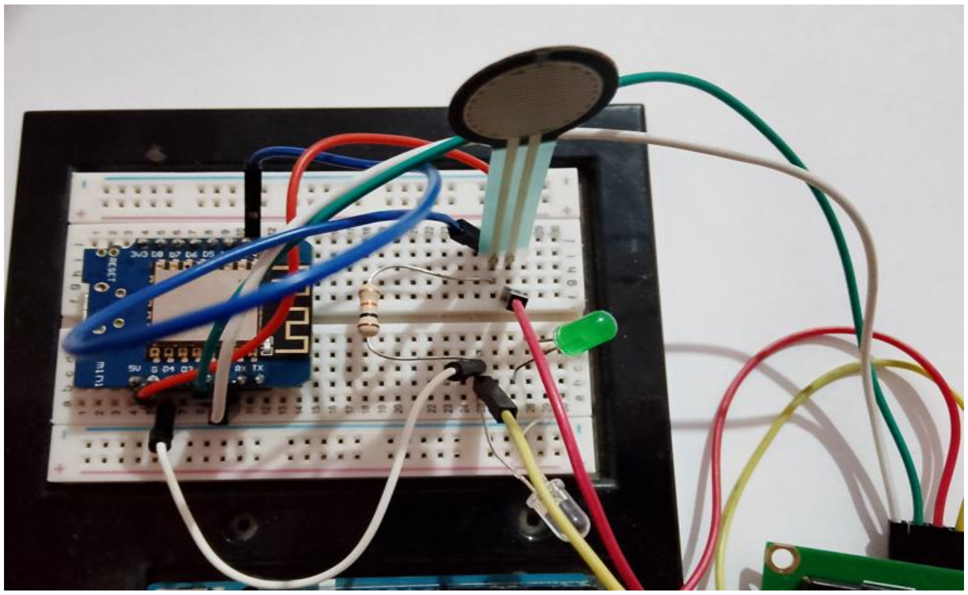
3. 2. Necessary materials

Before we dive into the code, it's important to know what components will be needed to complete this project. Here is a summary table of the components needed to complete the project:

Component	Description
ESP32 or ESP8266	Microcontroller with built-in Wi-Fi connectivity, used to read the sensor and host the web server.
FSR sensor	Pressure-sensitive sensor used to measure the force applied.
Resistance	Used to stabilize readings (optional depending on the mount).
Connection Cables	Allows components to be connected to each other.
Feeding	Power source (USB or battery) to power the ESP and sensor.

3. 2.1 Installation, operating principle

The **Force Sensitive Resistor (FSR)** sensor is an electronic component whose resistance decreases when a force is applied to its surface. This property makes it possible to convert a mechanical force into a measurable electrical variation. When subjected to pressure, the resistance of the FSR increases from several megaohms (without pressure) to a few hundred ohms (under high pressure). To exploit this property, the FSR is integrated into the **assembly in the figure below**. The esp8266 microcontroller reads this voltage through an analog input (ADC), converts it to a digital value, and then applies a **calibration** based on known measurements to infer the actual force.



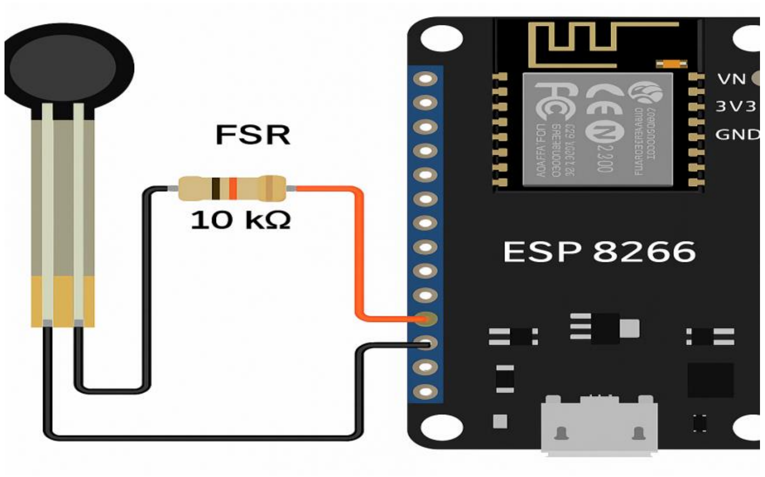


Figure 1: Circuit of test and measure de force (Strain)

3.2.2 Web-User Interface

The project includes a **web interface integrated** directly into the ESP8266. This interface allows you to consult the value of the force detected by the FSR sensor in real time, via a simple HTML page. The interface is composed of:

- An HTML page stored in the ESP's flash memory
- A dynamic display area of the force value
- A JavaScript script that regularly queries the ESP via HTTP (or WebSocket)

Example of the embedded HTML code in esp8266

```
<! DOCTYPE html>
<html>
<head>
<metacharset="UTF-8">
<title>Surveillance de Force FSR</title>
<script>
   setInterval(() => {
     fetch("/valeur").then(response => response.text()).then(data => {
       document.getElementById("fsr").innerHTML = data + " N";
     });
   }, 500);
</script>
</head>
<body>
<h2>Measured Force Value</h2>
<><strong><span id="fsr">0</span></strong></p>
</body>
</html>
```

- The server returns the HTML page to the URL /
- Requests to /value return the last analog measurement converted to Newton
- The use of fetch()allows a smooth update every 500 ms

In order to improve the interface, the following improvements are suggested

- ✓ Integration of **dynamic graphics** (e.g. with Chart.js)
- ✓ Choice of display unit (N, kg, g...)
- ✓ Secure access (simple authentication)
- ✓ WebSocket for faster, lighterrefresh

3.3. Main Program

The program used in this project is written in C++ for the Arduino IDE, making it simple to understand and modify. Let's take a look at how the code works step-by-step.

a) The necessary libraries

The program begins with the inclusion of libraries that allow the microcontroller to connect to the Wi-Fi network and act as a web server. Depending on whether you are using an ESP32 or a ESP8266, the libraries are chosen automatically.

Code:

```
#ifdef ESP32 #include <WiFi.h> #include <ESPAsyncWebServer.h>#else #include <Arduino.h>
#include <ESP8266WiFi.h> #include <ESPAsyncTCP.h> #include
<ESPAsyncWebServer.h>#endif
```

b) Definition of important parameters

Next, some constants are defined that will allow the system to be easily configured, such as the pin to which the sensor is connected, the supply voltage, and the range of values for the analog-to-digital converter (ADC).

Code:

```
#define FSR_PIN A0#define VCC 3.3#define MAX_ADC 1023.0
```

c) Connecting to the Wi-Fi network and starting the web server

Once the hardware is configured, the program attempts to connect to the Internet using the Wi-Fi network information. When the connection is successful, the IP address of the ESP is displayed, allowing the user to access the server through a browser.

```
WiFi.begin(ssid, password); while (WiFi.status() != WL_CONNECTED) {    delay(1000);
Serial.println("Connecting to WiFi...");}Serial.print("ESP IP Address: ");
Serial.println(WiFi.localIP());

server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){    request->send_P(200,
"text/html", index_html);});
server.on("/data", HTTP_GET, [](AsyncWebServerRequest *request){    request->send(200,
"application/json", getSensorData());});
server.begin();
```

d) Sensor Data Acquisition

The getSensorData() function is responsible for collecting data from the FSR sensor. It reads the raw value from the sensor, converts it into a voltage, and from that voltage calculates the force exerted.

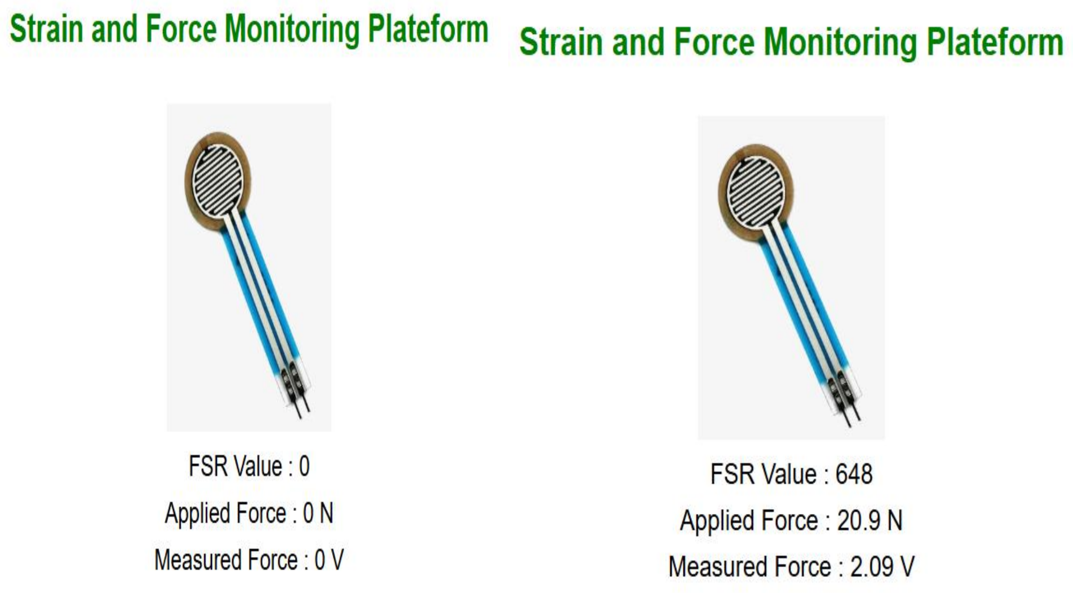
```
String getSensorData() {     int fsrValue = analogRead(FSR_PIN);     float voltage = (fsrValue /
MAX_ADC)       *       VCC;                          float       force       =       voltage       *       10;

   String  data  =  "{";       data  +=  ""fsr":"  +  String(fsrValue)  +  ",";       data  +=  ""force":"  +
String(force, 2) + ",";    data += ""voltage":" + String(voltage, 2);    data += "}";    return data;}
```

e) Displaying data in the browser

The web interface is designed to display sensor data in real-time. The HTML code includes a small JavaScript script that periodically queries the server to retrieve the data and displays it on the page.

Code:

```
<script>     setInterval(() => {          fetch('/data')              .then(response => response.json())
.then(data  =>  {                          document.getElementById('fsrValue').innerText  =  data.fsr;
document.getElementById('forceValue').innerText                        =                        data.force;
document.getElementById('voltageValue').innerText = data.voltage;          })          .catch(error
=>     console.error('Error     fetching     data:',     error));                  },     1000);     </script>
```



Strain and Force Monitoring Plateform

FSR Value : 0

Applied Force : 0 N

Measured Force : 0 V

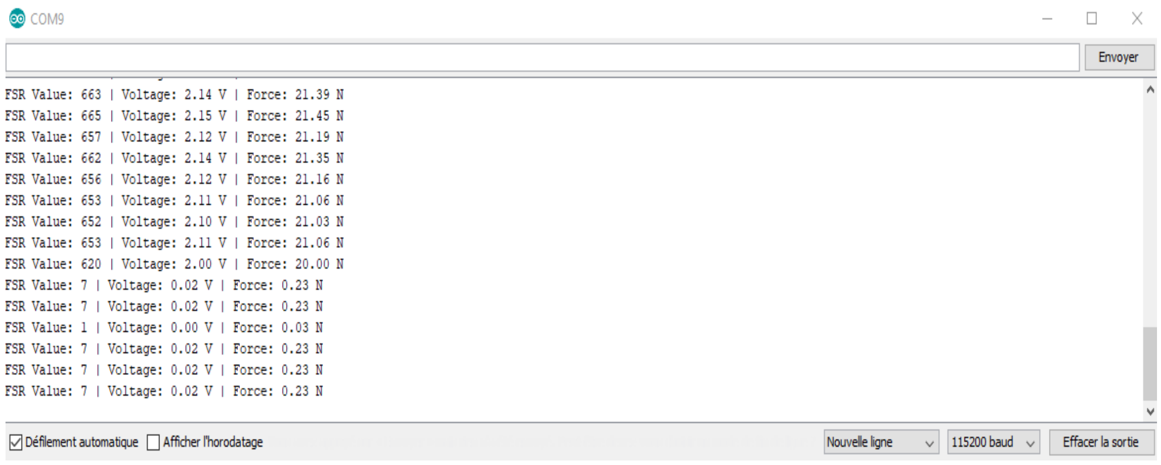
Strain and Force Monitoring Plateform

FSR Value : 648

Applied Force : 20.9 N

Measured Force : 2.09 V

Figure 2: Web interface for force and strain measurement: a) before applying strain and b) after applying strain



```
FSR Value: 663 | Voltage: 2.14 V | Force: 21.39 N
FSR Value: 665 | Voltage: 2.15 V | Force: 21.45 N
FSR Value: 657 | Voltage: 2.12 V | Force: 21.19 N
FSR Value: 662 | Voltage: 2.14 V | Force: 21.35 N
FSR Value: 656 | Voltage: 2.12 V | Force: 21.16 N
FSR Value: 653 | Voltage: 2.11 V | Force: 21.06 N
FSR Value: 652 | Voltage: 2.10 V | Force: 21.03 N
FSR Value: 653 | Voltage: 2.11 V | Force: 21.06 N
FSR Value: 620 | Voltage: 2.00 V | Force: 20.00 N
FSR Value: 7 | Voltage: 0.02 V | Force: 0.23 N
FSR Value: 7 | Voltage: 0.02 V | Force: 0.23 N
FSR Value: 1 | Voltage: 0.00 V | Force: 0.03 N
FSR Value: 7 | Voltage: 0.02 V | Force: 0.23 N
FSR Value: 7 | Voltage: 0.02 V | Force: 0.23 N
FSR Value: 7 | Voltage: 0.02 V | Force: 0.23 N
```

Figure 3: Display of measurement data on the serial monitor

3.4. How the system works?

The microcontroller operates continuously. As soon as it is powered, it connects to Wi-Fi and initializes the web server. Once connected, it starts responding to incoming requests and updating sensor data every second. This data is sent to the browser as JSON, where it is used to update the display in real-time.

The loop() function also allows real-time monitoring of sensor values on the serial monitor, providing an easy way to check that everything is working correctly.

```
void loop() {     Serial.print("FSR Value: ");     Serial.print(analogRead(FSR_PIN));
Serial.print(" | Voltage: ");     Serial.print((analogRead(FSR_PIN) / MAX_ADC) * VCC);
Serial.print(" V | Force: ");     Serial.print((analogRead(FSR_PIN) / MAX_ADC) * VCC * 10);
Serial.println(" N");

delay(500);}
```

3.5. Possible applications and extensions

This project is just a foundation, and it can be expanded in many ways. For example, multiple FSR sensors can be added to monitor different areas. In addition, the data can be sent to a remote database or mobile app for further processing. The system can also be enhanced to incorporate alerts, such as notifications in case of excessive force or rapid variation. The project is also a great starting point for more complex IoT applications, such as monitoring industrial machines or managing interactive devices.

3.6. Conclusion

This project demonstrates how to create a simple yet effective monitoring platform that allows the force applied to a sensor to be measured and the data to be displayed in real time via a web server. It highlights the use of ESP32 and ESP8266 microcontrollers for the creation of affordable and high-performance IoT systems. Through this project, you not only learned how to interface a load cell with a microcontroller, but also how to integrate a web server to share this data remotely. This type of system is ideal for applications where remote and real-time monitoring is essential.

General Conclusion

Chapter 1 lays the groundwork by explaining the fundamental principles of sensors, particularly force sensors, their classifications, and how they function. It emphasizes the crucial role of sensors in converting physical inputs into measurable electrical signals, which is essential for IoT applications.

Chapter 2 builds upon this by introducing the concept of the Internet of Things (IoT), detailing its architecture, key characteristics, and the communication protocols that enable data exchange between devices. It highlights the potential of IoT to revolutionize mechanical stress measurement through remote monitoring, real-time data acquisition, and enhanced accuracy.

Chapter 3 practically applies this knowledge by guiding the reader through the implementation of a constraint monitoring and control system. It demonstrates how to use a force-sensitive resistor (FSR) with a microcontroller to measure force and transmit the data to a web interface for real-time monitoring.

In essence, the chapters progress from the theoretical understanding of sensors and IoT to the practical application of building a system for measuring and monitoring mechanical stress. They illustrate the power of IoT in enabling remote and real-time monitoring of physical phenomena, with potential applications in various industries.

References

[1]. Fraden, J. (2016). Handbook of Modern Sensors: Physics, Designs, and Applications (5th ed.). Springer. https://doi.org/10.1007/978-3-319-19303-8

[2]. Webster, J. G. (Ed.). (2010). Measurement, Instrumentation, and SensorsHandbook. CRC Press.

[3]. Beckwith, T. G., Marangoni, R. D., & Lienhard, J. H. (2007). MechanicalMeasurements (6th ed.). Pearson.

[4]. Dally, J. W., & Riley, W. F. (1993). Experimental Stress Analysis (4th ed.). McGraw-Hill.

[5]. Ahmad, S., Akhtar, N., Ahmed, A., &Iftikhar, S. (2025). An ensemble approach to improveenergyefficiency in smart agriculture. IEEE Transactions on IndustrialInformatics. https://doi.org/10.1109/TII.2025.123456

[6]. Chauhan, N., &Subeesh, A. (2025). Deep learning-basedabiotic stress assessment in precision agriculture. Journal of Environmental Management, 323. https://doi.org/10.1016/j.jenvman.2025.117345

[7]. Gerguis, J. (2025). Photonicintegrated circuits for sensing applications. PurdueUniversityThesis Repository. https://hammer.purdue.edu/articles/thesis/28755164

[8]. Herrero, R., &Dasari, M. (2025). Edge-based cognitive digital twinsusing IoT frameworks. Internet of Things, Elsevier. https://doi.org/10.1016/j.iot.2025.100923

[9]. Jesus, B., Lins, F., &Laranjeiro, N. (2025). An approach to assessrobustness of MQTT-based IoT systems. Internet of Things, Elsevier. https://doi.org/10.1016/j.iot.2025.100891

[10]. Nankali, M., Soleimani, M., Enrique, P., & Peng, P. (2025). Direct laser synthesis of graphene for flexible temperaturesensors. Materials Today Nano, Elsevier. https://doi.org/10.1016/j.mtnano.2025.100321

[11]. Samplawski, C. (2025). Uncertainty-aware computer vision in resource-constrainedenvironments. University of Massachusetts Amherst. https://scholarworks.umass.edu/publication/a1be3b88

[12]. Sethuraman, R., Kavitha, D., & Pranav, K. R. N. (2025). IT2FLS-RSA: A novel QoS-drivenrouting and securityapproach. International Journal of Intelligent Systems Technologies and Applications. https://doi.org/10.1007/s40815-025-01980-8

[13]. Shah, K., Jadav, N. K., Gupta, R., Gupta, S., &Tanwar, S. (2025). A deeplearning-orchestratedgarlicrouting architecture for securetelesurgeryoperations in healthcare 4.0. EgyptianInformatics Journal. https://doi.org/10.1016/j.eij.2025.03.007

[14]. Wibowo, A., Amri, I., &Surahmat, A. (2025). Leveraging AI in disaster management using IoT. Jàmbá: Journal of Disaster Risk Studies, 17(1). https://jamba.org.za/index.php/jamba/article/view/1776

[Source1:] ITU-T Recommendation Y.2060 (06/2012), Overview of the Internet of Things]