Draft Study Material

विद्यया ऽ मृतमश्नुते

एन सी ई आर टी
NCERT

(Job Role)

Automotive IIOT Application Technician

(Qualification Pack: ASC/Q6413)

Sector: Automotive

Grade XII

PSS CENTRAL INSTITUTE OF VOCATIONAL EDUCATION

(a constituent unit of NCERT, under Ministry of Education, Government of India)

Shyamla Hills, Bhopal - 462 002, M.P., India

http://www.psscive.ac.in

Automotive IIOT Application Technician

(Job Role)

QUALIFICATION PACK - ASC/Q6413

SECTOR - AUTOMOTIVE

Draft Study Material for Grade XII

राष्ट्रीय शैक्षिक अनुसंधान और प्रशिक्षण परिषद
National Council of Educational Research and Training

PREFACE

Vocational Education is a dynamic and evolving fields and ensuring that every student has access to quality learning materials is of paramount importance. The journey of the PSS Central Institute of Vocational Education toward producing comprehensive and inclusive study material is rigorous and time-consuming, requiring thorough research, expert consultation, and publication by the National Council of Educational Research and Training (NCERT). However, the absence of finalised study material should not impede the educational progress of our students. In response to this necessity, we present the draft study material, a provisional yet comprehensive guide, designed to bridge the gap between teaching and learning, until the official version of the study material is made available by the NCERT. The draft study material provides a structured and accessible set of materials for teachers and students to utilise in the interim period. The content is aligned with the prescribed curriculum to ensure that students remain on track with their learning objectives.

The contents of the modules are curated to provide continuity in education and maintain the momentum of teaching-learning in vocational education. It encompasses essential concepts and skills aligned with the curriculum and educational standards. We extend our gratitude to the academicians, vocational educators, subject matter experts, industry experts, academic consultants, and all other people who contributed their expertise and insights to the creation of the draft study material.

Teachers are encouraged to use the draft modules of the study material as a guide and supplement their teaching with additional resources and activities that cater to their students' unique learning styles and needs. Collaboration and feedback are vital, therefore, we welcome suggestions for improvement, especially by the teachers, in improving upon the content of the study material.

Deepak Paliwal
(Joint Director)
PSSCIVE, Bhopal

Date:

STUDY MATERIAL DEVELOPMENT COMMITTEE

MEMBERS

- Dr. Rita Jain, *Founder & Director, AVRN Intellitech Pvt.Ltd.,* Bhopal, Madhya Pradesh, India.
- Dr. Kavita Kamerikar, *Consultant (ECE)* (Contractual), Department of Engineering and Technology (DET), PSS Central Institute of Vocational Education (PSSCIVE), NCERT, Shyamla Hills, Bhopal, Madhya Pradesh, India.
- Mr. Pulak Rijhwani, *Founder & CEO, Opsight AI Pvt. Ltd.*, Noida, Uttar Pradesh, India.
- Mr. Arun Kumar, *Founder & CTO, Opsight AI Pvt. Ltd.,* Noida, Uttar Pradesh, India.
- Mr. Ramkaran Sherawat, *Team Lead, Opsight AI Pvt. Ltd.*, Noida, Uttar Pradesh, India.
- Mr. Rajiv Tiwari, *Embedded Engineer, Opsight AI Pvt. Ltd.*, Noida, Uttar Pradesh, India.

MEMBER COORDINATOR

Dr. Vinod Kumar Yadav, Associate Professor, Department of Engineering and Technology, PSS Central Institute of Vocational Education, Bhopal, Shyamla Hills, Madhya Pradesh, India.

CONTENTS

MODULE 1 AUTOMOTIVE IIOT APPLICATIONS

Module Overview

This module introduces the key applications of the IIOT in the automotive sector, focusing on how connected technologies improve manufacturing, monitoring, and transportation systems.

Students will learn how IIOT integrates sensors, controllers, and communication networks to enable smart manufacturing, real-time monitoring, and automated material handling in automotive industries. They will also explore how IIOT supports smart transportation, including connected vehicles, traffic control, and intelligent mobility solutions.

By the end of this module, learners will understand how IIOT technologies contribute to efficiency, sustainability, and innovation in the automotive ecosystem, preparing them for emerging careers in connected and intelligent vehicle systems.

Learning Outcomes

After completing this module, you will be able to:

- Understand the basic concept and importance of IIOT in the automotive industry.
- Explain how IIOT helps in monitoring and controlling automotive systems.
- Describe the use of IIOT in material handling and automation in industries.
- Identify IIOT applications in smart and connected transportation systems.
- Recognize how IIOT improves safety, efficiency, and sustainability in vehicles and manufacturing.

Module Structure

Session 1: Introduction to Automotive IIOT
Session 2: Role of IIOT in Monitoring and Material Handling in Industries
Session 3: Smart Transportation and Predictive Maintenance

SESSION 1: INTRODUCTION TO AUTOMOTIVE IIOT

1.1 Importance of Automotive IIOT

The automotive industry has changed rapidly with the growth of digital technologies and automation. Earlier, most manufacturing work in automobile industries was done manually or with standalone machines. Today, industries are using smart technologies to improve production, safety, quality, and efficiency. One of the most important technologies used in modern industries is the Industrial Internet of Things (IIOT).

This transformation is made possible by the IIOT; a network of machines, sensors, and software connected through the internet. IIOT bridges the gap between the physical and digital worlds. It brings intelligence into every stage of automobile manufacturing, from design and production to maintenance and transportation.

1.1.1 Evolution of Automotive IIOT

Earlier, automobile plants relied on manual data collection. Operators physically inspected machines, noted temperatures, vibrations, and performance in logbooks, and reported issues to supervisors. This method was slow, error-prone, and reactive, i.e. problems were addressed only after failures occurred.

Today, with the power of IIOT, machines talk to each other. They continuously send real-time data about their condition and performance to a central monitoring system. Engineers and managers can see this information instantly on computers, tablets, or even mobile phones.

In this way, IIOT has turned factories into intelligent ecosystems where decisions are data-driven, and problems are detected before they cause losses.

1.1.2 How IIOT Works in the Automotive Industry

To understand how IIOT functions, imagine a car factory as a "living system." Every machine, robot, or assembly line act as an organ, and sensors act like nerves that sense what is happening.

Sensors and Controllers: Sensors are like the eyes and ears of the system. They measure key parameters such as temperature, pressure, vibration, speed, torque, humidity, and power consumption. Controllers (like PLCs or microcontrollers) collect this data and prepare it for transmission.

Connectivity: Data travels through communication networks such as Ethernet, Wi-Fi, Bluetooth, or industrial IoT protocols like MQTT, OPC-UA, or Modbus. This ensures that all devices are connected to a central system or cloud platform in real time.

Data Analysis: The data received is processed using smart software, analytics tools, or cloud-based AI systems. These platforms convert raw numbers into useful insights; for example, predicting when a machine might need maintenance or identifying which assembly line is running below efficiency.

Action and Control: Based on the analyzed data, automated systems can make real-time decisions such as slowing down a motor, adjusting temperature, or alerting an operator through a dashboard or SMS notification. Some systems even perform self-correction without human intervention.

Example: Consider a car manufacturing plant that produces 500 vehicles daily. The assembly line includes hundreds of robotic arms, welding stations, and paint booths. Each robot is equipped with vibration and temperature sensors. The data is sent to a central cloud system every second. When one welding robot shows abnormal vibration, the system automatically alerts the maintenance engineer. The engineer takes quick action, preventing a breakdown that could have stopped production for hours. This small intervention saves thousands of rupees in downtime and improves the reliability of the production line.

1.1.3 Benefits of IIOT in the Automotive Industry

The IIOT has brought major improvements in the automotive industry. IIOT helps industries improve productivity, safety, product quality, maintenance, and energy management. Smart sensors and connected systems continuously collect and analyze data, enabling industries to make faster and better decisions.

1. Higher Production Efficiency: Machines connected through IIOT continuously share their operational status and performance data. This helps industries quickly identify machine downtime, slow processes, and production bottlenecks. As a result, production managers can optimize workflow and improve overall efficiency.

Example: If a conveyor belt motor slows down due to excessive load, the IIOT system can automatically send an alert or adjust the operating conditions to avoid production delays.

2. Improved Safety: Industrial environments often involve high temperatures, heavy machinery, and hazardous materials. IIOT sensors help detect unsafe conditions such as overheating, gas leakage, or abnormal vibration at an early stage. Early detection helps prevent accidents and improves worker safety.

Example: In an automotive painting section, air quality sensors monitor harmful gas levels. If unsafe conditions are detected, the system automatically activates alarms and safety controls.

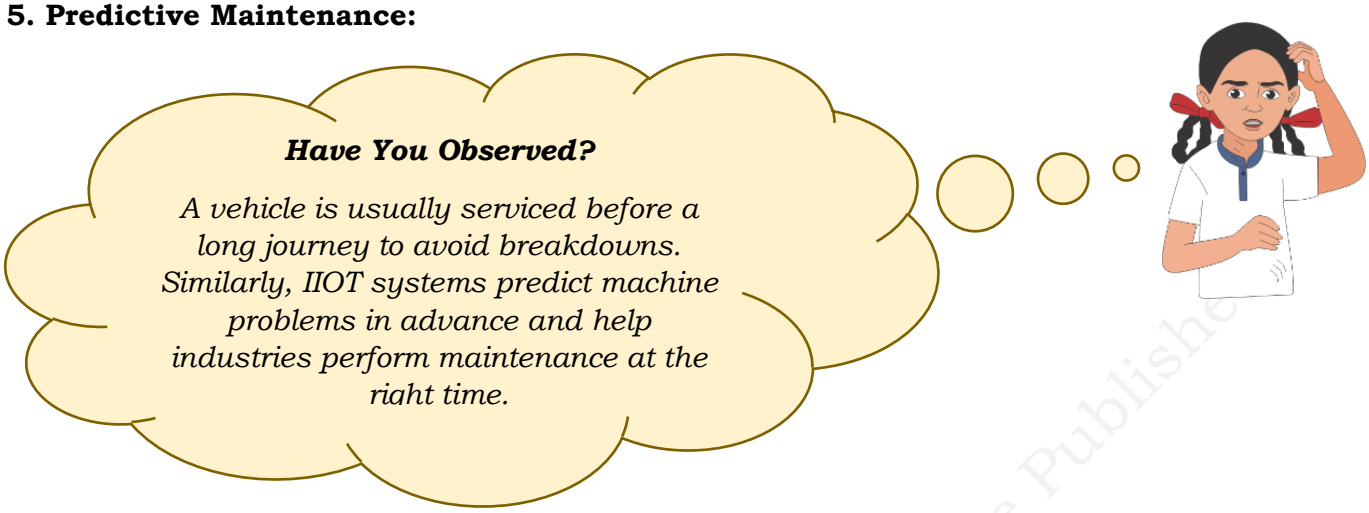
3. Better Product Quality: Automobile manufacturing requires high precision and accuracy. IIOT systems help monitor machine parameters such as temperature, pressure, torque, and alignment during manufacturing operations. This ensures consistent product quality and reduces manufacturing defects.

Example: In robotic welding systems, sensors continuously monitor welding temperature and alignment. If the welding conditions deviate from standard values, the system immediately alerts the operator.

4. **Energy Savings:** Large automobile industries consume significant amounts of electricity and fuel. IIOT helps monitor energy consumption and identify areas where energy is being wasted. Smart systems can automatically control lighting, ventilation, and machine operations to reduce energy usage.

Example: HVAC and lighting systems can automatically adjust according to room occupancy and environmental conditions, reducing unnecessary power consumption.

5. Predictive Maintenance:



Predictive maintenance is one of the most important advantages of IIOT. Instead of waiting for machines to fail, IIOT systems continuously monitor machine conditions and predict possible failures in advance. This helps reduce unexpected breakdowns and maintenance costs.

Example: If abnormal vibration or temperature is detected in a gearbox, the system predicts a possible bearing failure and informs the maintenance team before the machine stops working.

1.1.4 Smart Manufacturing in Automobile Industries

Smart manufacturing refers to the use of advanced technologies such as automation, robotics, artificial intelligence (AI), and IIOT in industrial production systems. Modern automobile industries use robotic arms, automated conveyor systems, and smart sensors to improve manufacturing operations. Smart manufacturing helps industries to:

- Increase production speed
- Improve accuracy and quality
- Reduce human errors
- Improve worker safety
- Optimize resource utilization

1.1.5 Connected Industrial Ecosystems

A connected industrial ecosystem is a network in which machines, devices, software systems, and people communicate and share information through industrial networks and cloud platforms.

In automotive industries, manufacturing units, warehouses, supply chains, maintenance departments, and monitoring systems are interconnected. This connectivity enables industries to monitor operations remotely and improve coordination between different departments.

- This connectivity helps industries:
- Monitor industrial operations remotely
- Improve communication and coordination
- Manage inventory efficiently
- Reduce operational delays
- Improve productivity and decision-making

1.1.6 Real-World Perspective

Leading automotive companies are demonstrating how IIOT is transforming the industry into a data-driven, intelligent ecosystem focused on efficiency, safety, sustainability, and innovation.

Real-World Example

Tesla connects its entire production line and vehicles through the cloud. Each car continuously sends performance data to Tesla's servers, enabling engineers to remotely diagnose issues and enhance vehicle performance.

Toyota utilizes IIOT-enabled robots for precision welding, painting, and assembly. Its "Smart Factory" system ensures high accuracy, minimal defects, and maximum productivity.

Bosch offers IIOT solutions like the Bosch IoT Suite to connect machines, monitor their condition, and manage maintenance schedules in real time.

1.2 Everyday Applications of IIOT in Automobiles

Industrial Internet of Things (IIOT) is becoming an important part of modern automobiles and transportation systems. Today's vehicles are equipped with smart sensors, communication systems, and monitoring devices that help improve safety, efficiency, and vehicle performance. IIOT allows vehicles and automotive systems to collect, analyze, and share data in real time.

Some common applications of IIOT in automobiles include fuel monitoring, electric vehicle battery monitoring, and real-time vehicle diagnostics.

A) Fuel Monitoring Using IoT Sensors

Fuel monitoring is one of the most useful applications of IIOT in automobiles. Smart fuel sensors are installed in vehicles to continuously measure the fuel level and fuel consumption. These sensors send data to monitoring systems through wireless communication networks.

Fuel monitoring systems help:

- Measure fuel consumption accurately
- Detect fuel leakage or theft
- Improve fuel efficiency
- Track vehicle performance
- Reduce operational costs

☞ *Around Us*

Many transport companies now track fuel usage and vehicle location in real time using smart monitoring systems.

For example, in transport vehicles, IoT fuel sensors send real-time fuel data to a central monitoring system. If fuel consumption suddenly increases, the system alerts the operator for inspection (Fig.1.1).

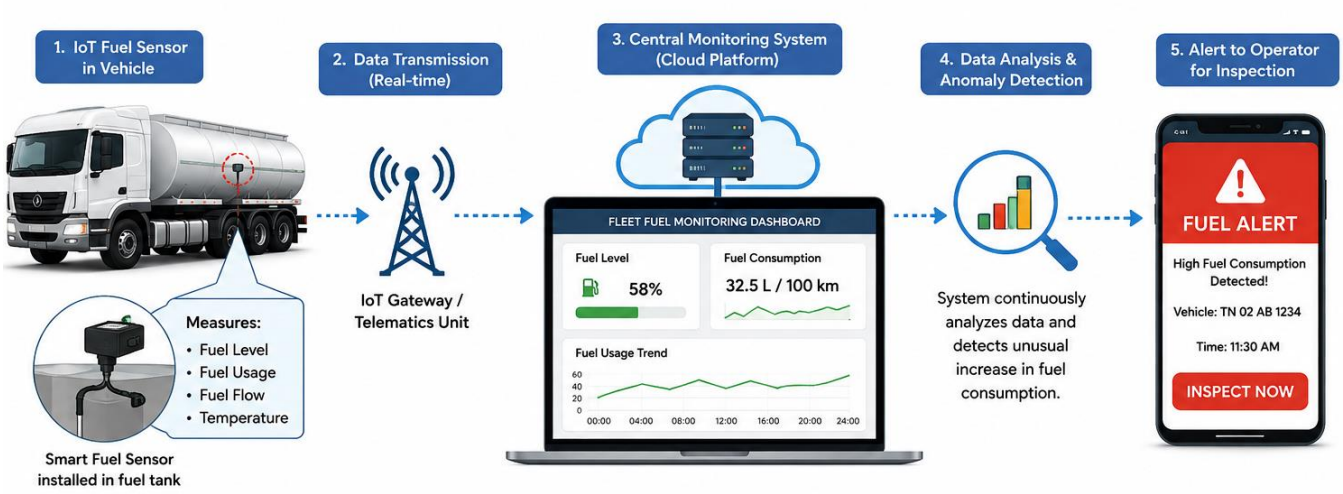


Fig.1.1: Fuel Monitoring Using IoT Sensors

B) Temperature Monitoring of EV Batteries

Electric vehicles (EVs) use rechargeable batteries as their power source. During charging and operation, EV batteries generate heat. Excessive temperature can reduce battery life and may create safety risks. Temperature sensors are used in EV battery systems to continuously monitor battery temperature. The collected data is analyzed in real time to maintain safe operating conditions.

Battery temperature monitoring helps:

- Prevent battery overheating
- Improve battery performance
- Increase battery life

- Ensure vehicle safety
- Improve charging efficiency

> ***Interesting Fact***
> *Electric vehicles use large battery packs that generate heat during charging and operation. Temperature sensors help maintain battery safety and improve battery life.*

For example, if the temperature of an EV battery becomes too high, the monitoring system automatically activates cooling systems or sends alerts to the driver (Fig.1.2).

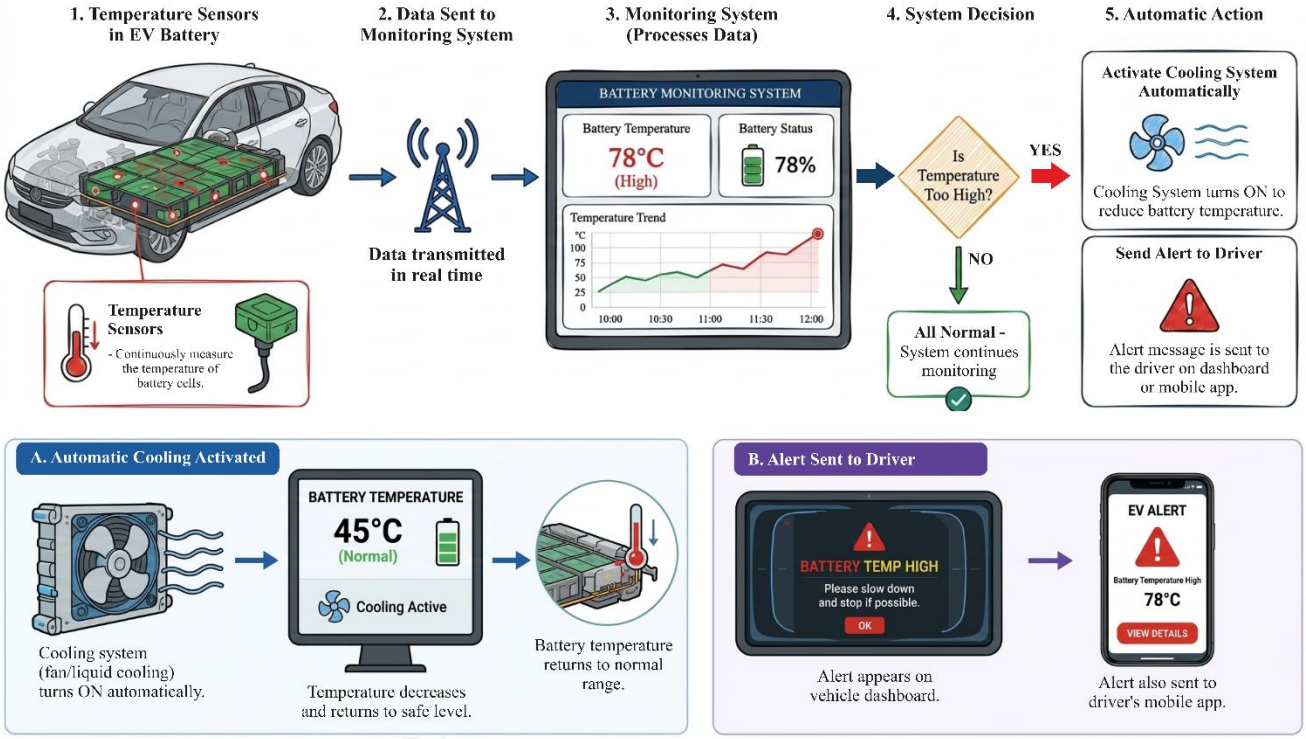


Fig.1.2: Temperature Monitoring of EV Batteries

C) Real-Time Vehicle Diagnostics and Performance Monitoring

Modern vehicles use multiple sensors to monitor engine condition, oil pressure, tire pressure, battery status, fuel efficiency, and engine temperature. IIOT systems collect and analyze this data in real time to detect abnormal conditions and improve maintenance efficiency.

Real-time diagnostics helps:

- Early fault detection
- Improve vehicle safety
- Reduce maintenance costs

- Improve driving performance
- Support predictive maintenance

For example, if a vehicle engine shows abnormal temperature or vibration, the system immediately displays a warning message and stores the fault information for maintenance purposes (Fig.1.3).

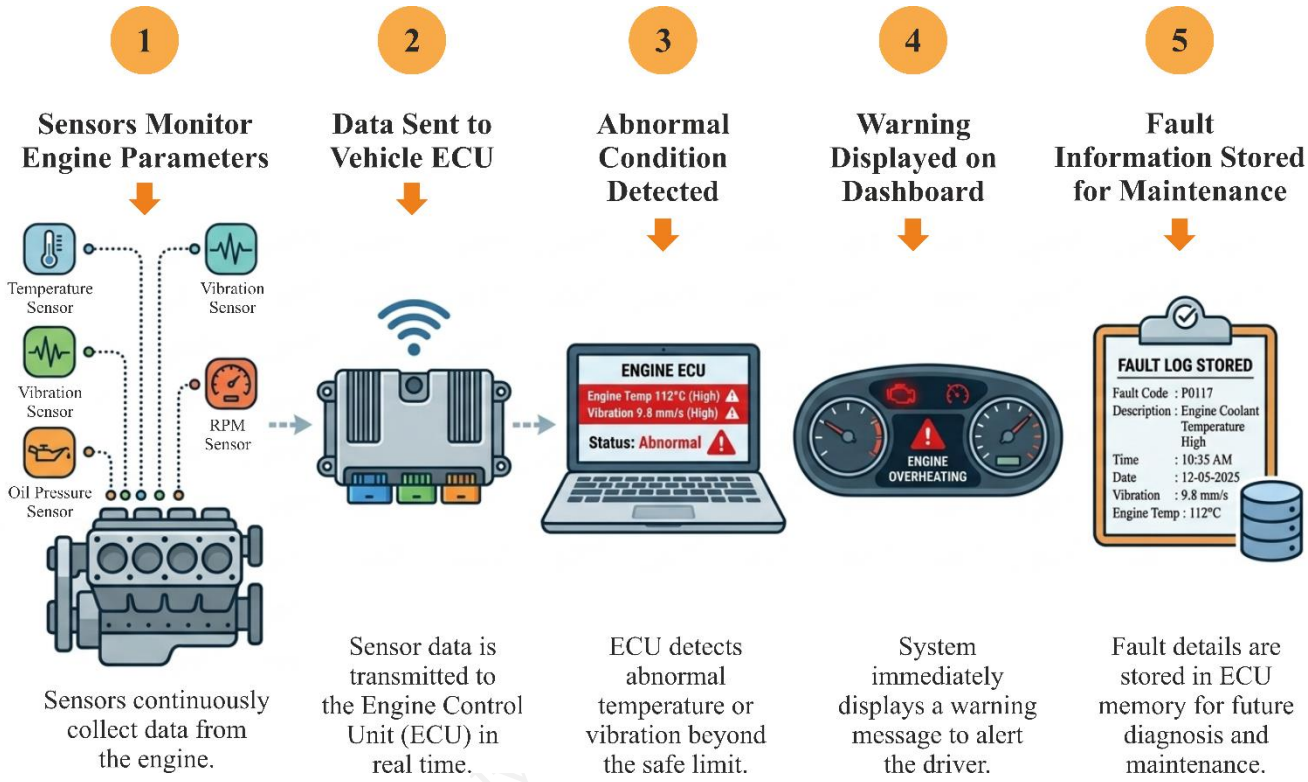


Fig.1.3: Real-Time Vehicle Diagnostics and Performance Monitoring

D) Remote Diagnostics and Updates

IIOT enables manufacturers and service centres to remotely monitor vehicle performance and diagnose problems without physically inspecting the vehicle. Modern vehicles can also receive software updates remotely through cloud-based systems.

Real-time diagnostics helps:

- Faster fault diagnosis
- Reduced service time
- Improved vehicle performance
- Better customer support

For example, some modern electric vehicles receive software updates remotely to improve battery performance and vehicle safety features.

1.3 Condition Monitoring in Automobiles Using IIOT Sensors

Have You Ever Wondered?

How does a modern vehicle detect engine problems before a breakdown occurs?
How does a car warn the driver about overheating or abnormal vibration

The answer lies in **Condition Monitoring using IIOT Sensors**.

Modern automobiles are equipped with smart sensors that continuously monitor the condition and performance of different vehicle components. These sensors collect real-time data related to vibration, temperature, pressure, speed, and engine condition. The collected data is analyzed to identify possible faults at an early stage and prevent unexpected failures.

Condition monitoring improves:

- Vehicle safety
- Machine reliability
- Maintenance efficiency
- Overall vehicle performance

1.3.1 Vibration Monitoring in Automobiles

Did You Notice?

Sometimes old vehicles make unusual sounds or vibrations while running. This may happen due to loosen parts, worn-out bearings, or engine problems. Modern vehicles use sensors to detect such issues early.

Vehicles contain many moving and rotating parts such as:

- Engines
- Motors
- Gearboxes
- Bearings
- Wheels and shafts

During operation, these parts naturally produce vibration. However, excessive or unusual vibration may indicate problems such as:

- Loose components
- Bearing wear
- Misalignment
- Mechanical imbalance

Vibration sensors continuously measure vibration levels and send the data to the monitoring system. If abnormal vibration is detected, the system generates an alert for inspection and maintenance.

Why is Vibration Monitoring Important?

- ❖ Early detection of faults
- ❖ Prevention of sudden breakdowns
- ❖ Improved machine performance
- ❖ Reduced maintenance cost
- ❖ Increased equipment life

Real-Life Example: *If the vibration level of a vehicle engine suddenly increases, the monitoring system immediately alerts the operator to inspect the engine before serious damage occurs.*

1.3.2 Engine and Machine Health Monitoring Using IIOT Sensors

Think About It!

Have you seen warning lights glowing on a vehicle dashboard? These warning indicators help drivers know about engine temperature, low oil level, or battery problems.

Modern automobiles use different types of sensors to continuously monitor engine and machine health.

Common Parameters Monitored

Parameter	Purpose
Engine Temperature	Detect overheating
Oil Pressure	Ensure proper lubrication
Fuel Consumption	Improve fuel efficiency
Engine Speed (RPM)	Monitor engine performance

Battery Condition	Ensure reliable power supply

Advantages of Engine Health Monitoring

- ❖ Continuous monitoring of engine condition
- ❖ Improved fuel efficiency
- ❖ Reduced engine failures
- ❖ Better vehicle performance
- ❖ Support for predictive maintenance

Real-Life Example: *If the engine temperature becomes too high, the system sends a warning message to the driver and stores the fault information for maintenance purposes.*

1.3.3 Sensor-Based Condition Monitoring and Fault Detection

Real-Life Connection

"When a mobile phone becomes too hot, it shows a warning message. In the same way, modern vehicles also use sensors and monitoring systems to detect abnormal conditions."

Sensor-based condition monitoring uses smart sensors and IIOT systems to detect faults before major failures occur.

The process works as follows:

Step 1: Data Collection: Sensors continuously collect data from vehicle systems.

Step 2: Data Transmission: The collected data is sent to the central monitoring system.

Step 3: Data Analysis: Software tools analyze the sensor data and compare it with normal operating conditions.

Step 4: Fault Detection: If abnormal conditions are detected, warning messages or alerts are generated automatically.

Common Faults Detected Using IIOT Sensors

- Overheating
- Excessive vibration
- Low oil pressure
- Battery failure

- Abnormal engine performance

Benefits of Fault Detection Systems

- ❖ Quick identification of problems
- ❖ Reduced downtime
- ❖ Improved vehicle safety
- ❖ Lower repair and maintenance costs
- ❖ Better operational efficiency

PRACTICAL ACTIVITY

ACTIVITY 1:
Smart Parking Assistance System Using Arduino Uno and Ultrasonic Sensor

Objective:

1. Understand how ultrasonic sensors measure distance
2. Interface an HC-SR04 ultrasonic sensor with Arduino
3. Calculate distance using echo time
4. Implement a basic parking assistance system
5. Use conditional logic to trigger output based on sensor readings

Components Required:

Component	Quantity
Arduino UNO	1
HC-SR04 Ultrasonic Distance Sensor	1
LED	1
Breadboard	1
Jumper Wires	6
USB Cable	1

Theory

Parking assistance systems help drivers avoid collisions while parking vehicles. These systems use distance sensors to detect nearby obstacles and warn the driver. In this activity, an HC-SR04 Ultrasonic Distance Sensor is interfaced with an Arduino Uno to measure the distance between the vehicle and an obstacle. If the object is too close (< 10 cm), the Arduino turns on an LED and displays a warning message on the Serial Monitor.

Working Principle:

1. The Arduino sends a trigger signal (10 µs HIGH pulse) to the Trig pin.
2. The sensor emits 8 ultrasonic pulses at 40 kHz from the transmitter.

3. These waves travel through the air and hit an obstacle.
4. The waves reflect back and are detected by the receiver.
5. The sensor measures the time taken for the echo to return via the Echo pin.
6. Arduino calculates distance using the formula below, then triggers the LED if < 10 cm.

Distance = (Time × Speed of Sound) / 2 Speed of Sound ≈ 0.034 cm/µs

Hardware Connection:

Ultrasonic Sensor Connection:

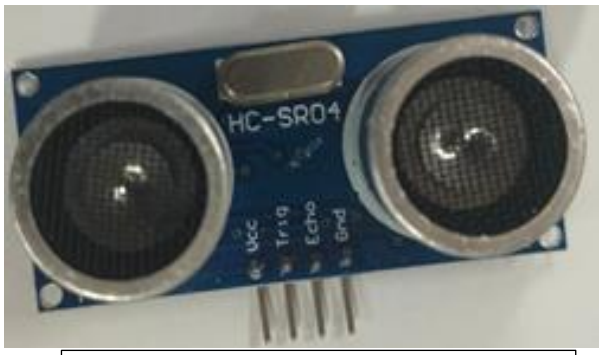


Fig.1: HC-SR04 Ultrasonic Sensor

Sensor Pin	Arduino Connection
V_{CC}	5V
GND	GND
Trig	Pin 9
Echo	Pin 10

- Transmitter — Sends ultrasonic waves
- Receiver — Receives reflected waves from objects

LED Connection:

LED Pin	Arduino Connection
Anode (+)	Pin 13
Cathode (−)	GND

Circuit Diagram:

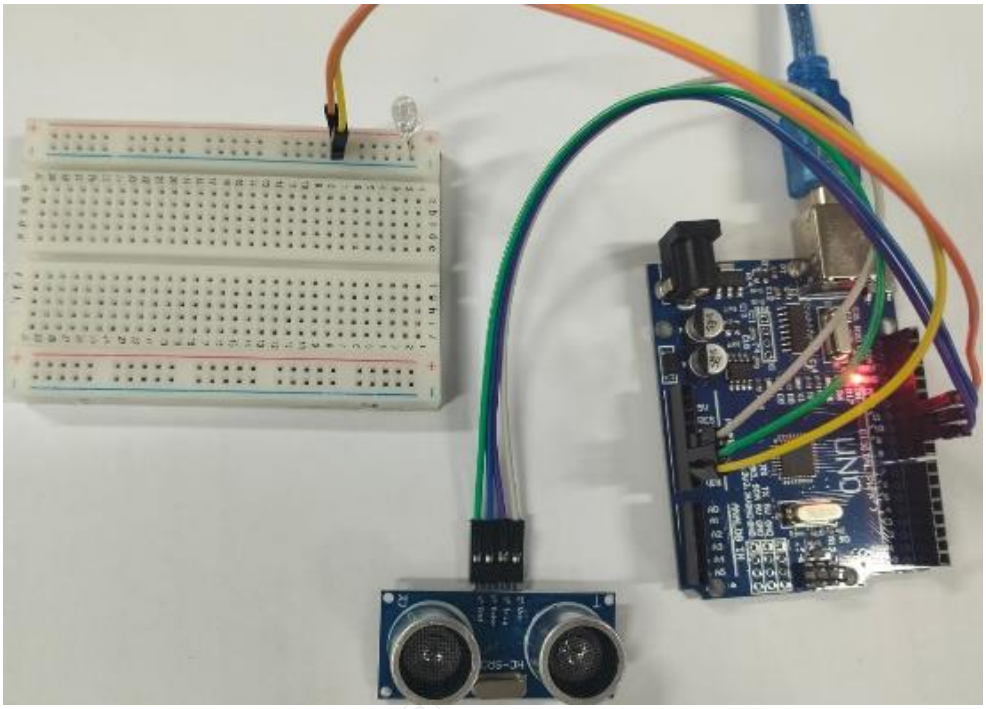


Fig.2: HC-SR04 to Arduino UNO

Procedure:

Step 1: Wire the Components

Connect the ultrasonic sensor and LED as per the Hardware Connection tables above.

Step 2: Connect Arduino to Computer

Plug in the USB cable and open Arduino IDE.

Step 3: Write the Program

Enter the following code:

```
// Arduino Code
const int trigPin = 9;
const int echoPin = 10;
const int ledPin = 13;
const int warningPin = 11;
void setup() {
  Serial.begin(19200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(warningPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  digitalWrite(warningPin, LOW);
```

```
}
void loop() {
  long duration;
  float distanceCm;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH, 30000);
  if (duration == 0) {
    Serial.println("No echo received");
    digitalWrite(ledPin, LOW);
    digitalWrite(warningPin, LOW);
  } else {
    distanceCm = duration * 0.0343 / 2.0;
    Serial.print("Distance: ");
    Serial.print(distanceCm);
    Serial.println(" cm");
    if (distanceCm < 10.0) {
      Serial.println("WARNING: Stop the vehicle to avoid collision or for safe
parking.");
      digitalWrite(ledPin, HIGH);
      digitalWrite(warningPin, HIGH);   // Pin 11 HIGH in warning zone
    } else {
      digitalWrite(ledPin, LOW);
      digitalWrite(warningPin, LOW);
    }
  }
  delay(500);
}
```

Step 4: Upload and Test

1. Select board Arduino UNO and correct COM port
2. Click Upload
3. Open Serial Monitor at baud rate 9600
4. Move an object towards the sensor. Observe the LED turn ON below 10 cm

Output:

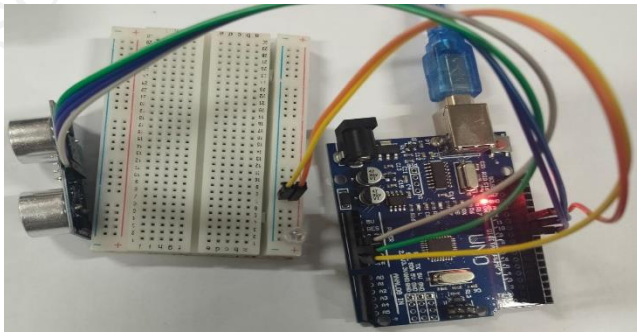


Fig.3: No Object Detection within 10 cm

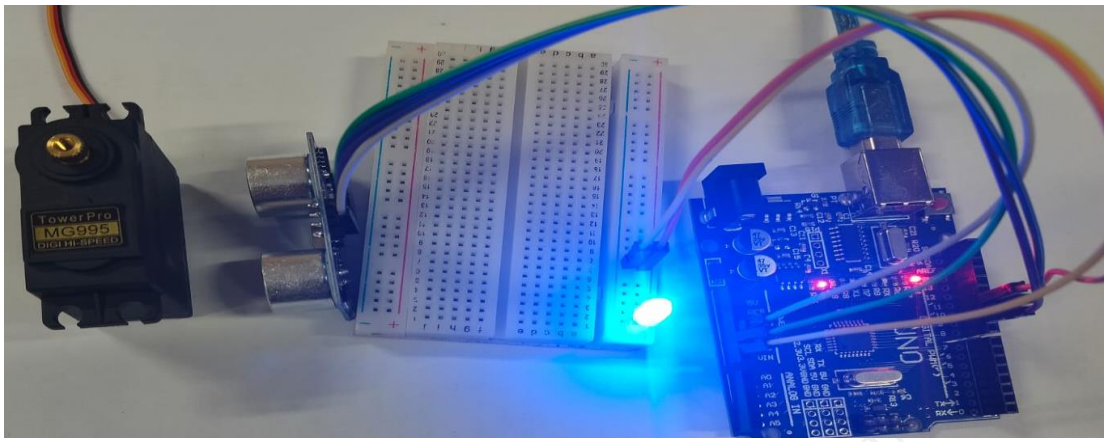


Fig.4: Object Detection Under 10 cm

Exercise

Modify the code to add two warning levels: turn the LED ON when distance < 10 cm and print 'VERY CLOSE!' when distance < 5 cm.

ACTIVITY 2:
Vibration Detection Using ADXL335 Accelerometer and Arduino UNO

Objectives:

1. Understand the working principle of an accelerometer sensor
2. Interface a 3-axis accelerometer with Arduino
3. Read analog sensor values from multiple pins
4. Display motion and tilt data using Arduino
5. Understand basic motion sensing applications

Components Required:

Component	Quantity
Arduino UNO	1
ADXL335 3-Axis Accelerometer Module	1
Jumper Wires	5
USB Cable	1

Theory

Accelerometers are sensors used to measure acceleration, tilt, and vibration in different directions. These sensors are widely used in mobile phones, robotics, gaming controllers, and motion detection systems. In this activity, an ADXL335 3-Axis Accelerometer Module is interfaced with an Arduino Uno to read acceleration data along the X, Y, and Z axes. The Arduino reads the analog signals from the sensor and displays the values on the Serial Monitor.

Working Principle:

The ADXL335 3-Axis Accelerometer measures acceleration using MEMS (Micro-Electromechanical Systems) technology.

1. Inside the accelerometer, a tiny sensing mass moves when acceleration occurs.
2. This movement changes the capacitance inside the sensor.
3. The sensor converts this change into analog voltage signals.
4. The Arduino Uno reads these voltages through analog pins A0, A1, and A2.
5. Arduino converts the analog voltages into digital values (0 – 1023).
6. The acceleration values for X, Y, and Z axes are displayed on the Serial Monitor.

Hardware Connection:

Accelerometer Pin	Arduino Connection
VCC	3.3V
GND	GND
X OUT	A0
Y OUT	A1
Z OUT	A2

Connect VCC to the 3.3V pin on Arduino UNO; do NOT use 5V as it may damage the ADXL335 module.

Circuit Diagram:

Fig.5:ADXL335 Module

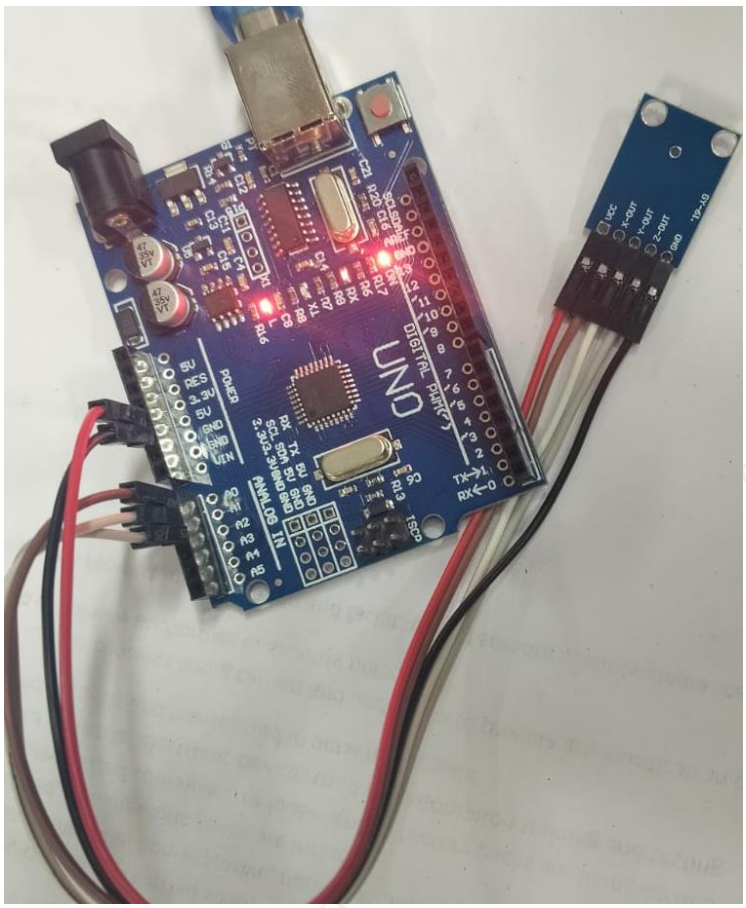


Fig.6: Module to Arduino Connection

The accelerometer provides analog voltage outputs corresponding to acceleration along each axis.

Procedure:

Step 1: Wire the Accelerometer

Make the connections from the Hardware Connection table above. Use 3.3V for VCC.

Step 2: Connect Arduino to Computer

Plug in the USB cable and open Arduino IDE.

Step 3: Write the Program

Enter the following code:

```
// Arduino Code
int xPin = A0;
int yPin = A1;
```

```
int zPin = A2;
void setup() {
  Serial.begin(9600);
}
void loop() {
  int xValue = analogRead(xPin);
  int yValue = analogRead(yPin);
  int zValue = analogRead(zPin);
  Serial.print(X);
  Serial.print(",");
  Serial.print(Y);
  Serial.print(",");
  Serial.print(Z);
  Serial.println(",");
  delay(50);
}
```

Step 4: Upload and Test

1. Select board Arduino UNO and correct COM port
2. Click Upload
3. Open Serial Monitor at baud rate 9600
4. Tilt the sensor in different directions and observe X, Y, Z values change

Output:

Fig.7: Serial plotter output — X, Y, Z axis acceleration values

Exercise

Modify the code to print a message 'Tilt Detected!' when any axis value exceeds 600 or falls below 400.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. An automobile manufacturing plant notices that one robotic welding machine is producing poor-quality welds. The IIOT system reports abnormal vibration and temperature values from that robot. What should the maintenance team do first?

   a) Replace all welding robots in the plant
   b) Ignore the warning until production stops
   c) Inspect the welding robot for possible faults before a breakdown occurs
   d) Increase the production speed of the assembly line

2. A transport company observes a sudden increase in fuel consumption in one of its trucks. The IIOT fuel monitoring system sends an alert to the fleet manager. What is the most appropriate action?

   a) Continue operating the truck without inspection
   b) Inspect the vehicle for fuel leakage, theft, or engine-related issues
   c) Reduce the number of trips made by the truck
   d) Replace the fuel tank immediately

3. An electric vehicle's battery temperature rises above the safe operating limit while charging. According to the IIOT-based monitoring system, what action should be taken?
   a) Disconnect all sensors from the vehicle
   b) Increase the charging current immediately
   c) Activate the cooling system or alert the driver about overheating
   d) Ignore the temperature reading if the battery is charging normally

4. A gearbox in an automobile assembly line shows increasing vibration levels over several days. The IIOT system predicts a possible bearing failure. How can this information be best utilized?
   a) Wait until the gearbox stops working completely
   b) Schedule maintenance before the bearing fails and causes downtime
   c) Increase the operating speed of the gearbox
   d) Turn off all sensors connected to the gearbox

5. An automobile factory wants to reduce electricity consumption without affecting production. Which IIOT-based solution would be most effective?
   a) Operate all machines continuously, even when not required
   b) Remove energy monitoring sensors from the factory
   c) Use smart systems to monitor energy usage and automatically control lighting and HVAC systems
   d) Increase the number of workers monitoring electricity consumption

B. Match the following

Column A	Column B
1. Vibration Sensor	A. Monitors battery overheating in EVs
2. Fuel Monitoring System	B. Detects abnormal movement in rotating parts
3. Battery Temperature Sensor	C. Tracks fuel usage and identify leakage
4. Cloud-Based Diagnostics	D. Sends vehicle data for remote analysis
5. Smart Conveyor System	E. Automates movement of components in factories

C. Fill in the blanks

1. Earlier, operators manually recorded machine temperatures, vibrations, and performance details in _____.
2. In an IIOT system, controllers collect sensor data and prepare it for _____ through communication networks.
3. Smart manufacturing uses technologies such as automation, robotics, artificial intelligence, and _____.
4. Fuel monitoring systems help detect fuel leakage and fuel _____ in transport vehicles.
5. In sensor-based fault detection, the collected sensor data is compared with normal _____ conditions to identify abnormalities.

D. Answer the following

1. A welding robot in an automobile plant shows abnormal vibration readings. How can the IIOT system help prevent production losses?
2. A fleet manager receives an alert indicating unusually high fuel consumption in a delivery vehicle. What actions should be taken using IIOT data?
3. An EV battery temperature rises beyond the safe operating range during charging. Explain how IIOT can help maintain battery safety.
4. A vehicle dashboard displays a warning for low tire pressure. How does sensor-based fault detection help improve vehicle safety in this situation?
5. An automobile factory wants to reduce machine downtime. How can predictive maintenance using IIOT sensors help achieve this objective?

SESSION 2: ROLE OF IIOT IN MONITORING AND MATERIAL HANDLING IN INDUSTRIES

Imagine visiting a modern automobile manufacturing plant. Hundreds of vehicle parts move from one workstation to another. Robotic arms assemble components, conveyor belts transport materials, and large screens display machine status in real time. Despite this complex activity, the entire system works smoothly and efficiently.

Have you ever wondered how industries coordinate thousands of machines, materials, and workers simultaneously?

The answer lies in the **Industrial Internet of Things (IIOT)**.

IIOT enables machines, sensors, robots, and software systems to communicate with each other and exchange information in real time. This helps industries monitor operations, automate material movement, improve productivity, and reduce operational costs.

In modern automotive industries, IIOT plays a significant role in:

- Material handling
- Warehouse automation
- Asset tracking
- Machine monitoring
- Production monitoring
- Predictive maintenance

Think About It!

When ordering a product online, how does the warehouse know where the item is stored and when it should be dispatched?

1.4 Smart Conveyor and Robotic Systems

One of the most visible applications of IIOT in industries is the use of smart conveyor systems and industrial robots.

A conveyor system is used to move materials, components, and products from one location to another. Traditional conveyors operated continuously without much intelligence. Modern IIOT-enabled conveyors can monitor their own performance and automatically respond to changing conditions (Fig.1.4).

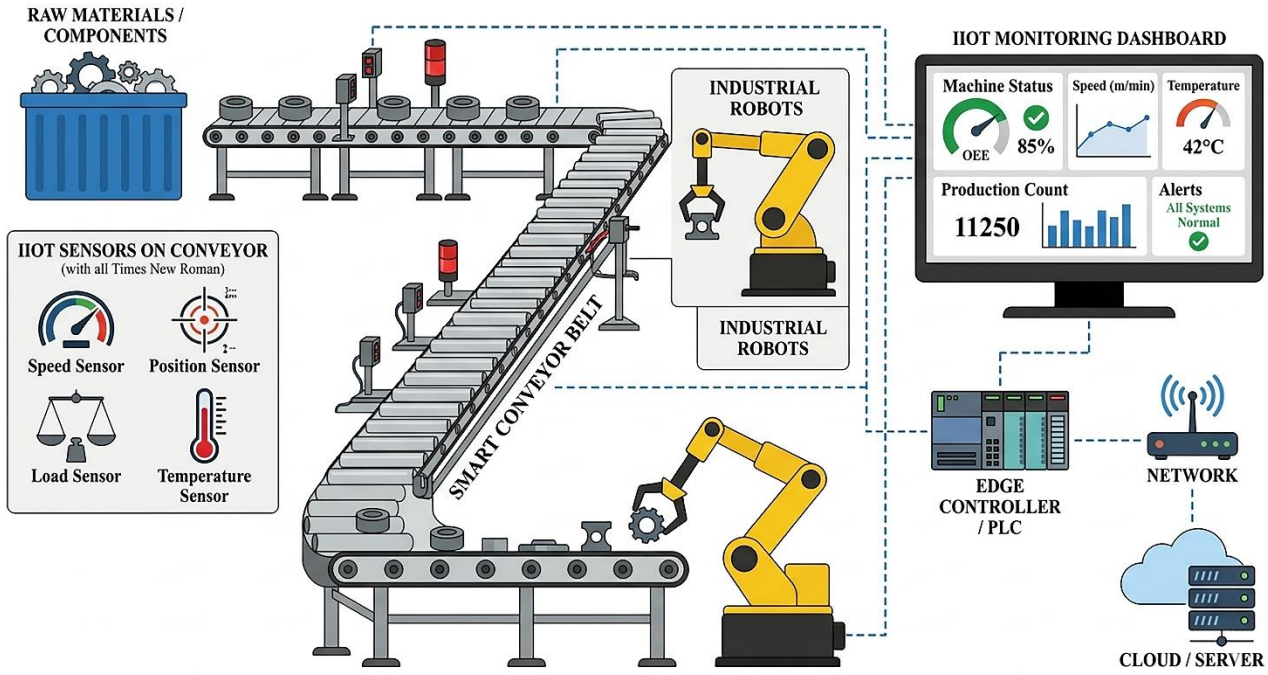


Fig.1.4: **Smart Conveyor and Robotic Systems**

How It Works?

1. Sensors installed on conveyor belts continuously monitor: conveyor movement, load, position, speed, motor temperature and machine status.
2. The collected data is transmitted to controllers and monitoring systems for analysis.
3. Edge Controller/PLC analyzes data and controls conveyor and robotic operations.
4. Robots pick, place, assemble, or move components automatically.
5. Dashboard displays machine status, production data, and alerts in real time. If any abnormal condition is detected, such as excessive load or conveyor blockage, the system immediately generates an alert or automatically stops operation.
6. Cloud Server stores data for monitoring, analysis, and maintenance planning.

Industrial Robots in Material Handling

Industrial robots work alongside conveyor systems to perform tasks such as:

- Picking and placing components
- Packaging products
- Sorting materials
- Welding vehicle parts
- Assembly operations

These robots use sensors and controllers to perform repetitive tasks accurately and efficiently.

A Day Inside an Automobile Factory

A vehicle door arrives at an assembly station through a smart conveyor system. Sensors verify its position and orientation. A robotic arm automatically picks the door and installs it on the vehicle body. The operation is completed within seconds with high precision (Fig.1.5).

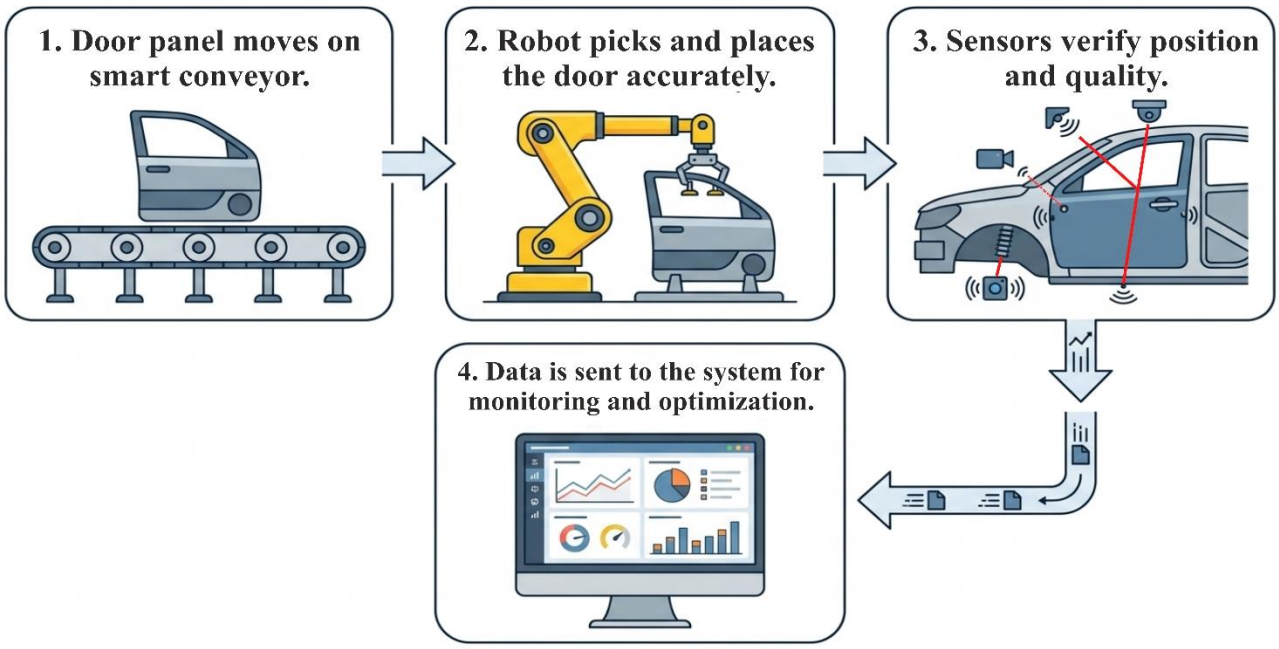


Fig.1.5: **Automation Example**

Benefits of Smart Conveyors and Robots

- ❖ Faster material movement
- ❖ Improved production speed
- ❖ Reduced human effort
- ❖ Better workplace safety
- ❖ Consistent product quality
- ❖ Reduced operational costs

Did You Know?

Modern automobile factories may use more than 500 industrial robots working simultaneously on a single production line. Without automation, this task would require more time and manual effort.

1.5 Asset Tracking and Warehouse Automation

Industries store thousands of components, spare parts, tools, and finished products in warehouses. Managing these assets manually is difficult and time-consuming. IIOT enables industries to track and manage assets automatically.

Asset Tracking: Asset tracking refers to monitoring the location and status of materials, equipment, and products in real time.

Common technologies used for asset tracking include:

- RFID Tags
- Barcode Systems
- QR Codes
- GPS Devices
- Wireless Sensors

Each asset is assigned a unique identification number, allowing industries to know exactly where it is located.

Did You Know?

Large automobile warehouses may track tens of thousands of components every day using RFID technology.

Warehouse Automation: Warehouse automation technologies automatically store, retrieve, and transport materials within the warehouse. Warehouse automation uses:

- Automated Guided Vehicles (AGVs)
- Robotic storage systems
- Smart shelves

- RFID readers
- Inventory management software

How it Works? (Refer Fig.1.6)

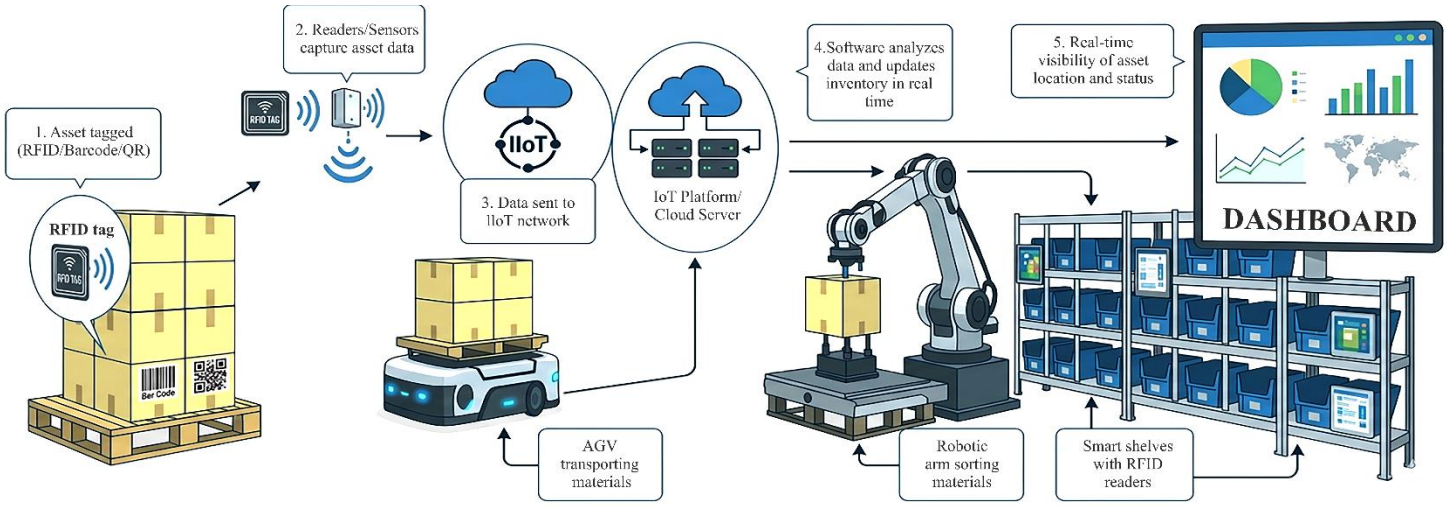


Fig.1.6: Asset Tracking and Warehouse Automation Implementation in smart Factory

Real-Life Example

Suppose a particular engine component is required for vehicle assembly. Instead of searching manually, the warehouse management system identifies the exact storage location and instructs an Automated Guided Vehicle (AGV) to retrieve and deliver the component to the production line.

Benefits of Asset Tracking and Warehouse Automation

- ❖ Accurate inventory management
- ❖ Faster material retrieval
- ❖ Reduced inventory errors
- ❖ Better space utilization

- ❖ Improved supply chain visibility
- ❖ Reduced operational costs

1.6 Real-Time Machine and Production Monitoring

Machines are the heart of any manufacturing industry. If a machine stops unexpectedly, production may be delayed and costs may increase. To avoid such situations, industries use IIOT-based real-time monitoring systems. Sensors continuously collect information about machine performance and operating conditions.

Modern IIOT systems monitor:

- Temperature
- Vibration
- Pressure
- Speed
- Energy consumption
- Production count
- Machine status

The collected data is displayed on dashboards and monitoring screens.

Imagine a factory manager sitting in a control room. Instead of physically visiting every machine, the manager can monitor:

- *Machine status*
- *Production output*
- *Energy consumption*
- *Fault alerts*

through a single dashboard.

How Real-Time Monitoring Works

1. Sensors collect machine data.
2. Controllers process the data.
3. Information is transmitted through industrial networks.
4. Dashboards display machine status.
5. Alerts are generated if abnormal conditions are detected.

Automotive Industry Example

A vibration sensor installed on a motor continuously monitors machine health.

If vibration exceeds the normal range:

- *An alert is generated.*
- *Maintenance personnel are notified.*
- *Corrective action is taken before machine failure occurs*

This reduces downtime and improves productivity.

Benefits of Real-Time Monitoring

- Continuous machine supervision
- Early fault detection
- Reduced downtime
- Better maintenance planning
- Improved productivity
- Enhanced operational efficiency
- Better decision-making

Observe Around You

Visit a supermarket, railway station, airport, or warehouse and identify examples of automated material handling systems. Discuss how sensors and automation may be used to improve their operation.

PRACTICAL ACTIVITY

ACTIVITY 1:
Temperature and Humidity Monitoring Using DHT11 Sensor and ESP32

Objectives:

1. Understand how sensors interface with microcontrollers
2. Identify ESP32 GPIO pins
3. Connect a DHT11 sensor to ESP32
4. Write Arduino code to read sensor data
5. Use the Serial Monitor for debugging
6. Understand basic IoT data collection principles

Components Required:

Component	Quantity
ESP32 DevKit V1	1
DHT11 Temperature & Humidity Sensor	1
Jumper Wires	3
USB Cable	1
Computer with Arduino IDE	1

Theory

In this activity, students will learn how to connect a DHT11 temperature and humidity sensor to an ESP32 microcontroller and read real-time environmental data. The ESP32 will read sensor values and display them in the Serial Monitor. This is a common IoT sensing activity used in smart homes, environmental monitoring, and industrial monitoring systems.

ESP32 Pin Diagram

- 3V3 pin — Power supply (3.3 V)
- GND pin — Ground connection
- D27 — Data pin used for the sensor

Wiring Steps:

Connect VCC (Red Wire) of DHT11 → 3V3 of ESP32

Connect DATA (Yellow Wire) of DHT11 → D27 of ESP32

Connect GND (Black Wire) of DHT11 → GND of ESP32

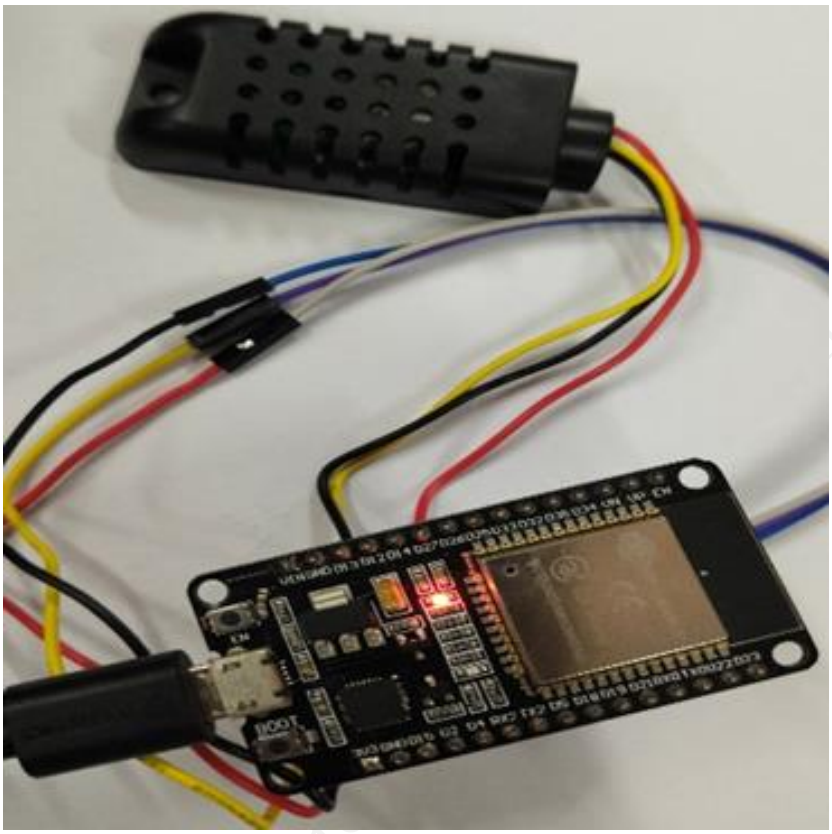


Fig.1: **ESP32 DevKit V1 Pin Diagram**

Procedure:

Step 1: Connect ESP32 to Computer

Connect the ESP32 board to your computer using a USB cable.

Red LED on the board should turn ON once powered.

Step 2: Install Required Library

Open Arduino IDE and navigate to:

1. Tools → Manage Libraries
2. Search for “DHT sensor library"
3. Install “DHT sensor library by Adafruit"

Step 3: Write the Program

Copy and paste the following code into a new Arduino IDE sketch:

```
// Arduino Code
#include <DHT.h>
```

```
#define DHTPIN    27
#define DHTTYPE   DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);
  dht.begin();
}

void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  Serial.print("Humidity: ");    Serial.print(h);
  Serial.print(" %  |  Temp: "); Serial.print(t);
  Serial.println(" C");
  delay(2000);
}
```

Step 4: Upload the Code

1. Select board: Tools → Board → ESP32 Dev Module
2. Select port: Tools → Port → correct COM port
3. Click the Upload button (→)
4. Wait for 'Done uploading'

Step 5: Open Serial Monitor & Observe

1. Go to Tools → Serial Monitor
2. Set baud rate to 115200
3. Observe temperature and humidity readings every 2 seconds

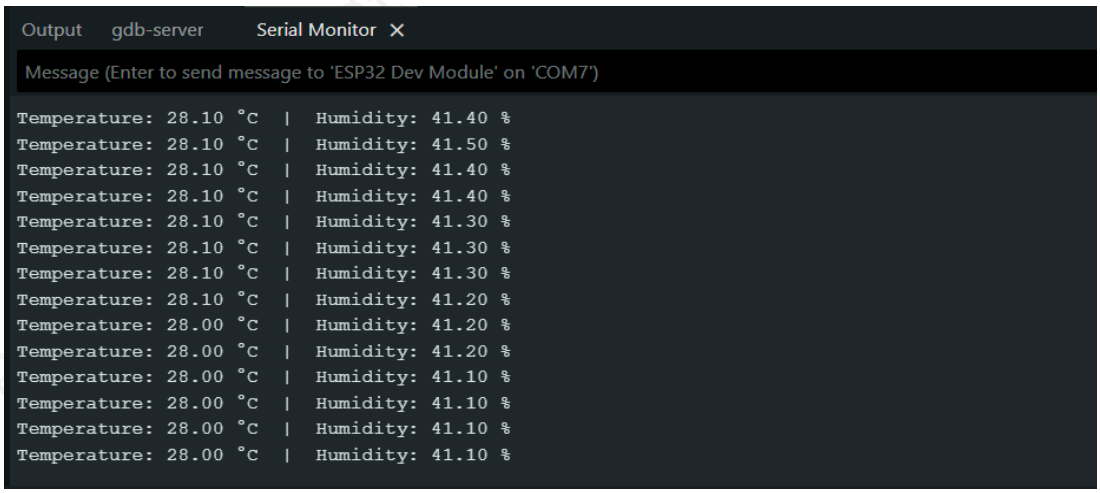


Fig.2: Serial Monitor output — Temperature & Humidity readings

Real-World Applications of Temperature and Humidity Monitoring

- Smart home climate monitoring
- Industrial environment monitoring
- Greenhouse automation
- Cold storage monitoring
- HVAC systems

Exercise

Modify the program to change the sensor reading interval from 2 seconds to 5 seconds. Observe the difference in the Serial Monitor output.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. A conveyor belt in an automobile factory suddenly becomes overloaded. How can an IIOT-enabled conveyor system respond?
   a) Increase conveyor speed automatically
   b) Generate an alert or stop operation to prevent damage
   c) Disconnect all sensors
   d) Shut down the entire factory

2. A robotic arm on an assembly line is repeatedly placing components in the wrong position. Which IIOT feature would help identify the problem?
   a) Asset tracking system
   b) Inventory management software
   c) Sensors and monitoring system
   d) GPS tracking

3. A warehouse manager needs to locate a specific automobile component immediately. Which technology would be most useful?
   a) Vibration sensor
   b) RFID tag
   c) Temperature sensor
   d) Pressure sensor

4. A production dashboard shows abnormal machine vibration levels. What is the most appropriate action?
   a) Ignore the reading
   b) Increase production speed
   c) Inspect the machine for possible faults
   d) Remove the vibration sensor

5. An industry wants to reduce delays in moving materials between workstations. Which IIOT-based solution would be most effective?
   a) Manual transportation
   b) Smart conveyor and robotic systems
   c) Paper-based inventory records
   d) Additional lighting

B. Match the following

Column A	Column B
1. RFID Tags	A. Displays machine status and alerts
2. Dashboard	B. Stores and retrieves materials automatically
3. AGV	C. Identifies and tracks assets
4. Robotic Storage System	D. Transports materials within warehouse
5. Vibration Sensor	E. Detects machine health problems

C. Fill in the blanks

1. Smart shelves and RFID readers are commonly used in _____ automation.

2. Assets are identified and tracked using a unique _____ number.

3. Industrial robots perform repetitive tasks with high _____ and efficiency.

4. Information collected from machines is displayed on monitoring _____.

5. Real-time monitoring helps industries improve maintenance _____ and Planning.

D. Answer the following

1. How can smart conveyor systems help prevent production delays in an automobile manufacturing plant?

2. A warehouse contains thousands of spare parts. How can RFID technology improve inventory management?

3. Explain how real-time machine monitoring can help reduce maintenance costs.

4. An AGV is used in a warehouse instead of manual transportation. What advantages does it provide?

5. A monitoring dashboard shows an abnormal increase in machine temperature. What actions can be taken using IIOT-based monitoring systems?

SESSION 3: SMART TRANSPORTATION AND PREDICTIVE MAINTENANCE

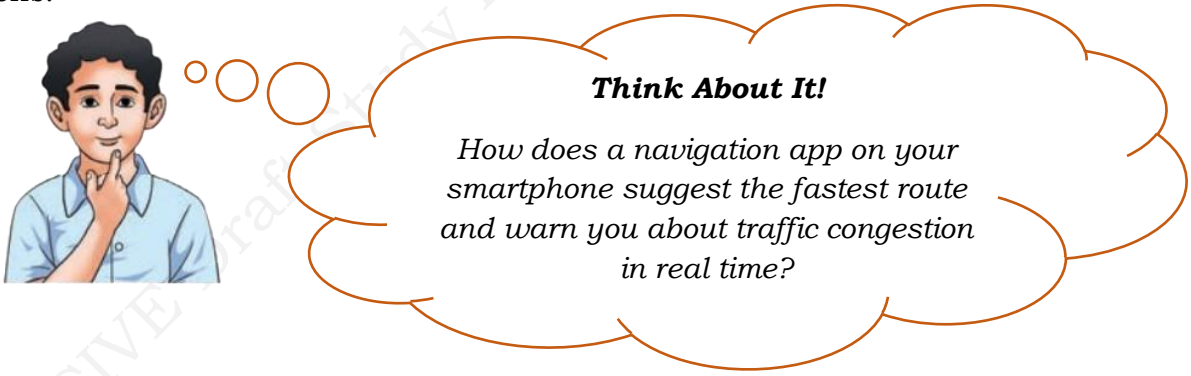
Introduction

Transportation systems around the world are becoming smarter with the use of modern digital technologies. Smart or Intelligent Transportation refers to the use of technologies such as the Industrial Internet of Things (IIOT), Artificial Intelligence (AI), sensors, communication networks, GPS, and data analytics to make transportation systems safer, more efficient, and environmentally sustainable.

These technologies enable vehicles, roads, traffic signals, and control centres to communicate and exchange real-time information. This helps in managing traffic efficiently, reducing accidents, optimizing routes, improving fuel efficiency, and enhancing public transportation services.

In simple terms, Intelligent Transportation is a smart, data-driven system that improves the movement of people and goods by combining digital technologies with transportation infrastructure. Modern connected vehicles continuously collect and share data, allowing operators, drivers, and transportation authorities to make better decisions and improve overall transportation performance.

As the automotive industry moves towards connected, electric, and autonomous vehicles, Intelligent Transportation Systems are becoming an essential part of modern mobility solutions.



1.7 Intelligent Transportation Systems (ITS)

An Intelligent Transportation System (ITS) uses sensors, communication networks, GPS technology, and data analytics to improve traffic management and transportation services.

ITS helps monitor traffic conditions, manage road congestion, improve passenger safety, and provide real-time information to drivers and transport authorities.

1.7.1 Applications of Intelligent Transportation Systems (Refer Fig.1.7)

- Smart traffic signals
- Traffic monitoring systems
- Electronic toll collection
- Smart parking systems
- Emergency vehicle management
- Public transport tracking

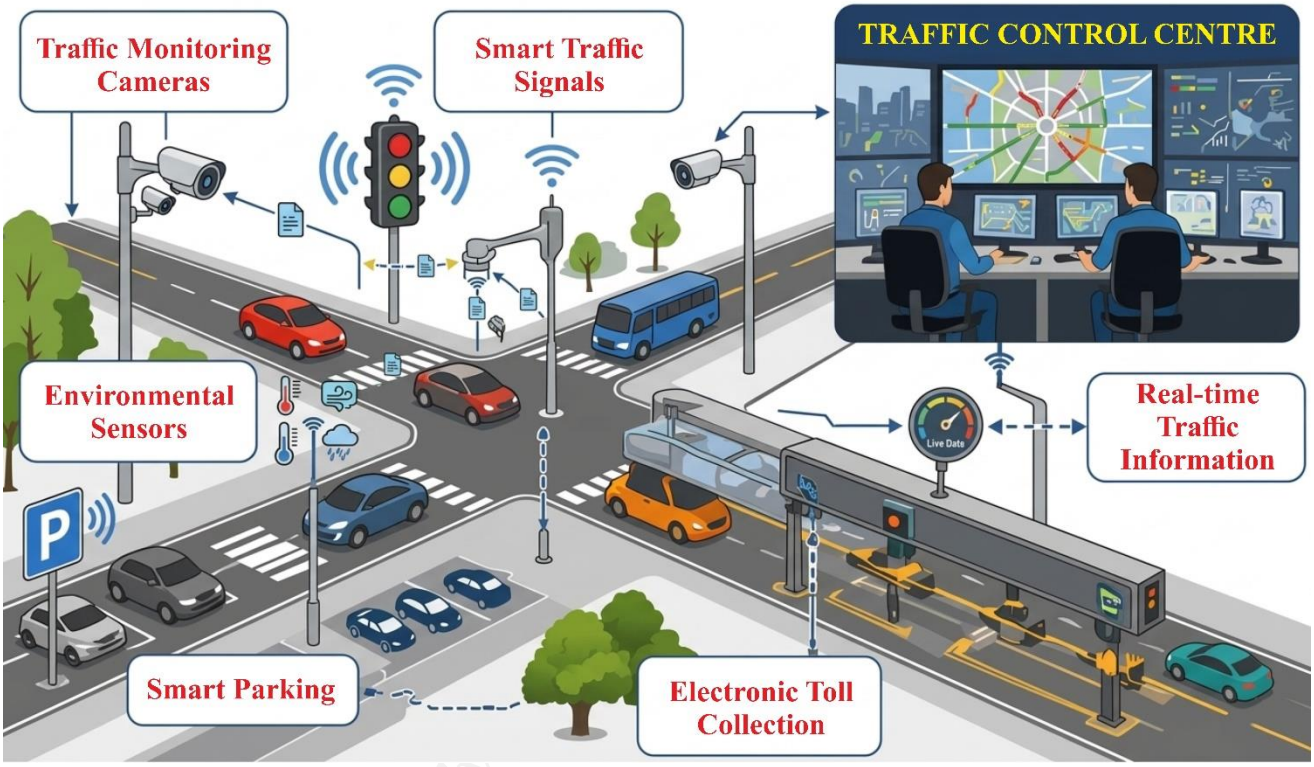


Fig.1.7: Applications of Intelligent Transportation Systems

Example: *In a smart city, traffic sensors installed at intersections continuously monitor vehicle movement. Traffic signals automatically adjust their timing based on traffic density, reducing congestion and improving traffic flow.*

1.7.2 Benefits of ITS

- Reduced traffic congestion
- Improved road safety
- Reduced travel time
- Better fuel efficiency
- Lower vehicle emissions

Think About It!

How can real-time traffic information help drivers save fuel and travel time?

1.8 Connected Vehicles and Fleet Monitoring

Connected vehicles are equipped with sensors, GPS modules, wireless communication systems, and onboard computers that continuously exchange information with other vehicles, cloud platforms, and transportation infrastructure (Fig.1.8).

This connectivity allows vehicle owners and fleet operators to monitor vehicle performance, location, fuel consumption, and driver behavior in real time.

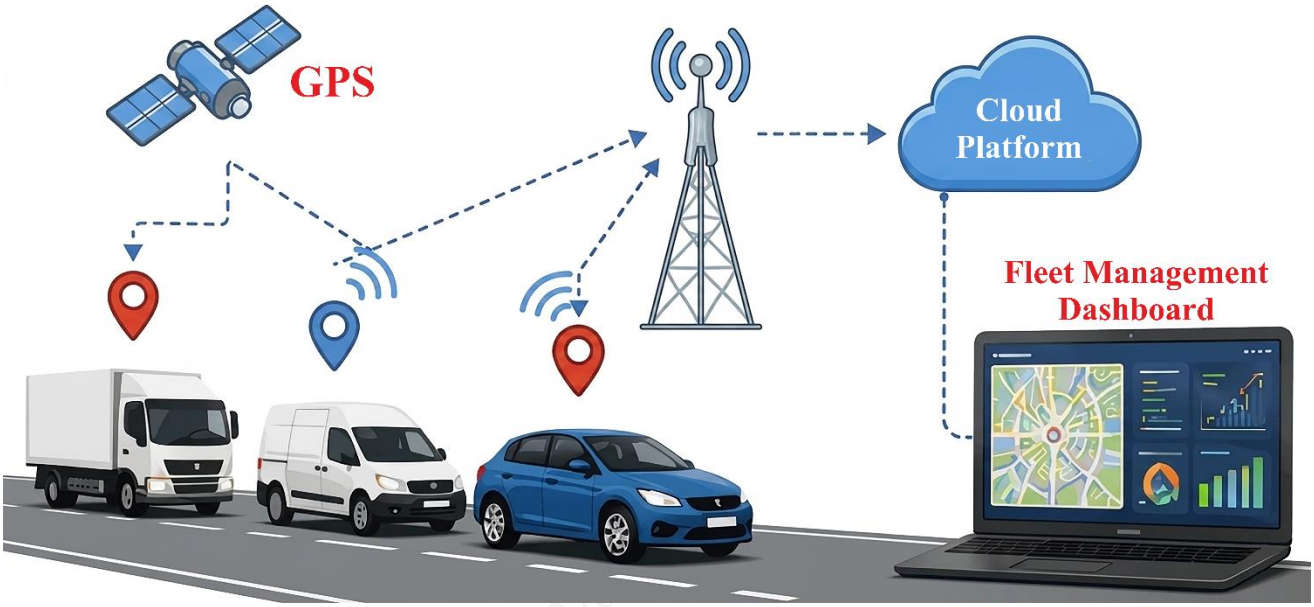


Fig.1.8: Connected Vehicles and Fleet Monitoring

1.8.1 Information Monitored in Connected Vehicles

- Vehicle location
- Speed
- Fuel consumption
- Engine condition
- Battery status
- Driver behavior
- Maintenance requirements

1.8.2 Fleet Monitoring

Fleet monitoring refers to the management of multiple vehicles through a centralized monitoring system. Fleet operators use dashboards to track vehicle movement, optimize routes, improve fuel efficiency, and ensure vehicle safety.

Benefits of Fleet Monitoring

- Real-time vehicle tracking
- Improved route planning
- Reduced fuel consumption
- Better vehicle utilization
- Improved driver safety
- Reduced operating costs

Example: *A logistics company uses GPS-enabled vehicles to monitor delivery of trucks. Fleet managers can track vehicle locations, estimate delivery times, and identify delays using a central monitoring dashboard.*

1.9 EV Infrastructure and Telematics

The increasing adoption of Electric Vehicles (EVs) has created a growing demand for intelligent charging infrastructure and advanced battery monitoring systems. Unlike conventional vehicles, EVs rely on battery power and charging networks for their operation. To ensure efficient charging, battery safety, and energy management, modern EV systems use IIOT technologies. IIOT enables electric vehicles, charging stations, and cloud-based platforms to communicate and exchange information in real time. This connectivity helps monitor charging activities, battery condition, energy consumption, and overall vehicle performance.

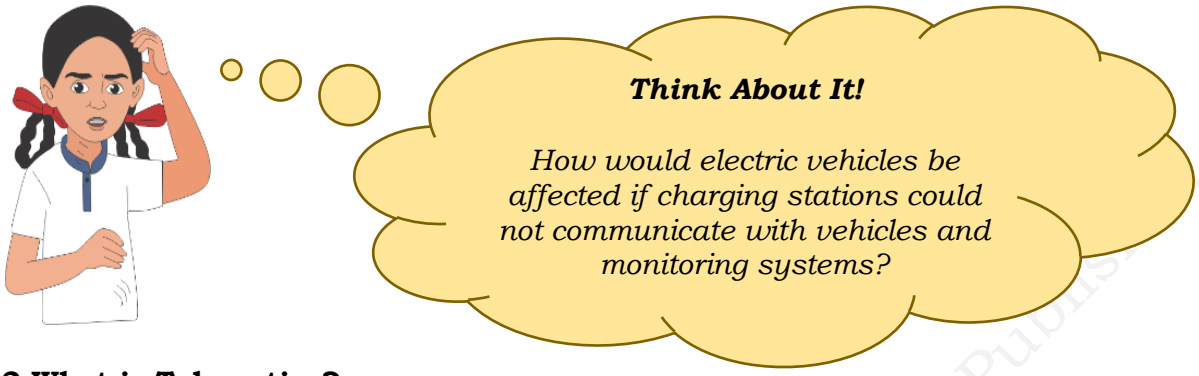
1.9.1 EV Infrastructure

EV infrastructure refers to the systems and facilities that support the operation and charging of electric vehicles. It includes charging stations, battery monitoring systems, smart energy management systems, charging management software, and renewable energy integration systems. These components work together to provide safe, efficient, and reliable charging services (Fig.1.9).

Fig.1.9: EV Infrastructure

Smart charging systems can automatically optimize charging schedules based on battery condition, electricity demand, and user requirements. They also monitor battery health continuously and help prevent problems such as overcharging, overheating, and excessive

battery degradation. As the number of EVs increases, intelligent charging infrastructure plays a vital role in ensuring sustainable and efficient transportation.



1.9.2 What is Telematics?

Telematics is a technology that combines telecommunications and informatics to collect, transmit, and analyze vehicle-related information through communication networks. It enables vehicles to exchange information with cloud servers, fleet management systems, and users in real time.

Telematics systems integrate technologies such as GPS, sensors, wireless communication, cloud computing, and data analytics. These technologies work together to monitor vehicle location, speed, battery status, energy consumption, driving behaviour, and maintenance requirements. The collected data is analyzed to improve vehicle performance, optimize routes, enhance safety, and support predictive maintenance.

In modern electric vehicles, telematics allows drivers and fleet operators to monitor vehicle information remotely through mobile applications and web dashboards. For example, an EV can continuously send information about battery charge level, charging status, and estimated driving range to a cloud platform, enabling users to make informed decisions about vehicle operation and charging (Fig.1.10).

Fig.1.10: Telematics systems

1.10 Predictive Maintenance in Automotive Systems

Traditional vehicle maintenance is usually performed at fixed intervals. However, vehicle components may wear out at different rates depending on usage conditions. Predictive maintenance uses IIOT sensors, real-time monitoring, and data analytics to identify potential problems before failures occur.

Instead of waiting for a breakdown, maintenance can be planned based on the actual condition of vehicle components. Following parameters are monitored in predictive maintenance:

- Engine temperature
- Vibration levels
- Oil condition
- Battery health
- Tire pressure
- Brake performance

Example: *If vibration sensors detect abnormal engine vibrations, the system alerts the vehicle owner and service center before a major engine failure occurs.*

Benefits of Predictive Maintenance

- Reduced vehicle breakdowns
- Improved vehicle reliability
- Lower maintenance costs
- Increased component lifespan
- Better vehicle availability

1.11 Sensor-Based Fault Detection

Modern vehicles are equipped with a large number of sensors that continuously monitor the health and performance of various vehicle systems. These sensors act as the “eyes and ears” of the vehicle by collecting real-time information about operating conditions and sending it to the Electronic Control Unit (ECU) or onboard monitoring system. Sensor-based fault detection helps identify abnormal conditions at an early stage, allowing corrective action to be taken before a minor issue develops into a major failure.

The primary objective of sensor-based fault detection is to improve vehicle safety, reliability, and operational efficiency. Sensors continuously measure important parameters such as temperature, pressure, vibration, speed, current, and battery condition. The measured values are compared with predefined operating limits stored in the vehicle's control system. Whenever a sensor detects a value outside the normal range, the system generates an alert, warning message, or fault code for the driver or maintenance personnel.

1.11.1 Common Sensors Used for Fault Detection in Automotive (Refer Fig.1.11)

Different types of sensors are used in vehicles to monitor various systems and components:

- **Temperature Sensors:** Monitor engine temperature, battery temperature, and coolant temperature.
- **Pressure Sensors:** Measure oil pressure, fuel pressure, and tire pressure.
- **Vibration Sensors:** Detect abnormal vibrations in engines, motors, bearings, and rotating parts.
- **Current Sensors:** Monitor electrical current flowing through circuits and battery systems.
- **Battery Sensors:** Measure battery voltage, charging status, and battery health.
- **Speed Sensors:** Monitor wheel speed, engine speed, and vehicle speed.

Fig.1.11: Common Sensors Used in vehicles

> ***Interesting Fact***
>
> *Modern vehicles can contain more than 100 sensors that continuously monitor different vehicle systems and operating conditions.*

1.11.2 Faults Detected by Sensors

Sensor-based monitoring systems can detect a wide range of vehicle problems, including:

- Engine overheating
- Low oil pressure
- Battery faults
- Tire pressure loss
- Excessive vibration
- Charging system problems
- Motor overheating in electric vehicles
- Brake system abnormalities

1.11.3 Working of Sensor-Based Fault Detection

The process of fault detection generally follows these steps:

1. Sensors continuously collect data from vehicle systems.
2. The data is transmitted to the vehicle's control unit.
3. The control unit compares the sensor readings with predefined safe operating limits.
4. If abnormal conditions are detected, the system generates an alert or warning.
5. The fault information may be displayed on the dashboard or transmitted to a cloud-based monitoring system.

This continuous monitoring enables drivers and service technicians to identify and resolve problems quickly.

Example: *A Tire Pressure Monitoring System (TPMS) continuously measures the air pressure inside each tire. If the pressure falls below the recommended level, the system immediately activates a warning light on the vehicle dashboard. The driver can then inflate the tire before it becomes unsafe or causes excessive tire wear.*

1.12 AI-Based Vehicle Analytics

Artificial Intelligence (AI) is playing an increasingly important role in modern transportation systems. Vehicles today generate large amounts of data through sensors, telematics systems, cameras, GPS devices, and onboard control units. AI technologies analyze this data to identify patterns, predict future events, and support intelligent decision-making.

AI-based vehicle analytics refers to the use of machine learning algorithms and data analytics techniques to process vehicle data and generate useful insights. These insights help improve vehicle performance, safety, maintenance planning, and transportation efficiency.

AI systems can analyze both real-time and historical data collected from connected vehicles. By studying this information, AI can detect hidden patterns that may not be visible to human operators and provide recommendations for improving vehicle operation.

1.12.1 Applications of AI-Based Vehicle Analytics

AI is used in several automotive applications, including:

- Predictive maintenance
- Driver behavior analysis
- Route optimization
- Fuel efficiency improvement
- Traffic prediction
- Battery performance analysis
- Fleet management
- Autonomous vehicle systems

1.12.2 How AI Helps Vehicles

Artificial Intelligence enhances vehicle performance in several ways:

- Predicts component failures before breakdowns occur
- Identifies unsafe driving behavior
- Recommends optimal routes based on traffic conditions
- Improves fuel and energy efficiency
- Supports autonomous and driver-assistance systems
- Optimizes battery charging and usage in electric vehicles
- Enhances fleet management and vehicle utilization

By continuously learning from collected data, AI systems become more accurate and effective over time.

1.12.3 AI in Predictive Maintenance

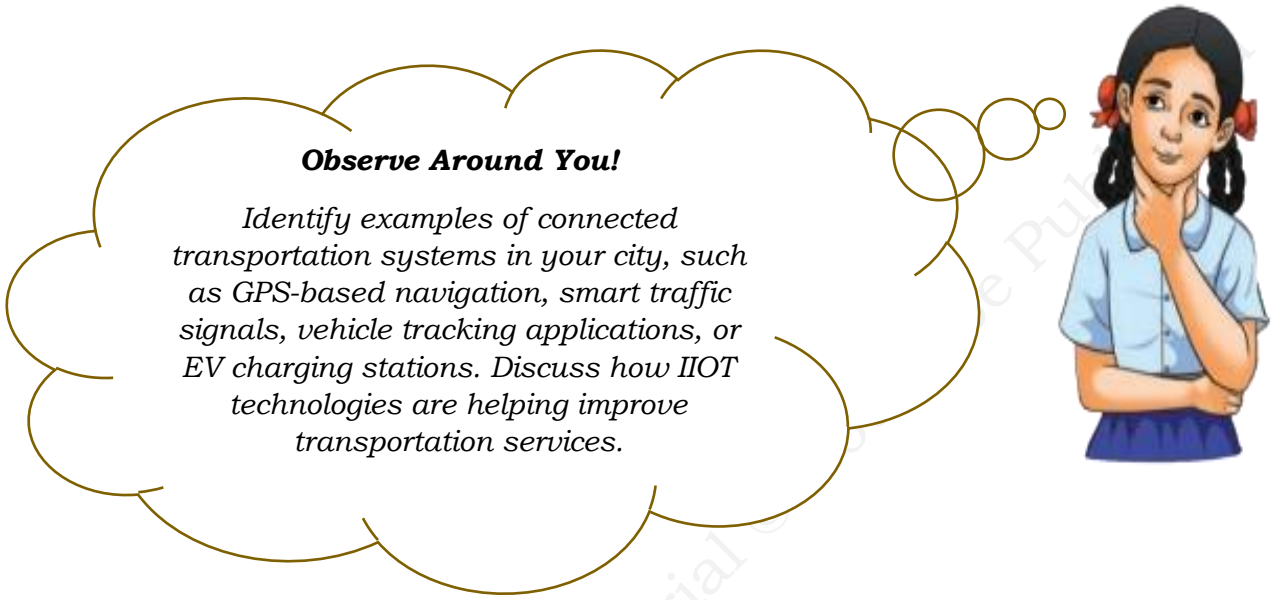
One of the most valuable applications of AI is predictive maintenance. AI algorithms analyze data from temperature, vibration, pressure, and battery sensors to identify signs of wear and deterioration. When abnormal patterns are detected, maintenance alerts are generated before a failure occurs.

For example, if an AI system observes a gradual increase in motor vibration over several weeks, it may predict a bearing failure and recommend maintenance before the vehicle experiences a breakdown.

1.12.4 AI for Driver Behavior Analysis

AI can also analyze driving habits such as sudden acceleration, harsh braking, speeding, and sharp turns. Based on this analysis, the system can provide recommendations that improve safety, reduce fuel consumption, and minimize vehicle wear.

Example: *An AI-based monitoring system analyzes driving patterns and identifies frequent sudden braking and rapid acceleration. The system provides feedback to the driver, suggesting smoother driving practices. As a result, fuel efficiency improves, maintenance costs decrease, and vehicle components experience less wear and tear.*



PRACTICAL ACTIVITY

ACTIVITY 1:
Automatic Rain Detection and Smart Wiper Control Using Rain Sensor and Microcontroller

Objectives:

1. Understand how a rain detection sensor works
2. Interface a rain sensor with Arduino
3. Control a servo motor using Arduino
4. Implement an automatic rain detection system
5. Develop basic sensor-based automation projects

Components Required:

Component	Quantity
Arduino UNO	1

Component	Quantity
Rain Drop Sensor Module	1
SG90 Micro Servo Motor	1
Jumper Wires	10
USB Cable	1

Theory

Rain detection systems are commonly used in automatic window closing systems, smart irrigation systems, and weather monitoring systems. A rain sensor detects water droplets on its surface and sends a signal to a microcontroller.

In this activity, a Rain Detection Sensor Module is interfaced with an Arduino Uno to control a SG90 Micro Servo Motor. When rain is detected, the Arduino rotates the servo motor to close a cover. When no rain is detected, the servo returns to its initial position.

Hardware Connection:

Rain Sensor to Arduino:

Rain Sensor Pin	Arduino Connection
VCC	5V
GND	GND
DO (Digital Output)	D2

Servo Motor Connection:

Servo Wire	Arduino Connection
Red	5V
Brown	GND
Yellow (Signal)	D9

Working Principle:

1. The rain sensor plate contains conductive tracks.
2. When water droplets fall on the plate, conductivity between tracks increases.
3. The sensor module detects this change and outputs a digital signal.

4. The Arduino UNO reads this signal through digital pin D2.
5. If rain is detected, the Arduino rotates the SG90 Servo Motor to 90°.
6. When no rain is detected, the servo returns to its original position (0°).

Circuit Diagram:

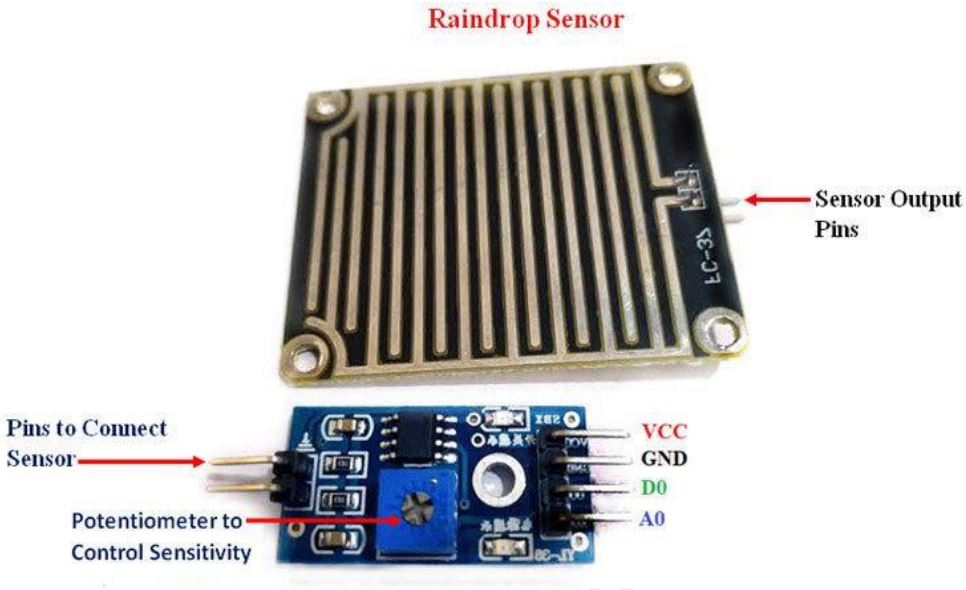


Fig.1: Rain Sensor Module

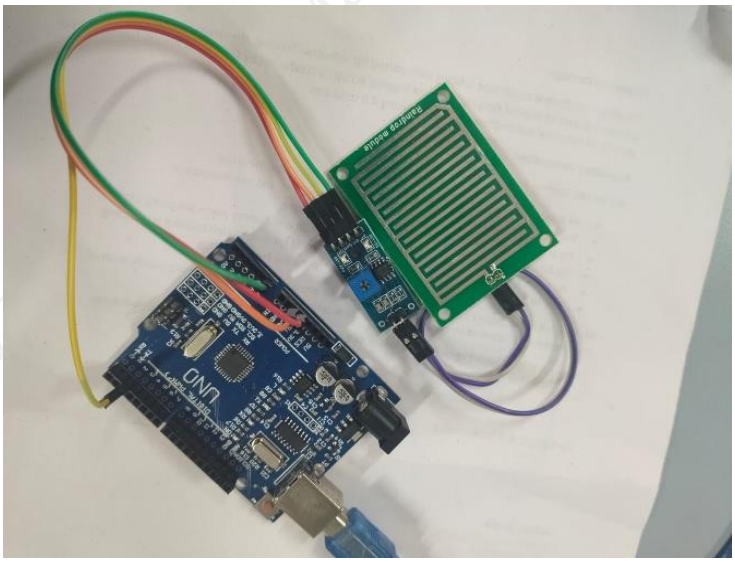


Fig.2: Circuit Diagram

Procedure:

Step 1: **Wire the Components**

Connect the rain sensor and servo motor as per the Hardware Connection tables above.

Step 2: **Connect Arduino to Computer**

Plug in the USB cable and open Arduino IDE.

Step 3: **Write the Program**

Enter the following code:

```
// Arduino Code
#include <Servo.h>
#define Servo_PWM 9
#define Rain_DO   2
Servo MG995_Servo;
const int rotate90Time = 500;
const int stopDelay = 500;
void setup() {
  Serial.begin(19200);
  pinMode(Rain_DO, INPUT);
  Serial.println("System Started");
}
void loop() {
  int rainState = digitalRead(Rain_DO);
  if (rainState == LOW) {
    Serial.println("Rain Detected -> Clockwise 90 deg");
MG995_Servo.attach(Servo_PWM);
    MG995_Servo.write(0);
    delay(rotate90Time);
    MG995_Servo.detach();
    delay(stopDelay);
    rainState = digitalRead(Rain_DO);
    if (rainState != LOW) {
      Serial.println("Rain Stopped");
      MG995_Servo.detach();
      return;
    }
    Serial.println("Rain Detected -> Counter Clockwise 90 deg");
MG995_Servo.attach(Servo_PWM);
    MG995_Servo.write(180);
    delay(rotate90Time);
    MG995_Servo.detach();
    delay(stopDelay);
  }
  else {
    Serial.println("No Rain");
    MG995_Servo.detach();
    delay(300);
  }
}
```

Step 4: Upload and Test

1. Select board Arduino UNO and correct COM port
2. Click Upload
3. Open Serial Monitor at baud rate 9600
4. Sprinkle a few drops of water on the sensor plate and observe the servo movement

Output:

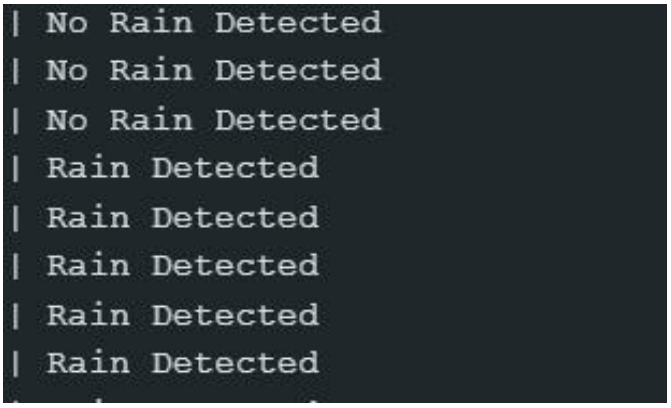


Fig.3: Rain Detected

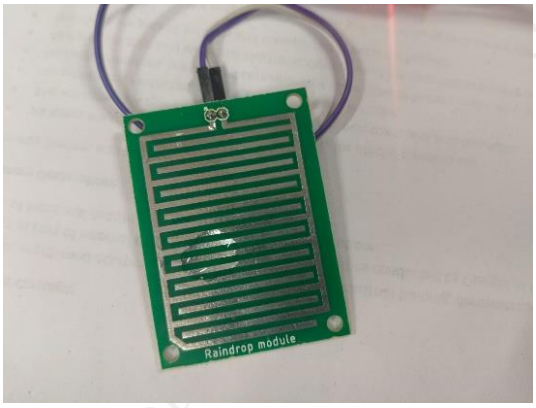


Fig.4: Water droplet on Sensor

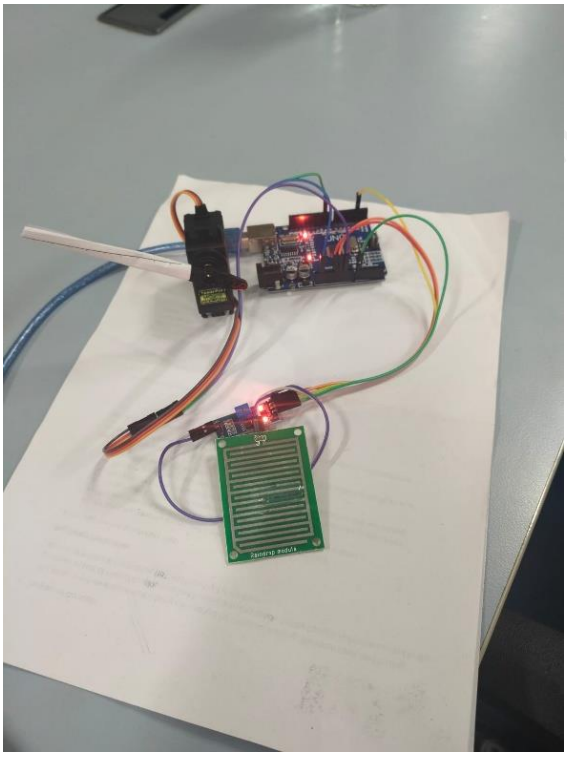


Fig.5: Servo at 0°

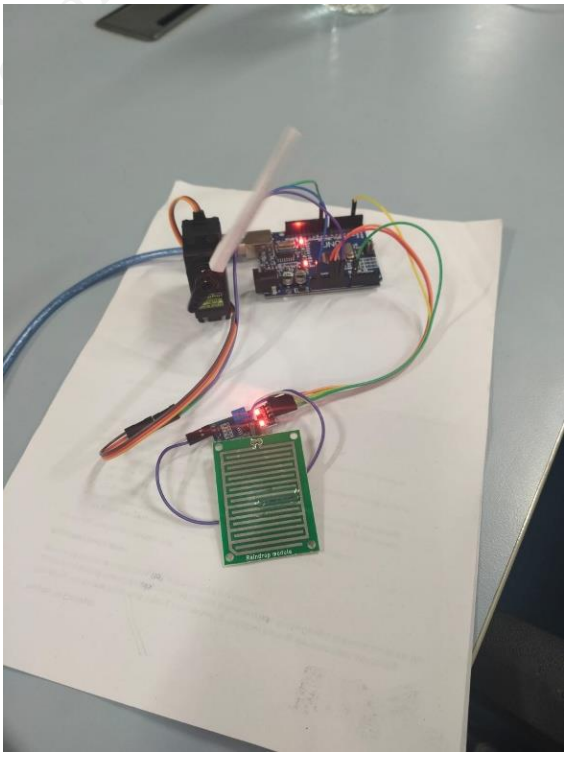


Fig.6: Servo at 70°

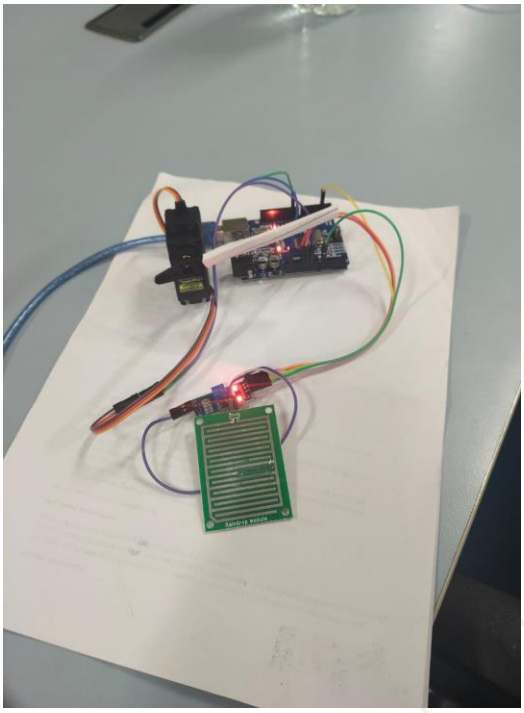


Fig.7: Servo at 90°

Exercise

Modify the code so the servo moves to 180° when rain is detected and returns to 0° when dry.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. A fleet manager notices that a delivery vehicle is frequently taking longer routes, resulting in higher fuel consumption. Which technology can help optimize the route?
   a) Pressure Sensor
   b) Telematics System
   c) Battery Sensor
   d) Tire Pressure Monitoring System

2. An EV charging station detects that a battery temperature is rising rapidly during charging. What should the smart charging system do?
   a) Increase charging current
   b) Disconnect all sensors
   c) Monitor battery condition and take corrective action to prevent overheating
   d) Turn off GPS services

3. A vehicle's vibration sensor reports increasing vibration levels over several weeks. Which maintenance approach would best utilize this information?
   a) Corrective maintenance after failure
   b) Manual inspection every year
   c) Predictive maintenance before breakdown occurs
   d) No maintenance required

4. A transport authority wants to reduce traffic congestion at busy intersections. Which Intelligent Transportation System application would be most useful?
   a) Battery monitoring system
   b) Smart traffic signals
   c) Tire pressure sensors
   d) Oil pressure sensors

5. An AI system detects frequent harsh braking and sudden acceleration by a driver. What is the likely benefit of this analysis?
   a) Increase vehicle weight
   b) Improve driving behaviour and fuel efficiency
   c) Increase engine temperature
   d) Reduce battery monitoring

B. Match the following

Column A	Column B
1. Smart Traffic Signals	A. Monitors battery charge and driving range
2. Fleet Monitoring	B. Reduces road congestion
3. EV Charging Infrastructure	C. Tracks multiple vehicles from a central dashboard
4. Tire Pressure Sensor	D. Supports charging and battery management
5. Telematics Dashboard	E. Warns about low air pressure

C. Fill in the blanks

1. Telematics combines telecommunications and _____ to collect and analyze vehicle information.
2. The Electronic Control Unit is commonly abbreviated as _____.
3. In predictive maintenance, maintenance decisions are based on the actual _____ of vehicle components.
4. Smart charging systems help prevent battery _____ and overheating.
5. AI systems analyze both real-time and _____ vehicle data to identify patterns and trends.

D. Answer the following

A. How can Intelligent Transportation Systems help reduce travel time and fuel consumption in urban areas?

B. A fleet operator notices increased fuel consumption in several vehicles. How can telematics help identify the possible causes?
C. How does predictive maintenance improve vehicle availability compared to traditional maintenance schedules?
D. A vehicle dashboard displays a warning about low oil pressure. Explain how sensor-based fault detection helps prevent serious engine damage.
E. An AI system predicts a possible bearing failure based on increasing motor vibration. What actions should the maintenance team take and why?

MODULE 2	Remote Monitoring and Controlling in IIOT Network

Module Overview

This module introduces the concept of Remote Monitoring and Control in IIOT networks, explaining how modern industries manage machines, sensors, and processes from remote locations. It provides foundational knowledge of IIOT-based connectivity, data collection, and communication systems used for real-time supervision and decision-making in industrial environments.

The module covers the architecture of remote data acquisition, explaining how sensors, gateways, and protocols work together to transmit data between field devices and cloud platforms. It also focuses on dashboards and data visualization, helping learners interpret machine data through key performance indicators (KPIs) and graphical interfaces for better operational insights.

Students will study remote control and command execution, understanding how industrial systems operate safely and efficiently over secure networks. The module further explores alerts, alarms, and proactive analysis, emphasizing how smart systems detect and respond to equipment faults or performance deviations automatically.

Additionally, learners will understand security measures for remote IIOT access, including authentication, encryption, and protection strategies to prevent unauthorized access and data breaches. Students will also gain awareness of best practices for network reliability, maintenance, and optimization, preparing them to handle real-world IIOT monitoring and control systems effectively.

Learning Outcomes

After completing this module, you will be able to:

- Explain the concepts, architecture, and importance of remote monitoring and control in IIOT-based industrial systems.
- Describe and analyze the process of remote data acquisition, including the roles of sensors, gateways, and communication protocols.
- Interpret and visualize real-time machines and process data using dashboards, performance metrics, and trend analysis tools.
- Execute and manage remote control operations and command responses while ensuring system safety and reliability.
- Implement effective alert mechanisms, proactive analysis methods, and security measures to protect remote IIOT networks from faults or unauthorized access.

Module Structure
Session 1: Importance of Remote Monitoring and Control **Session 2:** Remote Data Acquisition Architecture **Session 3:** Dashboards and Data Visualization **Session 4:** Remote Control and Command Execution **Session 5:** Alerts, Alarms and Proactive Analysis **Session 6:** Security Measures for Remote IIOT Access

SESSION 1: IMPORTANCE OF REMOTE MONITORING AND CONTROL

2.1 Centralized IIOT Operations

In earlier times, industries depended heavily on manual supervision. Operators had to move from one machine to another, take readings, record them in logbooks, and report to supervisors. This process was time-consuming, error-prone, and often resulted in delayed responses. Even a small mistake in reading or a delay in reporting could lead to production losses, equipment damage, or even safety hazards.

With the advancement of the IIOT, the concept of centralized monitoring and control has revolutionized industrial operations. In a centralized IIOT system, sensors and actuators are connected to machines to measure and control various parameters such as temperature, vibration, pressure, speed, humidity, and energy usage. These sensors continuously collect data and transmit it through industrial communication networks or cloud platforms to a central control system.

From a single control room or even through a laptop, tablet, or smartphone; engineers can now monitor hundreds of machines across multiple locations. This provides a "single window" view of the entire plant, improving efficiency and decision-making. If any machine operates outside its normal range, the system immediately generates an alert or notification, allowing maintenance teams to take instant corrective action. This reduces human dependency, minimizes downtime, and ensures safer and more reliable operations.

Centralized IIOT operations also help in data analysis and optimization. Managers can compare performance between different machines, departments, or even factories located in various regions. By studying patterns and trends, they can identify inefficient processes, schedule predictive maintenance, and plan resource allocation more effectively.

Furthermore, these systems can be integrated with Artificial Intelligence (AI) and Machine Learning (ML) algorithms to predict failures, optimize production schedules, and even control operations automatically without human intervention.

Future of Industries

In the future, factories may automatically correct machine problems before humans even notice them.

For example, in the automotive industry, modern assembly plants have hundreds of robotic arms, conveyor systems, and testing units working simultaneously. Without centralized monitoring, managing such a complex system would be extremely difficult. Today, all these machines are connected through an IIOT-based central platform. Supervisors can view real-time data, track machine health, and even control robotic movements remotely. If any robot malfunctions or a line slows down, the system instantly flags the issue and suggests corrective measures.

Centralized IIOT operations have therefore become the heart of modern smart factories, enabling industries to achieve real-time visibility, faster decision-making, improved safety, and greater productivity, all from one unified platform (Fig.2.1).

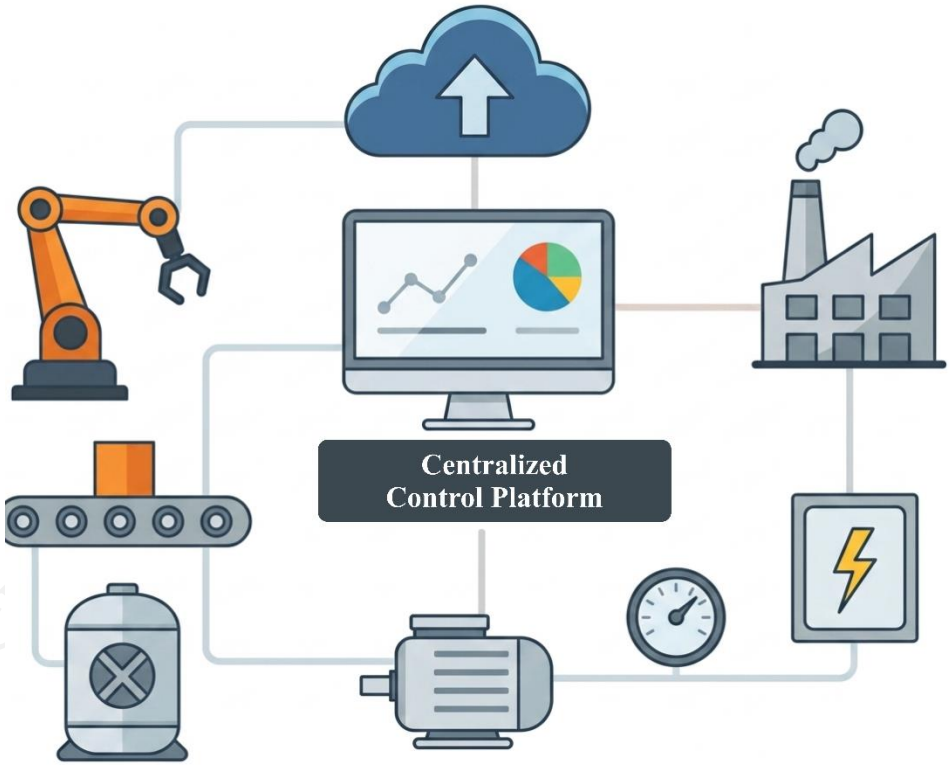
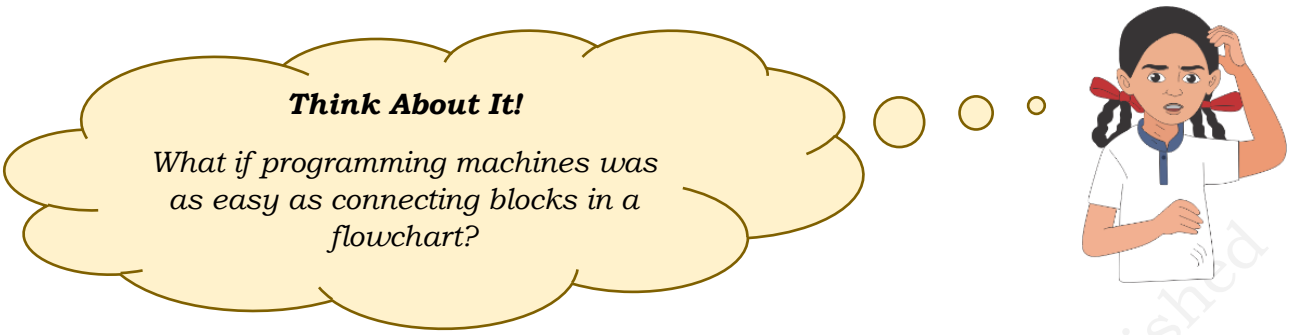


Fig.2.1: Centralized Monitoring

2.2 Introduction to Node-RED



Node-Red is a programming tool for wiring together hardware devices, APIs and online services in new & interesting ways. It provides a browser-based editor that make it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single click.

How We Can Use Node-Red

Node-RED is a flow-based development tool widely used in IIOT applications for connecting sensors, machines, cloud platforms, and dashboards. It provides a browser-based editor where users can create applications by simply dragging and connecting blocks called *nodes*. Each node performs a specific function such as reading sensor data, processing information, generating alerts, or controlling devices (Fig.2.2).

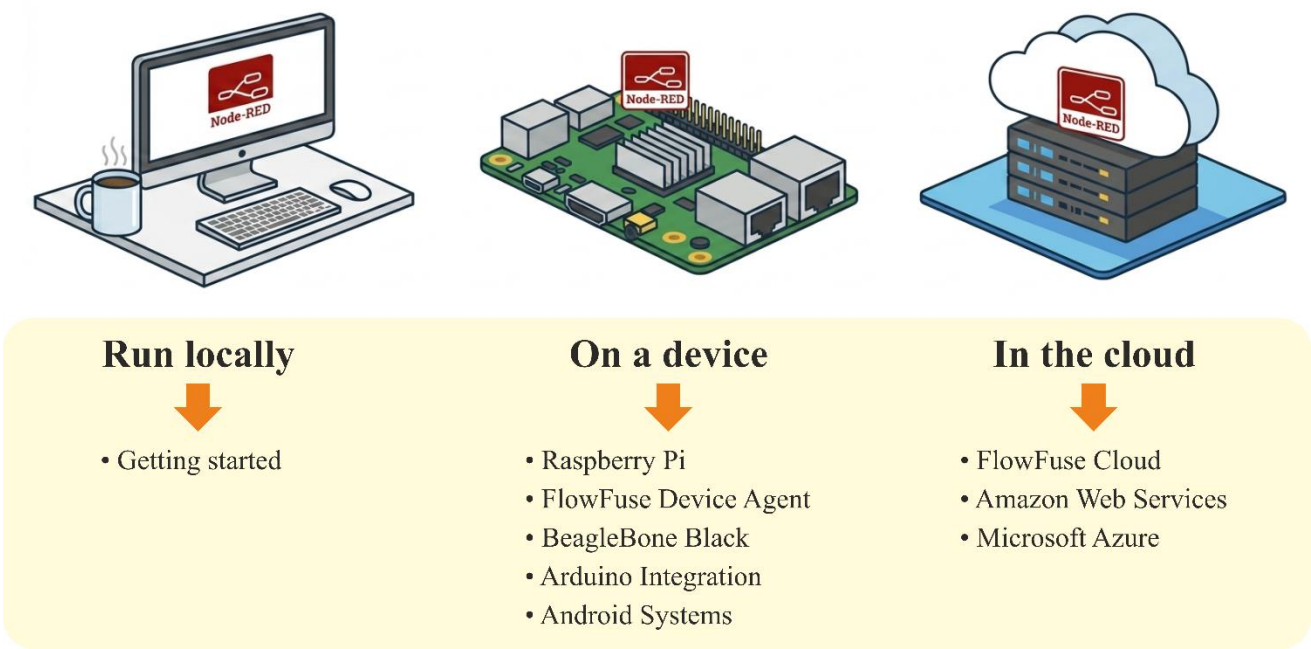


Fig.2.2: Introduction to Node-Red

In modern IIOT environments, Node-RED acts as an edge computing platform that collects data from industrial equipment, processes it locally, and sends it to dashboards or cloud

services. This enables real-time monitoring and control without needing complex programming.

Node-RED supports multiple industrial communication protocols such as:

- MQTT
- HTTP
- WebSocket
- Serial Communication
- Modbus
- OPC-UA

These protocols allow Node-RED to communicate with PLCs, microcontrollers like ESP32, industrial sensors, and cloud platforms.

Key Features of Node-RED (Refer Fig. 2.3)

1. Flow-based programming using drag-and-drop interface
2. Browser-based editor (no complex IDE required)
3. Easy integration with IoT hardware and cloud services
4. Real-time data processing
5. Built-in dashboard for visualization
6. Supports JavaScript for advanced logic
7. Large library of community nodes

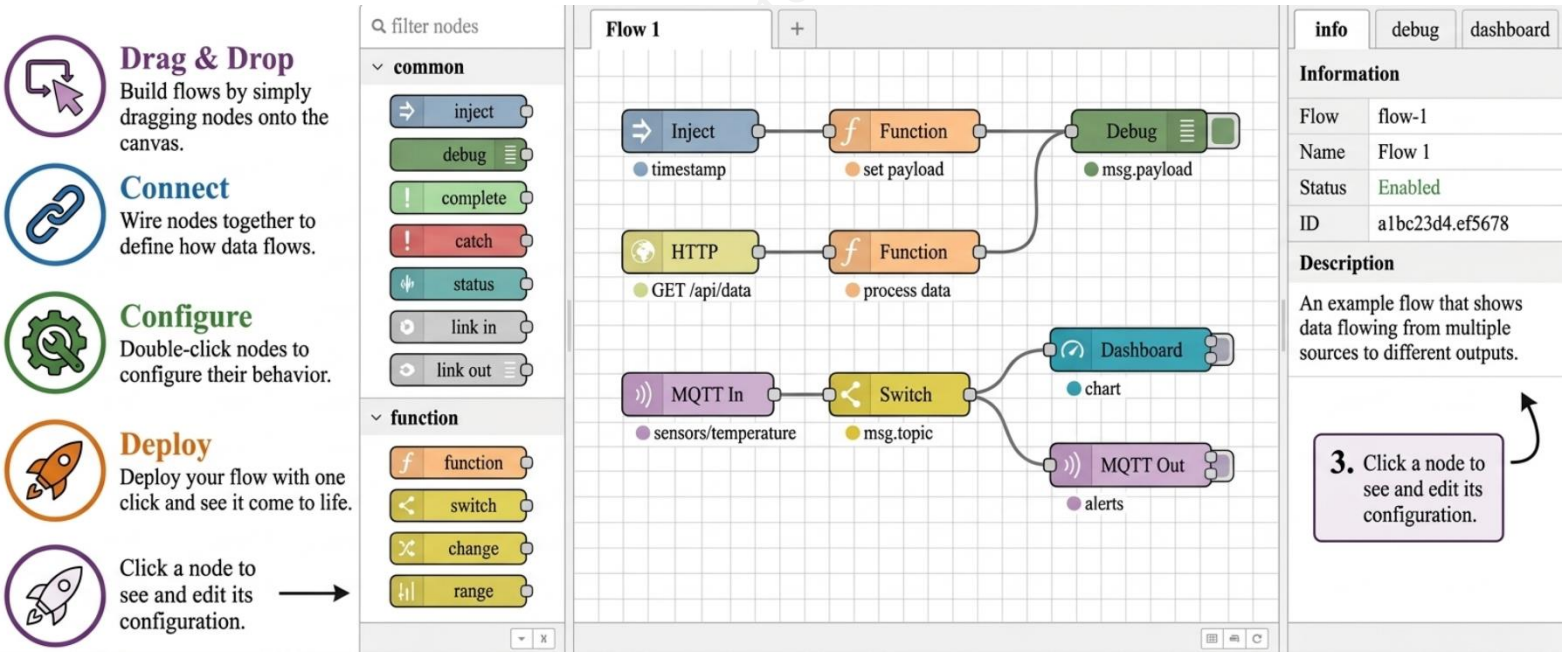


Fig.2.3: Node-Red Interface

Role of Node-RED in IIOT Systems

Node-RED performs multiple functions in an IIOT architecture. Some of them are listed as follows:

- Data acquisition from sensors and machines
- Data processing and filtering
- Alarm generation based on thresholds
- Dashboard creation for visualization
- Remote monitoring of industrial systems
- Remote control of actuators and devices
- Integration with cloud platforms

Some common Examples of IIOT Flow Using Node-RED are as follows:

Sensor → Node-RED → Processing → Dashboard
Sensor → Node-RED → Alarm → Notification
Dashboard → Node-RED → Control Command → Machine

For example, a temperature sensor installed on a motor sends data to Node-RED. If the temperature exceeds a predefined limit, Node-RED generates an alert and displays it on the dashboard. Engineers can then remotely stop the motor using a control button on the same dashboard.

Thus, Node-RED serves as a central platform for remote monitoring and control, making it an essential tool in modern smart factories.

2.3 Benefits of Centralized Monitoring

Centralized monitoring through IIOT offers a wide range of benefits that improve efficiency, safety, and productivity in industries. By connecting machines and systems to a single control platform, industries can manage complex operations more effectively and make faster, data-driven decisions.

1. Real-Time Visibility

All connected machines and sensors continuously send data to a central dashboard. Engineers can view live performance, equipment status, and alerts in real time without visiting the shop floor.

2. Early Fault Detection

The system instantly detects unusual conditions such as overheating, excessive vibration, or pressure drops. Early alerts allow maintenance teams to take timely action and prevent costly breakdowns.

3. Reduced Downtime

Since problems are detected early, repairs can be planned. This minimizes unplanned stoppages and ensures smooth, continuous operation.

4. Improved Safety

Remote monitoring reduces the need for operators to work near hazardous machines or environments. Automated alerts and shutdown systems enhance workplace safety.

5. Better Resource Management

Data collected from various machines helps managers analyze energy use, material flow, and production speed. This enables better planning and cost control.

6. Enhanced Decision-Making

Centralized systems provide complete visibility across all departments. Managers can compare performance between machines, identify bottlenecks, and make informed strategic decisions.

7. Remote Accessibility

Through cloud connectivity, authorized personnel can access data and control systems from any location using a computer or mobile device. This supports remote operations and supervision.

8. Integration with AI and Analytics

When combined with Artificial Intelligence (AI) and data analytics, centralized monitoring can predict machine failures, optimize performance, and suggest process improvements automatically.

2.4 Predictive Maintenance

Maintenance of machines is one of the most critical parts of industrial operations. It directly affects production quality, safety, and cost efficiency. Traditionally, industries relied on two primary maintenance approaches; reactive maintenance and preventive maintenance.

- **Reactive Maintenance:**

  This method involved repairing or replacing equipment only after it failed. While simple and low-cost initially, it often resulted in unexpected breakdowns, production losses, and expensive emergency repairs.
  Example: In an automotive paint shop, a conveyor motor failure can stop the entire paint line, cause production delays and wasted materials.

- **Preventive Maintenance:**

  Here, machines were serviced on a fixed schedule, such as every few months, regardless of their actual condition. Though it prevented some failures, it often led to over-maintenance, unnecessary part replacements, and wasted man-hours. Example: Replacing all hydraulic filters in a robotic assembly unit every 3 months, even when some were still in good condition (Fig.2.4).

With the IIOT, industries have adopted a smarter, data-driven method known as Predictive Maintenance. In this approach, sensors continuously monitor machine health parameters

such as vibration, temperature, current, pressure, sound, and lubrication levels. These values are sent to a centralized IIOT platform or cloud dashboard for real-time analysis using AI and machine learning algorithms.

If the system detects unusual behavior, like a slight increase in vibration, abnormal heat, or power fluctuation, it can predict a possible failure well before it happens. Maintenance teams are alerted instantly through dashboards or mobile notifications.

This approach offers several advantages:

- Early Fault Detection: Problems are identified before they cause breakdowns.
- Planned Downtime: Repairs can be scheduled during non-productive hours, avoiding disruption.
- Cost Efficiency: Reduces unnecessary maintenance and costly emergency repairs.
- Extended Equipment Life: Machines last longer due to timely intervention.
- Improved Safety: Faulty components can be fixed before posing a risk to workers or production

Examples:

1. In an automotive factory, a vibration sensor attached to a robotic arm motor detects a slight imbalance. The IIOT system analyzes this data and sends an alert to the maintenance team, predicting bearing wear. The team replaces the bearing during routine servicing, preventing a costly production halt.
2. In spinning and weaving units, temperature and current sensors track motor performance. When the motor's temperature starts to rise beyond a threshold, the system predicts overheating and alerts the technician, preventing motor burnouts.
3. IIOT sensors monitor turbine vibrations and oil pressure. When slight deviations are noticed, predictive algorithms forecast bearing wear, enabling engineers to schedule maintenance during off-peak hours—saving millions in repair costs.
4. Humidity and vibration sensors track packaging machines. A sudden change in vibration frequency alerts staff about misalignment in rollers, allowing quick correction before affecting packaging quality.
5. Forklifts and automated guided vehicles (AGVs) equipped with IIOT sensors monitor battery voltage and motor temperature. Predictive alerts help plan timely battery replacement, avoiding downtime during peak hours.
6. Vibration sensors installed on crushers and conveyor belts detect imbalances and misalignments early. Predictive maintenance ensures smoother operation and prevents costly mechanical damage.

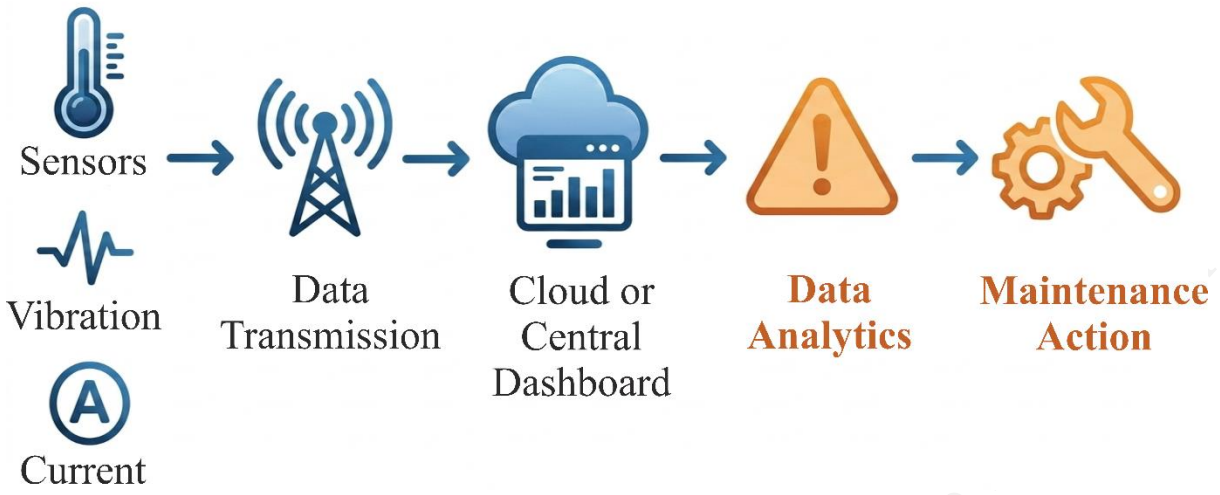


Fig.2.4: Predictive Maintenance

2.5 Key Performance Metrics

Think About It!

How can industries improve performance without measuring machine efficiency?

Dashboards use key performance metrics to visualize the health, productivity, and efficiency of industrial systems in real time. By tracking these indicators, managers and engineers can make informed decisions, identify bottlenecks, and optimize operations for better outcomes.

In IIOT environments, massive amounts of data are continuously generated by sensors, machines, and production systems. However, this data becomes meaningful only when it is analyzed through Key Performance Metrics (KPIs). Key Performance Metrics (KPIs) are the measurable indicators that show how effectively an operation, machine, or system is performing.

Dashboards convert this raw data into easy-to-understand visual indicators. Dashboards use these metrics to visualize the health, productivity, and efficiency of industrial systems in real time. By tracking these indicators, managers and engineers can track progress, identify problems, and make data-driven decisions.

By monitoring the right KPIs, industries can achieve better efficiency, reliability, and profitability as given here:

1. Overall Equipment Effectiveness (OEE)

OEE is a universal metric used to measure how effectively manufacturing equipment is being utilized. It combines three critical elements:

- **Availability:** Whether the machine is operational when required.
- **Performance:** How fast the machine operates compared to its designed speed.
- **Quality:** How many units meet the quality standards.

$$OEE = Availability \times Performance \times Quality$$

Example:
A CNC machine operates 90% of the time, performs at 95% of rated speed, and produces 98% quality parts.

$$OEE = 0.90 \times 0.95 \times 0.98 = 83.7\%$$

An OEE above 85% is considered excellent in manufacturing.

2. Production Output

This metric measures the total quantity of goods produced within a specific time period. It helps assess if the factory meets its production targets.

- Monitored in real time through dashboards.
- Can be displayed per shift, per hour, or per production line.

Example:
"Shift Output: 1,200 components produced (Target: 1,000)" – This indicates improved productivity

3. Downtime and Availability

Downtime measures the total time a machine is non-operational. It is categorized as:

- **Planned Downtime:** Scheduled maintenance or equipment upgrades.
- **Unplanned Downtime:** Machine breakdowns, sensor faults, or power failures.

Reducing downtime increases machine availability and ensures steady production flow. Dashboards often use color-coded indicators (green = active, red = stopped) for quick monitoring.

4. Machine Health and Condition Monitoring

Machine health metrics are essential for predictive maintenance.
Sensors track:

- Vibration levels (mechanical wear)
- Temperature (overheating detection)
- Current and voltage (electrical performance)

- Noise levels (bearing failure prediction)

Example:
If vibration readings exceed normal limits, the system sends an alert predicting motor bearing wear.

5. Energy Consumption and Efficiency

Energy usage is a major cost factor in industries. Monitoring power consumption per machine or process helps identify inefficiencies and save costs. Dashboards can display:

- Energy used per product (kWh/unit)
- Peak load periods
- Energy efficiency rating of equipment

Example:
An IIOT dashboard shows that Machine A consumes 20% more power than others performing similar tasks, prompting an inspection for mechanical issues.

6. Cycle Time

Cycle Time represents the time taken to produce one unit from start to finish. It helps evaluate process efficiency and identify slow stages.

Example:
If the expected cycle time for a robotic welding operation is 25 seconds but current performance shows 32 seconds, optimization is required to restore speed.

7. Quality Metrics

Quality metrics measure how effectively a process delivers defect-free products.
Key indicators include:

- **First Pass Yield (FPY):** Percentage of products made correctly without rework.
- **Defect Rate:** Ratio of defective items to total produced.
- **Rework and Scrap Rate:** Cost and time spent on fixing or discarding poor-quality items.

Formula:

$$QualityRate = \frac{Good\ Units}{Total\ Units\ Produced} \times 100$$

8. Throughput

Throughput measures the rate of production output over time. It reflects the actual production speed compared to designed capacity. Higher throughput indicates smoother workflow and better resource utilization.

Example:
A packaging line processes 5,000 bottles per hour (up from 4,500 last week) after process optimization, showing improved throughput.

9. Maintenance Metrics

Maintenance performance directly impacts productivity. Key indicators include:

- **Mean Time Between Failures (MTBF):** The average time between equipment breakdowns.
- **Mean Time to Repair (MTTR):** The average time needed to repair and restart the machine.
- **Maintenance Cost per Hour:** The total maintenance cost compared to productive machine hours.

Dashboards displaying these values help identify unreliable machines and plan better maintenance schedules.

10. Equipment Utilization Rate

Utilization shows how much of the available production capacity is being used.

$$UtilizationRate = \frac{ActualOperatingTime}{AvailableTime} \times 100$$

Low utilization often indicates idle equipment, poor scheduling, or overcapacity.

11. Inventory and Material Flow Metrics

In smart factories, IIOT dashboards also monitor:

- Inventory levels (raw materials, work-in-progress, and finished goods)
- Material movement using RFID or barcode tracking
- Lead Time – time from order placement to delivery

This ensures smooth supply chain management and reduces storage costs.

12. Safety and Environmental Metrics

IIOT dashboards can track safety-related data such as gas leak sensors, machine guarding status, or emergency stop counts. Environmental parameters like CO_2 emissions, air quality, and energy intensity are also displayed to ensure compliance with sustainability goals.

13. Alarm and Event Summary

Dashboards include alarm histories showing how often and how long specific faults occur. By analyzing this data, engineers can identify repetitive issues and take permanent corrective actions.

14. Productivity Index and Cost Metrics

These metrics relate production output to total operational cost.

- **Production Efficiency:** Ratio of actual output to potential output.
- **Cost per Unit Produced:** Tracks how much cost is incurred to produce a single unit.
- **Return on Equipment (ROE):** Measures profitability of machinery investment

15. Worker and Process Performance

With IIOT-enabled systems, even human efficiency can be monitored through:

- Task completion time
- Error frequency
- Operator-machine interaction metrics

These insights help in training, load balancing, and process improvement.

2.6 Impact on Production Efficiency

IIOT has revolutionized the way industries achieve productivity, quality, and efficiency. By integrating smart sensors, automation systems, and data analytics, IIOT enables real-time visibility and control over production operations, helping industries make faster, more informed decisions and achieve higher output with lower waste.

Earlier, production efficiency largely depended on human supervision, manual data collection, and fixed maintenance schedules. Operators had to inspect machines physically and record data such as temperature, pressure, or speed. This process was not only time-consuming but also prone to human error. Delays in identifying problems often resulted in reduced output, increased downtime, or compromised product quality.

With the introduction of IIOT-enabled systems, factories have become more connected and intelligent. Sensors installed on machines continuously collect and transmit performance data, such as vibration, energy consumption, speed, and load to a centralized monitoring platform. Engineers can instantly identify performance bottlenecks or irregularities from a single dashboard, even across multiple production lines or plants.

A. Real-Time Data for Quick Decision-Making

IIOT provides continuous data flow, allowing production managers to analyze machine performance in real time. If a conveyor slows down, a motor overheats, or an assembly robot works below capacity, the system generates an instant alert. This enables immediate corrective action, preventing downtime and ensuring smooth workflow.

Example: In an automotive assembly plant, IIOT sensors detect a drop in robotic welding speed. The system automatically readjusts the operation sequence and alerts maintenance staff to check the affected robot, preventing a production halt.

B. Predictive Maintenance to Reduce Downtime

IIOT's predictive maintenance feature plays a major role in improving production efficiency. Sensors monitor machine health parameters like vibration and temperature to predict potential failures. Instead of stopping production for scheduled or emergency maintenance, repairs are carried out only when needed during planned downtime.

This reduces both unplanned stoppages and maintenance costs, keeping the production line running longer and more efficiently.

C. Process Optimization and Automation

IIOT platforms use advanced analytics and AI algorithms to optimize process parameters automatically. For instance, if material flow or machine speed needs adjustment, the system can fine-tune it instantly based on sensor feedback.

This ensures uniform production, reduces energy waste, and maintains consistent product quality across batches.

Example: In a bottling plant, flow sensors connected to an IIOT network automatically adjust pump speed to match production demand, minimizing overflow or wastage.

D. Improved Resource Utilization

Through centralized monitoring, industries can track machine utilization, energy consumption, and operator efficiency. This helps managers identify underused equipment, schedule workloads effectively, and cut down idle time. IIOT systems also analyze trends to suggest ways to improve line balancing or redistribute tasks for maximum throughput.

Example: In a multi-line automotive factory, the IIOT dashboard compares productivity across lines. It detects that one line has a higher idle time and automatically reallocates tasks from overloaded machines to underused ones.

E. Enhanced Product Quality and Traceability

Production efficiency is not only about speed, it also depends on consistent quality. IIOT systems record every step of the production process, allowing traceability of materials, machine parameters, and quality checkpoints. Any deviation from standards triggers an alert, reducing rework and scrap rates.

Example: In a tire manufacturing unit, IIOT sensors track cure temperature and pressure. If conditions vary from standard limits, the system adjusts controls automatically, ensuring every tire meets quality specifications.

F. Continuous Improvement through Analytics

IIOT systems store large volumes of historical production data, enabling industries to analyze trends, compare performance, and plan process improvements. Insights from this data can be used to fine-tune production schedules, energy use, and material flow for even greater efficiency over time.

PRACTICAL ACTIVITY

ACTIVITY 1:
Installation and Setup of Node.js and Node-RED

Objective:

- Install Node.js
- Install Node-RED
- Launch Node-RED editor

Equipment/ Software Required:

- Computer/Laptop
- Internet connection
- Web Browser

Installation Procedure:

Step 1. Download and install Node.js

1. Open the official Node.js website: https://nodejs.org/en/download
2. Download the recommended version for Windows by clicking **Windows Installer (.msi)** (Fig.1).

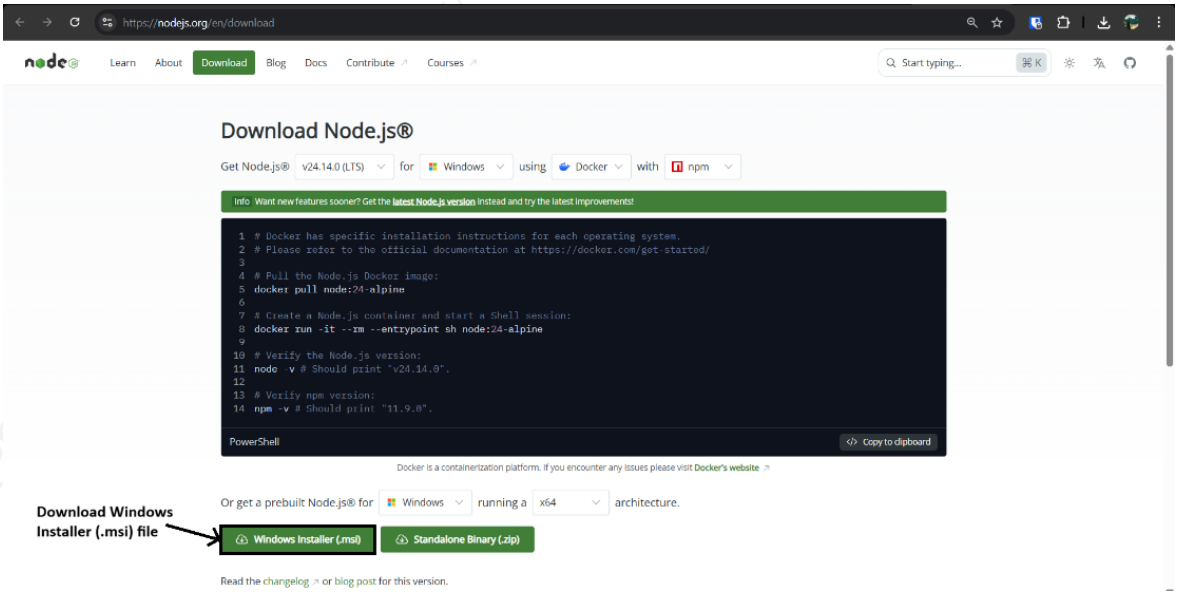


Fig.1: Nodejs Website

3. Run the downloaded installer.

Purpose	Command
Check Node.js version	**node --version**
Check npm version	**npm --version**
Install Node-RED	**npm install -g node-red**

4. Complete the setup using the default options by click on Next at each step and in the last click on Install.

Step 2. Install Node-RED

After Node.js installation is complete, open Command Prompt and run the following command (Fig.2):

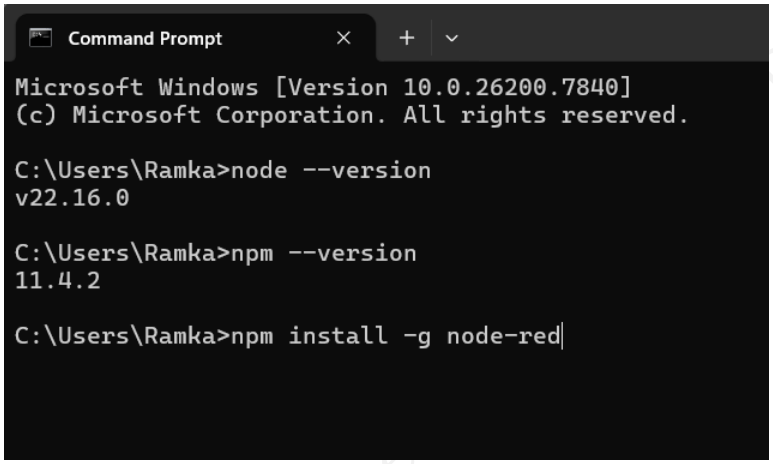


Fig.2: Command Prompt Install Reference image

Step 3. Start Node-RED

5. In Command Prompt, start Node-RED by running the command: **node-red**
6. Wait until the console shows that the server is running.
7. Open your browser and go to: **http://localhost:1880**

Step 4. Install extra nodes in Node-RED

In the Node-RED editor, go to Menu > Manage palette > Install (or press Ctrl + Shift + P to open the palette installer), then search and install the required nodes and click on Install (Fig.3).

- node-red-contrib-mqtt-broker
- node-red-contrib-ui-led
- node-red-dashboard
- node-red-node-random
- node-red-node-serialport

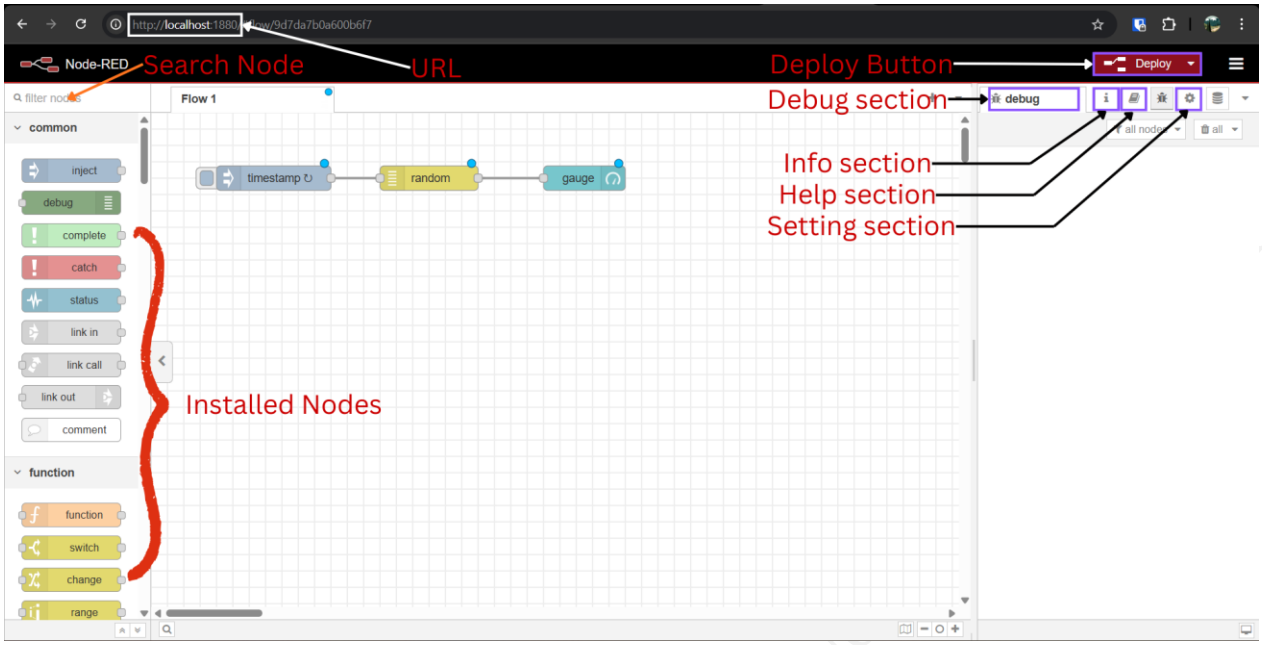


Fig.3: Node-Red Introduction

ACTIVITY 2:
Creating a Basic Flow Using Inject, Function, and Debug Nodes

Objective: Creating a Basic Flow Using Inject, Function, and Debug Nodes

Equipment/Software Required:

- Computer/Laptop
- Internet connection
- Web Browser

Procedure:

Step 1: Access the node-red editor

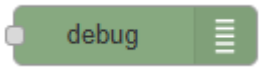
With Node-RED running, open the editor in a web browser by **http://localhost:1880**.

Step 2: Add an Inject node

- The Inject node allows you to inject messages into a flow, either by clicking the button on the node, or setting a time interval between injects.
- Drag one onto the workspace from the palette.
- Select the newly added Inject node to see information about its properties and a description of what it does in the Information sidebar panel.

Step 3: Add a Debug node

The Debug node causes any message to be displayed in the Debug sidebar. By default, it just displays the payload of the message.



Step 4: Wire the two together

Connect the Inject and Debug nodes together by dragging between the output port of one to the input port of the other.

Step 5: Deploy

At this point, the nodes only exist in the editor and must be deployed to the server. Click the Deploy button

Step 6: Inject

With the Debug sidebar tab selected, click the Inject button (the small square button next to inject node). You should see numbers appear in the sidebar (Fig.1 and 2).

Fig.1: Flow1

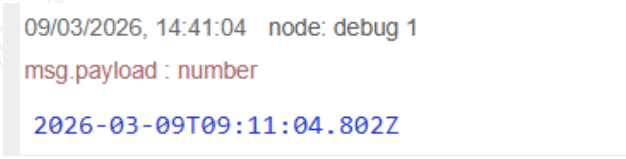


Fig.2: Output of Flow1

By default, the Inject node uses the number of milliseconds since January 1st, 1970 as its payload.

Step 7: Add a Function node

- The Function node allows you to pass each message though a JavaScript function.
- Delete the existing wire (select it and press delete on the keyboard).
- Wire a Function node in between the Inject and Debug nodes.
- Double-click on the Function node to bring up the edit dialog. Copy the following code into the function field:

```
// Create a Date object from the payload

var date = new Date(msg.payload);

// Change the payload to be a formatted Date string

msg.payload = date.toString();

// Return the message so it can be sent on

return msg;
```

Click Done to close the edit dialog and then click the deploy button (Fig.3).

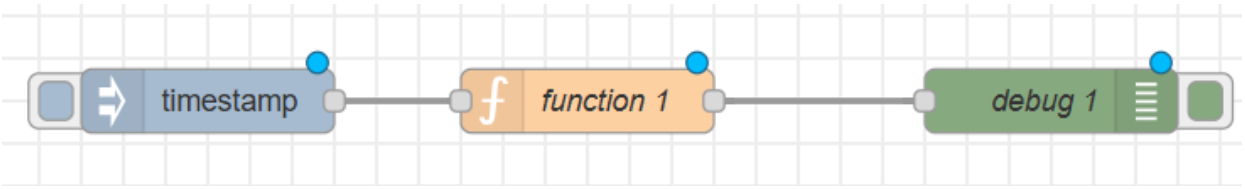


Fig.3: Flow2

Now when you click the Inject button, the messages in the sidebar will now be formatted is readable timestamps (Fig.1.7(ii)).

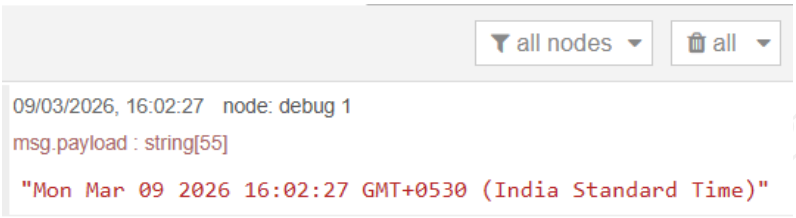


Fig.4: Output of Flow2

ACTIVITY 3:
Exploring Core Node-RED Nodes: Inject, Random, Change, Function, Switch, Delay, and Range.

Objective: Explore key built-in nodes and learn how to use the Node-RED documentation to understand each node (https://nodered.org/docs/user-guide/nodes).

Equipment/Software Required:

- Computer/Laptop (Node-Red installed)
- Internet connection
- Web Browser

Create a single flow named “Core Nodes Tour” with these mini-tests (each must end in Debug):

1. Inject → Debug
2. Inject → Random → Debug
3. Inject → Change → Debug
4. Inject → Function → Debug
5. Inject → Switch → Debug (2 outputs, each to its own Debug)
6. Inject → Delay → Debug
7. Inject → Range → Debug

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. In a factory, operators manually record machine readings, leading to delays and frequent errors. Which solution would best improve this situation?
   a) Increase number of operators
   b) Implement centralized IIOT monitoring system
   c) Reduce number of machines
   d) Stop data recording

2. A production manager wants to monitor machines located in different plants from a single dashboard and take quick decisions. What should be implemented?
   a) Manual inspection system
   b) Separate monitoring systems for each machine
   c) Centralized IIOT control system
   d) Paper-based reporting

3. An engineer wants to connect sensors, process data, and generate alerts using a drag-and-drop interface without complex coding. Which tool should be used?
   a) PLC Ladder Logic only
   b) Node-RED platform
   c) Spreadsheet software
   d) HDMI interface

4. A motor shows a slight increase in vibration, and the system alerts the maintenance team before failure occurs. Which maintenance approach is being applied?
   a) Reactive maintenance
   b) Preventive maintenance
   c) Predictive maintenance
   d) Breakdown maintenance

5. A factory wants to evaluate how efficiently its machines are performing by considering availability, performance, and quality together. Which metric should be used?
   a) Throughput
   b) Cycle Time
   c) Overall Equipment Effectiveness (OEE)
   d) Downtime

B. Match the following

Column A	Column B
1. Centralized IIOT System	A. Drag-and-drop programming tool
2. Node-RED	B. Predicts failures before breakdown
3. Predictive Maintenance	C. Measures equipment efficiency

4. OEE	D. Single platform for monitoring machines
5. Sensors	E. Collect machine data

C. Fill in the blanks

1. Node-RED supports industrial communication protocols such as MQTT, HTTP, and _____.
2. In centralized IIOT systems, engineers can monitor machines remotely using devices like laptops, tablets, or _____.
3. In dashboards, machine status is often represented using color codes where green indicates active and _____ indicates stopped.
4. _____ maintenance may lead to unnecessary replacement of parts even when they are still in good condition.
5. Cycle time refers to the time taken to produce _____ unit from start to finish.

D. Answer the following

1. Different machines in a plant are giving uneven performance outputs. What approach can managers use through IIOT systems to compare and improve their performance?
2. An organization notices unusually high electricity consumption in certain equipment. Explain how connected monitoring systems can help identify the cause and reduce energy usage.
3. Sudden voltage fluctuations are affecting the smooth running of machines on a production line. How can sensor-based monitoring systems be utilized to manage this situation effectively?
4. Workers are exposed to risky conditions while checking machines manually. Suggest how modern monitoring techniques can minimize their direct exposure to such environments.
5. A manufacturing company is experiencing a rise in defective products. How can data collected from machines be used to control and improve product quality?

SESSION 2: REMOTE DATA ACQUISITION ARCHITECTURE

2.7 Introduction

In modern automotive industries, machines, production lines, robots, testing equipment, and utility systems generate a large amount of operational data every second. To improve productivity, quality, safety, and maintenance efficiency, industries need systems that can collect this data remotely and make it available for monitoring and decision-making. This system is known as Remote Data Acquisition Architecture**.**

Remote Data Acquisition Architecture refers to the structured arrangement of sensors, controllers, communication networks, gateways, cloud platforms, and dashboards used to collect machine data from distant locations and deliver it to users in real time.

In an IIOT environment, remote data acquisition helps monitor CNC machines, paint booths, conveyor belts, robotic arms, assembly stations, compressors, power systems, and vehicle testing units without requiring constant manual inspection.

How to know that a robotic welding machine is overheating if the machine is located in another plant hundreds of kilometres away?

2.8 Need for Remote Data Acquisition in Automotive Industry

The automotive industry operates with highly automated production systems where machine downtime can lead to production losses. Remote data acquisition is important because it provides:

- Real-time monitoring of machines and production lines
- Early fault detection and maintenance alerts
- Improved machine utilization
- Better production planning
- Energy consumption monitoring
- Remote supervision of multiple plants
- Historical data for analysis and quality control
- Reduced manual inspection effort

> ***Real-World Example***
>
> *A robotic welding cell reports an abnormal temperature rise through the IIOT system. Technician receive an alert on their dashboard and perform maintenance before production stops.*

2.9 Basic Architecture of Remote Data Acquisition System

A typical remote data acquisition system consists of the following layers (Fig.2.5):

1. Field Layer – Sensors and actuators
2. Control Layer – PLCs and controllers
3. Communication Layer – Network and gateways
4. Processing Layer – Cloud or local server
5. Application Layer – Dashboard, reports, alarms

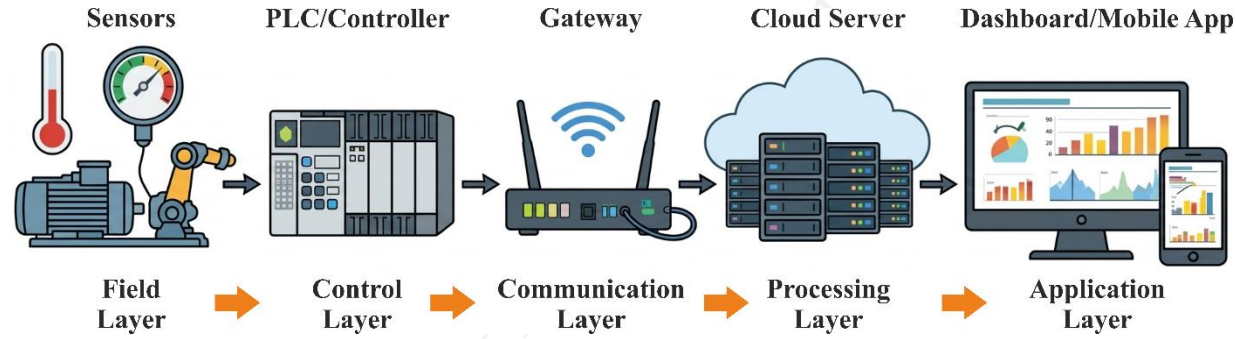


Fig.2.5: Basic Remote Data Acquisition System Architecture

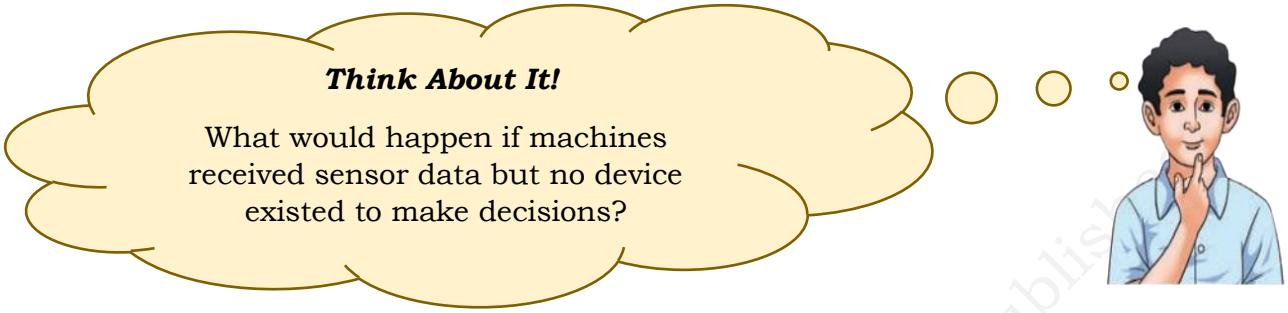
2.10 Main Components of Architecture

A. Sensors: Sensors detect physical parameters from machines and processes. For example, a proximity sensor counts car body movement on conveyor lines. The most common sensors used in automotive plants are:

- Temperature sensor
- Pressure sensor
- Vibration sensor
- Proximity sensor
- Flow sensor
- Current sensor
- Speed encoder

> ***Observe Around You***
>
> *Modern vehicles, elevators, and home appliances also use different types of sensors.*

B. Controllers:



Controllers receive sensor signals and control machine operations. Like a PLC controls conveyor motors and collects sensor inputs. Common controllers are as follows:

- PLC
- Microcontroller
- RTU

C. Edge Devices: Edge devices process data near machines before sending it to cloud systems. Functions of edge devices are:

- Noise filtering
- Data compression
- Fast decision-making
- Temporary storage
- Protocol conversion

D. Gateway: Gateway connects factory devices to enterprise or cloud systems. One gateway may connect 100 machines to cloud monitoring software. A gateway can perform following functions:

- Collect data from many machines
- Convert industrial protocols
- Secure communication
- Internet connectivity

E. Cloud / Server: The cloud stores and processes large amounts of machine data. Popular platforms include Microsoft Azur, Amazon Web Services, and Google Cloud.

It can be used in following tasks:

- Data storage
- Analytics
- Trend reports
- Alarm management
- Predictive maintenance

F. Dashboard

Dashboards present data visually to operators and managers. Some important features of dashboard are:

- Live machine status
- Production count
- Alarm screen
- Energy reports
- OEE charts
- Mobile monitoring

2.11 Sensor-to-Cloud Data Flow

In a modern IIOT system, the ability to send data from sensors on machines all the way to the cloud is what makes remote monitoring and intelligent control possible. This end-to-end process known as the "Sensor-to-Cloud Data Flow", connects the physical and digital worlds, allowing industries to collect, analyze, and act on data in real time from anywhere in the world (Fig.2.6).

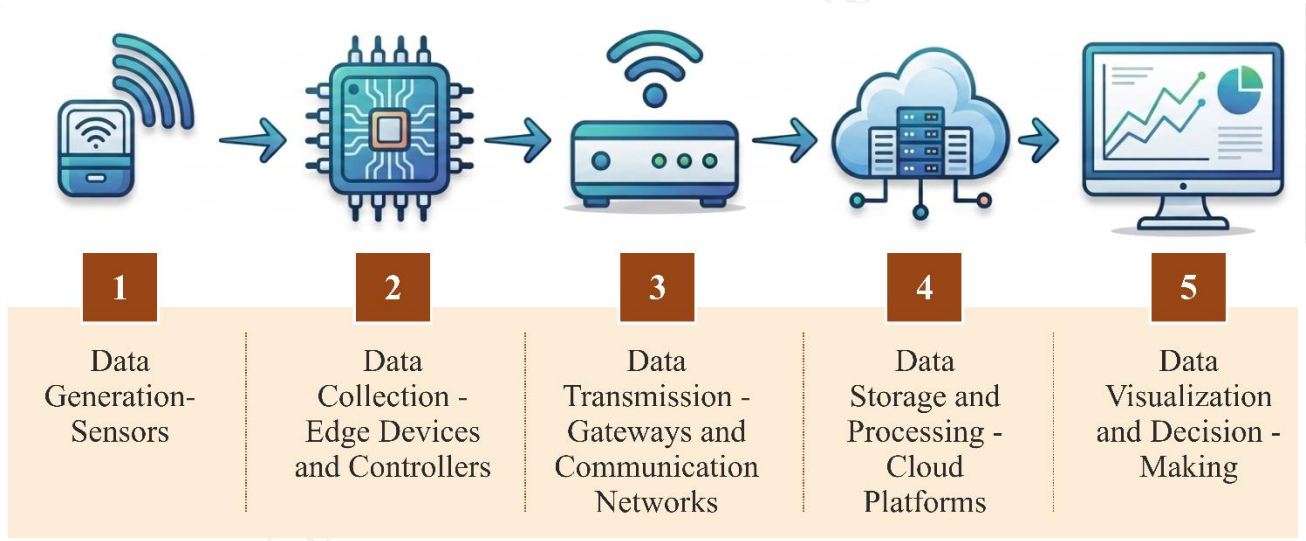


Fig.2.6: Sensor-to-Cloud Data Flow

The step wise flow can be described as follows:

A. Data Generation – Sensors

The process begins at the sensor level. Sensors are the primary data sources in an IIOT environment. They continuously measure physical parameters such as temperature, pressure, humidity, vibration, speed, or voltage from machines or environments.

These sensors convert real-world signals (analog or digital) into electrical data that can be processed by electronic systems.

Example: A vibration sensor on a motor generates data about its mechanical movement, while a temperature sensor measures heat levels inside the motor housing.

B. Data Collection – Edge Devices and Controllers

The raw data from multiple sensors is first collected by edge devices, microcontrollers (like Arduino or NodeMCU), or PLCs (Programmable Logic Controllers).

These devices perform preprocessing such as filtering noise, averaging values, or converting analog data into digital forms to make the data suitable for transmission.

Edge devices may also make quick decisions locally (called edge computing) without waiting for cloud instructions. This reduces latency and allows faster responses.

Example: An edge device can instantly stop a conveyor if a proximity sensor detects an obstacle, even before the cloud receives the data.

C. Data Transmission – Gateways and Communication Networks

Think About It!

How does machine data travel from a factory floor to cloud servers located far away?

Once collected, the processed data is sent to the internet or industrial network using IIOT gateways. A gateway acts as a bridge between local devices and the cloud, handling data formatting, security, and protocol conversion. Common communication technologies include:

Wired: Ethernet, Modbus TCP, PROFINET

Wireless: Wi-Fi, LoRaWAN, Bluetooth, Zigbee, and 5G

Gateways ensure reliable data transfer by packaging sensor information into standard formats and transmitting it securely to cloud servers.

Example: A factory's IIOT gateway may collect data from hundreds of sensors and send it to a cloud dashboard every few seconds via an encrypted internet connection.

D. Data Storage and Processing – Cloud Platforms

Once data reaches the cloud, it is stored in large databases and analyzed using AI, Machine Learning (ML), and analytics tools. The cloud processes huge volumes of sensor data to identify trends, detect faults, and generate insights. Visualization dashboards allow users to see key performance indicators (KPIs), graphs, and alerts in real time. Popular cloud platforms used in IIOT include Microsoft Azure IoT, AWS IoT Core, Google Cloud IoT, and Siemens MindSphere.

Example: Temperature and energy consumption data from multiple machines are analyzed in the cloud to detect abnormal behavior, enabling predictive maintenance planning.

E. Data Visualization and Decision-Making

Finally, the processed information is displayed to users through web dashboards, mobile apps, or SCADA systems. Engineers, managers, or operators can monitor machine health, production status, and alerts in real time, even from remote locations. This enables data-driven decision-making, improving efficiency, safety, and reliability.

Example: If the cloud dashboard shows rising vibration in a pump, maintenance engineers can plan an inspection before a breakdown occurs.

Table 2.1 shows different stages involved in Sensor-to-Cloud Data Flow.

Table 2.1: Stages in Sensor-to-Cloud Data Flow

Stage	Function	Example Device/Technology
1. Sensing	Measures physical parameters	Temperature, Vibration, Pressure sensors
2. Collection	Converts and preprocesses data	Arduino, PLC, Edge Controller
3. Transmission	Sends data to the internet/cloud	IIOT Gateway, Wi-Fi, Ethernet, LoRa
4. Cloud Processing	Stores, analyzes, and visualizes data	AWS IoT, Azure IoT, Google Cloud
5. Action	Displays alerts and enables control	Dashboard, SCADA, Mobile App

PRACTICAL ACTIVITY

ACTIVITY 1:
Reading ESP32 Sensor Data in Node-RED (Serial Communication)

Introduction to the Activity

In the previous activity (completed in Grade 11), students interfaced a DHT11 temperature and humidity sensor with an ESP32 microcontroller and observed the sensor readings through the Arduino IDE Serial Monitor.

Sample Output:

Temperature: 26.90 °C | Humidity: 41.50 %

While the Serial Monitor is useful for basic testing and verification, Industrial IoT (IIoT) applications require sensor data to be collected, processed, visualized, and shared through dedicated platforms.

In this activity, students will build upon their previous experiment by integrating the ESP32 with **Node-RED**, a popular flow-based programming tool used in IoT and automation systems.

Instead of viewing the sensor readings in the Arduino IDE Serial Monitor, students will:

1. Receive sensor data from the ESP32 using the **Node-RED Serial In** node.
2. Monitor the incoming sensor data within the Node-RED environment.
3. Transform the raw text data into a structured **JSON** format.
4. Prepare the data for visualization, analysis, storage, and further IoT applications.

This workflow reflects a typical Industrial IoT architecture, where data generated by field devices is collected at the edge, processed by platforms such as Node-RED, and then forwarded to dashboards, databases, or cloud services for monitoring and decision-making.

Objective:

• Understand how Node-RED communicates with hardware devices
• Use the Serial In node in Node-RED
• Read raw sensor data from ESP32
• Convert text data into structured data
• Use a Function node to process sensor data
• Visualize data in Node-RED debug panel

Prerequisites:

Students should have completed the previous activity:

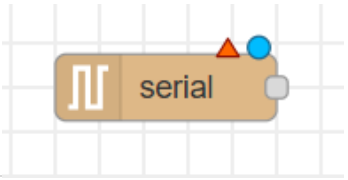
ESP32 + DHT11 Sensor Activity
The ESP32 must already be running the Arduino program that outputs:

Temperature: XX.X °C | Humidity: XX.X %

Equipment/Software Required:

- Computer/Laptop (Node-Red installed)
- Internet connection
- Web Browser

Procedure: Understanding Node-RED Serial Communication
Node-RED can communicate with microcontrollers using **Serial Ports**. The **Serial In node** reads data from devices connected via USB or serial communication.



Example Communication Flow

ESP32 → USB Serial → Node-RED Serial In Node → Processing → Debug

Step 1 – Start Node-RED

1. Open Node-RED in your browser: http://localhost:1880
2. Create a new flow.

Step 2 – Add Serial In Node

From the **palette Section**, drag the following node in workspace: **Serial In**
This node reads data from the ESP32 serial port.

Step 3 – Find Serial Port

How to Find the ESP32 COM Port / Serial Port

Before configuring the Node-RED Serial In node, we must identify the serial port used by the ESP32.
When the ESP32 is connected to the computer using a USB cable, the operating system assigns it a COM port (Windows) or serial device (Linux/Mac).

Method 1 – Using Arduino IDE (Recommended)

1. Connect the ESP32 to your computer using a USB cable.
2. Open Arduino IDE.
3. Navigate to: Tools → Port
   You will see a list of available serial ports.
   Example:COM3, COM5, COM7
4. Select the port that shows ESP32 Dev Module or the port that appears after connecting the ESP32.
   Example: COM7 (ESP32 Dev Module)
   This is the serial port used in Node-RED.

Method 2 – Using Device Manager (Windows)

1. Right-click the Start Menu.
2. Open: Device Manager
3. Expand the section: Ports (COM & LPT) and you should see something like: USB-SERIAL CH340 (COM7) Or Silicon Labs CP210x USB to UART Bridge (COM7)
4. The number in brackets is your COM port.
   Example: COM7

Step 4 – Configure Serial In Port

1. Double click the **Serial In node**.
2. Click on **Edit Serial port config node** for more settings
3. Configure the settings as:

Setting	Value
Serial Port	**Serial Port based on step 3**

Baud Rate	115200
Data Bits	8
Parity	None
Stop Bits	1

Input settings:

Split input: \n

Deliver: ASCII strings

This means Node-RED will read each line sent by ESP32.

Step 5 – Connect Debug Node

1. Drag a Debug node.
2. Connect it like this:
   Serial In node → Debug
3. Deploy the flow (Fig.1).

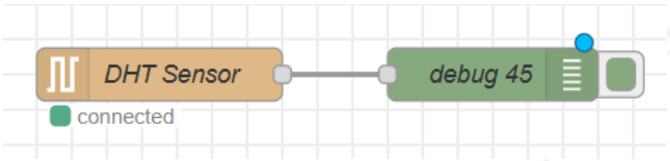


Fig.1: Connect Debug Node

Step 6 – Observe Raw Sensor Data

- Once the ESP32 starts sending data, the Debug panel will show:
- **Temperature: 27.10 °C | Humidity: 40.30 %**
- This is the raw data received from the ESP32.
- At this stage, the data is still a string and not structured.

Step 7 – Convert Data Using Function Node

- To extract the temperature and humidity values, we use a **Function Node** (Fig.2)
- Drag a **Function node** between the **Serial In** and **Debug** nodes.
- Flow structure: **Serial In → Function → Debug**

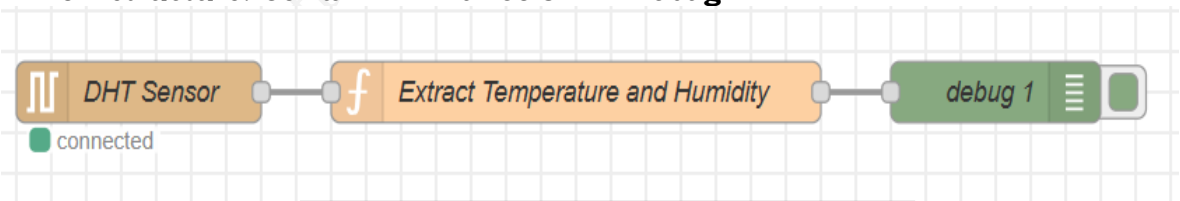


Fig.2: Convert Data Using Function Node

Step 8 – Add Data Processing Code in Function Node

Open the **Function Node** and paste the following code:

```
let data = msg.payload;

let temp = data.match(/Temperature:\s*([0-9.]+)/);
let hum = data.match(/Humidity:\s*([0-9.]+)/);

msg.payload = {
    temperature: parseFloat(temp[1]),
    humidity: parseFloat(hum[1])
};
```

Understanding the Code:

Extract Temperature

data.match(/Temperature:\s*([0-9.]+)/)
This searches the text and extracts the temperature value.

Extract Humidity

data.match(/Humidity:\s*([0-9.]+)/)
This extracts the humidity value.

Convert to JSON Format:

The output becomes as shown:

```
{
  "temperature": 26.9,
  "humidity": 41.5
}
```

This structured format is easier to use for dashboards, databases, and cloud systems.

Step 9 – Deploy the Flow

Click **Deploy**.

- Observe the Debug panel again.
- Now the output will appear as (Fig.3):

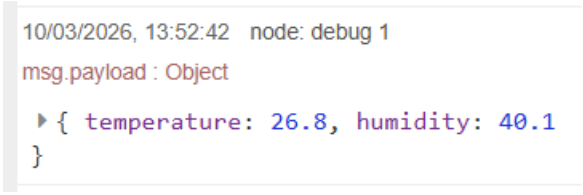


Fig.3: Flow Output

Let's Try this!

Modify the Function node to also include timestamp.

Example output:

```
{
 temperature:                    26.9,
 humidity:                       41.5,
 timestamp:          "2026-03-10T12:55:15"
```

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. A factory installs sensors to monitor machine temperature, but the system needs immediate action when overheating occurs without waiting for cloud response. What should be implemented?
   a) Cloud-only processing
   b) Edge computing at controller level
   c) Manual monitoring
   d) Data storage system

2. An industry wants to securely send data from multiple machines to a cloud platform using different communication protocols. Which component should be used to manage this process?
   a) Sensor
   b) Dashboard
   c) IIOT Gateway
   d) Mobile App

3. A supervisor wants to quickly identify critical machine issues using visual indicators instead of reading raw data tables. Which dashboard feature should be applied?
   a) Detailed spreadsheets
   b) Text-based reports
   c) Color-coded visual elements
   d) Manual logs

4. A company needs different dashboard views for operators, supervisors, and managers based on their roles. What design principle should be applied?
   a) Standardization
   b) User-focused design
   c) Data storage
   d) Protocol conversion

5. An engineer notices that sensor data displayed on the dashboard is outdated, leading to wrong decisions. Which feature should be ensured to solve this issue?
   a) Static data display
   b) Auto-refresh and real-time updates
   c) Manual data entry
   d) Increased storage capacity

B. Match the following

Column A	Column B
1. Edge Device	A. Converts protocols and connects to cloud
2. IIOT Gateway	B. Processes data locally for quick response
3. Cloud Platform	C. Stores and analyzes large data
4. Dashboard	D. Displays information to users
5. Communication Network	E. Transfers data between devices

C. Fill in the blanks

1. Edge computing helps in reducing _____ by processing data locally.
2. A gateway ensures secure and reliable data _____ between devices and the cloud.
3. Dashboards should include _____ to show when the data was last updated.
4. Color coding such as green, yellow, and red is used to indicate _____ levels in dashboards.
5. Sensors convert physical parameters into _____ signals for processing.

D. Answer the following

1. Sensor readings from a machine contain random fluctuations that affect analysis. What steps can be taken at the controller level to make this data reliable before sending it forward?
2. An industry uses both Ethernet-based machines and wireless devices in the same system. How can the data from these different sources be effectively integrated and transmitted to the cloud?
3. A supervisor wants to study production patterns over time to detect inefficiencies. In what way can cloud platforms support this requirement?
4. Users complain that important warnings are getting lost among too much information on the screen. How can dashboard organization be improved so that critical data stands out clearly?
5. A monitoring system must work smoothly on both mobile phones and large control room screens. What design approach should be used to ensure proper display across devices?

SESSION 3: DASHBOARDS AND DATA VISUALIZATION

2.12 Design Principles for Dashboards

In the IIOT environment, dashboards serve as the visual control center of a smart factory. They display key data collected from sensors, machines, and cloud platforms in a clear, easy-to-understand format. A well-designed dashboard helps engineers, operators, and managers monitor performance, identify issues, and make informed decisions in real time (Fig.2.7).

Fig.2.7: Dashboards Example

Designing an effective dashboard requires careful planning and understanding of user needs. A cluttered or poorly organized dashboard can cause confusion and delay decision-making. Therefore, applying good design principles ensures that data is not only visible but also meaningful and actionable. These are discussed as follows:

❖ Clarity and Simplicity

The most important rule in dashboard design is clarity. The purpose of a dashboard is to present complex data in a simple and easy-to-read format.

- Use clear labels, units, and color coding.
- Avoid unnecessary graphics or decorations.
- Present only relevant information that supports quick understanding.

Example: Instead of showing all machine parameters at once, display only critical metrics such as temperature, speed, and energy consumption in the main view, with detailed data available on click.

❖ **Hierarchy and Layout**

Organize information logically. Place the most important metrics such as alarms, machine status, or key performance indicators (KPIs) at the top or center of the screen where users naturally look first.

- Use consistent alignment and spacing.
- Group related data together (e.g., temperature, vibration, and current for one machine).
- Use panels or sections for different production areas or machines.

❖ **Use of Visual Elements**

Dashboards should rely on visual elements rather than long tables or text. Humans process visuals faster than numbers.

- Use charts (bar, line, pie, or gauge) to show trends.
- Use icons and color indicators (green = normal, yellow = warning, red = critical).
- Use maps or plant layouts for location-based data visualization.
- Include real-time animations where necessary, such as motor status or energy flow.

Example: A temperature gauge that changes color as temperature rises provides instant insight compared to reading numerical data.

☞Industrial dashboards use gauges similar to speedometers and fuel meters seen in cars because circular indicators are easier to understand quickly.

❖ **Real-Time Data and Interactivity**

IIOT dashboards should display real-time updates to help users respond quickly to changes on the factory floor.

- Enable auto-refresh for live data.
- Include interactive elements such as filters, drop-down menus, and zoom-in graphs.
- Allow users to click on a device or alert them for more detailed information.

Example: An operator clicks on a red alert for a robotic arm to view its vibration trend graph and suggested maintenance schedule.

❖ **Consistency and Standardization**

Maintain a consistent visual style across all dashboard screens. This includes using the same colors, icons, and terminology for similar data types.

- Define standard units (°C, RPM, kW).
- Use uniform date/time formats and graph styles.
- Apply the same alert levels and thresholds across all machines.

Consistency improves readability and reduces training time for new users.

❖ **Focus on User Needs**

Different users require different levels of information:

- Operators need live machine status and alerts.
- Supervisors need performance summaries and downtime reports.
- Managers need production trends and cost analysis.

Design dashboards that fit the needs of each user type. Avoid overloading everyone with the same data.

Example: Create separate dashboards, one for maintenance (machine health), one for production (output rate), and one for management (overall efficiency).

❖ **Data Accuracy and Reliability**

A dashboard is only as useful as the data it displays. Always ensure that the data is accurate, updated, and synchronized across systems.

- Implement data validation checks.
- Highlight any communication errors or missing sensor data.
- Display the timestamp of the latest update.

Example: A small indicator showing "Data last updated: 10:32 AM" helps users verify that information is current.

❖ **Responsive and Scalable Design**

Modern dashboards should work seamlessly across different devices like desktop computers, tablets, or smartphones.

- Use responsive design to automatically adjust layout and charts to screen size.
- Ensure performance remains smooth even when handling large volumes of data.

Example: Supervisors can view machine status on their mobile phones while walking through the plant.

❖ **Alerts and Notifications**

Effective dashboards not only show data but also notify users about critical events.

- Set up visual and audio alerts for high-priority issues.
- Use color-coded alarm indicators and blinking icons for faults.
- Allow alerts to be sent via email, SMS, or mobile app for remote access.

❖ **Continuous Improvement**

Dashboards should evolve as industrial needs change. Regular feedback from users helps improve layout, features, and data presentation.

- Review user behavior and update dashboard design accordingly.
- Add new visualizations as new sensors or machines are added to the system.

2.13 Real-Time and Historical Data Analysis

In an IIOT environment, data is continuously generated by sensors, machines, and control systems. This data can be viewed and analyzed in two major ways, real-time analysis and historical analysis. Both are essential for effective decision-making, process optimization, and predictive maintenance in smart factories.

2.13.1 Real-Time Data Analysis

Real-time data analysis focuses on monitoring and interpreting data the moment it is produced. In modern industries, sensors continuously send live readings of parameters such as temperature, vibration, pressure, speed, humidity, and energy consumption to IIOT dashboards. This allows engineers and operators to observe system performance instantly and take immediate action if anything goes wrong.

a) Objective of Real-Time Data Analysis

The purpose of real-time data analysis is;

- To detect and respond to abnormal situations immediately
- To ensure safety, efficiency, and smooth production
- To support instant decision-making during operations

Key Characteristics of real-time data analysis can be summarized as:

- Data is updated every second or even in milliseconds.
- Dashboards use live gauges, color-coded indicators, and alerts.
- Decision-making happens on the spot — often automatically.

For example,

- In an automotive assembly plant, robotic arms are monitored using real-time data. If a robot's torque sensor shows abnormal resistance, the IIOT dashboard instantly highlights the fault in red. The system may automatically slow down the line or alert maintenance teams before any damage occurs.
- In a power plant, if a turbine's temperature exceeds its safety threshold, a real-time alert can trigger an immediate cooling action or emergency shutdown.

b) Benefits of Real-Time Data Analysis

Real-time analysis ensures that industrial operations remain proactive instead of reactive, improving both reliability and performance.

- Quick fault detection and response
- Reduced machine downtime
- Improved process control
- Enhanced workplace safety
- Better quality assurance

c) Visualization Examples (refer Fig.2.8)

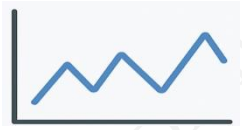
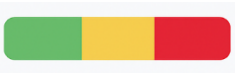
	Live gauges showing motor RPM or temperature
	Real-time energy consumption graphs
	Blinking alarm icons for system faults
	Machine health color indicators(green/yellow/red)

Fig.2.8: Visualization Examples- Real-Time Data Analysis

2.13.2 Historical Data Analysis

While real-time analysis focuses on what is happening now, historical data analysis examines what has happened over time. Data collected from sensors, controllers, and production systems is stored in databases or cloud servers. This stored data is later analyzed to identify long-term trends, performance patterns, and recurring issues.

a) Objective of Historical Data Analysis

- To evaluate past performance and efficiency
- To detect patterns leading to failures
- To plan predictive maintenance
- To improve production planning and resource utilization

Key characteristics of historical data analysis are as follows:

- Data is stored in databases, servers, or cloud systems.
- Analysis is performed periodically (daily, weekly, monthly).
- Focus on understanding performance over time.

For example,

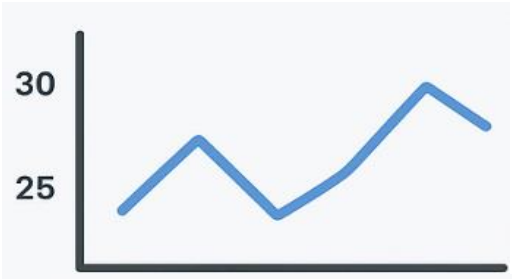
- In a vehicle testing facility, temperature and vibration data from test rigs are recorded over several months. Historical analysis reveals that a specific batch of motors consistently shows vibration peaks after 300 hours of operation, indicating a design issue.
- In a food processing plant, analyzing historical humidity and temperature data helps identify why certain batches spoiled faster, leading to process improvements.
- In a vehicle testing facility, temperature and vibration data from test rigs are recorded over several months. Historical analysis reveals that a specific batch of motors consistently shows vibration peaks after 300 hours of operation — indicating a design issue.
- In a food processing plant, analyzing historical humidity and temperature data helps identify why certain batches spoiled faster, leading to process improvements.

b) Benefits of Historical Data Analysis

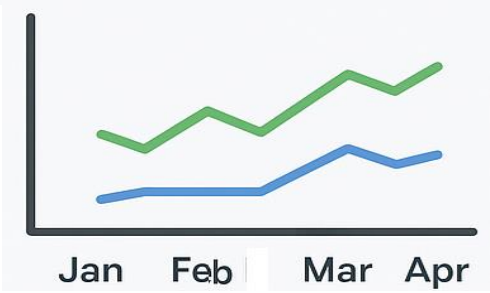
Historical analysis transforms raw data into actionable insights, supporting continuous improvement and innovation.

- Supports data-driven decision-making
- Helps identify hidden inefficiencies
- Enables predictive maintenance and future planning
- Provides evidence for audits and reports

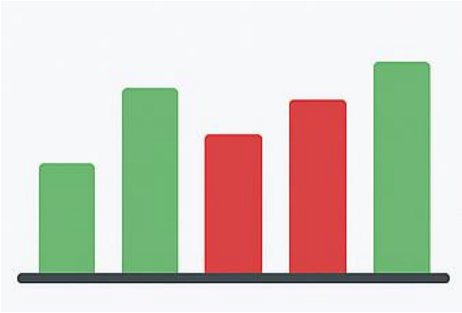
c) Visualization Examples (refer Fig.2.9)



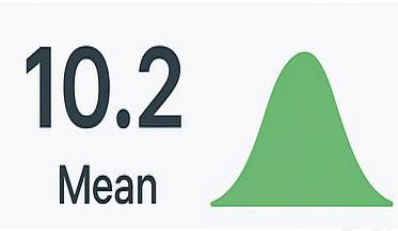
Trend charts showing temperature overtime



Comparative graphs of machine performance across months



Production volume and defect rate reports



Statistical dashboards showing mean, peak, and deviation data

Fig.2.9: Visualization Examples- Historical Data Analysis

2.13.3 Integration of Real-Time and Historical Analysis

The most effective IIOT dashboards combine both types of analysis for complete visibility and control.

- Real-time data helps operators monitor and act immediately.
- Historical data helps engineers and managers analyze and plan strategically.

One example of integration is a car manufacturing plant, where;

- The dashboard shows the current temperature of a welding robot (real-time).

- It also displays a trend graph comparing this data over the last 30 days (historical). If the temperature trend shows a gradual increase, it predicts that the robot may need maintenance soon.

By integrating both data types, the factory achieves instant awareness and long-term intelligence.

2.14 Technical Architecture Behind Real-Time and Historical Analysis

The IIOT system architecture supporting these analyses typically includes:

1. **Sensors and Edge Devices** – collect live data from machines.
2. **Gateways** – filter, aggregate, and transmit data to servers or the cloud.
3. **Cloud Databases or Data Historians** – store long-term data for historical analysis.
4. **Analytics Software or Dashboards** – visualize both real-time and historical data.

Protocols such as MQTT, Modbus, OPC UA, and HTTP enable continuous and reliable communication between these layers.

Example Data Flow:
Sensors → Gateway → Cloud Platform → Dashboard → Analytics → Decision Action (Fig.2.10)

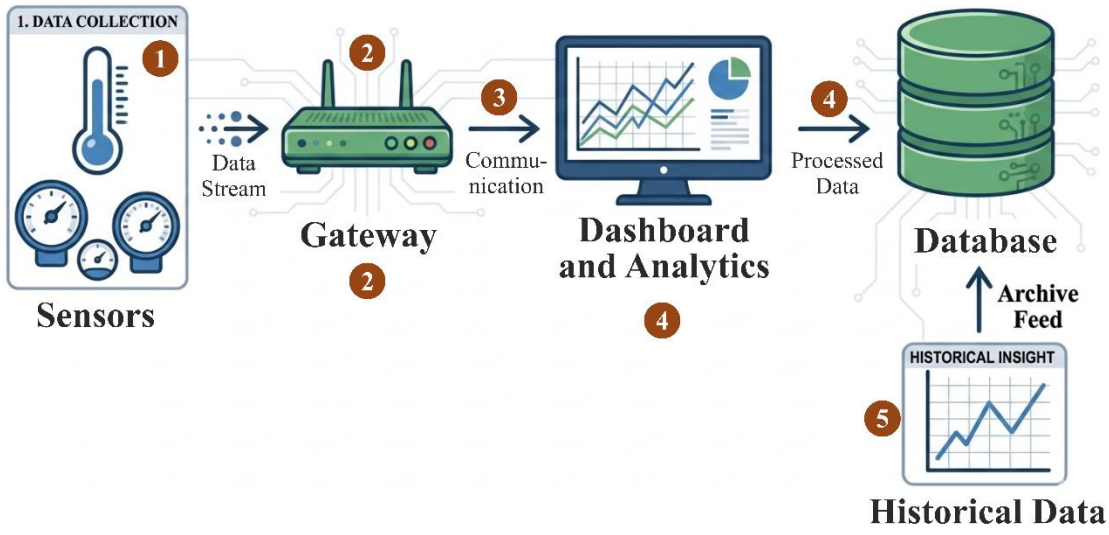
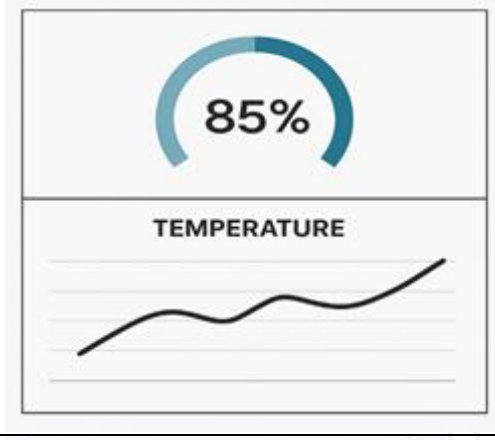


Fig.2.10: Integration of Real-Time and Historical Analysis

2.15 Visualization on Dashboards

Dashboards use different visualization methods for both data types. Table 2.2 shows comparison of both visualizations.

Table 2.2: Visualization on Dashboards of Real-Time and Historical Analysis

Type of Data	Typical Visuals	Visualization on Dashboards	Purpose
Real-Time Data	Gauges, live graphs, digital counters, alerts		Monitor ongoing processes instantly
Historical Data	Trend lines, bar charts, comparison graphs, heat maps		Identify long-term patterns and performance changes

Dashboards may also include filters that allow users to switch between live view and historical trends; for example, viewing "Last 10 minutes," "Last 24 hours," or "Last 30 days."

2.16 Benefits of Combining Real-Time and Historical Data

When industries use both real-time and historical data together, they gain:

- Comprehensive visibility: Full understanding of both current and past system behavior.
- Informed decisions: Immediate data for quick fixes and trend data for long-term planning.
- Reduced downtime: Early fault detection through predictive insights.
- Increased efficiency: Data-backed adjustments to speed, load, and scheduling.
- Continuous improvement: Learning from historical data to refine processes.

2.17 Role in Predictive Analytics and Decision-Making

The integration of real-time and historical data forms the foundation for predictive analytics, where machine learning algorithms analyze both past and live data to predict future events.

For example, a predictive model may analyze vibration history and real-time temperature data from a gearbox to forecast when it will likely fail, allowing timely maintenance. This approach supports data-driven decision-making, where insights from past trends and present conditions guide optimized industrial strategies.

2.18 Sensors for Hazard Detection

Modern industrial environments demand continuous monitoring and rapid detection of potential hazards to ensure worker safety, machine protection, and operational efficiency. Among the various technologies used, Flow Sensors, LiDAR Sensors, and Proximity Sensors play a vital role in identifying abnormal conditions that could lead to accidents or equipment damage. These sensors are widely integrated into IIOT systems to provide real-time hazard detection, automatic control actions, and data-driven maintenance insights.

When combined with real-time dashboards and visual analytics, these sensors help operators monitor risk zones, detect anomalies, and take immediate corrective action.

A. Flow Sensor

Flow sensors measure the rate at which liquids or gases move through pipelines or systems. In industrial plants, monitoring flow is vital for safety, especially in systems carrying fuel, chemicals, or compressed gases (Fig.2.11). Any sudden increase or drop in the flow rate can indicate a leak, blockage, or pump malfunction — all of which pose potential hazards.

Fig.2.11: Flow Sensor

Working Principle: A flow sensor converts the movement of fluid into electrical signals, which are then transmitted to a control unit or cloud dashboard. Dashboards visualize this data using real-time flow graphs, gauges, or alarms.

Role in Hazard Detection:

- Detect fuel leaks in automotive manufacturing lines to prevent fire hazards.
- Monitor coolant flow in robotic welding systems for stable operation.
- Measure air pressure in pneumatic systems to ensure proper machine function.
- Identify abnormal pressure drops indicating possible burst or cracked pipes.
- Prevent overheating or fire risks caused by insufficient coolant circulation.
- Trigger automatic shut-off valves when unsafe flow or pressure is detected.

Visualization on Dashboards:

- Real-time flow rate graphs with alarm indicators for abnormal readings.
- Color-coded alerts (green for normal, yellow for warning, red for critical).
- Comparison charts showing flow rate trends over time for predictive analysis.

B. LiDAR Sensor

LiDAR (Light Detection and Ranging) sensors are advanced devices that use laser beams to measure distance and detect surrounding objects with high precision. They are widely used in autonomous vehicles, smart factories, and warehouse automation for obstacle detection and mapping (Fig.2.12).

Fig.2.12: LiDAR Sensor

Working Principle: LiDAR sensors emit rapid laser pulses and measure the time it takes for the light to bounce back from an object. The collected data is processed to create a 3D map of the environment, enabling accurate hazard detection.

Role in Hazard Detection:

- Detecting obstacles or humans in robot movement paths.
- Mapping factory layouts for automated guided vehicles (AGVs).
- Monitoring safety perimeters in high-risk areas (e.g., around conveyor belts or cranes).

Visualization on Dashboards:

- Real-time 3D visual maps showing obstacles or restricted zones.
- Motion tracking overlays display moving objects in real time.
- Heat maps identify frequently obstructed or high-risk areas.

☞ LiDAR technology is also used in self-driving cars, drones, and modern surveying systems.

C. Proximity Sensor

Proximity sensor detect the presence or absence of nearby objects without physical contact. They are essential in preventing collisions, ensuring safe machine operations, and maintaining correct assembly line functioning (Fig.2.13).

Fig.2.13: Proximity Sensor

Working Principle: These sensors emit electromagnetic fields or infrared signals. When an object enters the detection range, the sensor sends a signal to the controller or dashboard, triggering an alert or stopping a machine automatically.

Role in Hazard Detection:

- Detect human presence near moving machinery to trigger automatic safety shutdowns.
- Ensure correct placement of objects on conveyor belts during automated operations.
- Prevent collisions between robotic arms and other equipment in automotive assembly lines.
- Identify unauthorized entry into restricted or hazardous areas.
- Prevent machine-to-machine collisions by detecting nearby equipment movement.
- Verify safe positioning of mechanical components before initiating any process.
- Activate emergency stops instantly if an object or person comes too close to operating machinery.

Visualization on Dashboards:

- Indicators showing active/inactive zones or detected objects.
- Real-time alerts or blinking icons for hazard proximity.
- Statistical reports showing frequency of detection or near-miss events.

2.18.1 Integration in IIOT-Based Hazard Monitoring

In an IIOT system, sensors such as flow sensors, proximity sensors, and LiDAR sensors continuously send real-time data to a central controller or cloud platform. This integrated data helps industries monitor equipment conditions, worker safety, and machine movement from a single platform. By applying analytics and machine learning techniques, the system can detect abnormal patterns and provide early warnings before hazards develop.

Examples of Hazard Identification

- A sudden reduction in flow rate with increasing temperature may indicate coolant leakage or pipeline blockage.
- Repeated collision signals from a LiDAR sensor may suggest poor AGV path alignment.
- Frequent obstruction signals from proximity sensors may indicate unsafe movement near machines or improper material placement.

2.18.2 Dashboard-Based Safety Monitoring

When integrated into an IIOT dashboard, data from flow, LiDAR, and proximity sensors offers a comprehensive safety monitoring system. Dashboards combine their readings to provide:

- Real-time monitoring: Continuous updates about flow irregularities, object detection, or obstructions.
- Predictive alerts: Identifying early warning signs from historical data trends.

- Visual analytics: Graphs, 3D maps, and color-coded alerts that help operators quickly interpret complex data.
- Automated response: Triggering alarms or stopping machinery when dangerous conditions are detected.

For example, in an automotive plant,

- Flow sensors supervise coolant circulation in machining and welding systems.
- Proximity sensors protect workers near robotic arms and conveyors.
- LiDAR sensors detect obstacles on Automated Guided Vehicle (AGV) routes.

All sensor data appears on a central dashboard, where color-coded indicators show system health and potential hazards. If an anomaly occurs, the system automatically sends an alert to the operator's screen or mobile device (Table 2.3).

Table 2.3: Integration of Sensors in IIOT-Based Hazard Monitoring

Sensor Type	**Measured Parameter**	**Visualization on Dashboard**	**Purpose in Hazard Detection**
Flow Sensor	Liquid/Gas flow rate	Live flow meter, gauge, and alerts	Detect leaks or blockages
LiDAR Sensor	Distance and mapping	3D view or heat map	Detect obstacles and unsafe zones
Proximity Sensor	Object presence	Blinking icon or alarm	Prevent collisions and injuries

2.19 Temperature and Pressure Sensors for Firefighting Systems

In modern industrial environments, fire safety systems have evolved from basic smoke detectors to smart, sensor-based firefighting networks. Among these, temperature and pressure sensors play a crucial role in detecting fire hazards early and ensuring the automatic operation of suppression systems.

Role of Temperature Sensors

Temperature sensors act as the "early warning" components of a firefighting system. They continuously monitor heat levels in different sections of a plant such as engine bays, welding zones, transformer rooms, or paint booths. When the ambient temperature begins to rise beyond a safe threshold, these sensors send real-time data to the central IIOT dashboard.

- If the rise is gradual, it may indicate equipment overheating or poor ventilation.
- If it's sudden, the system can identify it as a potential fire outbreak.

The dashboard immediately displays a visual alert, typically a red indicator or flashing icon; and triggers an alarm. In fully automated systems, the same data can activate sprinklers, gas suppression units, or emergency fans without human intervention.

Some modern temperature sensors also include infrared (IR) or thermocouple-based detection for highly accurate readings in high-temperature environments like foundries and automotive paint shops.

Role of Pressure Sensors

Pressure sensors ensure that firefighting systems are always operational and ready to respond. They monitor:

- The pressure levels in firewater pipelines, CO_2 tanks, and foam systems.
- The flow rate during discharge to confirm that extinguishing agents are reaching the target area.

A drop in the pressure is instantly reflected on the dashboard as a low-pressure warning, alerting maintenance teams before the system fails. In some advanced setups, pressure data is automatically logged, compared with historical readings, and analyzed to detect slow leaks or blockages in the system.

2.19.1 Integration with IIOT Dashboards

In an IIOT-enabled firefighting setup, data from temperature and pressure sensors is transmitted through a communication network (like Modbus, MQTT, or OPC UA) to a central dashboard. This dashboard presents live information using:

- Color-coded gauges for temperature and pressure levels.
- Animated alerts for high-temperature or low-pressure events.
- Graphs and trend lines show variations over time.
- Event logs with timestamps for post-incident analysis.

For operators, this dashboard serves as a command center, allowing them to monitor multiple locations simultaneously, even from remote devices like tablets or smartphones.

2.19.2 Benefits of Data Monitoring

- Real-Time Data: Supports immediate detection and quick response to fire hazards, reducing damage and downtime.
- Historical Data: Helps engineers study repeated temperature rise, pressure drops, or delayed response events for maintenance planning.
- Predictive Analytics: Uses past and present data to forecast low pressure conditions or identify zones vulnerable to overheating.

For example, in an automotive manufacturing plant, temperature sensors are installed near paint booths, battery testing areas, and storage units for flammable materials. Pressure sensors are fitted within the water and foam pipelines of the fire suppression network.

If the temperature in the paint booth exceeds safety limit and the pressure in the fire line is optimal, the system automatically releases foam. The dashboard logs this event, showing:

- Temperature before and after discharge,
- Pressure variation during activation, and
- Time taken for system response.

This integration ensures a quick, effective response, protecting both workers and equipment from fire hazards (Fig.2.14).

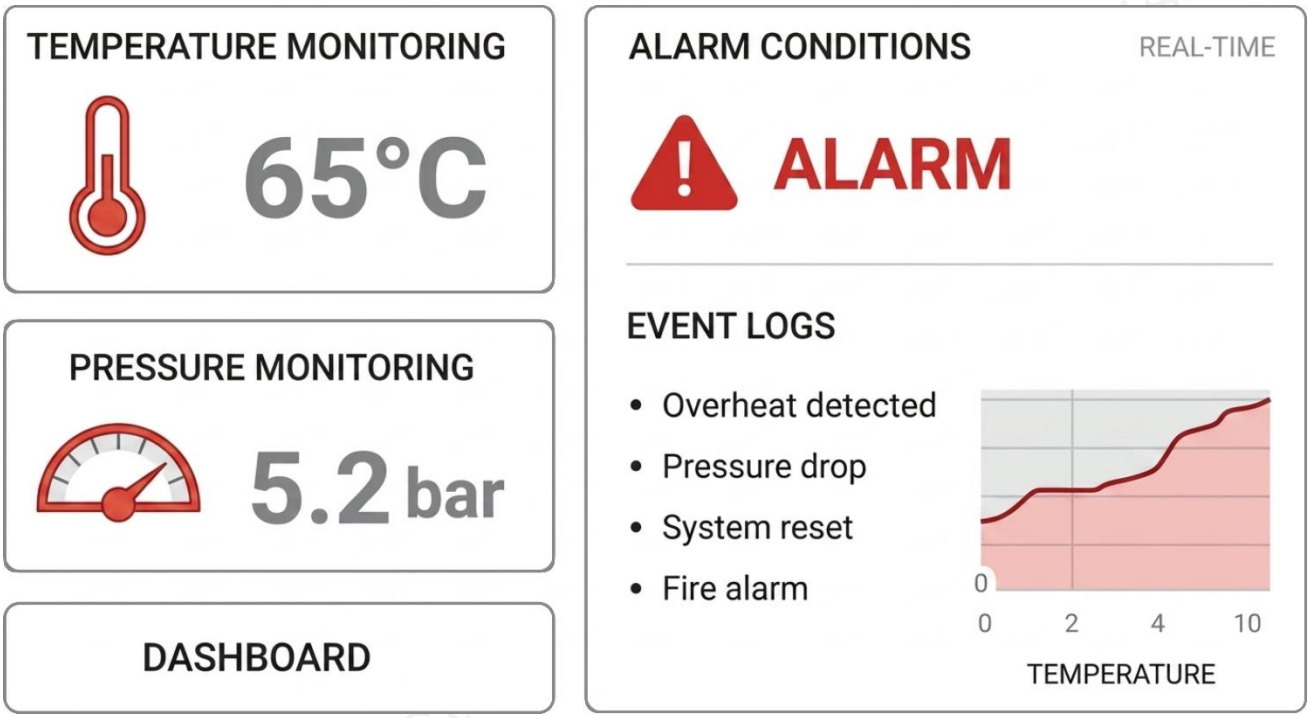


Fig.2.14: Smart Firefighting Systems

Advantages of IIOT-Based Firefighting Systems

- Early Detection: Sensors detect temperature or pressure anomalies before fire spreads.
- Remote Monitoring: Dashboards allow centralized control and instant alerts.
- Reduced Human Error: Automation minimizes the need for manual operation.
- Predictive Maintenance: Analyzes historical trends to schedule maintenance before failure.
- Compliance and Reporting: Automatically generates safety logs and reports for audits.

PRACTICAL ACTIVITY

ACTIVITY 1:
To Create a Simple Dashboard in Node-RED using Simulated Data

Objective:

- To generate the data using the **Random** node and displayed on the dashboard using **gauges** and a **chart**.
- To learn how to change the dashboard theme and layout.

Equipment/Software Required:

- Computer/Laptop (Node-Red installed)
- Internet connection
- Web Browser

Procedure:

Step 1: Understand the Random Node

The **Random** node is used to generate random numeric values automatically. It is useful when no real sensor is connected and you want to simulate live data for testing and dashboard design.

Purpose of the Random node:

- Generates sample numeric values
- Simulates live sensor readings
- Helps test dashboard elements such as gauges and charts

Step 2: Add the Required Nodes

From the left-side palette, drag the following nodes into the workspace:

- **Inject** node
- **Random** node
- **Gauge** node
- **Chart** node

Step 3: Configure the Inject Node

1. Double-click the **Inject** node.
2. Set it to trigger at a regular interval, such as every **1 second** or **2 seconds**.
3. Click **Done**.

This node will continuously trigger the Random node to generate values.

Step 4: Configure the Random Node

1. Double-click the **Random** node.
2. Set the required minimum and maximum values.
   - Example: Minimum = 0

- Maximum = 100

3. Click **Done**.

Step 5: Understand the Gauge Node

The **Gauge** node is used to display a numeric value on the Node-RED dashboard in a visual meter form. It is useful for showing live values such as temperature, humidity, pressure, or level. The gauge reads the value from random node and updates automatically whenever a new value arrives.

In this activity, the Gauge node will display the random value generated by the **Random** node.

Configuration

1. Open the **Gauge** node by double-clicking it (Fig.1).

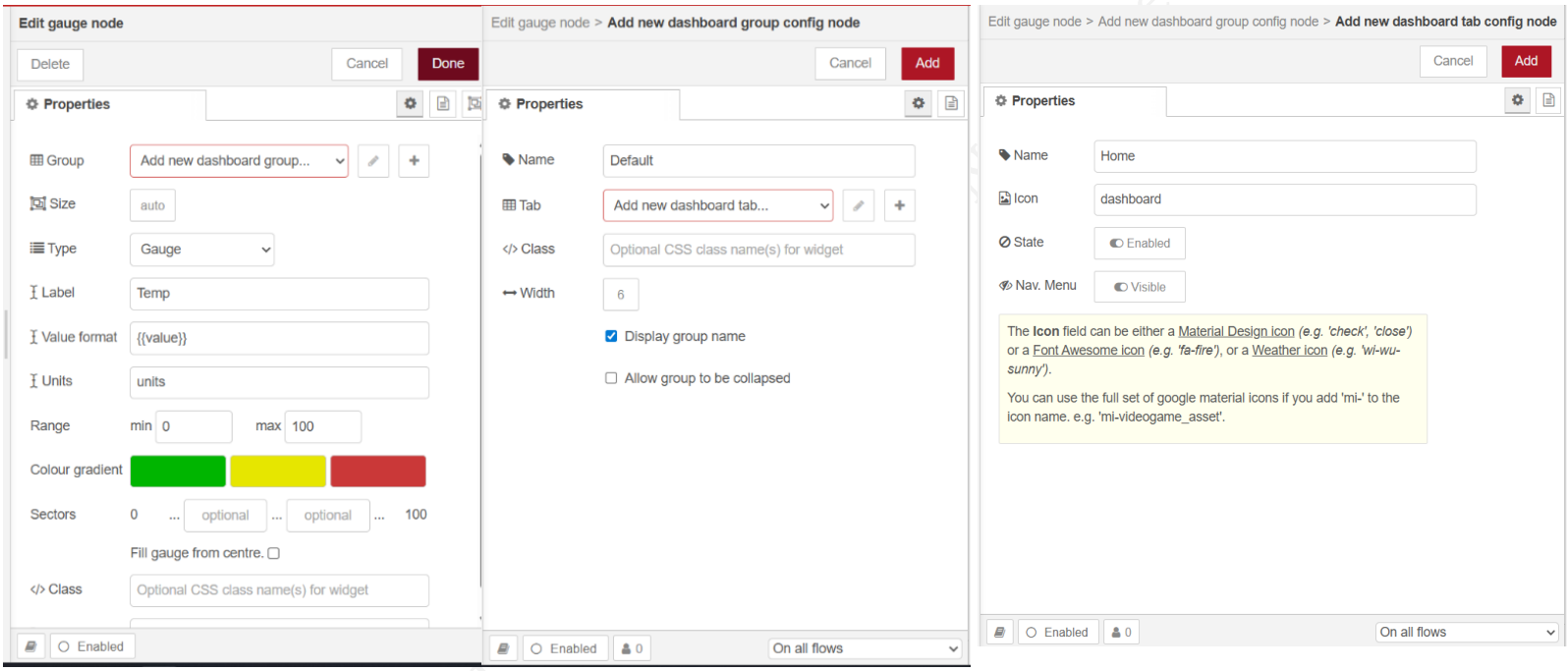


Fig.1: Gauge Node in Dashboard

2. In **Group**, select an existing group or create a new one.
3. If needed, create a new **Dashboard Tab** such as **Home**.
4. Set the **Type** to **Gauge**.
5. Enter the **Label** as **Temp**.
6. Keep **Value format** as {{value}}.
7. Set **Units** as °C.
8. Set the **Range**:
   - Minimum: 0
   - Maximum: 100
9. Set color zones:

- Green for low values
- Yellow for medium values
- Red for high values

10. Click **Done**.

The Gauge node will now show the simulated value on the dashboard in real time.

Step 5: Add Multiple Gauges

1. Drag **Gauge** nodes into the workspace.
2. **Label** is the name that will be visible to the dashboard
3. Connect:
   - **Inject → Random → Gauge**
4. If needed, connect the same Random node output to multiple Gauge nodes.

You can configure each gauge with different labels such as (Fig.2 and Fig.3):

- Temperature
- Pressure
- Humidity
- Tank Level

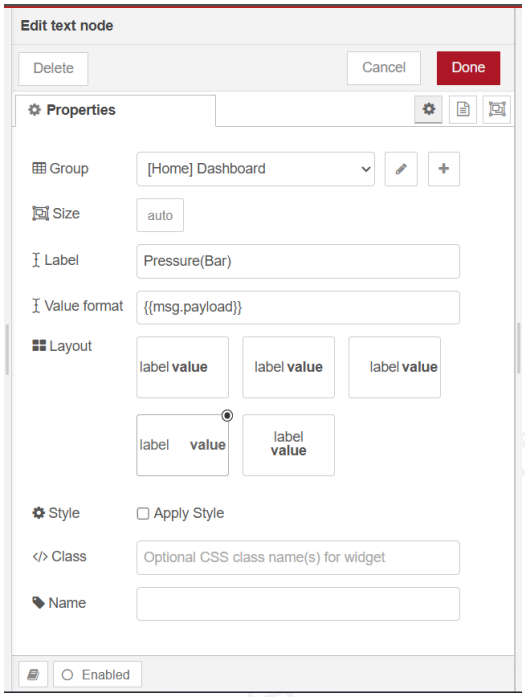


Fig.2(i): Text Node

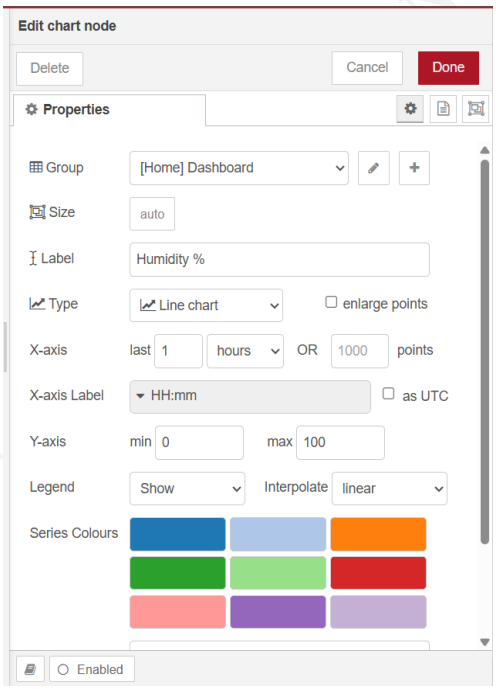


Fig.2(ii): Chart Node

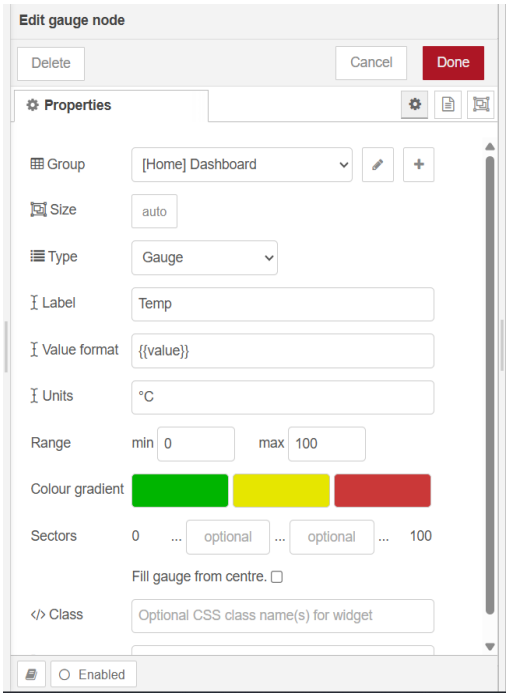


Fig.2(iii): Gauge Node

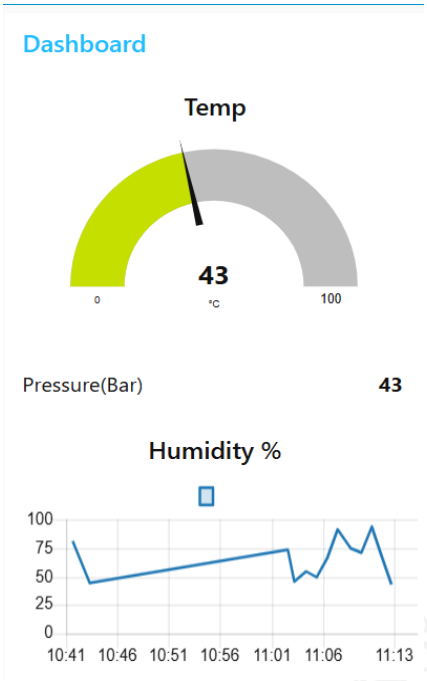


Fig. 3(i): Dashboard Sample

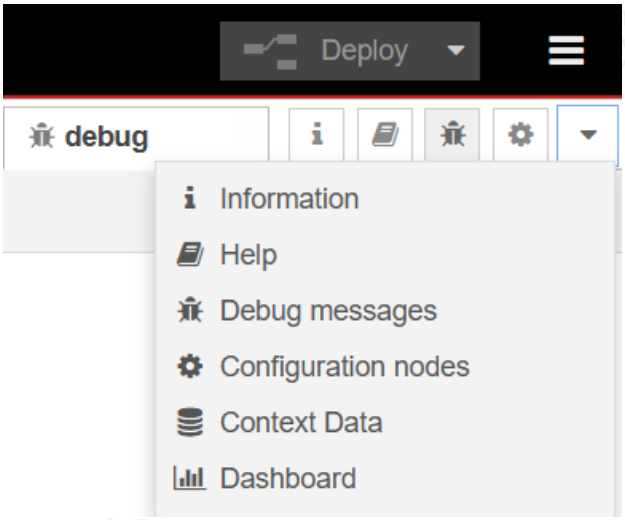


Fig. 3 (ii): More Options

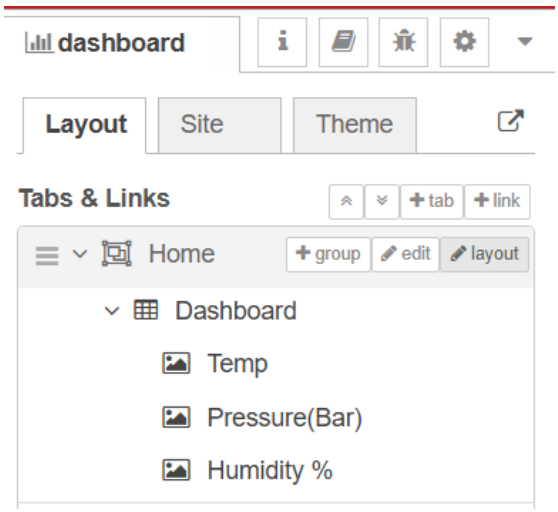


Fig.3 (iii): Dashboard Page

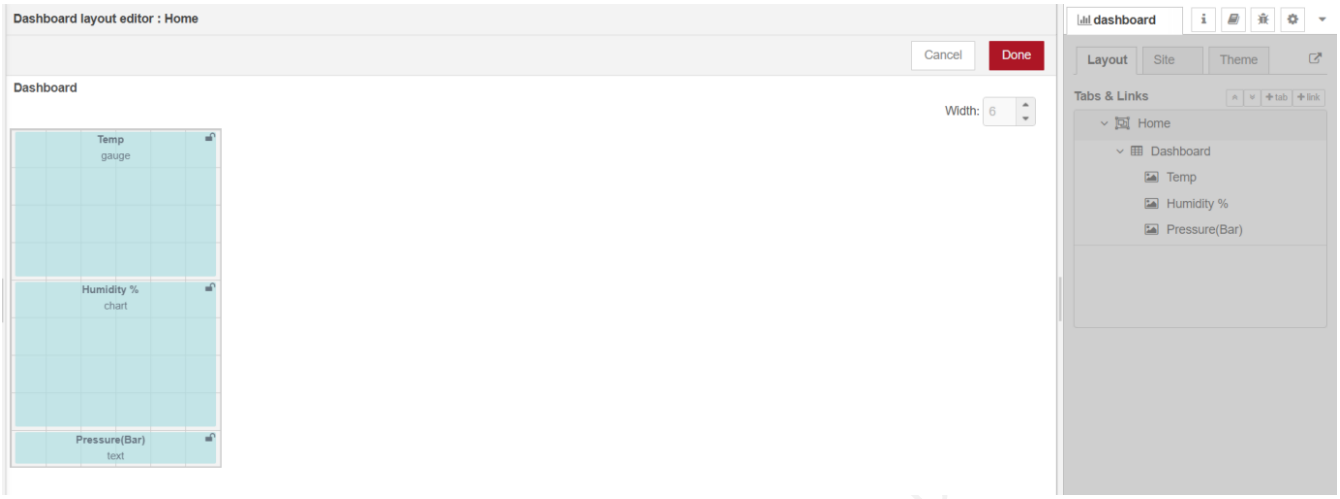


Fig.3(iv): Dashboard Layout Page

ACTIVITY 2:
Create a Node-RED Dashboard Gauge to Display Temperature and Humidity Data from a Physical DHT Sensor

Objective:

To display real-time temperature and humidity readings from a DHT sensor on a Node-RED Dashboard using gauge widgets.

Equipment/Software Required:

- Computer/Laptop (Node-Red installed)
- Internet connection
- Web Browser

Procedure: (refer Fig.1)

1. Open the Node-RED flow created in the previous activity.
2. Verify that the temperature and humidity data is being received in Node-RED.
3. Add a Function node to separate the temperature and humidity values.
4. Add two Dashboard Gauge nodes—one for Temperature and one for Humidity.
5. Configure the gauges with appropriate labels, units, and ranges.
6. Connect the Function node outputs to the corresponding gauges.
7. Deploy the flow and open the Node-RED Dashboard.
8. Observe the real-time temperature and humidity values displayed on the gauge.

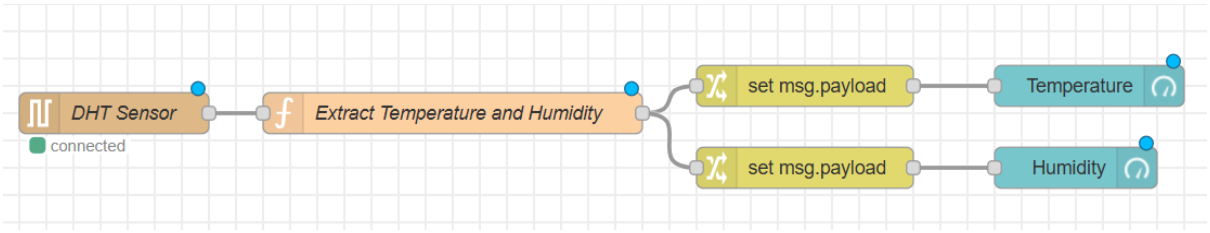


Fig.1(i): Example Flow

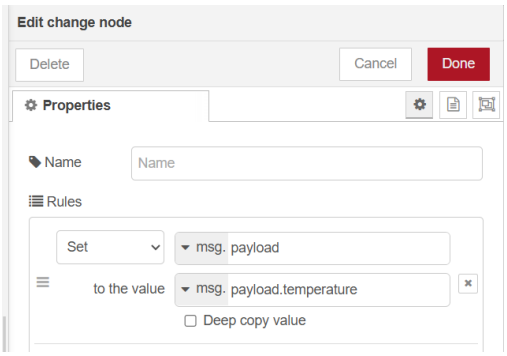


Fig.1(ii): Filter Temperature

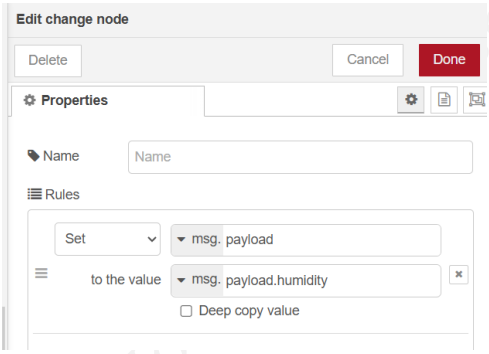


Fig.1iii): Filter Humidity

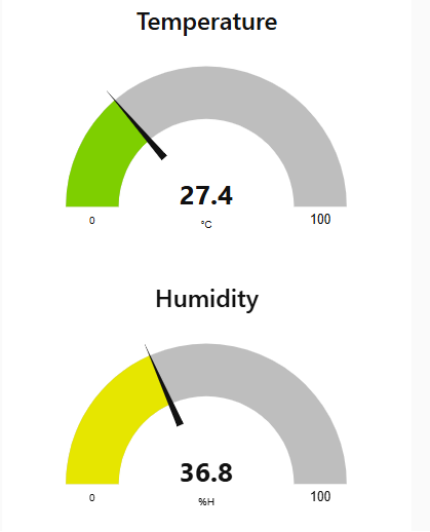


Fig.4 (iv): Final Output

👉👉 Exercises

Create four bar charts on the dashboard and label them Availability, Utilization, Productivity, and OEE. Each bar chart should display values within a range of 0 to 100 and use percent (%) as the unit. Apply color coding to represent performance levels: 0–60% in red, 60–90% in yellow, and 90–100% in green. This visualization helps users quickly understand machine performance and efficiency levels.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. An operator finds that the dashboard is overloaded with too many parameters, making it difficult to quickly understand machine status. What should be done to improve usability?
   a) Add more graphs
   b) Show all parameters on one screen
   c) Display only critical metrics with detailed view on demand
   d) Remove all labels

2. A supervisor wants the most critical alerts and KPIs to be visible immediately upon opening the dashboard. Which design approach should be followed?
   a) Place all data randomly
   b) Use bottom-heavy layout
   c) Apply logical hierarchy with key data at top/center
   d) Hide alerts in menus

3. A factory wants operators to instantly identify abnormal conditions without reading numerical values. Which solution should be implemented?
   a) Use long data tables
   b) Replace visuals with text
   c) Use color-coded indicators and gauges
   d) Disable alert

4. A maintenance engineer needs to study machine performance over several months to identify recurring faults. Which type of data analysis should be applied?
   a) Real-time analysis
   b) Historical data analysis
   c) Manual observation
   d) Edge processing

5. In an automated plant, a robot suddenly encounters an obstacle in its path. The system must detect and respond immediately. Which sensor should be applied in this situation?
   a) Flow sensor
   b) Temperature sensor
   c) LiDAR sensor
   d) Pressure sensor

B. Match the following

Column A	Column B
1. Real-time analysis	A. Monitors system pressure levels
2. Historical analysis	B. Adjusts layout for different devices

3. LiDAR sensor	C. Immediate monitoring and action
4. Responsive design	D. Detects objects using laser
5. Pressure sensor	E. Studies long-term performance trends

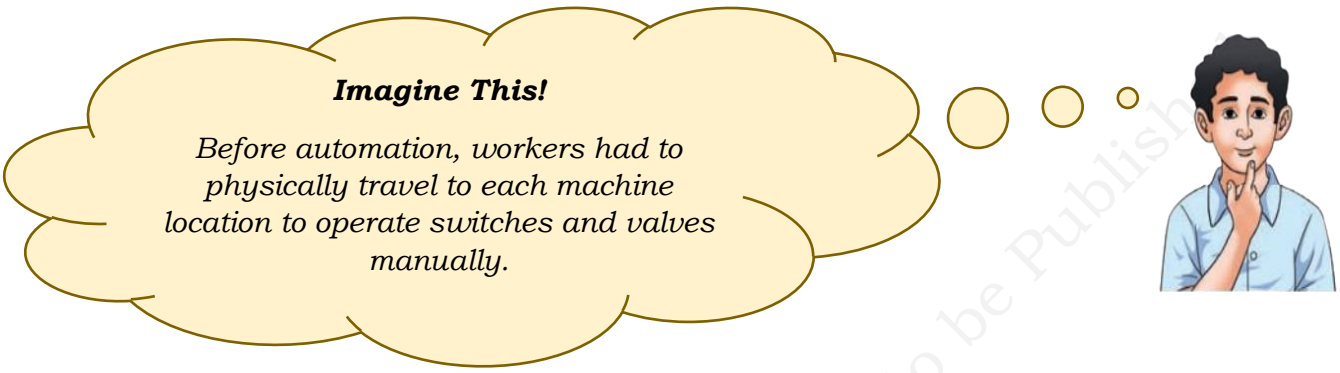
C. Fill in the blanks

1. Dashboards should avoid unnecessary graphics to maintain _____ and simplicity.
2. In real-time analysis, data is updated in _____ or milliseconds.
3. LiDAR sensors use _____ beams to detect objects and measure distance.
4. A _____ sensor detects the presence of nearby objects without physical contact.
5. A sudden drop in pressure may indicate a _____ in the system.

D. Answer the following

1. Operators are overwhelmed because too much information is displayed on a single screen. What changes can be made using dashboard design principles to make important data easier to understand?
2. A company wants to identify patterns in machine failures that occur over long periods. How can stored data be utilized to support this requirement?
3. In an automated warehouse, moving equipment must avoid collisions with obstacles and workers. How can sensing technology be used to handle this situation effectively?
4. A plant requires instant alerts when abnormal conditions occur in machines. What type of data handling approach should be implemented to enable quick response?
5. During a fire emergency, the system must both detect rising heat and ensure proper functioning of the suppression system. How can different types of sensors work together to manage this scenario?

SESSION 4: REMOTE CONTROL AND COMMAND EXECUTION



2.20 Principles of Remote Actuation

In an IIOT environment, machines and devices across the factory floor are not only monitored remotely but also controlled and actuated from a central system.

Remote actuation refers to the ability to operate or control machines, motors, valves, and actuators from a distance using digital commands transmitted through a network. This concept allows industries to automate processes, respond quickly to changing conditions, and ensure safety without requiring manual intervention.

A) Components of a Remote Actuation System

A typical IIOT-based remote actuation system includes the following components (Table 2.4):

Table 2.4: Components of a Remote Actuation System

Component	Function
Sensors	Measure real-world parameters like temperature, speed, or pressure and send data to the controller.
Controller / PLC / Microcontroller	Processes sensor data and decides whether an action is needed based on programmed logic.
Communication Network	Transmits commands between the controller and the actuator using protocols like MQTT, Modbus, or OPC-UA.
Actuators	Perform the physical action — such as rotating a motor, opening a valve, or pushing a piston.
Dashboard / HMI	Allows human operators to monitor system conditions and issue remote control commands.

B) Working Principle

The operation of a remote actuation system generally follows these steps:

1. **Data Collection** – Sensors collect environmental or process data.
2. **Decision Making** – The central controller analyzes this data against predefined conditions.
3. **Command Transmission** – If an action is required, a digital control signal is sent via the network.
4. **Actuation** – The actuator receives this signal and performs the physical operation.
5. **Feedback** – The actuator or associated sensor sends confirmation or status data back to the controller to close the loop. This feedback loop ensures accuracy, safety, and reliability by confirming that each action has been executed as intended.

For example, in an automotive paint shop, humidity and temperature must be tightly controlled. If sensors detect high humidity, the IIOT controller sends a remote command to activate the dehumidifier fan. Once the condition stabilizes, another command switches the fan off automatically. Operators can also manually override these actions through a dashboard; demonstrating both automation and human control in remote actuation systems.

C) Communication and Control Protocols

For remote actuation to be reliable, secure, and timely, industries use standard communication protocols such as:

- **MQTT (Message Queuing Telemetry Transport):** Lightweight protocol ideal for cloud-based actuation.
- **Modbus / EtherCAT / PROFINET:** Used for real-time industrial control between PLCs and field devices.
- **HTTP / REST APIs:** Common for web-based dashboards and mobile control applications.

These protocols ensure that control commands are delivered accurately even in complex or large-scale industrial networks.

D) Typical Patterns & Message Flows

In a typical IIOT system, data and control messages follow a structured pattern to ensure reliable and safe operation. The sequence begins with the sensor transmitting a measured value (such as temperature, pressure, or flow rate) to the controller. The controller evaluates this data and determines whether any action is required based on pre-defined logic or conditions. If an action is necessary, the controller sends an actuation command to the respective actuator through the gateway, which serves as the communication bridge between devices. The actuator, upon receiving the command, performs the required action and sends back an acknowledgment along with its updated status to the controller. This

feedback loop ensures that every action is verified and logged. Finally, the controller updates the dashboard with the current operational status and stores the event for future analysis. Commands in such systems can take several forms — immediate commands (e.g., "Stop Now"), scheduled commands (e.g., "Start at 02:00"), or conditional commands (e.g., "If temperature exceeds 70°C, activate cooling"). This structured communication flow ensures efficient coordination, transparency, and real-time responsiveness in IIOT-based automation environments.

E) Safety and Security Principles

Since remote actuation involves operating equipment from a distance, safety and cybersecurity are critical. Key principles include:

- **Authentication and Authorization:** Only authorized users or systems can send control commands.
- **Fail-Safe Design:** If communication fails, actuators revert to a safe default state (e.g., valves close automatically).
- **Redundancy:** Backup communication channels or power supplies ensure uninterrupted control.
- **Data Encryption:** Prevents unauthorized interception or manipulation of control signals.

F) Applications of Remote Actuation

Remote actuation is widely used in smart industries and automotive systems, such as:

- Automotive Manufacturing: Controlling robotic arms, conveyor systems, and painting robots remotely.
- Process Industries: Operating pumps, valves, and compressors from centralized control rooms.
- Smart Buildings: Managing HVAC, lighting, and fire suppression systems.
- Energy Systems: Opening/closing circuit breakers or solar tracking panels remotely.
- Agriculture: Controlling irrigation pumps or nutrient delivery systems automatically.

G) Advantages

- **Improved Safety:** Minimizes human exposure to hazardous environments.
- **Faster Response:** Immediate control actions reduce downtime.
- **Higher Efficiency:** Enables automation and centralized supervision.
- **Predictive Maintenance:** Combines sensor feedback with actuation to prevent failures.
- **Remote Accessibility:** Allows engineers to operate systems from anywhere via IIOT dashboards or mobile devices.

H) Examples

a. Automotive Paint Booth

- Sensors: Temperature, solvent vapor
- Actuation: Ventilation fans, spray valves
- Principle: If solvent vapor > limit → trigger exhaust fans and reduce spray rate; if 113overtempt, shut down spray nozzles and alert.

b. Conveyor System

- Sensors: Proximity, photoelectric
- Actuation: Motor drive inverter (start/stop), pneumatic pushers
- Principle: On jam detection → stop conveyor, reverse short, alert operator.

c. Fire Suppression

- Sensors: Heat and smoke detectors, pressure sensors on firewater lines
- Actuation: Release valves, pumps
- Principle: If multiple sensors confirm fire → trigger suppression and log activation; if pressure low → switch to backup pump.

2.21 Secure Two-Way Communication

In modern IIOT-based remote control systems, secure two-way communication forms the backbone of reliable and safe operations. It enables continuous data exchange between devices, controllers, and cloud platforms, ensuring that every command sent to a machine and every response received from it is both authentic and protected from unauthorized access or tampering.

In a typical setup, field devices such as sensors and actuators communicate with controllers or gateways, which then connect to supervisory systems or cloud platforms. This communication occurs in two directions, from devices to the cloud (uplink) for transmitting operational data, and from the cloud to devices (downlink) for sending commands or configuration updates. Maintaining the integrity and confidentiality of both these data flows is essential, especially in remote or safety-critical operations such as automotive manufacturing, power grids, or oil and gas systems.

To ensure secure two-way communication, IIOT systems implement several security principles:

- Confidentiality: All communication between devices and networks is encrypted to prevent data exposure. Protocols such as TLS (Transport Layer Security) and SSL are used to secure MQTT, HTTP, and OPC UA communications.
- Integrity: Data integrity ensures that the message received is exactly what was sent. Techniques such as hashing, digital signatures, and message authentication codes (MACs) are used to detect any unauthorized modifications.

- Authentication and Authorization: Before a device or user can exchange data, their identity must be verified using digital certificates, API keys, or multi-factor authentication. Authorization policies then determine what actions each user or system component is allowed to perform.
- Non-repudiation: Every command and response is logged with timestamps and digital proofs to ensure accountability, so that no user or device can deny sending or receiving data.

A) Communication Protocols for Secure Two-Way Communication in IIOT

Modern industrial systems rely on robust communication protocols designed with built-in security features:

- MQTT over TLS-lightweight, encrypted, and ideal for IIOT gateways and cloud systems.
- OPC UA (Unified Architecture)- provides encryption, authentication, and complex data modeling for industrial automation.
- Modbus TCP Secure-enhances the traditional Modbus protocol with TLS encryption.
- HTTPS / REST APIs- used for secure web-based dashboards and remote command execution.

Each of these protocols ensures that both control commands (e.g., starting motors, adjusting valves) and sensor feedback (e.g., temperature, flow, vibration) are securely transmitted without interruption or compromise.

B) Network Architecture and Protection

A secure IIOT network is divided into layers, with each layer performing specific functions and security checks:

- Device Layer: Field sensors and actuators are assigned unique IDs and cryptographic keys.
- Control Layer: PLCs and controllers perform logic control and act as the first line of defense, filtering incoming commands.
- Gateway Layer: Converts protocols, applies encryption, and manages authentication between devices and the cloud.
- Cloud / SCADA Layer: Stores and analyzes data, issues control commands, and enforces access policies through role-based control.

Firewalls, Intrusion Detection Systems (IDS), VPN tunnels, and network segmentation further protect data movement between these layers. For example, separating the Operational Technology (OT) network from the Information Technology (IT) network reduces the risk of external cyberattacks reaching industrial assets.

C) Command Verification and Acknowledgment

A defining feature of secure two-way communication is the command acknowledgment mechanism. When a remote operator issues a command (e.g., "Start Pump 3"), the actuator executes the instruction and sends back an acknowledgment (ACK) confirming successful execution. If the command fails, a Negative Acknowledgment (NACK) or error message is returned. This verification prevents command duplication, ensures traceability, and provides a full audit trail in case of malfunction or misuse.

All such transactions are timestamped and logged by the controller or cloud dashboard. In critical operations, redundant acknowledgments from multiple sensors or feedback loops may be required to confirm that an action was safely completed; for instance, a valve not only receives an open command but must also report its new position via a feedback sensor (Fig.2.15).

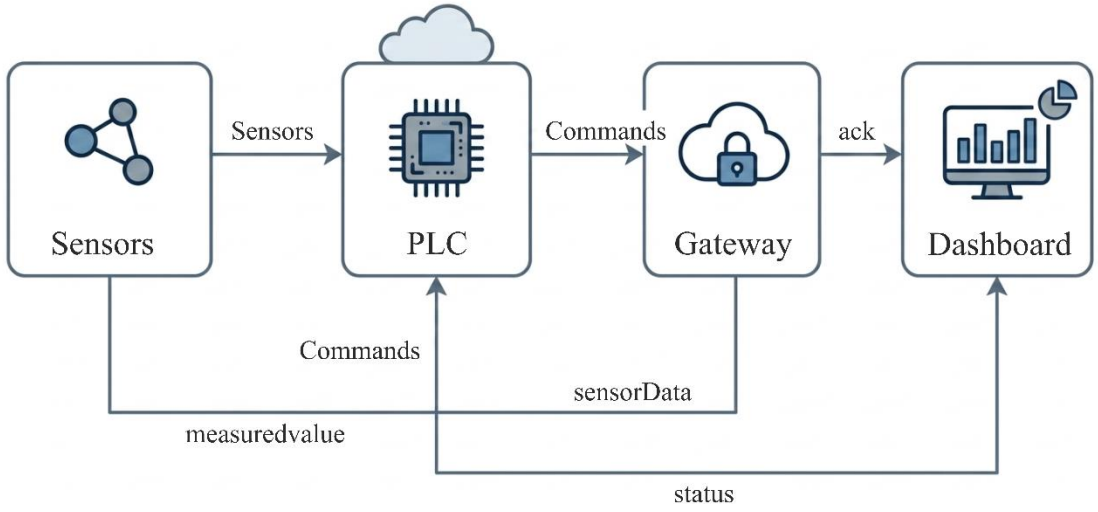


Fig.2.15: Secure Two-Way Communication

For example, in a modern automotive assembly plant, robotic arms, conveyors, and spray booths are all connected through a centralized IIOT system. Secure two-way communication allows the central controller to send real-time commands: such as starting a painting robot or adjusting conveyor speed, while receiving immediate feedback about machine status, torque levels, or paint flow rate.

- Commands are sent over MQTT with TLS encryption, ensuring data cannot be intercepted.
- The robot controller verifies command integrity via digital signature.
- The actuator executes the movement and sends a secure acknowledgment back.
- The dashboard logs the event and displays a confirmation message to the operator.

This seamless, secure feedback loop ensures that every control action is both traceable and tamper-proof, maintaining operational safety and production continuity.

D) Benefits of Secure Two-Way Communication

- Enhanced Reliability: Commands and feedback are protected from corruption or delay.
- Cybersecurity: Reduces vulnerability to hacking and malware.
- Operational Transparency: Every command and acknowledgment is logged for audits.
- Remote Accessibility: Authorized users can monitor and control operations from anywhere.
- Compliance: Meets industrial cybersecurity standards like IEC 62443 and ISO 27001.

2.22 Practical Applications of Remote Control

Remote control is one of the most useful features of the IIOT. It allows operators and engineers to control machines, systems, and processes from a distance using computers, tablets, or even mobile phones. Instead of manually operating each machine, commands can be sent through a network by saving time, reducing errors, and increasing safety. Some practical Applications of remote control in IIOT networks are as follows:

i) Automotive Manufacturing

In modern car factories, there are hundreds of machines and robots working together on assembly lines, paint booths, and conveyor systems. With remote control:

- Operators can start or stop robotic arms or conveyor belts from a central control room.
- If a robot overheats or moves incorrectly, the system immediately alerts the operator.
- The operator can then send a shutdown command remotely to prevent damage or accidents.
- For example, in a paint booth, air flow, temperature, and spray guns can all be monitored and adjusted from a computer dashboard, without a need to enter the booth (Fig.2.16).

Fig.2.16: Applications of Remote Control in Automotive Manufacturing

ii) Power and Energy Systems

Remote control is used to operate and monitor electric grids and transformers.

- Engineers can turn power lines on or off, reroute electricity, or isolate faulty sections remotely.
- This helps in reducing power outages and improves safety. In solar power plants, remote systems are used to track sunlight, adjust panel angles, and monitor battery charge levels automatically.

iii) Water and Waste Management

In water treatment plants, many pumps, valves, and sensors are installed across large areas.

- Through remote control, valves can be opened or closed, and pump speeds adjusted according to flow rate and pressure readings.
- If there's a leakage or low water level, the system can automatically stop pumps and send alerts to the control center.

iv) Warehousing and Material Handling

Remote control is widely used in automated warehouses and logistics systems.

- Conveyor belts, cranes, and robotic pickers are operated remotely.
- Operators can change the route of goods, stop equipment during jams, or restart systems after maintenance, all from a single dashboard. This reduces manual labor and ensures smooth material flow.

v) Safety and Emergency Systems

Remote control is also vital for safety and emergency actions.

- If sensors detect gas leaks, fire, or unauthorized entry, the IIOT system can automatically:
    - Activate ventilation fans
    - Sound alarms
    - Close emergency valves or doors
    - Send alerts to staff mobile devices. This quick response helps prevent accidents and keeps workers safe.

2.23 Benefits of Remote control in IIOT

Remote control in IIOT systems provides several operational and safety advantages. It allows operators to monitor and control machines from a central location, improving productivity, response time, and reliability. Benefits of use of remote control in IIOT are as follows:

Faster Decision-Making: Remote control enables operators to send commands instantly without physically visiting the machine location. This reduces delay in decision-making and allows quick adjustments during production, maintenance, or emergency situations.

Better Safety: IIOT-based remote systems can respond rapidly to hazardous conditions such as overheating, gas leakage, overload, or machine malfunction. Operators can shut down equipment or activate safety systems immediately, helping protect workers and machinery.

Higher Efficiency: Multiple machines, production lines, or plant sections can be managed from one control room or dashboard. This improves coordination, reduces manual effort, and allows smoother operation of industrial processes.

Reduced Downtime: Remote diagnostics and control help identify faults at an early stage. Corrective actions can be taken before minor problems become major failures, reducing machine stoppage and production loss.

PRACTICAL ACTIVITY

ACTIVITY 1:
Controlling an LED Using Node-RED Dashboard and ESP32

Introduction of the Activity
In the previous activities, students learned how to:

- Acquire temperature and humidity data from a DHT sensor using an ESP32.
- Transmit sensor data through serial communication.
- Process and visualize data in Node-RED.
- Generate machine alarms based on predefined threshold values.

In this activity, students will extend these concepts by controlling a physical device through Node-RED. Rather than only receiving data from the ESP32, Node-RED will send control commands back to the microcontroller.

Using a **Node-RED Dashboard Switch** widget, students will remotely turn an LED ON and OFF. This demonstrates **bidirectional communication** between Node-RED and the ESP32, a fundamental concept in IIOT systems where monitoring and control are performed through a centralized interface.

Such remote-control mechanisms are widely used in industrial applications for operating machines, actuators, indicators, and other field devices.

Objectives:

- Understand bidirectional communication between Node-RED and ESP32
- Use the Node-RED Dashboard Switch node
- Send commands through Serial communication
- Convert dashboard values using a Function node
- Control hardware devices from Node-RED interface

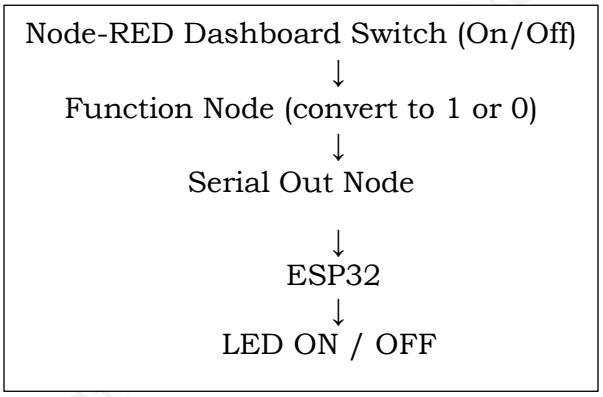
Components Required:

Component	Quantity
ESP32 DevKit	1
LED	1
Breadboard	1
Jumper wires	2
USB cable	1
Computer with Node-RED	1

Hardware Connections:
Connect the LED to the ESP32 as shown below.

Component	ESP32 Pin
LED Anode (+)	GPIO 2
LED Cathode (−)	GND

Working Principle:
The system works as follows:



The Node-RED dashboard sends commands to the ESP32:

Command	Action
"1"	LED ON
"0"	LED OFF

The ESP32 continuously reads the serial port and controls the LED accordingly.

Procedure:

Step 1 – Upload ESP32 Program
Upload the following program to the ESP32 using the Arduino IDE.

ESP32 Code: Write the given code:

```
#define LED_PIN 2         // GPIO 2 (built-in LED on many ESP32 boards)
char data;
void setup() {
  pinMode(LED_PIN, OUTPUT);     // Set LED pin as output
  Serial.begin(115200);        // Start serial communication
  Serial.println("ESP32 LED Control Ready");
}
void loop() {
  if (Serial.available() > 0) {
    data = Serial.read();     // Read serial data
    if (data == '1') {
      digitalWrite(LED_PIN, HIGH);   // LED ON
      Serial.println("LED ON");
    }
    if (data == '0') {
      digitalWrite(LED_PIN, LOW);   // LED OFF
      Serial.println("LED OFF");
    }
  }
}
```

After uploading the code, the ESP32 will wait for commands from Node-RED

Step 2 – Open Node-RED

- Open Node-RED in your browser: http://localhost:1880
- Create a new flow.

Step 3 – Add Dashboard Switch Node

- From the Node-RED palette, add: **ui_switch**
- Rename it: LED
- This switch will allow users to control the LED from the dashboard.

Step 4 – Add Function Node

- Add a Function node after the switch.
- Rename it: Convert User input for ESP
- This node converts the dashboard value into serial commands.

Function Node Code :

```
if (msg.payload == true) {
    msg.payload = "1";
} else {
    msg.payload = "0";
}
return msg;
```

Step 5 – Add Serial Out Node

- Add a Serial Out node.
- Rename it: WRITE TO ESP32

- Configure the serial port:

Setting	Value
Serial Port	(ESP32 port) Check in Control Panel
Baud Rate	115200
Data Bits	8
Parity	None
Stop Bits	1

- And keep the rest default
- Connect:
  Switch node → Function Node → Serial Out
- This will send commands to the ESP32.

Step 6 – Add Debug Node

- Add a Debug node to monitor the data sent to the ESP32.
- Connect: Function Node → Debug
- Example debug output: "1" or "0"

Step 7 – Deploy the Flow

- Click **Deploy** in Node-RED.
- Open the Node-RED Dashboard at:
  http://localhost:1880/ui
- You will see the **LED switch** (Fig.1(i)).

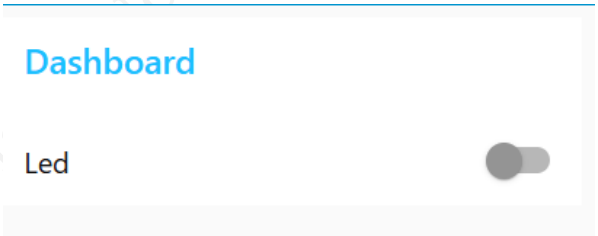


Fig.1(i): LED switch in Dashboard

Step 8 – Test the System

a) LED Switch ON

- When the LED switch is turned ON: Node-RED sends: "1" (Fig.1(ii)).

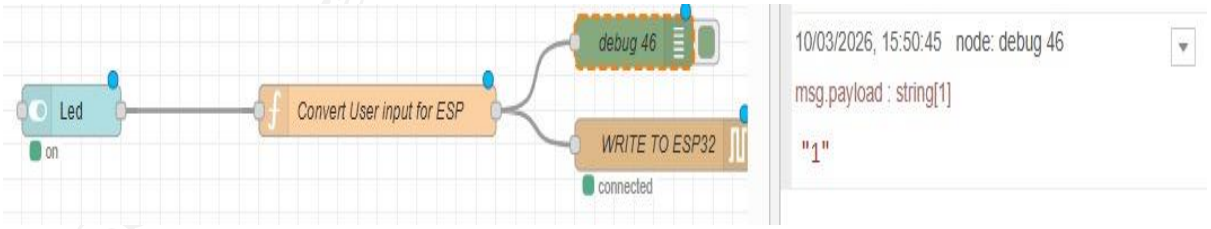


Fig.1 (ii): LED Switch ON

- ESP32 receives the command and turns the LED **ON** (Fig.1(iii)).

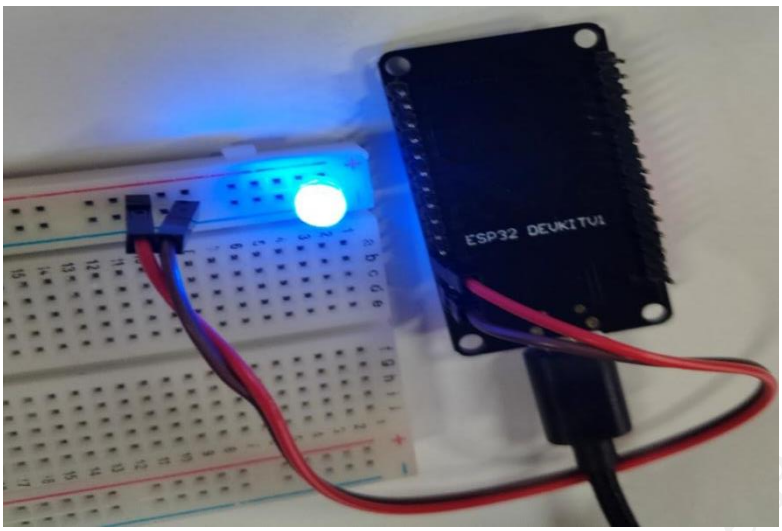


Fig.1(iii): LED ON

b) LED Switch OFF

- When the switch is turned OFF: Node-RED sends: "0" (Fig.1.23(iv)).

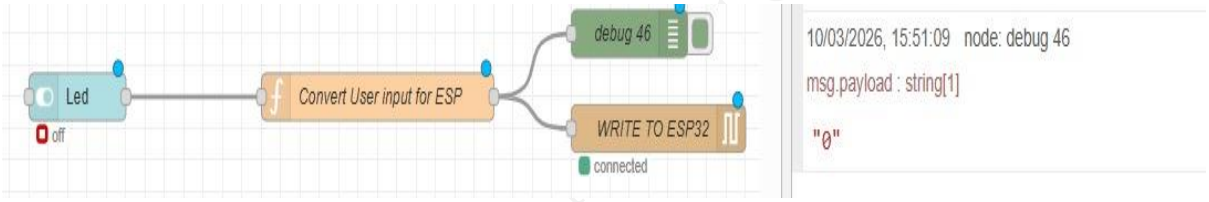


Fig.1(iv): LED Switch OFF

- ESP32 receives the command and turns the LED **OFF** (Fig.1.23(v)).

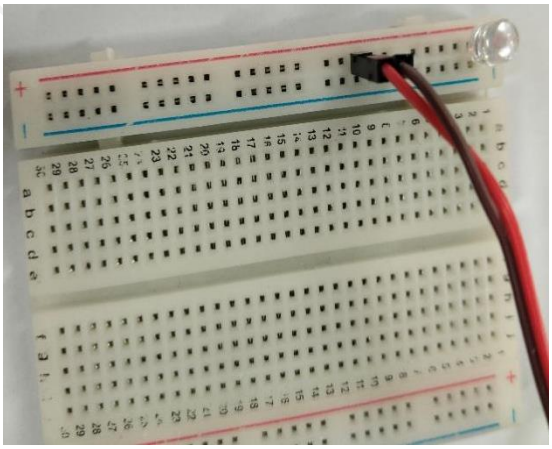


Fig.1(v): LED OFF

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. A temperature sensor in a plant detects overheating, and the system automatically turns on a cooling fan. This situation demonstrates
   a) Data storage
   b) Remote actuation using sensor input
   c) Manual intervention
   d) Visualization

2. An engineer schedules a command to start a motor at 2:00 AM without being physically present. Which type of command is being used?
   a) Immediate command
   b) Conditional command
   c) Scheduled command
   d) Manual command

3. Consider a system where a controller sends a command to an actuator, and the actuator responds back with its status. What is the main purpose of this response?
   a) Increase network traffic
   b) Provide confirmation of action
   c) Store historical data
   d) Improve visualization

4. A factory installs backup communication channels so that control commands continue even if the primary network fails. This is an example of
   a) Encryption
   b) Redundancy
   c) Authentication
   d) Visualization

5. A system is designed so that only authorized users can send commands to machines through a dashboard. Which principle is being applied here?
   a) Data logging
   b) Authentication and authorization
   c) Real-time monitoring
   d) Edge processing

B. Match the following

Column A	Column B
1. Controller	A. Transfers commands between devices
2. Actuator	B. Interface for user control and monitoring
3. Communication Network	C. Processes data and makes decisions

4. Dashboard / HMI	D. Confirms action and updates system status
5. Feedback Loop	E. Executes physical action

C. Fill in the blanks

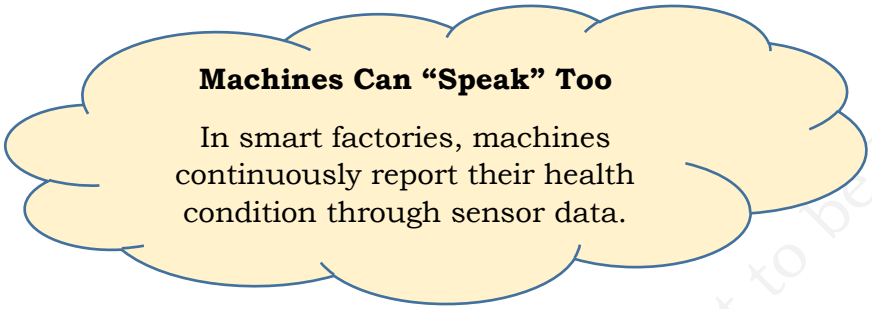
1. Remote actuation allows machines to be controlled from a _____ location.
2. MQTT is a _____ protocol commonly used in IIOT systems.
3. If communication fails, actuators return to a _____ state for safety.
4. Commands such as “Start at 02:00” are examples of _____ commands.
5. In secure communication, data is protected using _____ to prevent unauthorized access.

D. Answer the following

1. In a water treatment plant, pump operation must adjust automatically when flow conditions change. Explain how remote actuation can be implemented in this scenario.
2. A control system rejects commands from unknown users trying to access industrial machines. Identify the concept being applied and state its importance.
3. During operation, a machine executes a command and then sends its status back to the controller. Why is this step necessary in automated systems?
4. A factory installs backup communication links to avoid system shutdown during network failure. What principle is being followed, and how does it benefit operations?
5. Instead of manually operating each device, a supervisor manages multiple machines through a single interface. What advantages does this approach offer in industrial environments?

SESSION 5: ALERTS, ALARMS AND PROACTIVE ANALYSIS

2.24 Threshold-Based Alert Configuration



Threshold-based alert configuration is one of the most effective and widely used techniques in IIOT systems for detecting potential issues early. It allows industries to define specific limit values, known as thresholds, for key process parameters such as temperature, pressure, vibration, current, or speed. When any parameter exceeds or falls below its predefined safe range, the system automatically generates an alert or alarm, notifying operators to take timely corrective action before a breakdown or safety incident occurs.

In simple terms, this concept enables machines to "self-report" abnormal conditions without constant human supervision. For instance, if a motor's temperature rises beyond its safe limit, the IIOT platform immediately sends an alert to the control dashboard or operator's device, prompting preventive action.

Threshold-based alert configuration is a core feature of modern smart factories and industrial automation systems. It ensures continuous, real-time monitoring, reduces downtime, and enhances both operational reliability and worker safety by allowing early detection and quick response to emerging problems.

Thresholds can be set for both upper and lower limits:

- Upper Threshold (High Limit): Maximum allowed value before an alert is triggered (e.g., high temperature or high pressure).
- Lower Threshold (Low Limit): Minimum allowed value before an alert is triggered (e.g., low voltage or low flow rate).

These thresholds are defined based on the machine's design, safety standards, and operating environment.

2.24.1 How Threshold-Based Alerts Work

Each sensor in the IIOT network continuously measures a specific parameter (like temperature or flow rate) and sends the data to the controller or cloud platform. The system compares these readings to the threshold values set by engineers.

The alert logic generally follows this pattern:

1. **Normal Condition:** The value stays within the safe range - no action needed.
2. **Warning Condition:** The value approaches the limit - the system sends a *warning alert*.
3. **Critical Condition:** The value crosses the limit - the system sends a *critical alert* or automatically performs a safety action.

For example, if a fan motor's safe temperature range is 30°C–70°C:

- At 65°C → the system issues a warning alert ("Motor temperature rising").
- At 75°C → the system issues a critical alert and can automatically stop the motor to prevent damage.

For example, in an automotive paint booth, A temperature sensor might have a threshold range of 25°C–35°C.

- If the temperature rises above 35°C, a high-temperature alert is triggered.
- If it falls below 25°C, a low-temperature alert warns that the heating system may have failed.
- A pressure sensor monitoring spray nozzles could trigger an alert if the air pressure falls below 60 psi, indicating a possible blockage or leak.
- Air Quality: If solvent vapor concentration (measured by a gas sensor) goes beyond a safe limit, the system automatically turns on exhaust fans and alerts the supervisor.

These alerts help technicians take corrective action immediately, preventing defective paint finishes or costly downtime.

2.24.2 Configuration and Visualization of Threshold-Based Alert

Modern IIOT dashboards make it easy for operators to set up and monitor thresholds.

- The dashboard shows real-time readings from each sensor.
- Operators can set upper and lower limits directly from the screen.
- Alerts can be configured to appear on the dashboard, flash as visual signals, or be sent as SMS or email notifications to maintenance teams.

To make it easier to understand, dashboards often use color indicators:

- Green Zone: Normal operation
- Yellow Zone: Warning or near limit

- Red Zone: Critical condition requiring immediate action

This visual system helps operators quickly identify which machine or process needs attention.

2.24.3 Types of Threshold-Based Alert

There are three main types of alerts based on how critical the situation is:

1. Informational Alerts: Indicate normal changes or updates in system performance.
2. Warning Alerts: Suggest that the system is nearing an unsafe condition.
3. Critical Alerts: Triggered when the value goes beyond the threshold — may include automatic shutdown or emergency responses.

Some systems also support auto-response alerts, where the system can take immediate corrective action. For example, if temperature rises too high, it can automatically turn on a cooling fan or reduce motor speed.

2.24.4 Benefits of Threshold-Based Alerts

- Early Problem Detection: Helps detect abnormal conditions before they cause breakdowns.
- Improved Safety: Reduces risks of overheating, overpressure, or mechanical failure.
- Reduced Downtime: Maintenance teams can act before a full shutdown occurs.
- Better Efficiency: Machines operate within their best performance range.
- Predictive Maintenance: Frequent alerts on certain parameters can signal wear and tear, helping plan maintenance in advance.

It can be easily understood by taking a real word example of a smart factory. Here threshold-based alerts work as follows:

- Vibration sensors on a motor continuously measure vibration levels.
- The system's threshold is set to 5 mm/s.
- When vibration exceeds 4.5 mm/s, a warning alert appears on the dashboard.
- If vibration reaches 6 mm/s, a critical alert is generated, and the controller automatically stops the motor to avoid damage. This simple process helps the maintenance team fix the issue before it turns into a costly failure.

2.25 Alarm Management Strategies

In any IIOT system, alarm management is an essential process that ensures industrial operations remain safe, reliable, and efficient. With thousands of sensors and devices continuously transmitting data, a well-organized alarm strategy prevents confusion, avoids false alerts, and ensures that critical warnings are acted upon immediately.

A good alarm management system not only alerts operators to problems but also helps analyze patterns, prevent recurring issues, and optimize system performance over time.

The main goals of alarm management in IIOT systems are to:

- Detect abnormal or unsafe conditions in real time.
- Inform operators with accurate and timely alerts.
- Prioritize alarms based on severity and potential risk.
- Support corrective action before system failure occurs.
- Reduce false or nuisance alarms that cause distraction or fatigue.

By following structured alarm management strategies, industries can enhance both machine uptime and worker safety while maintaining productivity.

2.25.1 Classification of Alarms

Alarms are categorized according to their level of urgency and impact on production and safety. (Table 2.4)

Table 2.4: Classification of Alarms

Type	Description	Example
Critical Alarms	Immediate danger or breakdown; requires instant action	Motor overheating, gas leakage, high pressure
Warning Alarms	Deviation from normal range; may become critical if ignored	High vibration, low coolant level
Information Alarms	Status or performance notifications	Machine started, maintenance due

A) Alarm Prioritization

In a smart factory, hundreds of alarms may be triggered simultaneously. Without proper prioritization, operators can easily become overwhelmed. Each alarm should have a priority level (High, Medium, Low) assigned based on the following factors:

- Safety impact
- Production loss potential
- Equipment damage risk
- Environmental hazard

For example, a “Boiler Overpressure” alarm should always override a “Tank Level Low” alarm. This ensures that urgent issues are handled first.

B) Alarm Filtering and Suppression

Sometimes, one event can trigger a cascade of alarms — known as alarm flooding. To prevent this, IIOT systems use filtering and suppression techniques that:

- Show only the root cause alarm (e.g., "Power Supply Failure") and suppress dependent alarms (e.g., "Motor Stopped," "Sensor Offline").
- Use logic-based rules to identify and hide repetitive or non-critical alarms.

This keeps dashboards clear and prevents distraction from false or redundant alerts.

C) Smart and Predictive Alarms

Modern IIOT systems use artificial intelligence (AI) and machine learning (ML) to make alarms more intelligent. Instead of relying only on static threshold limits, predictive alarms analyze trends over time.
For example:

- A pump's vibration gradually increases over days.
- The system identifies the trend and predicts that bearing failure may occur soon.
- A predictive alarm alerts maintenance teams before the actual breakdown.

Such intelligent alerts reduce downtime and allow for condition-based maintenance rather than reactive repair.

D) Alarm Escalation and Notification

If an alarm is not acknowledged within a certain time, it should escalate automatically to higher authorities through:

- SMS or email notifications.
- Mobile push alerts to supervisors.
- Integration with voice call or emergency siren systems for critical events.

This ensures that important warnings never go unnoticed, even during night shifts or unmanned hours.

E) Visualization and Dashboard Design

Clear visual presentation of alarms is equally important. Dashboards typically display (Fig.2.17):

- **Color codes:** Red for critical, yellow for warning, green for normal.
- **Flashing icons** or sound indicators for urgent alarms.
- **Time stamps** showing when the alarm occurred.
- **Recommended corrective actions** for faster resolution.

Operators can view the complete alarm history, filter by type or equipment, and analyze recurring issues.

Fig.2.17: Visualization and Dashboard Design

F) Continuous Review and Optimization

Alarm systems must evolve with operational changes. Periodic review helps in:

- Identifying frequent nuisance alarms.
- Updating thresholds and logic as machinery ages or production patterns shift.
- Removing obsolete alarms after equipment upgrades.
- Creating standard operating procedures (SOPs) for consistent responses.

Regular evaluation ensures that the alarm management system remains accurate, relevant, and efficient.

Example: Alarm Management in a Smart Automotive Plant

In a car manufacturing plant, hundreds of machines work simultaneously.

- If a robotic arm overheats, a high-priority alarm appears on the dashboard.
- The system automatically logs the event, alerts the maintenance team, and, if not acknowledged, escalates it to the floor supervisor.
- Meanwhile, related low-priority alarms (e.g., “Robot idle”) are suppressed.
- Data from this incident is analyzed later to adjust preventive maintenance schedules.

2.26 Predictive Analytics & Anomaly Detection

In IIOT systems, predictive analytics and anomaly detection are two powerful techniques that make IIOT systems smarter and more proactive. Instead of waiting for problems to

happen or relying only on fixed threshold alarms, these techniques analyze patterns in real-time and historical data to identify early warning signs of failure.

These technologies combine data science, sensor analytics, and artificial intelligence (AI) to continuously learn from machine behavior, detect abnormal trends, and provide actionable insights, making factories smarter, safer, and more efficient.

2.26.1 Predictive Analytics: Seeing Problems Before They Happen

Predictive Analytics uses data collected from sensors, machines, and networks to forecast potential failures or maintenance needs. By studying trends such as temperature rise, vibration fluctuations, or energy usage patterns, predictive analytics can estimate when a component might fail or when maintenance should be scheduled. This helps industries move from reactive or preventive maintenance to a more efficient predictive maintenance approach, reducing downtime and saving costs.

For example, if the vibration pattern of a motor shows a gradual increase over time, the system predicts that a bearing might fail soon. Engineers receive an alert days before the actual breakdown, allowing them to plan a repair without interrupting production.

Key Benefits of Predictive Analytics

- Early identification of faults and performance degradation.
- Reduced unplanned downtime and emergency repairs.
- Better inventory planning for spare parts.
- Extended asset lifespan and improved operational reliability.

2.26.2 Anomaly Detection: Spotting the Unexpected

While predictive analytics forecasts possible issues based on trends, anomaly detection focuses on identifying unusual or abnormal behavior in real time. It compares live sensor data with what is considered “normal” for a particular machine or process.

When data deviates significantly from normal patterns — even if it hasn’t crossed alarm limits — the system flags it as an anomaly. This helps detect subtle or emerging issues that might otherwise go unnoticed.

Common Techniques Used in Anomaly Detection

- Statistical Analysis: Identifies deviations using mean, variance, and standard deviation of data.
- Machine Learning Algorithms: The system learns normal patterns over time using models like neural networks or clustering.
- Correlation Analysis: Checks if related sensors show consistent readings — for example, a rise in motor temperature should correlate with increased current draw.

For example, in a car manufacturing plant, predictive models analyze robotic arm torque data and detect slight increases in resistance. The system identifies this as an early sign of

joint wear, notifies the maintenance team, and schedules service before a production delay occurs.

2.26.3 Integrating Predictive Analytics and Anomaly Detection

When combined, predictive analytics and anomaly detection form the core of a proactive maintenance ecosystem. Here's how they work together in an IIOT network (Fig.2.18):

1. Sensors continuously collect real-time data (vibration, temperature, current, flow, etc.).
2. Edge devices or gateways preprocess the data and send it to cloud-based analytics platforms.
3. Predictive models evaluate the health of equipment based on historical and live data trends.
4. Anomaly detection systems monitor for sudden or subtle deviations in performance.
5. When a risk is detected, the system generates a proactive alert or maintenance recommendation.
6. Dashboards visualize these insights, helping engineers make informed decisions.

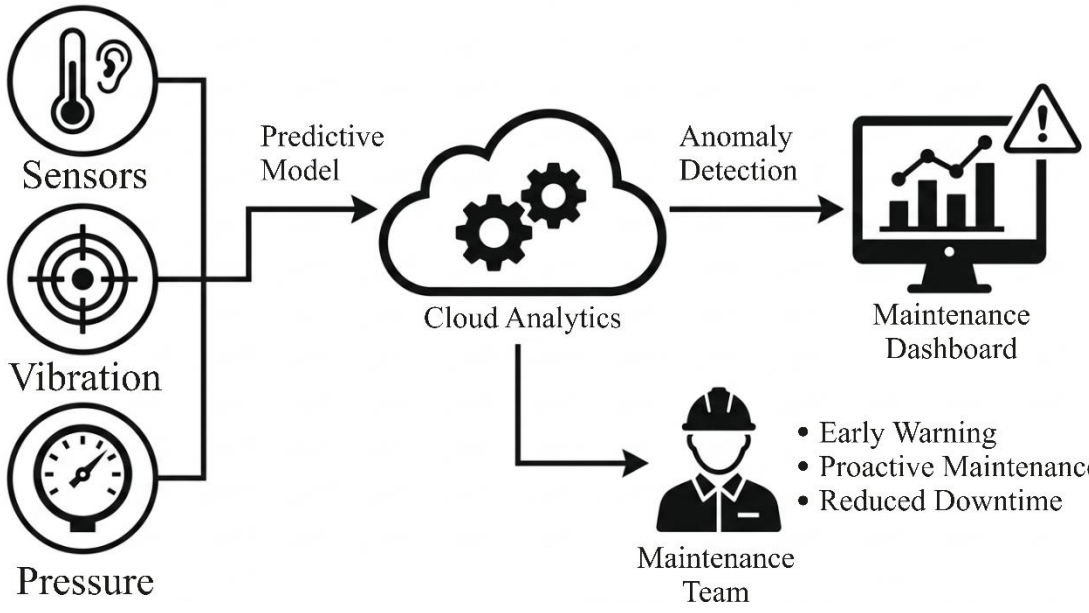


Fig.2.18: Integrating Predictive Analytics and Anomaly Detection

2.26.4 Applications in Industry

Predictive analytics and anomaly detection are used across multiple sectors:

- Automotive Manufacturing: Predicting robot arm wear, paint system air flow issues, and conveyor motor faults.

- Power Plants: Identifying transformer overheating, generator imbalance, or turbine vibration anomalies.
- Oil and Gas: Detecting pipeline leaks or compressor inefficiency before failures occur.
- Water Treatment: Forecasting pump degradation and identifying abnormal flow rate changes.
- Smart Buildings: Monitoring HVAC systems for predictive maintenance and energy optimization.

For example, in a smart automotive factory, thousands of sensors monitor robotic welding, painting, and assembly systems. Temperature, vibration, and current data from each robot are sent to a predictive analytics platform. The system identifies subtle variations in arm torque and movement speed that suggest mechanical wear. A predictive alert is sent to the maintenance dashboard three days before a potential breakdown. The maintenance team schedules a quick repair during non-production hours; preventing costly downtime.

2.26.5 Benefits of a Predictive IIOT System

- Zero Unplanned Downtime: Detects equipment issues at an early stage and allows repairs to be scheduled before failure occurs.
- Improved Safety: Identifies hazardous operating conditions before they become serious risks to workers or machines.
- Optimized Maintenance Cost: Components are replaced only when required, reducing unnecessary routine maintenance expenses.
- Better Decision-Making: Real-time and historical data provide accurate insights for maintenance planning and resource allocation.
- Higher Productivity: Machines operate efficiently with fewer interruptions, leading to smoother production processes.

PRACTICAL ACTIVITY

ACTIVITY 1:
Machine Alarm Generation Using DHT Sensor Data in Node-RED

Objectives:

• Understand the concept of alarm thresholds
• Create alarm logic in Node-RED
• Use Function nodes for condition checking
• Generate machine alarm messages
• Display alarm messages in the Debug panel
• Understand how alarms are used in industrial monitoring systems

Prerequisites

Students should have completed the previous activity on reading and visualizing DHT sensor data using Node-RED Dashboard. In that activity, they learned how to:

1. Read temperature and humidity data from a DHT11 sensor.
2. Transmit sensor data through serial communication.
3. Receive the data in Node-RED using the **Serial In** node.
4. Extract temperature and humidity values using a **Function** node.
5. Display the sensor readings on a Node-RED Dashboard.

Building on the same setup, this activity introduces a simple machine alarm system. Alarms will be generated based on predefined threshold values:

- A **Temperature Alarm** will be triggered when the temperature exceeds **25°C**.
- A **Humidity Alarm** will be triggered when the humidity falls below **25%**.

This activity demonstrates a common Industrial IoT (IIOT) application where alarms are automatically generated when process parameters exceed safe operating limits, enabling timely monitoring and corrective action.

Alarm Logic
In this activity, following alarm conditions are defined:

Parameter	Alarm Condition	Alarm Message
Temperature	Above 25°C	Temp has reached above 25°C, please check!
Humidity	Below 25%	Humidity has reached below 25%, Please check!

Flow Overview

The Node-RED flow will work like this:

DHT Sensor → Extract Temperature and Humidity → change node → Check Alarm Threshold using function node → Debug

Procedure:
Step 1 – Open the Existing Node-RED Flow (Fig.1(i))
Open the flow created in the previous activity.

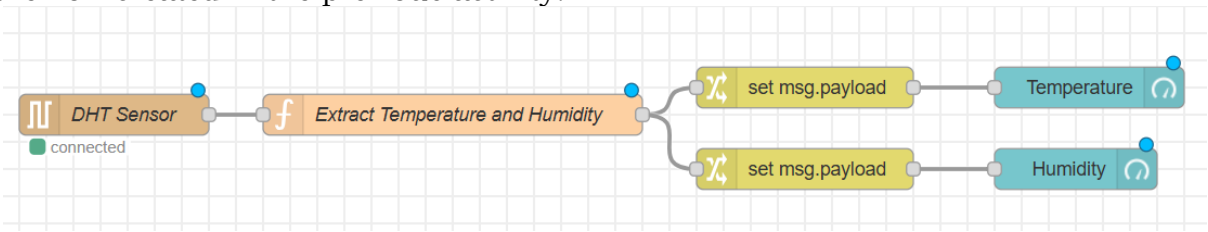


Fig.1(i): Existing Node-RED Flow

Example output from each change nodes(Fig.1(ii))
:

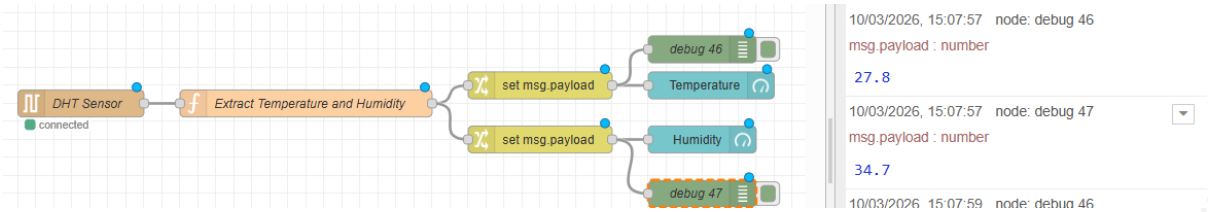


Fig.1(ii): Output from each change nodes

Step 2 – Understand Why We Need Separate Alarm Paths
The extracted data contains both values together:

```
{
  "temperature": 26.9,
  "humidity": 41.5
}
```

To check separate alarm conditions, we create two paths:

- One path for temperature
- One path for humidity

Each path will use its own alarm threshold logic.

Step 3 – Add Two Change Nodes to Split the Data
After the Extract Temperature and Humidity node, add two Change nodes.
Change Node 1

```
Set = msg.payload
to the value = msg.payload.temperature
```

This sends only the temperature value to the temperature path.

Note: Create the same for Humidity Node also by yourself.

Step 4 – Add Function Node for Temperature Alarm
Add a Function node after the temperature path.
Name it: **Temperature Alarm threshold**
Paste this code inside this function node:

```
let temp = msg.payload;
let Alarm = {};
if (temp >= 25) {
    Alarm = {Alarm: "Temp has reached above 25°C, Please check !"};
    msg.payload = Alarm;
    return msg;
}
msg.payload = null;
return msg;
```

Step 5 – Add Debug Node for Temperature Alarm
Connect a Debug node after the Temperature Alarm threshold function node. This will display the temperature alarm message in the debug panel.
Expected alarm output (Fig.1(iii)):

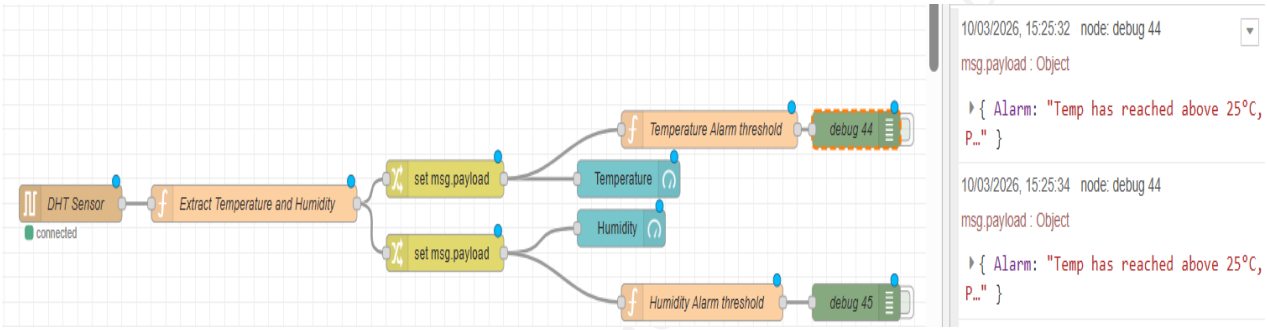


Fig.1(iii): Expected alarm output

Step 8 – Perform the same steps for Humidity also

☞☞ Exercises

1. Change the temperature alarm threshold from 25°C to 30°C.
2. Change the humidity alarm threshold from 25% to 35%.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. A motor's vibration level rises slightly but has not crossed the critical limit. The system still generates an alert for early action. What concept is being applied?
   a) Manual monitoring
   b) Anomaly detection
   c) Data storage

d) Actuation

2. A machine parameter crosses its upper safe limit, and the system automatically shuts down the equipment. This is an example of:
   a) Informational alert
   b) Warning alert
   c) Critical alert with auto-response
   d) Historical analysis

3. A plant receives too many repeated alarms from related faults, making it difficult to identify the root cause. Which strategy should be applied?
   a) Alarm escalation
   b) Alarm suppression and filtering
   c) Predictive analytics
   d) Data encryption

4. An engineer studies vibration data over time and predicts bearing failure before it happens. Which technique is being used?
   a) Threshold monitoring
   b) Predictive analytics
   c) Manual inspection
   d) Visualization

5. If an alarm is not acknowledged within a fixed time, it is automatically sent to higher authorities. This process is known as:
   a) Alarm filtering
   b) Alarm escalation
   c) Data logging
   d) Threshold setting

B. Match the following

Column A	Column B
1. Upper Threshold	A. General system update
2. Lower Threshold	B. Maximum safe limit
3. Warning Alert	C. Immediate action required
4. Critical Alert	D. Near unsafe condition
5. Informational Alert	E. Minimum safe limit

C. Fill in the blanks

1. Threshold values define the _____ range of machine operation.

2. Dashboards use _____ colors like green, yellow, and red to indicate system status.

3. Predictive analytics mainly uses _____ data patterns to forecast failures.
4. An alarm indicating “machine started” is an example of an _____ alert.
5. Anomaly detection identifies _____ behavior that deviates from normal patterns.

D. Answer the following

1. A machine parameter begins to move close to its limit but has not yet crossed it. In what way can alert configuration assist technicians in taking timely action?
2. Multiple alarms are triggered for a single fault, making it difficult to identify the main issue. Which method can be used to handle such situations effectively?
3. Over time, data shows a steady increase in vibration in a motor system. How can this information be used to avoid unexpected breakdowns?
4. A warning notification is generated but no one responds to it immediately. What system feature can ensure that the issue still gets attention?
5. Even when readings appear normal, the system flags unusual patterns in machine behavior. How can advanced data techniques help detect such hidden problems?

MODULE 3	MAINTENANCE AND TROUBLESHOOTING OF I/O LINK MASTER AND IIOT NETWORK DEVICES

Module Overview

This module focuses on the maintenance and troubleshooting of I/O Link Master and IIOT network devices that enable communication between sensors, actuators, controllers, and cloud platforms in modern industrial automation systems. Learners will gain an understanding of IIOT connectivity, machine alarms, status monitoring, and machine performance analysis to identify faults and improve system efficiency.

The module also develops practical skills in network diagnostics, hardware testing, preventive maintenance, and network optimization. Through hands-on activities, learners will learn to troubleshoot communication and device-related issues, minimize downtime, and ensure reliable operation of IIOT-based industrial systems.

In addition, the module introduces best practices for maintaining network reliability, improving equipment availability, and supporting data-driven decision-making. By the end of the module, learners will be equipped to manage and maintain IIOT-enabled industrial environments effectively, preparing them for careers in industrial automation and smart manufacturing.

Learning Outcomes

After completing this module, you will be able to:

- Explain the architecture, communication flow, and operational principles of I/O Link Master and IIOT network devices used in industrial automation systems.
- Diagnose and interpret machine alarms, device faults, and network performance data to identify the root causes of system malfunctions.
- Carry out systematic troubleshooting and testing of sensors, actuators, and communication modules using appropriate diagnostic tools and procedures.
- Implement and manage preventive and predictive maintenance activities to ensure continuous and reliable operation of IIOT-enabled systems.
- Optimize and document network performance, maintenance records, and corrective actions in compliance with standard industrial safety and quality practices.

Module Structure
Session 1: Foundational IIOT Connectivity **Session 2:** Machine Alarm and Status Analysis **Session 3:** Advanced Machine Performance Analytics **Session 4:** IIOT Network Monitoring and Evaluation **Session 5:** IIOT Network Diagnostics, Troubleshooting, and Optimization **Session 6:** IIOT Hardware Testing, Safety, and Maintenance Practices

SESSION 1: Foundational IIOT Connectivity

3.1 Device-to-Cloud Protocols

In IIOT, device-to-cloud communication plays a vital role in connecting machines, sensors, controllers with centralized servers or cloud platforms. These protocols define how data collected by field devices is transmitted securely and efficiently to the cloud, where it can be stored, analyzed, and visualized for decision-making.

Device-to-cloud protocols act as the digital bridge between physical equipment on the factory floor and digital analytics systems. Choosing the right communication protocol ensures reliable data transfer, low latency, and energy efficiency; all crucial for real-time monitoring and automation.

Device-to-cloud communication allows:

- Continuous data flow from sensors and devices to cloud applications.
- Remote monitoring and control from anywhere via dashboards or mobile apps.
- Integration of multiple devices into a unified IIOT ecosystem.
- Scalability, enabling thousands of devices to connect and operate together efficiently.

3.1.1 Common Device-to-Cloud Protocols

a) MQTT (Message Queuing Telemetry Transport)

MQTT (Message Queuing Telemetry Transport) is one of the most widely used communication protocols in IIOT. It is a lightweight publish–subscribe protocol designed for efficient data exchange between devices, especially in networks with limited bandwidth or unreliable connectivity. In an MQTT system, devices known as *clients* send (or "publish") their data to a central server called a *broker*. Other devices or applications that need this data *subscribe* to specific topics managed by the broker. This model reduces network traffic and simplifies communication management.

For example, a temperature sensor can publish its readings every second to a broker, and a cloud-based monitoring application subscribes to that topic to receive real-time updates. This allows seamless and continuous data flow without requiring direct device-to-device communication. MQTT is particularly well-suited for remote or battery-powered devices, as it consumes very little data and power.

The main advantages of MQTT include its efficiency, reliability, and resilience in challenging network environments. It ensures data delivery even when the connection is weak or intermittent, making it ideal for industrial monitoring, smart grids, and remote asset management in IIOT applications.

Think About It!

How can thousands of sensors send data continuously without overloading the network?

b) HTTP/HTTPS (Hypertext Transfer Protocol Secure)

HTTP/HTTPS (Hypertext Transfer Protocol Secure) is a request–response communication protocol widely used for web-based data exchange in IIOT systems. It allows devices to send or receive information from cloud servers using REST APIs (Representational State Transfer Application Programming Interfaces). In this setup, an IIOT device acts as a client that sends data, such as sensor readings or status updates, to a web server, which then processes, stores, or displays the information on a dashboard.

The main advantages of HTTP/HTTPS include ease of implementation and compatibility with existing web technologies, making it ideal for applications that require periodic data uploads rather than continuous streaming. Since HTTPS is the secure version of HTTP, it also provides data encryption and authentication, ensuring safe communication between devices and servers.

However, one limitation of HTTP/HTTPS is that it relies on multiple handshakes for each data exchange, which makes it less efficient for real-time or continuous data transmission. Therefore, while it is suitable for remote monitoring and data reporting applications, it is not ideal for time-critical industrial processes that demand constant connectivity and instant response.

Real-Life Connection

When you open a website or use online banking, your browser also uses HTTPS for secure communication.

c) CoAP (Constrained Application Protocol)

CoAP (Constrained Application Protocol) is a lightweight web transfer protocol designed specifically for resource-constrained devices used in IIOT environments, such as small sensors, controllers, and embedded systems. It operates similarly to HTTP but is optimized for low-power and low-bandwidth networks, making it ideal for devices with limited memory or processing capacity.

CoAP runs over UDP (User Datagram Protocol) instead of TCP, allowing faster communication with lower latency and reduced power consumption. This makes it particularly suitable for real-time applications where quick message delivery is more important than guaranteed delivery. It also supports device discovery and multicast communication, enabling one device to send data to multiple receivers simultaneously—useful in large-scale sensor networks.

For example, CoAP is often used in smart energy meters, HVAC controllers, and home automation systems, where devices send small, compact data packets like temperature or energy readings to a central server. Its simplicity, efficiency, and compatibility with IoT frameworks make CoAP a preferred protocol for connecting numerous small devices within industrial and smart infrastructure networks.

d) AMQP (Advanced Message Queuing Protocol)

AMQP (Advanced Message Queuing Protocol) is a message-oriented middleware protocol designed for reliable, secure, and structured communication between devices and systems in IIOT environments. It is commonly used in enterprise-level industrial applications, where consistent message delivery, transaction control, and data integrity are critical.

AMQP works by organizing data into queues, ensuring that every message is stored, routed, and delivered to the correct destination even if temporary network failures occur. This makes it highly dependable for systems that require guaranteed delivery and acknowledgment, such as logistics management, energy distribution, and large-scale manufacturing networks.

One of the key advantages of AMQP is its ability to prioritize, queue, and route messages intelligently based on defined rules, allowing for efficient data handling across complex industrial systems. It also supports encryption and authentication, ensuring that communication remains secure. Overall, AMQP provides a robust foundation for managing high-volume, critical IIOT data exchanges in modern industrial operations.

e) OPC UA (Open Platform Communications Unified Architecture)

OPC UA (Open Platform Communications Unified Architecture) is a machine-to-machine and device-to-cloud communication protocol widely used in industrial automation and IIOT systems. It is platform-independent, meaning it can operate across various hardware and software environments, and is designed with a strong focus on security, reliability, and interoperability.

One of the most powerful features of OPC UA is its structured data modeling, which allows machines and systems from different manufacturers to exchange information in a standardized and meaningful way. This makes it ideal for integrating complex automation systems such as PLCs, SCADA, and MES within a unified industrial network.

> ☞ *Large automobile industries use OPC UA to connect PLCs, SCADA systems, and cloud platforms together.*

f) LwM2M (Lightweight Machine-to-Machine)

LwM2M (Lightweight Machine-to-Machine) is an IoT communication protocol specifically designed for device management and telemetry in IIOT environments. Built on top of the CoAP (Constrained Application Protocol), it is optimized for lightweight, efficient communication between devices with limited power and processing capabilities. LwM2M enables industries to remotely configure, monitor, and update firmware on connected devices, reducing the need for manual intervention.

This protocol is particularly useful for smart grids, industrial gateways, and remote monitoring systems, where large numbers of devices must be maintained efficiently over wide areas. Its low bandwidth usage and support for remote control and maintenance make it a preferred choice for scalable and energy-efficient IIOT deployments.

All above mentioned protocols are summarized in Table 3.1.

Table 3.1: Types of Protocols

Protocol	**Type**	**Key Features**	**Applications**
MQTT	Lightweight publish–subscribe	Reliable in poor networks	Remote, low-power sensors
HTTP/HTTPS	Request–response	Secure but high overhead	Web APIs, simple data transfer
CoAP	Lightweight web transfer	Fast and energy-efficient	Embedded devices
AMQP	Message queue	Reliable, ordered delivery	Enterprise systems
OPC UA	Industrial automation	Secure, structured data	PLCs, SCADA, MES
LwM2M	Device management	Supports remote updates	Smart grids, gateways

3.1.2 Factors for Selecting the Right Protocol

When selecting a device-to-cloud protocol, following parameters must be considered:

- Data size and frequency (e.g., continuous vs periodic updates)
- Network bandwidth and power consumption
- Security requirements (e.g., encryption and authentication)

- Scalability for handling many devices simultaneously
- Compatibility with existing systems and cloud platforms (like AWS IoT, Azure IoT, Google Cloud IoT)

For instance, MQTT is ideal for lightweight communication in remote monitoring, while OPC UA is preferred for large-scale industrial automation that requires structured data and high security.

The real-world example is an automotive manufacturing plant, where hundreds of IIOT sensors measure parameters such as motor speed, air pressure, and temperature (Fig.3.1).

- These sensors use MQTT to publish real-time data to a cloud broker.
- The broker forwards the data to an analytics platform hosted on the cloud.
- If an anomaly is detected, the system sends an alert back to the operator dashboard using HTTPS or WebSocket communication.
- This real-time device-to-cloud link ensures that the entire plant remains connected, monitored, and optimized for performance.

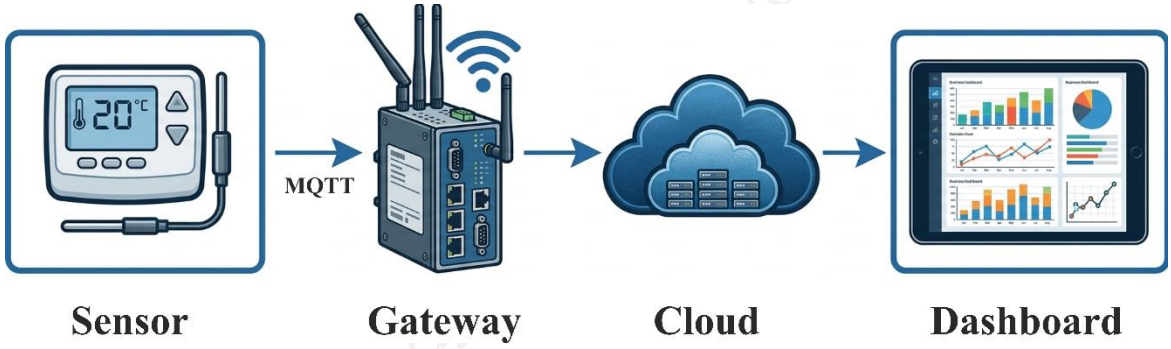


Fig. 3.1: **Real-time *Device-to-Cloud Transmission***

3.2 Device-to-Device & Gateway Protocols

In IIOT, communication is the foundation that connects sensors, controllers, machines, and cloud platforms. While cloud-based communication enables remote access and analytics, real-time operations inside a factory depend heavily on Device-to-Device (D2D) and Gateway protocols. These protocols ensure that machines can communicate instantly and that local data is efficiently shared, processed, and transmitted to higher systems when needed.

3.2.1 Understanding Device-to-Device (D2D) Communication

Device-to-Device communication allows industrial devices, such as sensors, actuators, robots, and controllers; to exchange information directly within a local network, without depending on cloud connectivity. This ensures ultra-fast response times, which is crucial in automated processes where even a one-second delay can cause errors or safety risks.

A) Characteristics of D2D Communication

- Low latency: Direct data exchange enables near-instant actions.
- Reliability: Local communication continues even if internet access fails.
- Deterministic performance: Ensures predictable timing for real-time control.
- Closed-loop control: Devices can detect an issue and react automatically.

B) Common D2D Protocols

i) Modbus RTU / Modbus TCP

Modbus RTU / Modbus TCP is one of the oldest and most reliable industrial communication protocols widely used in IIOT and automation systems. It enables the exchange of small data packets between field devices such as sensors, actuators, and controllers like PLCs. Modbus RTU operates over serial communication lines (RS-232 or RS-485), while Modbus TCP functions over Ethernet networks, offering faster and more flexible data transfer.

Despite its simplicity, Modbus remains highly effective for monitoring and control applications where real-time performance is essential. For example, in an industrial furnace system, a temperature sensor can send measured data to a PLC, which then adjusts heating levels accordingly. Its straightforward structure, wide compatibility, and ease of implementation make Modbus a standard choice for machine-to-machine communication in industrial environments.

ii) CAN (Controller Area Network)

Controller Area Network (CAN) is a robust and efficient communication protocol widely used in automotive systems, robotics, and medical equipment. It enables microcontrollers and devices to exchange data directly with each other without requiring a central computer, making it ideal for real-time control and coordination.

In a typical automotive application, the Engine Control Unit (ECU) communicates seamlessly with brake sensors, airbags, and transmission systems through the CAN bus. For instance, when the driver presses the brake pedal, the brake sensor instantly shares data with the ECU to adjust engine performance or activate safety systems. CAN's reliability, error detection, and ability to handle multiple nodes on a single network make it one of the most trusted protocols in modern embedded and automotive systems.

> ☞ *Modern vehicles contain multiple electronic systems that continuously exchange data using the CAN network.*

iii) EtherCAT and PROFINET

EtherCAT and PROFINET are high-speed, Ethernet-based communication protocols developed specifically for industrial automation and control applications. They provide real-time data exchange between controllers, sensors, and actuators, ensuring precise and synchronized machine operations. These protocols are especially useful in environments where timing and coordination are critical, such as robotic assembly lines, CNC machines, and packaging systems.

For example, in an automotive assembly plant, multiple robotic arms use EtherCAT or PROFINET networks to move in perfect synchronization; welding, assembling, or painting parts with millisecond precision. Their ability to handle large data volumes with low latency and high reliability makes them essential for modern Industry 4.0 and IIOT-based smart manufacturing systems.

> *In robotic assembly lines, even a delay of a few milliseconds may affect synchronization and product quality.*

iv) IO-Link

IO-Link is a modern digital communication standard that connects sensors and actuators directly with controllers in industrial automation systems. It enables two-way communication, allowing not only the transmission of sensor data but also diagnostic and configuration information; all through simple, low-cost 3-wire cables. This makes IO-Link both efficient and easy to implement.

Unlike traditional wiring systems that only transmit basic signals (such as ON/OFF), IO-Link allows sensors to share detailed measurement values, device IDs, and health status with the controller. It also enables remote parameterization, meaning operators can change sensor settings without physically accessing the device. This enhances predictive maintenance, reduces downtime, and simplifies troubleshooting, making IO-Link a key technology for smart factories and IIOT-enabled automation.

3.2.2 Role and Importance of Gateways

While D2D communication is excellent for local control, industries also need to connect these local networks with enterprise systems, cloud platforms, or SCADA. This is where Gateways play a critical role. They act as communication bridges between different network layers and technologies.

A) Functions of Gateways

Different functions of gateways are as follows:

1. **Protocol Conversion:** Gateways translate data between incompatible systems. For example, they can convert Modbus data into MQTT or OPC UA format for cloud transmission.
2. **Edge Processing:** Gateways can process and filter data locally before sending it to the cloud, reducing bandwidth usage and improving response time.
3. **Data Aggregation:** A single gateway can collect inputs from multiple sensors or PLCs, combine them, and send summarized information to a central system**.**
4. **Security Management:** Gateways add an additional layer of cybersecurity by implementing encryption, user authentication, and firewalls to prevent unauthorized access.

5. **Offline Operation:** If the cloud connection fails, gateways can continue local operations and sync data later, ensuring system continuity.

   For example, in a packaging line, sensors measure product presence, while motors control conveyor speed (Fig.3.2).

- Local D2D Communication: Sensors and PLCs use EtherCAT to synchronize belt speed with product flow.
- Gateway Communication: A Modbus-to-MQTT gateway sends production data to the cloud dashboard.
- If a motor overheats, the gateway triggers a local stop signal and sends an alert to the central control room.

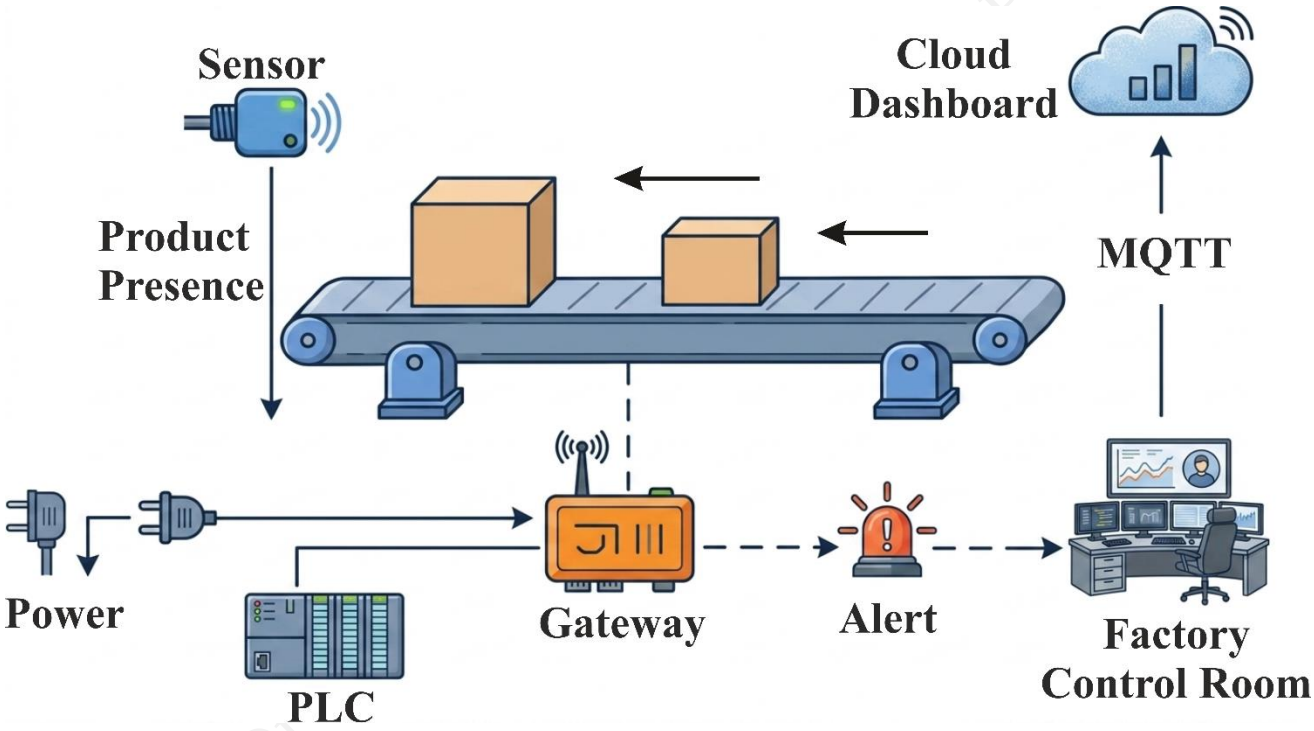


Fig. 3.2: **Smart Conveyor in a Factory**

3.2.3 Integration Between D2D and Gateways

Both types of communication, D2D and Gateway, work together in modern IIOT systems. This integrated approach allows industries to achieve real-time control while maintaining strategic visibility and predictive analytics.

- D2D ensures that operations at the machine level remain fast, stable, and coordinated.
- Gateways ensure that decision-makers at higher levels have visibility into production performance and equipment health.

Benefits of D2D and Gateway Protocols

- Fast Decision-Making: Machines respond instantly to sensor data.
- High Reliability: Operations continue even with network disruptions.
- Efficient Data Flow: Gateways filter and format data before cloud upload.
- Scalability: Easy to add new devices without redesigning the system.
- Interoperability: Supports both legacy and modern equipment.
- Enhanced Security: Encrypted data exchange protects industrial networks.

Real-World Example: Automotive Paint Booth

In an automotive paint booth:

- Temperature, humidity, and air pressure sensors communicate directly with the PLC using PROFINET.
- The PLC controls spray robots and fans locally (D2D).
- A gateway connects the PLC to a cloud dashboard using MQTT, where engineers can monitor and adjust settings remotely.
- This layered approach ensures real-time local control and global visibility, improving both efficiency and safety.

Observe Around You

Automatic doors, elevators, and smart traffic systems also depend on sensor communication.

3.2.4 Integration in Industrial Networks

Integration in IIOT networks is the process of connecting sensors, machines, controllers, and cloud systems so that they can share data seamlessly and work together as a single, coordinated ecosystem. It ensures that every component from a small temperature sensor on the shop floor to a cloud-based analytics platform, communicates effectively to support automation, monitoring, and decision-making.

Concept of Industrial Network Integration

In traditional factories, machines often worked in isolation, each with its own control system and limited data sharing. Today, IIOT integration enables interconnected networks where all devices exchange information in real time. This integration combines operational technology (OT) like sensors, PLCs, and actuators with information technology (IT) such as cloud servers, databases, and dashboards.

The result is a smart industrial environment where data flows smoothly from the production floor to the management level, supporting predictive maintenance, energy optimization, and better production planning.

Layers of Industrial Network Integration

To achieve seamless data flow, integration takes place across several layers of the IIOT architecture (*Fig.3.3)*:

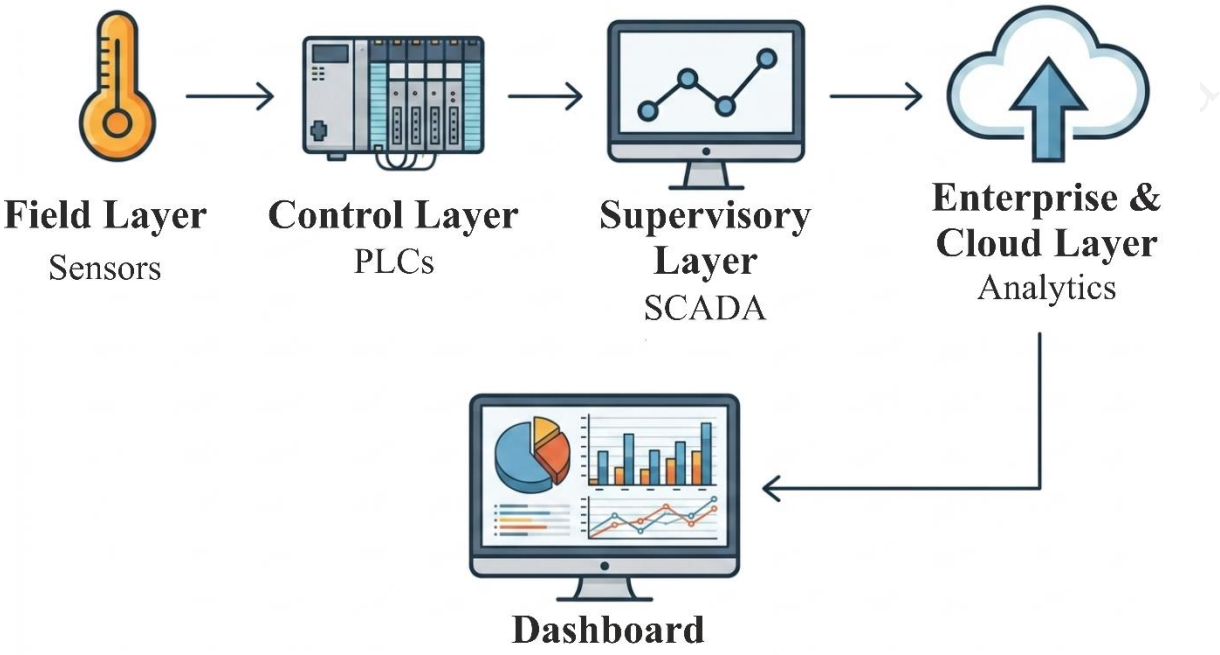


Fig.3.3: Integration of industrial networks

a) Field Layer (Device Layer)

- Includes sensors, actuators, and transducers that measure parameters like temperature, pressure, vibration, and flow.
- Devices use protocols like IO-Link, Modbus, or CAN to send data to controllers.
- Integration at this level focuses on converting physical measurements into digital data.

b) Control Layer

- Consists of PLCs (Programmable Logic Controllers), microcontrollers, and robotic controllers that make real-time decisions based on sensor data.
- Communication protocols such as EtherCAT, PROFINET, and Modbus TCP are commonly used.
- This layer ensures local automation, for example, adjusting motor speed or controlling a valve automatically.

c) Supervisory Layer

- Supervisory systems like SCADA (Supervisory Control and Data Acquisition) and HMI (Human-Machine Interface) gather and visualize real-time data from multiple controllers.
- Integration at this level allows operators to monitor equipment status, receive alarms, and control processes remotely.

d) Enterprise & Cloud Layer

- Data from the factory floor is transmitted through gateways to cloud servers or enterprise databases using protocols such as MQTT, OPC UA, or HTTP.
- Advanced analytics, machine learning, and dashboards operate here to support decision-making, maintenance scheduling, and performance tracking.

3.2.5 Importance of Integration

Proper integration of industrial networks offers several advantages:

- Real-time visibility: Managers can see what's happening in the plant at any moment.
- Efficient automation: Machines coordinate actions without human intervention.
- Predictive maintenance: Early fault detection through continuous data monitoring.
- Data consistency: Information is uniform across all levels of the organization.
- Enhanced security: Standardized and controlled communication prevents unauthorized access.
- Energy and cost savings: Optimized operations reduce waste and downtime.

3.2.6 Common Integration Challenges

While integration brings many benefits, industries must also manage challenges such as:

- Compatibility issues between old (legacy) and new (smart) equipment.
- Maintaining cybersecurity while connecting to cloud networks.
- Handling large volumes of data from multiple sensors and devices.
- Ensuring reliable communication in noisy or high-interference environments.

Example: Integrated Automotive Manufacturing Line

In an automotive manufacturing plant, integration connects multiple systems:

- Sensors measure paint thickness, temperature, and humidity in the paint booth.
- PLCs control robotic arms and conveyor speeds based on sensor inputs.
- SCADA systems monitor real-time production flow and alert operators to any issues.

- Gateways send data to the cloud, where dashboards display production metrics and energy usage.
- If a sensor detects a temperature rise beyond a set limit, the PLC automatically slows the conveyor, SCADA logs the event, and the cloud dashboard updates supervisors instantly, demonstrating full vertical integration.

PRACTICAL ACTIVITY

ACTIVITY 1:
Visualize Temperature and Humidity Data and Connect ESP32 to Node-RED Dashboard Using MQTT

Objectives:

• Understand the concept of MQTT communication
• Configure ESP32 to publish sensor data using MQTT
• Understand MQTT topics and message structure
• Use MQTT In node in Node-RED
• Display sensor data in the Node-RED Debug panel

In this activity:

- The ESP32 reads temperature and humidity from the DHT sensor
- The ESP32 publishes the data to an MQTT broker
- Node-RED subscribes to the MQTT topic and receives the data
- The received data is displayed in the Debug panel

This demonstrates communication over the same Network between Node-RED and ESP32, which is commonly used in Industrial IoT systems for machine control.

System Architecture

The system will work as follows (Fig.1):

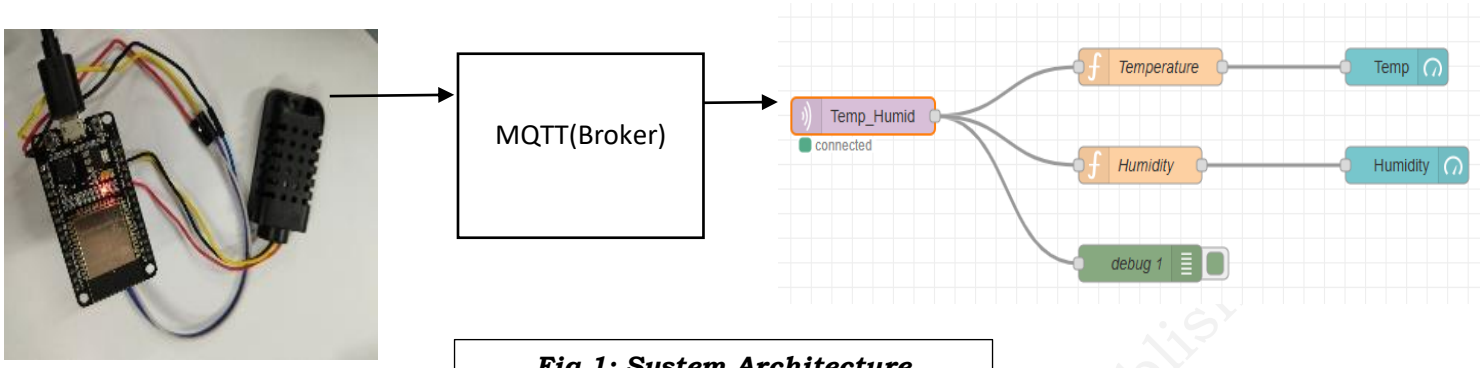


Fig.1: System Architecture

DHT Sensor → ESP32 → MQTT Publish → MQTT Broker → Node-RED MQTT In Node → Debug Panel

- ESP32 reads the sensor data and publishes it to the MQTT broker.
- Node-RED subscribes to the same topic and receives the data.

Step 1 - Components Required

Component	**Quantity**
ESP32 DevKit	1
DHT Sensor	1
Jumper wires	3
MQTT(Broker)	1
Node-RED	1

Step 2 - Hardware Connections
Connect the DHT Sensor to the ESP32 as shown below.

Component	**ESP32 Pin**
Data	GPIO 27
VCC	3V3
GND	GND

Step 3 – Install Required Libraries
Open Arduino IDE and install the following libraries:
• DHT sensor library by Adafruit
• Adafruit Unified Sensor
• PubSubClient (MQTT Library)

Step 4 – Configure MQTT Broker
Install and start an MQTT Broker such as Mosquitto.
Note: MQTT Broker can be on Cloud (e.g. HiveMQ).

Ensure:
• ESP32 and Node-RED system are connected to the same network
• Broker IP address is correctly added in the ESP32 code

Introduction to MQTT

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol widely used in IoT (Internet of Things). It allows devices like ESP32, Raspberry Pi, or sensors to communicate efficiently using a publish/subscribe model.

- Publisher → Sends messages to a topic
- Broker → Central server that routes messages
- Subscriber → Receives messages from the topics it subscribes to

MQTT is ideal for small devices because it is lightweight, fast, and reliable over local networks or the internet.

i) Install Mosquitto MQTT Broker on Windows

1. Go to the official Mosquitto website: https://mosquitto.org/download/
2. Download the latest Windows Installer (.exe).
3. Run the installer. Make sure to:
   - Install as a service (optional)
   - Include all dependencies (like OpenSSL if you want encryption)

ii) Test Mosquitto Locally

1. Open two Command Prompt windows.
2. In the first window, subscribe to a test topic:

```
mosquitto_sub -t
```

3. In the second window, publish a message:

```
mosquitto_pub -t "test/topic" -m "Hello
```

4. You should see "Hello MQTT" appear in the subscriber window, confirming the broker is working.

Step 3.

1. Change config file for Anonymous Device
2. Configuration File Location
3. On Windows, the default Mosquitto config file is usually:

```
C:\Program Files\mosquitto\mosquitto.conf
```

4. Open config file in administrator mode and change these commands and save.

```
allow_anonymous true
```

A. Allow Anonymous Connections

```
# Defaults to false, unless there are no listeners defined in the configuration
# file, in which case it is set to true, but connections are only allowed from
# the local machine.
  allow_anonymous true
```

- true → anyone on your network can connect without a password
- false → only authorized users can connect

B. Bind to All Network Interfaces

```
bind_address 0.0.0.0
```

```
#include_dir

bind_address 0.0.0.0
```

- 0.0.0.0
- → Mosquitto listens on **all network interfaces** (LAN + localhost)
- If you only want it accessible locally on your PC

Step 5 – ESP32 Code for Reading Temp and Humidity Data and Sending via MQTT.

Upload the following program to the ESP32 using the Arduino IDE:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>

#define DHTPIN 27
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
// Wifi Credentials of same Network
const char* ssid = "Admin";
const char* password = "12345678";
const char* mqtt_server = "192.168.55.113"; // IP of the System on which Broker is Running
WiFiClient espClient;
PubSubClient client(espClient);
void setup() {
  Serial.begin(115200);
  dht.begin();
  Serial.println("Connecting to WiFi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWiFi Connected");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());
  client.setServer(mqtt_server, 1883); // 1883 is the port using by mqtt
}
void reconnect() {
  while (!client.connected()) {
    Serial.print("Connecting to MQTT...");
    if (client.connect("ESP32_DHT")) {
      Serial.println("connected");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" retrying...");
      delay(2000);
    }
  }
}
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();
  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
```

To be continued...

```
  String payload = "{\"temperature\":";
  payload += temperature;
  payload += ",\"humidity\":";
  payload += humidity;
  payload += "}";
  client.publish("Temp_Humid", payload.c_str());
  Serial.println(payload);
  delay(2000);
}
```

This code performs the following functions:
• Connects ESP32 to WiFi
• Reads temperature and humidity from the sensor
• Publishes the data to MQTT topic **Temp_Humid**

Step 6– Create Node-RED Flow and Dashboard
Create the following Node-RED flow (Fig.2):

- Drag an MQTT In node into the workspace
- Configure the MQTT Broker IP Address
- Set the topic as: Temp_Humid
- Add a Debug node to display the data
- Deploy the flow

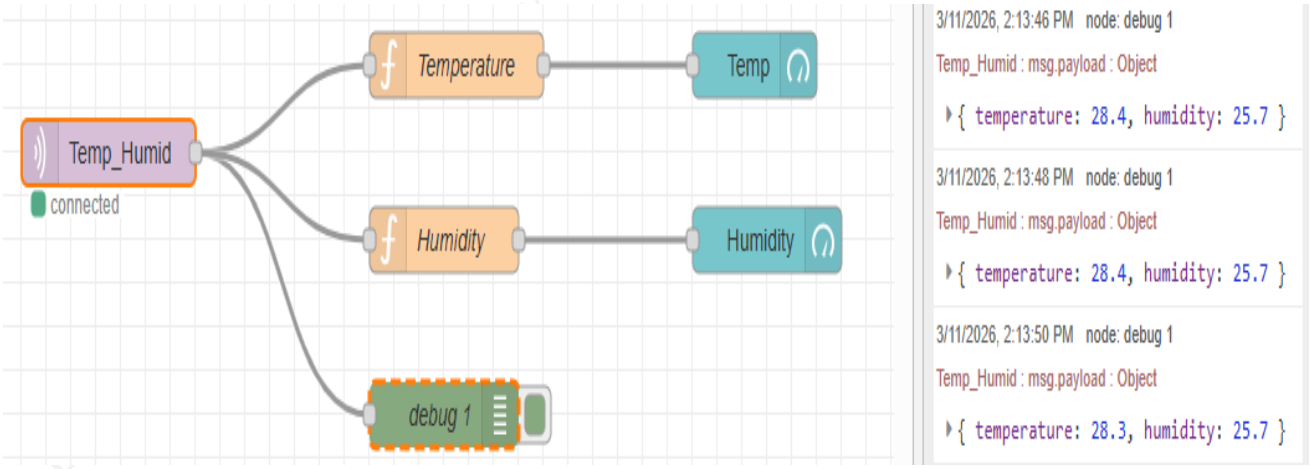


Fig.2: Node-RED Flow and Output

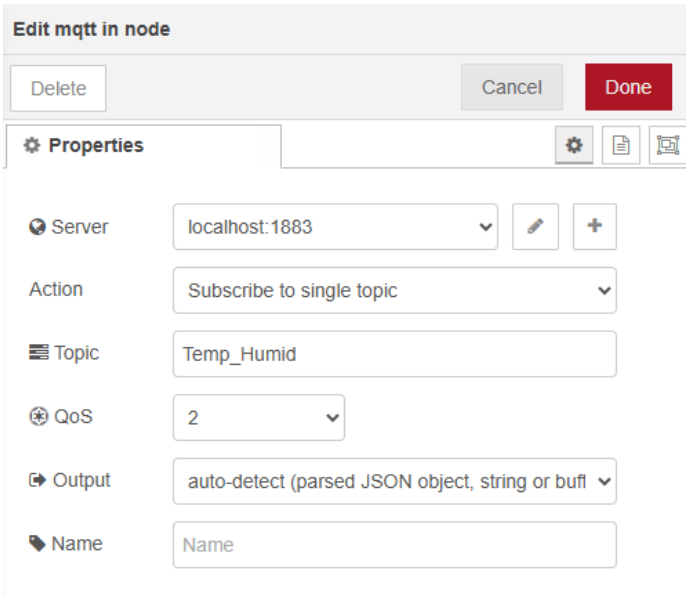


Fig.3: Edit mqtt in Node window

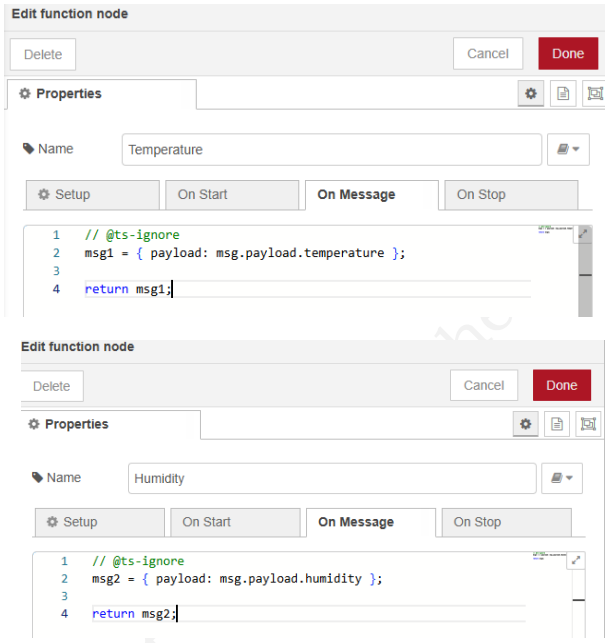


Fig.4: Edit function Node window

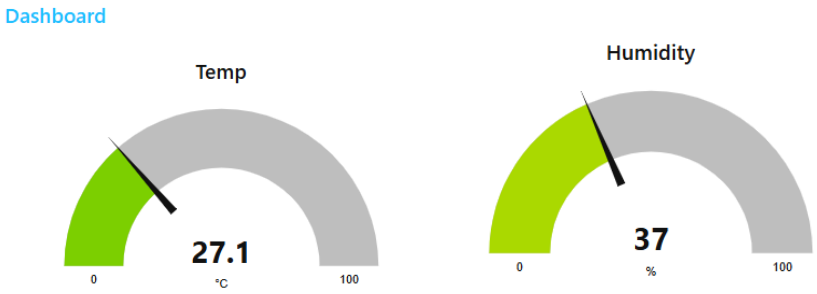


Fig.5: Dashboard

☞☞ Exercises

1. Change the MQTT topic from Temp_Humid to Sensor and update the Node-RED flow accordingly.
2. Change the data publishing interval from 2000ms to 10000ms.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. A remote sensor installed in a low-bandwidth area must send frequent updates with minimal power consumption. Which protocol should be selected?
   a) HTTP
   b) AMQP
   c) MQTT
   d) OPC UA

2. A web-based dashboard collects data periodically from devices using REST APIs. Which protocol is most suitable here?
   a) CoAP
   b) HTTP/HTTPS
   c) CAN
   d) EtherCAT

3. In an industrial system, messages must be delivered reliably even during network interruptions. Which protocol best supports this requirement?
   a) CoAP
   b) MQTT
   c) AMQP
   d) Modbus

4. A robotic assembly line requires precise synchronization between multiple machines with very low delay. Which communication type should be used?
   a) Cloud communication
   b) Device-to-Device communication
   c) Email alerts
   d) HTTP requests

5. A gateway converts Modbus data into MQTT format before sending it to the cloud. This function is known as
   a) Data visualization
   b) Protocol conversion
   c) Data storage
   d) Device control

B. Match the following

Column A	Column B
1. MQTT	A. Lightweight UDP-based protocol
2. HTTP/HTTPS	B. Structured industrial data exchange
3. CoAP	C. Message queuing with reliability
4. AMQP	D. Request–response communication

5. OPC UA	E. Publish–subscribe model

C. Fill in the blanks

1. MQTT uses a _____ model where devices publish and subscribe to topics.
2. CoAP operates over _____ protocol instead of TCP.
3. Gateways help in _____ data from multiple devices before sending it to the cloud.
4. Device-to-device communication ensures _____ response without relying on cloud connectivity.
5. OPC UA enables _____ between machines from different manufacturers.

D. Answer the following

1. A remote monitoring system needs to operate efficiently even in areas with unstable internet connectivity. Which type of protocol would you recommend and why?
2. Different machines on a production line must exchange data instantly to maintain synchronization. How can local communication methods support this requirement?
3. A factory wants to connect legacy equipment using Modbus with modern cloud platforms.
   What role can an intermediate system play in enabling this integration?
4. A company plans to expand its IIOT system by adding hundreds of new devices. Which factors should be considered while selecting a communication protocol?
5. Production data is collected at machine level and later analyzed in the cloud for decision-making. How does integration across different layers help achieve this process?

SESSION 2: MACHINE ALARM AND STATUS ANALYSIS

3.3 Understanding Machine Alarms

Think About It!

Can you imagine driving a car without a fuel gauge, temperature indicator, or warning lights? Machine alarms play a similar role in industrial systems.

Machine alarms are the first line of defense in an IIOT system. They provide real-time information about machine health and process conditions, helping industries detect problems early and take timely action. Unlike traditional systems that rely on manual inspection, IIOT-based alarms are automated, data-driven, and connected, enabling faster response and better monitoring.

A machine alarm is an automatic signal or notification generated when a machine operates outside its safe or expected limits. These alarms help operators quickly identify the location and nature of a problem.

Machine alarms play a key role in improving:

- Safety: Prevent accidents by giving early warnings about unsafe conditions.
- Productivity: Reduce unplanned downtime by allowing quick response to faults.
- Maintenance: Help technicians identify and repair problems faster.
- Data tracking: Store alarm history to find repeating issues and prevent future failures.

In modern IIOT systems, all alarms are connected to a central monitoring dashboard. Operators can see all machines and their alarm status from one screen; whether it's a red light for a serious fault or a yellow one for a warning.

Examples:

- If a motor overheats, a temperature alarm is generated.
- If a pump stops, a pressure or flow alarm is triggered.
- If a sensor fails or disconnects, a communication alarm appears.

Features of machine alarms used in modern IIOT systems are as follows:

i) Context-Aware Alarms

Modern IIOT alarm systems are context-aware, meaning alarms are generated based on actual operating conditions. For example, a high-temperature alarm may only be valid when the machine is running. This avoids unnecessary alerts when the machine is idle, reducing confusion and false alarms.

ii) Integration with Data Analytics

IIOT alarm systems are connected with data analytics tools that record:

- Time of occurrence
- Frequency of faults
- Trends over time

This data helps engineers identify recurring issues, such as repeated overheating of a motor. Over time, this information can be used in machine learning models to predict failures even before alarms occur.

Did You Know?

Modern factories use Artificial Intelligence (AI) to study alarm history and identify patterns that may indicate future failures.

iii) Alarm Communication Hierarchy

Alarms are delivered through multiple levels to ensure timely response:

- Local display on machine panels
- Centralized dashboards in control rooms
- Mobile notifications via cloud systems

This ensures that the right personnel receive alerts, whether they are on-site or working remotely.

iv) Automated Safety Actions

In advanced systems, alarms are directly linked to automatic safety responses. This helps prevent accidents and equipment damage.

Examples:

- If overheating is detected, the system shuts down the motor automatically.
- If pressure exceeds limits, a safety valve is opened.
- If a gas or chemical leak is detected, alarms, exhaust fans, and sirens are activated.

These actions reduce risk and prevent the situation from worsening.

v) Visual and Audible Alerts

To ensure quick attention, alarms use both visual and sound signals:

- Visual Indicators: Red/yellow lights, flashing icons, dashboard alerts
- Audible Indicators: Buzzers, alarms, sirens

These signals are especially useful in noisy industrial environments where immediate awareness is critical.

Observe Around You

Emergency exits, fire alarms, traffic signals, and vehicle dashboards all use visual and audible alerts to attract immediate attention.

3.3.1 How Do Machine Alarms Work?

Machine alarms work through a simple but smart process:

1. Sensing: Sensors continuously monitor machine conditions such as temperature, vibration, pressure, or current.
2. Comparison: The system compares these sensor readings with safe values (called setpoints or thresholds).
3. Detection: If any reading goes above or below the setpoint, an alarm condition is detected.
4. Notification: The system immediately sends an alert to the operator through a light indicator, buzzer, dashboard message, or mobile app notification.
5. Action: The operator or control system takes corrective steps, such as slowing down the machine, performing maintenance, or shutting it off safely.

3.3.2 Types of Machine Alarms

Machine alarms are usually divided into three main types based on their importance. This classification helps operators know which issue to fix first and how serious it is (Table 3.2).

Table 3.2: Types of Machine Alarms

Type of Alarm	Description	Example
Critical Alarm	Needs immediate action to prevent danger or damage	Motor overheating, gas leakage
Warning Alarm	Alerts about a potential problem that could become serious	Low oil level, high vibration
Information Alarm	Gives status updates or reminders	Maintenance due, machine started

3.3.3 Smart and Connected Alarms

With IIOT, machine alarms are becoming intelligent and data driven. Smart alarms can:

- Analyze sensor data to predict problems before they happen.
- Filter out false alarms caused by temporary changes.
- Automatically escalate notifications if the operator doesn't respond (e.g., send an SMS or email to a supervisor).
- Connect with cloud systems to show real-time alarms across multiple factories.

For example, in an automotive manufacturing plant, if one robot arm stops due to overheating, the alarm system instantly informs the operator, logs the event in the cloud, and notifies the maintenance engineer. This prevents larger issues and keeps production running smoothly.

3.3.4 Visualization of Alarms

Alarms are displayed in easy-to-understand formats:

- Color-coded indicators: Red (critical), Yellow (warning), Green (normal).

Did You Know?

☞ Red is commonly used for critical alarms because the human eye notices it quickly during emergencies.

- Dashboard icons or graphs: Show which machine has a problem.
- Sound or flashing alerts: Grab the operator's attention in noisy areas.

These visuals make it easier for staff to take quick and correct action.

3.3.5 Automatic Safety Response

In some cases, alarms trigger automatic responses to avoid serious damage. For example:

- If pressure is too high — the system automatically opens a relief valve.
- If temperature crosses the safe limit — the system stops the machine instantly.
- If gas is detected — exhaust fans and emergency sirens activate.

This combination of early warning and automatic control makes IIOT-based alarm systems powerful and reliable.

3.3.6 Visualizing Alarm Data

In modern machines and IIOT-based systems, a large number of alarms are generated whenever abnormal conditions occur, such as sensor faults, communication failures, or power issues. Instead of relying on lengthy and complex alarm logs, these alarms can be represented through visual tools such as graphs, charts, and dashboards. This process is known as alarm data visualization.

Visualization simplifies complex data and enables users to easily identify patterns, monitor system performance, and understand machine conditions in real time. It helps maintenance engineers and operators quickly detect issues and respond effectively, thereby improving system reliability and efficiency.

The primary objective of visualizing alarm data is to transform raw and complex alarm information into clear, meaningful insights. This enables maintenance personnel and engineers to:

- Instantly identify fault-prone machines, devices, or sensors
- Detect trends and recurring alarm patterns over time
- Analyze the sequence, frequency, and duration of faults
- Prioritize critical alarms that require immediate attention

In addition, alarm visualization provides several practical benefits:

- It helps identify machines or sensors generating frequent alarms.
- It represents the severity of issues using color coding (e.g., red for critical, yellow for warning, and green for normal conditions).
- It enables technicians to take timely corrective actions before system failure or breakdown.
- It supports preventive and predictive maintenance by highlighting repeated or emerging issues over time.

Let's Think!

Would it be easier to understand 1000 alarm records in a spreadsheet or through a graph?

3.3.7 Visualization Techniques

Several visualization techniques are employed for analyzing alarms in IIoT networks and I/O Link systems. Some common visualization techniques are as follows:

Alarm Dashboards: It is a single screen that shows all current and past alarms in a clear, color-coded form such as red for critical, orange for warning, and green for normal

Trend and Time-Series Graphs: These show how alarm frequency or intensity changes over time, allowing predictive analysis of potential equipment failures.

Alarm Timelines: A timeline shows when each alarm started and ended, making it easy to trace events in order. Presenting alarms in sequence helps in identifying the root cause and understanding how one event triggers another within the system.

Heat Maps: Display the concentration of alarms across devices or time periods, highlighting hotspots that need maintenance focus. They use colors to show which machines or time periods have the most alarms.

Pie and Bar Charts: These summarize alarms by type, device, or network area, giving a quick overview of problem sources.

3.3.8 Alarm Reporting for Maintenance Planning

Alarm reporting means creating systematic records and summaries of machine or network alarms that occur during operation. These reports are generated automatically by systems such as PLC, SCADA, or IIOT platforms that are connected to I/O Link Masters and smart devices.

Each alarm entry provides useful information such as:

- When the alarm occurred (date and time)
- Which machine or sensor caused it
- The type of fault (communication, hardware, or process-related)
- The severity of the alarm (minor, major, or critical)
- How long the fault lasted
- What action was taken or is required

These reports help the maintenance team plan and prioritize their work effectively. Alarm Reporting is Important due to following reasons:

- **Helps in Maintenance Scheduling:** Alarm reports provide a clear picture of which machines or devices face frequent problems. Maintenance teams can plan servicing or inspections based on this data.
- **Improves Fault Diagnosis:** By studying repeated alarms, technicians can identify the root causes—for example, loose connections, sensor failures, or communication issues in the network.
- **Supports Predictive Maintenance:** Alarm trends over time show warning signs before major failures occur, allowing predictive maintenance to be carried out.
- **Enhances System Reliability:** Regular monitoring and analysis of alarms prevent small issues from becoming major breakdowns, ensuring continuous production.
- **Improves Decision-Making:** Maintenance managers can use alarm reports to allocate resources, plan spare part inventories, and track team performance.

3.3.9 Structure of an Alarm Report

Reports can be generated automatically on a daily, weekly, or monthly basis and shared with engineers and management for analysis. A well-designed alarm report should be clear, concise, and data-rich, including:

- ✓ Machine or device name
- ✓ Alarm ID or type
- ✓ Time and duration of alarm

- ✓ Alarm severity level
- ✓ Possible cause
- ✓ Corrective action taken
- ✓ Remarks for future maintenance

3.3.10 Alarm Reporting in IIOT and I/O Link Systems

In an I/O Link-based IIOT environment, sensors and actuators are connected to an I/O Link Master, which continuously collects data about the device's condition.

When a fault or warning is detected, it sends alarm data through the network to the central IIOT platform or cloud dashboard.

Here, the data is:

- Stored for record-keeping
- Analyzed using software tools (like Grafana, Ignition, or Siemens MindSphere)
- Displayed visually through dashboards, charts, or graphs
- Compiled into automated alarm reports for review

These reports can show alarm frequency, locations of faulty devices, communication errors, and performance trends.

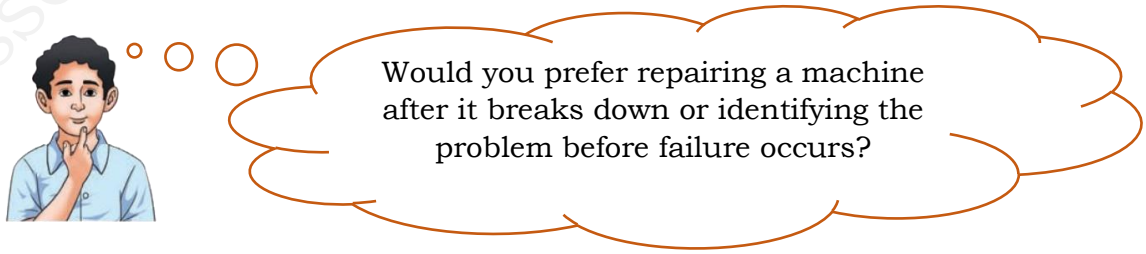
3.3.11 Types of Maintenance Based on Alarm Reporting

i) Reactive Maintenance: Maintenance performed after an alarm indicates a fault that has already occurred.

ii) Preventive Maintenance: Scheduled at regular intervals based on alarm frequency and machine usage data.

iii) Predictive Maintenance: Uses alarm history and trend analysis to forecast failures and plan maintenance before breakdowns happen.

iv) Condition-Based Maintenance: Conducted when certain alarm thresholds are reached, such as vibration limits, temperature rise, or voltage fluctuation.



3.3.12 Benefits of Alarm Reporting for Maintenance Planning

Alarm reporting helps reduce unexpected breakdowns and production losses by identifying issues early. It enables efficient use of resources by focusing maintenance efforts on critical machines. It also improves safety by detecting potential hazards in advance.

Additionally, it saves time and cost through better maintenance scheduling and minimizes emergency repairs. Alarm data provides proper documentation for audits and compliance, ensuring traceability. It also supports continuous improvement by allowing performance analysis and identifying recurring issues over time.

For example, if an I/O Link temperature sensor triggers multiple "overheat" alarms within a week, the alarm report will highlight this trend. The maintenance team can plan to:

- Inspect the cooling system,
- Check the sensor calibration, and
- Schedule preventive maintenance before a machine failure occurs.

Thus, alarm reporting helps convert raw data into actionable insights.

PRACTICAL ACTIVITY

ACTIVITY 1:
Identify Machine Alarms

Objective: To identify common machine alarms, understand their causes, and determine the appropriate corrective actions required for safe and efficient machine operation.

Procedure:

1. Study the machine alarm log provided by the instructor.
2. Identify the alarm displayed in the log.
3. Refer to the table below to understand the meaning of the alarm.
4. Determine the probable cause of the alarm.
5. Suggest the appropriate corrective action.
6. Record your observations and discuss the findings with your instructor.

Table (a): Common Machine Alarms and Corrective Actions

Alarm	What it Means	Corrective Action
Temperature High	Temperature is more than the allowed operating limits	Check Cooling System/Coolant Level/PID Controller Status
Low Lubrication	Oil or Grease Level is low due to which there is friction in moving parts	Fill Lubricant/Check Lubrication System Leakage.
Motor/Drive Overload	Mechanical System load is abruptly high	Check for any resistance/breakage in the mechanical system

Emergency Stop	Someone has stopped the machine purposefully.	There might be a safety/quality/job or tool change event due to which the emergency was engaged. Verify the same & restart the machine only after due verification.
Air Pressure Low	Air Pressure is insufficient to drive the machine sub-systems	Check Compressor Pressure/ Air Pressure Line Leakage
Door Open Alarm	Machine is stopped due to safety reason as the door is opened	Close the machine door properly before re-operating on the machine
Power Failure Alarm	Machine has lost power & abruptly stopped	Wait till Mains/DG Power to kick-in before restarting the machine
Wear Alarm	Machine Tool/Mould has been used beyond its allowed usage limits.	Replace the tool/mould before restarting the operations

☞☞ Exercise

Identify the alarm shown in the sample log and write the corresponding corrective action.

Fig.1: Sample Alarm Log of a Machine

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. An operator has to stop the machine immediately for safety reasons, what should he do?
   a) Switch Power Off
   b) Engage Emergency
   c) Both of the above
   d) None of the above

2. While the machine is under operation, the air pipe gets detached suddenly, due to which the machine stops, what possible alarm will pop up?
   a) Wear Alarm
   b) Temperature High
   c) Air Pressure Low
   d) None of the above

3. The moving mechanical slide of a machine starts making abrupt sound after which it stops suddenly, which possible alarm will pop up?
   a) Low Lubrication
   b) Power Failure Alarm
   c) Emergency Stop
   d) None of the above

4. The transformer of a factory goes into breakdown due to which the power supply goes down suddenly, which of the following alarms will likely pop up?
   a) Wear Alarm
   b) Temperature High
   c) Drive Overload
   d) None of the above

5. An operator stopped the machine to change a part, and tried restarting the machine but it did not start, which could be the possible reasons for the same?
   a) Emergency was still engaged
   b) Door was still open
   c) Could be either of a or b or both a & b
   d) None of the above

B. Match the following

Column A	Column B
1. Context-aware alarms	A. Displays alarms in visual format
2. Alarm dashboard	B. Alerts based on machine operating condition

3. Predictive alarms	C. Sends alerts to higher authority
4. Alarm escalation	D. Uses past data to forecast failures
5. Heat map	E. Shows alarm concentration visually

C. Fill in the blanks

1. Machine alarms are generated when system parameters go beyond _____ limits.
2. A _____ alarm indicates a serious condition that needs immediate action.
3. Alarm data stored over time helps in identifying _____ patterns of faults.
4. In noisy industrial environments, _____ alerts like sirens help grab attention quickly.
5. Alarm reports include details such as time, duration, and _____ of the fault.

D. Answer the following

1. A machine continues to generate alerts even when it is not running. How can alarm logic be improved to avoid such unnecessary notifications?
2. Maintenance staff notice that the same type of alarm occurs frequently on a particular machine. How can stored alarm data help in solving this issue?
3. An operator fails to respond to an important alarm during a shift. What system feature can ensure that the issue still gets addressed in time?
4. In a busy plant, it is difficult to quickly identify which machine is causing the most problems. How can visual tools assist in locating fault-prone equipment?
5. A system automatically shuts down equipment when unsafe conditions are detected. Explain how this feature improves both safety and reliability in industrial operations.

SESSION 3: ADVANCED MACHINE PERFORMANCE ANALYTICS

3.4 Key Metrics for Productivity and Reliability

Key performance metrics are essential for understanding how well machines perform and how reliable they are. These metrics help engineers and maintenance teams make data-driven decisions to improve machine uptime, reduce downtime, and enhance production performance.

Think About It!

How can a factory improve machine performance if it does not measure machine performance?

In modern industries that use IIOT systems, sensors, and automation, machines continuously generate operational data. By analyzing this data, we can measure performance through key metrics such as:

- Availability – How often the machine is operational.
- Performance Efficiency – How well it performs compared to its designed capacity.
- Quality Rate – How many products are produced without defects.
- Reliability – How consistently a machine operates without failure.

Among these, MTBF (Mean Time Between Failures) is a crucial metric for reliability analysis.

3.4.1 What is MTBF?

MTBF (Mean Time Between Failures) is the average time that a machine or system operates before a failure occurs. It helps in understanding how reliable a machine is and how often it might break down during normal operation.

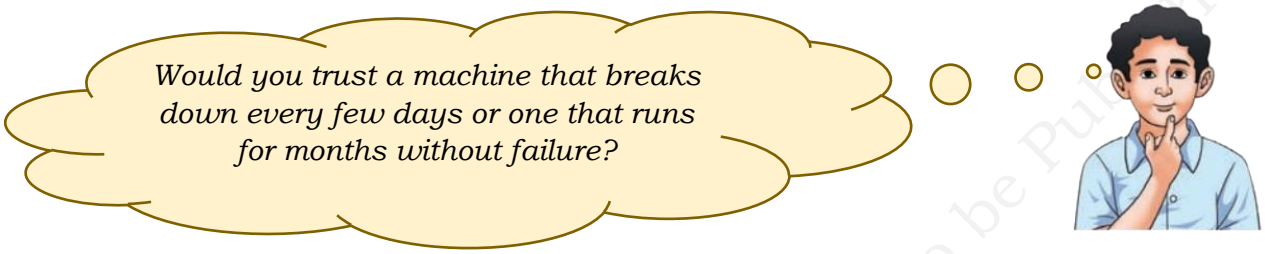
Formula:

$$MTBF = \frac{\text{Total Operating Time}}{\text{Number of Failures}}$$

For example, if a machine runs for 1,000 hours and experiences 5 failures,

$$MTBF = \frac{1000}{5} = 200 \text{ hours}$$

This means the machine operates for about 200 hours on average before a failure occurs. Example: If the operational time between 2 breakdown events is 20 days, MTBF will be 20 days. Similarly, if the total operating time in a month is 25 days & the no of breakdown events are 5, MTBF = 25/5 = 5 days



Why MTBF is Important

- Measures Reliability: A higher MTBF means the machine is more reliable.
- Supports Maintenance Planning: It helps plan preventive and predictive maintenance schedules.
- Reduces Downtime: Knowing when failures are likely to occur helps avoid unexpected stoppages.
- Improves Productivity: Reliable machines ensure continuous production with fewer interruptions.
- Helps Compare Equipment Performance: MTBF allows industries to evaluate and choose better-performing machines or components.

3.4.2 Other Related Metrics

1. **MTTR (Mean Time To Repair**): MTTR is the average time required to repair a machine after a failure. It can be calculated as:

$$MTTR = \frac{\text{Total Repair Time}}{\text{Number of Repairs}}$$

A lower MTTR means faster recovery from faults. For example, if total repair time is 120 minutes & Total Breakdown Events are 4 in a month, then

MTTR = 120/ 4 = 30 minutes

Did You Know?

Airlines, power plants, and automobile industries use MTBF extensively to evaluate equipment reliability.

2. **Availability:** It is the percentage of time a machine is available for operation. The formula to calculate availability is as follows:

$$Availability = \frac{MTBF}{MTBF + MTTR} \times 100$$

High availability shows both good reliability and quick maintenance response. For example, if a production robot in an assembly line has an MTBF of 300 hours and an MTTR of 3 hours, its Availability is:

$$Availability = \frac{300}{300 + 3} \times 100 = 99\%$$

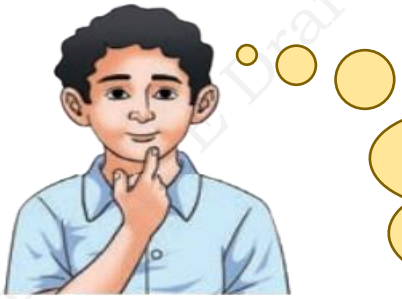
This shows that the robot is available for operation almost all the time, indicating high reliability and performance.

Can a machine be highly reliable but still have poor availability?

3. **OEE (Overall Equipment Effectiveness):** A combined metric that measures machine performance based on Availability, Performance, and Quality. It shows how effectively a machine or system is being used.

With the help of IIOT, sensors, and data analytics, industries can automatically calculate MTBF and other reliability metrics in real time as follows:

- Smart sensors record operating hours, fault occurrences, and repair durations.
- The data is sent to cloud dashboards or SCADA systems for analysis.
- Predictive algorithms identify patterns that lead to failures.
- Maintenance teams receive alerts to act before downtime happens.



Think About It!

"How were machine performance records maintained before sensors and digital monitoring systems became common?"

3.4.3 Benefits of Monitoring Key Metrics

- Enhances machine reliability and lifespan.
- Improves maintenance scheduling and planning.
- Minimizes unplanned stoppages.

- Reduces maintenance and production costs.
- Increases overall plant productivity and efficiency.

PRACTICAL ACTIVITY

ACTIVITY 1:
Calculate MTTR and MTBF of a Machine

Objective: To calculate the Mean Time to Repair (MTTR) and Mean Time Between Failures (MTBF) of machines using breakdown data and analyze their maintenance performance.

Procedure:

1. Study the monthly breakdown details provided in Table (a).
2. Count the number of breakdown events for each machine.
3. Calculate the total repair time for each machine.
4. Compute the MTTR using the given formula.

$$MTTR = \frac{\text{Total Repair Time}}{\text{Number of Repairs}}$$

5. Determine the total operating days in January 2026.
   - Consider only Sundays as non-working days.
6. Calculate the MTBF for each machine using the given formula.

$$MTBF = \frac{\text{Total Operating Time}}{\text{Number of Failures}}$$

7. Record the calculated values in Table(b).
8. Compare the MTTR and MTBF values of both machines and answer the questions provided.

Table (a): Input Data (Monthly Breakdown Details)

Machine Name	Date (DD/MM/YY)	Breakdown Reason	Total Repair Time (In Minutes)
Paint Shop	02/01/2026	Oven Fault	120
Injection Moulding	05/01/2026	Hopper Fault	60
Injection Moulding	15/01/2026	Screw Tip Damaged	30
Injection Moulding	22/01/2026	Heater Failure	45
Paint Shop	26/01/2026	Conveyor Chain Damage	240

Paint Shop	31/01/2026	Oven Fault	60
Injection Moulding	31/01/2026	Heater Failure	90

Table (b): Maintenance KPIs (Jan-25)

Machine Name	**MTTR (Jan-25) (In Minutes)**	**MTBF (Jan-25) (In Days)**
Paint Shop		
Injection Moulding		

Hint:

- Only Sundays are work-off Days for calculating total operating time in a month
- If the operational time between 2 breakdown events is 20 days, MTBF will be 20 days. Similarly, if the total operating time in a month is 25 days & the no of breakdown events are 5, MTBF = 25/5 = 5 days

On the basis of the above findings, answer the following questions:

a) Which machine is easier to repair once the breakdown has happened?
b) Which machine has higher probability of breakdown as per the trends?

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. A machine runs for long durations before experiencing any failure. Which parameter best reflects this behavior?
   a) MTTR
   b) MTBF
   c) Availability
   d) Quality Rate

2. After a breakdown, one system is restored very quickly compared to others. This situation highlights improvement in
   a) Reliability
   b) Performance efficiency
   c) Repair time metric
   d) Production rate

3. A production manager wants a single value that reflects availability, performance, and quality together. Which metric should be used?
   a) MTBF
   b) OEE

c) Availability
d) Reliability

4. Consider two machines with identical operating hours, but one fails more often than the other. What conclusion can be drawn?
   a) Both have equal MTBF
   b) One has lower reliability
   c) Both have equal availability
   d) One has higher MTTR

5. Sensors automatically record uptime, failures, and repair duration in a smart factory. What is the key benefit of using such data?
   a) Manual tracking of performance
   b) Delayed maintenance decisions
   c) Data-driven maintenance planning
   d) Increased machine downtime

B. Match the following

Column A	Column B
1. MTBF	A. Time taken to repair
2. MTTR	B. Average time between failures
3. Availability	C. Percentage of operational time
4. OEE	D. Overall equipment effectiveness
5. Quality Rate	E. Defect-free production

C. Fill in the blanks

1. The metric that measures how well a machine performs compared to its designed capacity is called _____ efficiency.
2. A machine that produces more defect-free products has a higher _____ rate.
3. The total operating time of a machine divided by number of failures gives _____.
4. A lower repair time helps in improving overall machine _____.
5. IIOT systems use _____ to automatically collect machine data like uptime and failures.

D. Answer the following

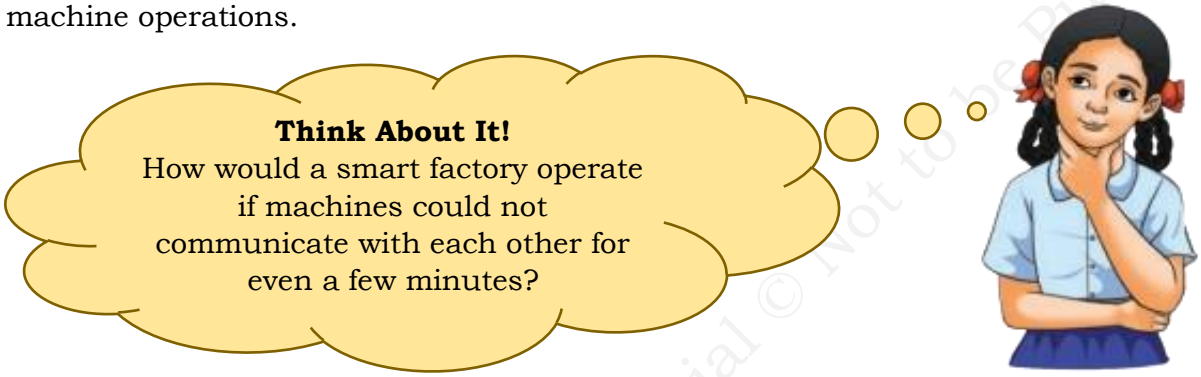
1. A machine is breaking down frequently during production. Which metric would you analyze first to understand this issue and why?
2. Two machines have the same MTBF, but one takes longer to repair. How will this difference affect their operational performance?
3. A production unit wants to improve output quality while maintaining efficiency. Which performance indicators should be monitored to achieve this goal?

4. An engineer uses real-time sensor data to track failures and repairs. How can this approach improve maintenance planning?
5. A machine shows high availability but still produces defective products. Which additional metric should be considered to evaluate overall performance?

SESSION 4: IIOT NETWORK MONITORING AND EVALUATION

3.5 Network Health Monitoring

Network Health Monitoring means continuously checking the condition and performance of all devices, connections, and communication links in an IIOT system. It ensures that data from sensors, controllers, and machines travel correctly and efficiently across the network. In simple terms, it is like a health check-up for your industrial network; helping you detect early signs of problems such as delays, connection drops, or faulty devices before they affect machine operations.



3.5.1 Importance of Network Health Monitoring

Network health directly impacts the performance and reliability of smart factories and IIOT systems. If the network is slow or unstable, machines cannot share data properly, causing errors, delays, or even system shutdowns. Following are the Key reasons for why monitoring is important:

- Ensures Continuous Communication: Prevents data loss between sensors, PLCs, and cloud platforms.
- Reduces Downtime: Detects and resolves issues before they lead to system failure.
- Improves Productivity: Keeps machines synchronized and running efficiently.
- Supports Predictive Maintenance: Identifies weak links that may cause future faults.
- Enhances Security: Detects unusual network activities that might indicate cyber threats.

3.5.2 Network Health Monitoring Parameters

To assess network health, the system monitors several technical parameters as listed in Table 3.3.

Table 3.3: Network Health Monitoring Parameters

Parameter	Purpose
Network Uptime	Measures how long the network stays active without failure.
Latency	Checks delay in data transfer between devices.

Packet Loss	Identifies data packets that fail to reach their destination.
Bandwidth Utilization	Monitors how much network capacity is being used.
Signal Strength	Checks communication quality for wireless or I/O Link devices.
Error Rates	Detects CRC errors, retries, or failed data transmissions.
Device Connectivity	Ensures all devices are properly connected and responsive.
Jitter	Measures timing variations in data flow, affecting real-time control.

3.5.3 Tools and Technologies Used

Network health monitoring in IIOT systems is supported by specialized tools and devices, such as:

- I/O Link Masters – Collect diagnostic data from field sensors and transmit to controllers.
- Network Diagnostic Tools – Like PROFINET Analyzer, Wireshark, or EtherNet/IP Scanner.
- SCADA/IIOT Dashboards – Display real-time network status and device health.
- Cloud-Based Monitoring Systems – Analyze data remotely using predictive analytics and AI.
- Edge Gateways – Process network data locally for fast decision-making.

3.5.4 Role of IIOT in Network Health Monitoring

In an IIOT environment, every device (sensor, actuator, PLC, etc.) is connected and capable of sending diagnostic information to a central dashboard.

The process typically includes:

1. Data Collection: IIOT sensors and I/O Link Masters gather device health and signal data.
2. Data Transmission: This data is transferred securely through industrial Ethernet or wireless networks.
3. Data Visualization: Dashboards display color-coded indicators — Green (Healthy), Yellow (Warning), Red (Fault).
4. Analysis and Alerts: The system automatically detects abnormal trends and sends alerts to technicians.
5. Action and Maintenance: The maintenance team can take corrective steps before the problem spreads or causes downtime.

3.5.5 Practical Examples of Network Health Monitoring

To understand how network health monitoring helps in real-world industrial environments, let's look at some common examples of communication issues and how they are identified and resolved using diagnostic tools and IIOT dashboards.

Example 1: Communication Delay in a PLC Network

In a packaging plant, machines connected through a Programmable Logic Controller (PLC) network start showing slower response times. Conveyor belts and robotic arms, which normally work in synchronization, begin to operate with slight delays, causing irregular packaging and reduced throughput.

Diagnosis: Maintenance engineers use a network diagnostic tool and observe:

- High latency between PLC and I/O modules.
- Packet loss during data transfer.
- Further inspection reveals that one Ethernet switch in the communication line is generating frequent retransmissions and signal drops.

Action Taken: The faulty Ethernet switch is replaced with a new industrial-grade switch. Network traffic is retested, showing stable communication with minimal latency and zero packet loss.

Result: Machine coordination improves instantly, production speed returns to normal, and the network health score rises from 70% to 98%.

Example 2: Sensor Communication Errors

In an automated assembly line, an I/O Link Master reports repeated CRC (Cyclic Redundancy Check) errors from one of the connected sensors. This sensor is responsible for detecting the position of a moving component, and its inconsistent data causes misalignment in the process.

Diagnosis: The maintenance team checks the device diagnostics in the I/O Link Master dashboard. The sensor's data packets are either incomplete or corrupted. Using a continuity tester, they trace the issue to a damaged sensor cable with exposed shielding near the connector.

Action Taken: The damaged section of the cable is replaced, and the sensor connection is re-established. The I/O Link Master is re-scanned, and the CRC error count drops to zero.

Result: The sensor starts transmitting data accurately, machine alignment improves, and the line resumes normal operation.

Example 3: Bandwidth Overload

An IIOT dashboard in an automotive plant shows bandwidth utilization reaching 85% during peak production hours. Data transfer between controllers, sensors, and cloud servers becomes slower, leading to delayed data visualization and alarms.

Diagnosis: Network analysis reveals that several devices are transmitting redundant data due to outdated firmware. These devices are sending unnecessary status updates every second, flooding the network with repetitive packets.

Action Taken: Engineers schedule a firmware update for the affected devices and reconfigure data reporting intervals through the network management system. The update reduces data traffic by 40%.

Result: Network bandwidth usage falls to 50%, latency is minimized, and all devices communicate efficiently without delay.

☞ *Bandwidth is like a highway. Too many vehicles on the road can cause traffic congestion and delays.*

Example 4: Wireless Signal Drop

A maintenance team receives multiple alerts about intermittent data loss from wireless vibration sensors mounted on industrial motors. The sensors are part of a predictive maintenance system that tracks vibration and temperature data in real time.

Diagnosis: Network logs show frequent signal strength fluctuations and short connection losses. Using a wireless analyzer, the team discovers that nearby Wi-Fi routers and access points are operating on the same frequency band, causing radio interference.

Action Taken: Technicians adjust the frequency channels of the wireless sensors to a less crowded band (e.g., from 2.4 GHz to 5 GHz) and reposition the antenna to minimize physical obstructions.

Result: The signal becomes stable, data loss is eliminated, and sensor connectivity reliability improves from 80% to 99%.

☞ *Metal structures, motors, and nearby wireless devices can interfere with industrial wireless communication.*

3.5.6 Network Health Indicators

To make monitoring simple, dashboards use graphical indicators such as:

- Network Health Score (0–100%)
- Color-coded status lights:
  - Healthy: All devices communicating normally.
  - Warning: Minor issues like latency or low signal.
  - Critical: Device or link failure detected.
- Trend Graphs: Show long-term performance of communication parameters.
- Automatic Alerts: Sent via email, SMS, or app notification for immediate action.

3.5.7 Integration with Maintenance Systems

Network health data can be linked with:

- Machine Performance Analytics – To see how communication affects production.

- Alarm Management Systems – To trigger alerts automatically during network faults.
- Maintenance Planning Tools – To schedule checks for switches, routers, or cables based on their diagnostic data.

3.5.8 Benefits of Network Health Monitoring

- Prevents communication failures by detecting weak links or faulty devices at an early stage
- Improves reliability by ensuring continuous and accurate data transmission
- Supports predictive maintenance by analyzing patterns to forecast potential failures
- Enhances troubleshooting by providing quick visibility into problem areas
- Increases productivity by reducing downtime and minimizing data errors
- Optimizes network performance by balancing traffic and improving efficiency
- Boosts cybersecurity by identifying unusual network behavior or potential intrusions

3.6 Comprehensive Network Evaluation

A Comprehensive Network Evaluation is the process of thoroughly assessing the performance, reliability, and security of an IIOT communication network.

It goes beyond basic troubleshooting and focuses on understanding how well the entire network (devices, cables, switches, sensors, and software) works together to support continuous and efficient industrial operations.

This evaluation helps maintenance engineers identify weak points, predict potential failures, and ensure that the network meets the required speed, data accuracy, and uptime standards for smart manufacturing environments.

The purpose of comprehensive network evaluation is to gain a complete picture of the network's condition. In short, it acts as a "health check-up" for the entire communication infrastructure as following:

- To detect hidden issues affecting performance (such as latency, data loss, or congestion).
- To verify that all IIOT devices are communicating correctly.
- To ensure the network design meets industrial reliability and safety standards.
- To plan maintenance, upgrades, or replacements based on real data.

3.6.1 Key Steps in Network Evaluation

A proper network evaluation involves following steps:

Step 1: Network Mapping

- Create a detailed network topology map showing all connected devices — PLCs, sensors, I/O Link Masters, routers, switches, and gateways.
- Identify communication paths, network zones, and data flow routes.
- This map helps visualize the structure and detect missing or misconfigured nodes.

Step 2: Connectivity Verification

- Check physical connections like cables, connectors, and ports.
- Use cable testers or link monitors to ensure signal quality.
- Confirm that all devices are online and reachable through the network.

Step 3: Performance Measurement

- Measure latency, packet loss, jitter, and bandwidth utilization using network diagnostic tools.
- Observe how the network behaves under normal and high-load conditions.
- Ensure communication speed meets industrial standards.

Step 4: Device Diagnostics

- Access device-level data from I/O Link Masters, PLCs, or smart sensors.
- Check for error messages, retries, or CRC faults.
- Identify outdated firmware or misconfigured devices causing instability.

Step 5: Network Traffic Analysis

- Use software tools (e.g., Wireshark, PROFINET Analyzer) to analyze data packets.
- Detect unnecessary broadcast traffic or loops that slow down communication.
- Separate control data from non-essential traffic to optimize flow.

Step 6: Security and Access Control Review

- Examine firewall settings, user permissions, and device authentication policies.
- Identify weak passwords or unauthorized access points.
- Ensure compliance with cybersecurity standards such as IEC 62443.

Step 7: Reporting and Recommendations

- Summarize findings in a Network Evaluation Report.
- Highlight issues (critical, moderate, minor) and propose corrective measures.
- Suggest future improvements like upgrading switches, adding redundancy, or implementing predictive monitoring.

3.6.2 Tools and Technologies Used for Comprehensive Evaluation

A comprehensive evaluation makes use of both hardware and software tools, as given in Table 3.4.

Table 3.4: Tools and Technologies Used in Comprehensive Network Evaluation

Tool Type	Examples	Purpose
Network Analyzers	PROFINET Inspector, Wireshark, EtherNet/IP Scanner	Capture and analyze data packets
Cable Testers	Fluke LinkIQ, Continuity Testers	Verify cable quality and signal strength
Diagnostic Software	Siemens TIA Portal, Schneider EcoStruxure	Check device configuration and communication
IIOT Dashboards	Ignition, ThingWorx	Visualize real-time performance
Cloud Monitoring Systems	Azure IoT Hub, AWS IoT Core	Evaluate remote network health and analytics

3.6.3 Parameters Checked During Evaluation

- Network Uptime: Measures how long the network remains continuously operational without interruption.
- Latency: Indicates the time delay between sending and receiving data packets.
- Packet Loss: Shows the number of data packets that fail to reach their destination.
- Bandwidth Usage: Evaluates how much of the available network capacity is being utilized.
- Error Rate: Displays the frequency of transmission errors or communication retries.
- Signal Quality: Tests communication strength and stability for wired and wireless devices.
- Device Health: Reports diagnostic information from sensors, actuators, and controllers.
- Security Integrity: Confirms secure data transmission, authentication, and access control.

For example, a food processing plant notices irregular communication between temperature sensors and the central PLC, leading to occasional delays in process control. In this case following actions will be taken:

1. Network Mapping: Engineers identify that the temperature sensors and PLC are connected through an intermediate unmanaged switch.
2. Performance Check: High latency and packet loss are observed on one network segment.
3. Device Diagnostics: One I/O Link Master shows repeated CRC errors and unstable voltage.
4. Security Review: Default passwords were still active on several nodes.
5. Corrective Actions:
    - Replaced the faulty switch.
    - Updated firmware on I/O Link Masters.

- Set up strong authentication policies.

After implementing these improvements, the network's reliability score increases from 75% to 98%, and the system runs with stable real-time communication.

3.6.4 Integration with IIOT Systems

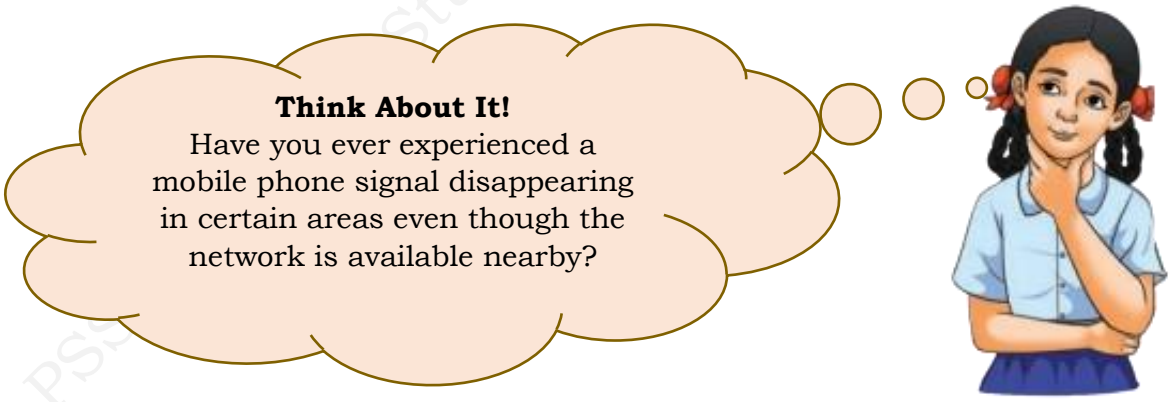
A comprehensive network evaluation complements IIOT infrastructure by:

- Providing data for predictive analytics and AI-based monitoring.
- Linking with SCADA dashboards for real-time visibility.
- Supporting edge computing nodes to process diagnostics locally.
- Enabling maintenance alerts through email or mobile notifications.

3.6.5 Benefits of Comprehensive Network Evaluation

- Improved Reliability: Early detection of hidden faults helps prevent sudden system failures
- Optimized Performance: Ensures devices communicate at optimal speed and efficiency
- Reduced Downtime: Issues are identified and resolved before affecting production
- Enhanced Security: Protects the system from unauthorized access and cyber threats
- Data Accuracy: Reliable communication ensures correct and consistent sensor readings
- Predictive Maintenance Support: Enables condition-based maintenance scheduling for network components

3.7 Identifying Connectivity Blind Spots



A connectivity blind spot is any part of the network where devices fail to communicate effectively due to physical, electrical, or configuration-related issues. Just like a blind spot in driving prevents you from seeing certain areas, a network blind spot prevents visibility of data or device status within an IIOT system.

These blind spots can exist in:

- Wired networks – due to faulty cables, loose connectors, or overloaded switches.
- Wireless networks – due to signal interference, weak antenna placement, or obstacles blocking transmission.
- Logical networks – due to misconfigured addresses, missing routes, or software errors that stop data flow.

Finding and addressing these blind spots is essential because:

- They disrupt communication between sensors and controllers.
- Lead to missing or delayed data, affecting real-time decisions.
- Increase downtime and maintenance costs.
- Create safety risks in automated systems that rely on continuous monitoring.
- Reduce network visibility, making diagnostics and maintenance difficult.

In Fig.3.4 the red/orange heat region in the center represents strong signal/coverage. **Blind spots** are the areas where signal strength is weak or absent—typically outside or at the edges of this heat zone.

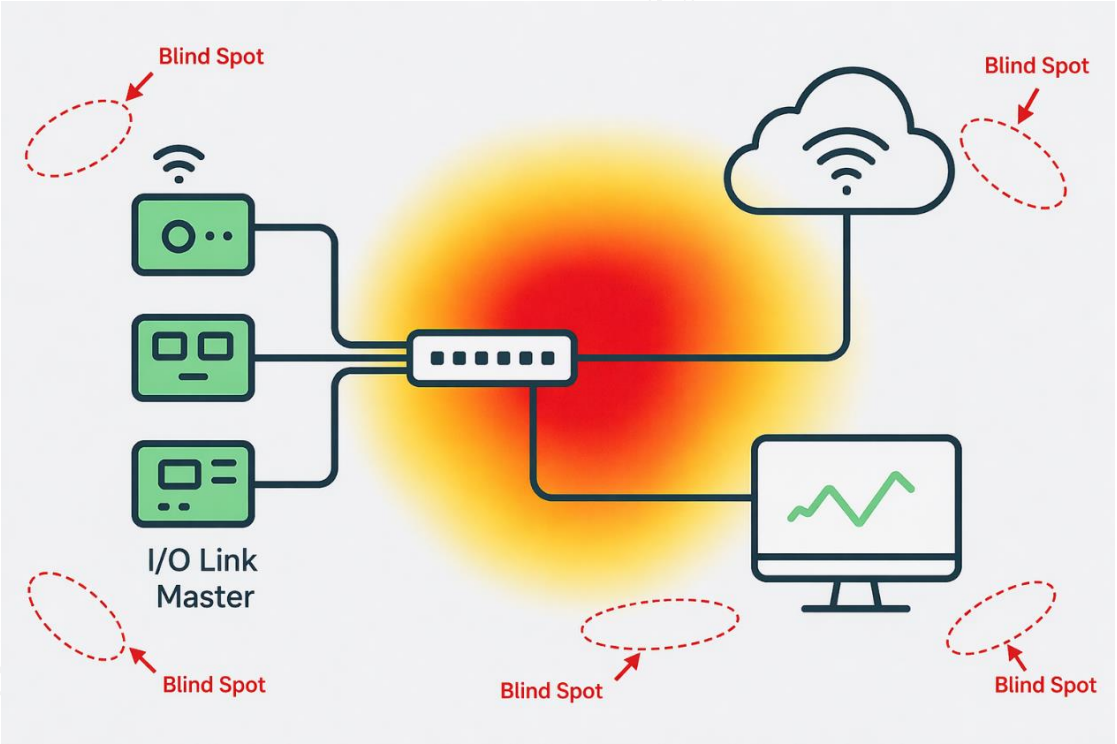


Fig.3.4: Connectivity Blind Spots

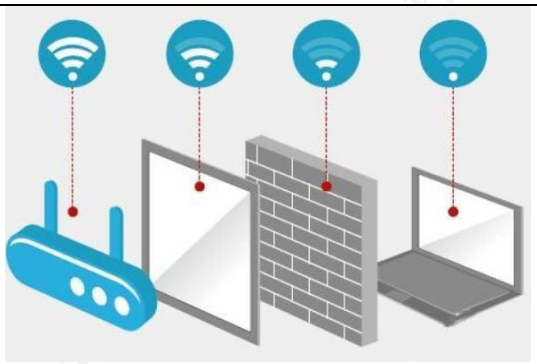
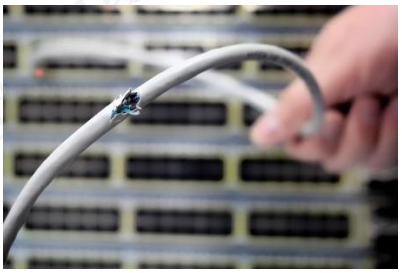
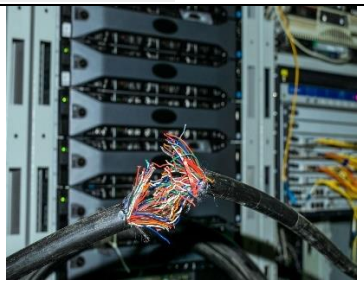
In the Fig.3.4, the blind spots can be identified as:

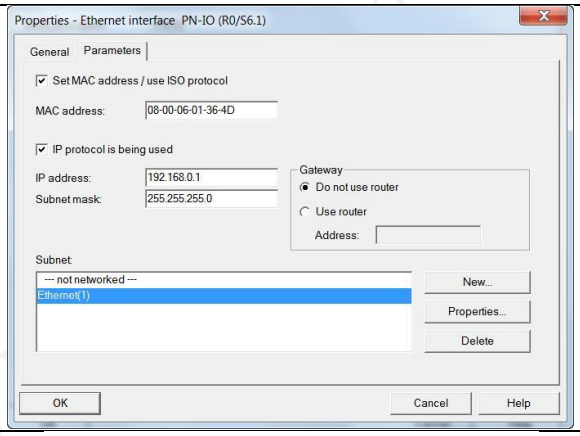
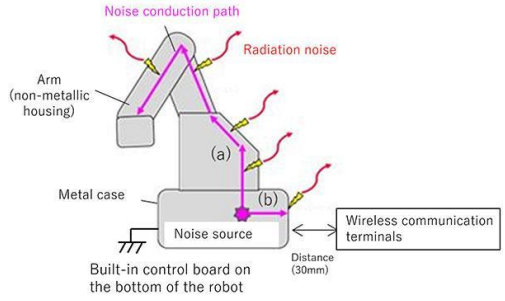
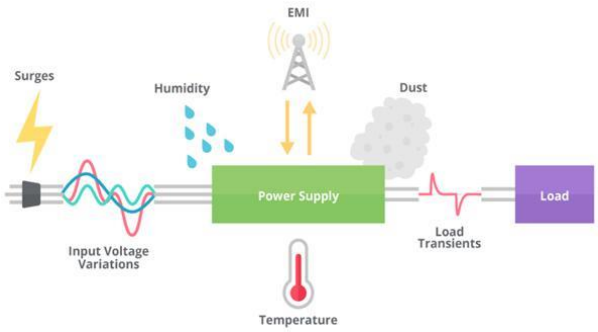
- **Outer grey regions (corners and edges):** These areas lie outside the colored heat map, indicating little to no signal coverage.
- **Far-left device areas:** The devices connected to the I/O Link Master that are located away from the central hotspot may experience weak communication.
- **Bottom-right near the monitor (outer edge):** As distance increases from the central node, signal strength reduces, creating potential blind zones.
- **Upper-right near the cloud boundary:** The region beyond the colored gradient suggests reduced connectivity with cloud communication.

3.7.1 Common Causes of Connectivity Blind Spots

The most common causes for connectivity blind spots are listed in Table 3.5.

Table 3.5: Common Causes of Connectivity Blind Spots

Cause	Description	Image
Physical Barriers	Walls, metal structures, or machinery that block wireless signals.	
Cable Faults	Damaged or poorly shielded cables causing signal loss in wired networks.	  

Loose or Corroded Connectors	Weak electrical contact interrupts data transmission.	 
Network Overload	Too many devices send data simultaneously, causing delays or dropouts.	
Improper Device Configuration	Wrong IP addresses or subnet settings leading to communication gaps.	
Interference	Overlapping Wi-Fi channels or electromagnetic noise from motors and drives.	
Power Supply Issues	Inconsistent voltage causes devices to disconnect intermittently.	

Firmware or Software Errors	Outdated or incompatible firmware affects communication protocols.	

3.7.2 Techniques for Identifying Connectivity Blind Spots

How can engineers improve communication if they do not know where weak signal areas exist?

A systematic evaluation of the network can help pinpoint these hidden communication gaps. This can be done in following way:

a) Network Mapping and Visualization

- Create a network topology map using IIOT dashboards or diagnostic software.
- Identify areas where devices appear “offline” or not sending data.
- Missing nodes or unclear connections often indicate blind spots.

b) Signal Strength and Quality Testing

- Use I/O Link diagnostics or wireless signal analyzers to check the communication strength of devices.
- Measure RSSI (Received Signal Strength Indicator) values to locate weak signal areas.
- For wireless networks, move sensors or antennas to improve line-of-sight coverage.

c) Data Flow Monitoring

- Observe real-time data updates on SCADA or IIOT platforms.
- Devices that show inconsistent or delayed updates may be located in a blind spot.
- Compare actual data flow with expected transmission rates.

d) Cable and Port Testing

- Use continuity testers or network cable analyzers to check wired connections.
- Identify high resistance, poor insulation, or damaged sections of cables.

e) Network Traffic Analysis

- Tools like Wireshark or PROFINET Inspector can detect dropped packets, retransmissions, or network congestion.
- These patterns often point toward blind spots in data flow.

f) Device Diagnostic Reports

- Modern I/O Link Masters and smart sensors provide diagnostic data on communication quality.
- Reviewing this data helps locate nodes with poor connectivity or frequent retries.

Real-World Examples of Identifying Connectivity Blind Spots

Example 1: Weak Wireless Signal in a Factory Area

A plant's vibration sensors installed near heavy machinery often lose connectivity. Network analysis reveals that metal surfaces and motor interference are weakening wireless signals.

By repositioning antennas and switching to a 5 GHz band, communication is stabilized and blind spots are removed (Fig.3.5).

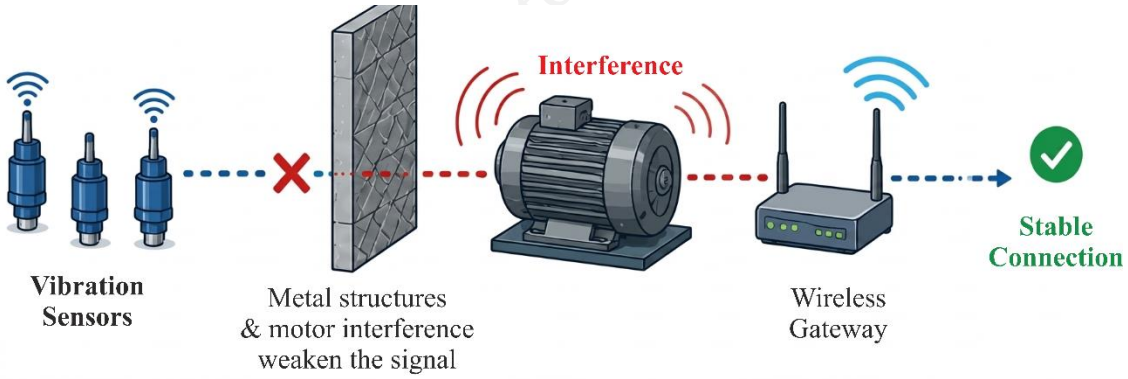


Fig.3.5: Weak Wireless Signal in a Factory Area

Example 2: Unresponsive Node in a Wired Network

An I/O Link Master reports missing data from one connected sensor. Cable testing identifies a broken wire inside the protective sleeve. After replacing the cable, the device reconnects successfully, restoring complete data visibility (Fig.3.6).

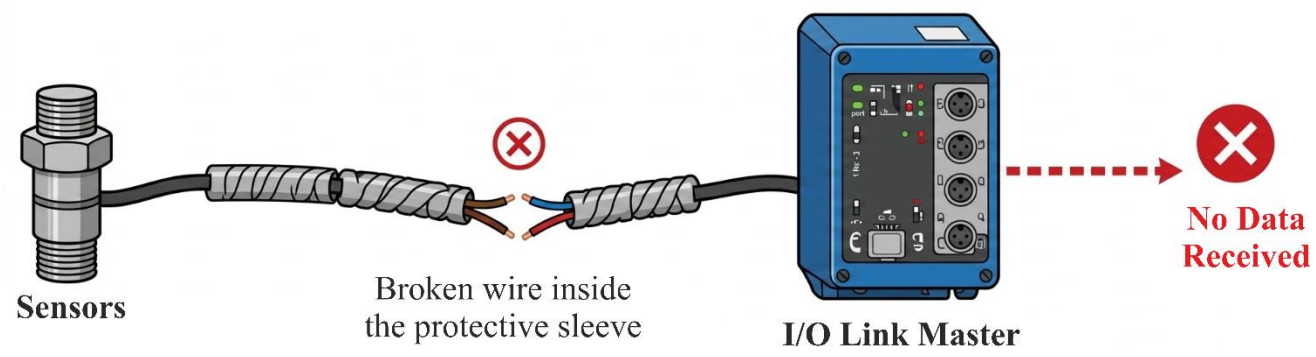


Fig.3.6: Unresponsive Node in a Wired Network

Example 3: Misconfigured Devices Causing Data Gaps

Some PLCs are not communicating with the central dashboard. Inspection shows duplicate IP addresses assigned during setup, leading to conflicts. Reassigning proper network addresses resolves the issue and eliminates data blind spots (Fig.3.7).

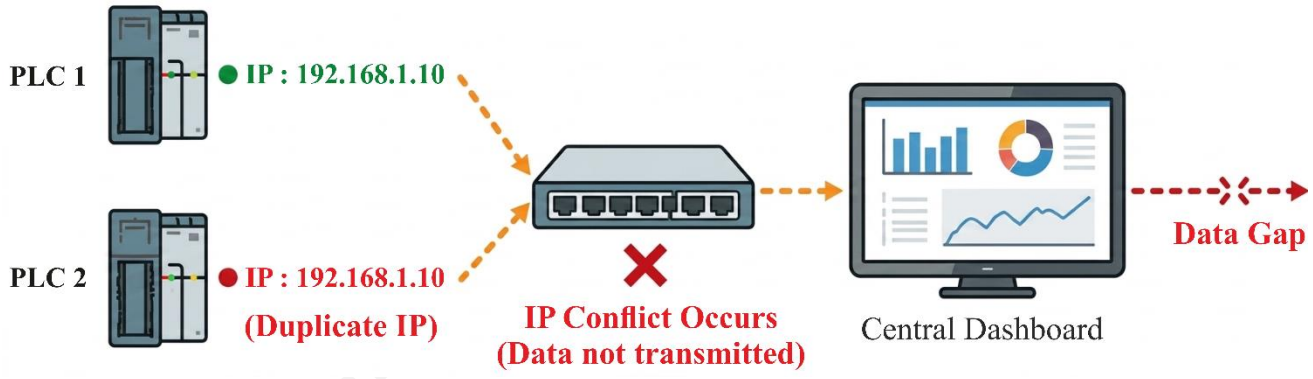


Fig.3.7: Misconfigured Devices Causing Data Gaps

3.7.3 Preventive Measures for Communication Failures and Blind Spots

Preventive measures are essential for maintaining a reliable and efficient IIOT network by minimizing the chances of communication failures and blind spots. By adopting simple yet effective practices such as proper device placement, regular monitoring, and timely updates, industries can ensure smooth data flow, reduce downtime, and enhance overall system performance. These measures help in identifying potential issues early and taking corrective actions before they impact operations (Fig.3.8).

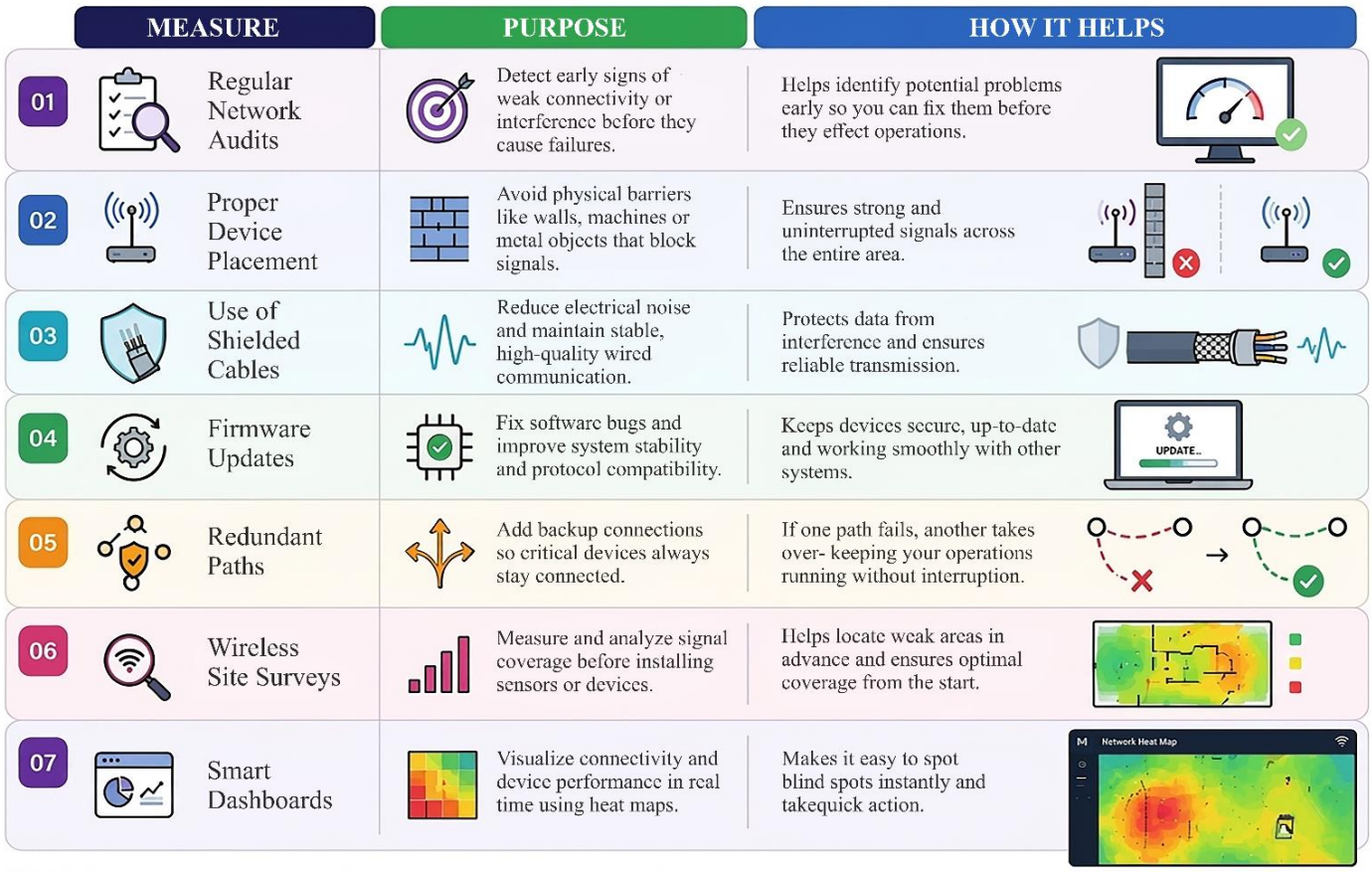


Fig.3.8: Preventive Measures for Communication Failures and Blind Spots

3.7.4 Role of IIOT in Detecting Blind Spots

IIOT systems make it easier to detect and correct blind spots automatically through:

- **Real-time visibility:** IIOT dashboards provide a live view of all connected devices, making blind spots easy to identify
- **Smart visual alerts:** Color-coded signals (red, yellow, green) quickly highlight weak or lost connections
- **Predict before failure:** Analytics detect patterns and warn about potential blind spots in advance
- **Cloud intelligence:** Stored data helps reveal recurring connectivity issues over time
- **Remote monitoring:** Engineers can detect and resolve issues from anywhere
- **Instant notifications:** Automatic alerts ensure quick response to connectivity problems
- **Smarter network planning:** Insights help improve device placement and network design
- **Reliable operations:** Reducing blind spots ensures smooth and uninterrupted system performance

3.7.5 Benefits of Identifying Connectivity Blind Spots

1. Improved Communication Reliability: Ensures all sensors and devices remain properly connected, maintaining stable and uninterrupted communication across the network.
2. Enhanced Data Accuracy: Prevents loss of important real-time data and ensures that the information received from sensors is precise and consistent.
3. Reduced Downtime: Early detection of blind spots helps avoid process interruptions and allows corrective actions before system failure occurs.
4. Better Maintenance Planning: Enables timely repair or replacement of faulty network links, supporting well-planned and efficient maintenance schedules.
5. Higher Productivity: Stable communication supports continuous and efficient operations, reducing delays and improving overall system output.
6. Optimized Network Design: Assists engineers in planning better layouts and device placements, ensuring maximum coverage and efficient use of network resources.

PRACTICAL ACTIVITY

ACTIVITY 1:
Identify a System's IP Address in an IIOT Network

Objective: To identify the IP address of the computer connected to an IIOT network using Command Prompt or Terminal commands.

Tools Required:

- Computer or Laptop connected to network
- Operating System: Windows / Linux

Procedure:

1. Connect the computer to the IIOT network through Ethernet or Wi-Fi.
2. Open the **Command Prompt** in Windows.
    - Press **Windows + R**
    - Type **cmd**
    - Press **Enter**
3. In the Command Prompt window, type the following command:

    `ipconfig` (for Windows), `ifconfig` (for Linux)

4. Press **Enter** to execute the command.

5. The system will display network configuration details as shown in the image (Fig.1).

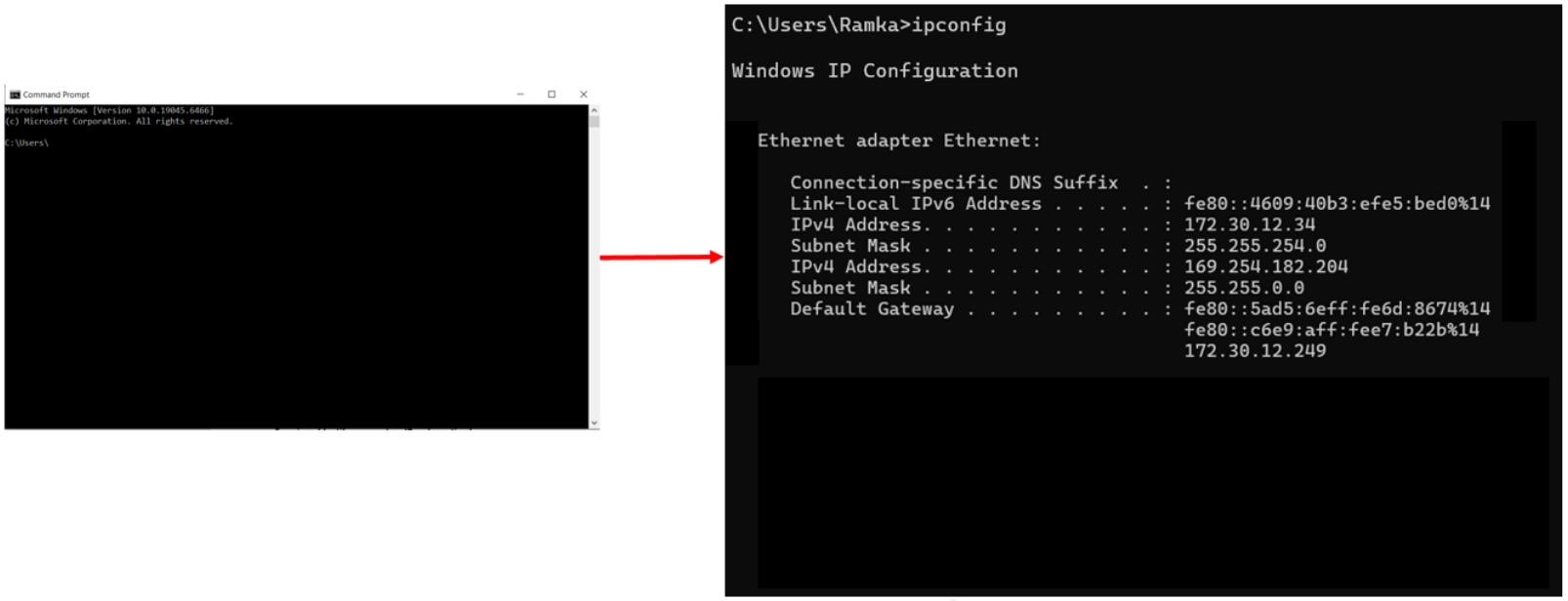


Fig.1: Screenshot of Commands to obtain system's IP Address

6. Under the active network adapter (Ethernet or Wireless LAN), observe the following details:

   - **IPv4 Address:** Current IP address of the system
   - **Subnet Mask:** Network mask
   - **Default Gateway:** Router or gateway IP address

7. In the given image, the displayed IPv4 address is: **172.30.12.34**

8. Record the IP address for network communication, IIOT device connection, or troubleshooting purposes.

ACTIVITY 2:
Identify and List All Devices Currently Connected to the Network

Objective: To scan the local network and identify all connected devices using a network scanning tool.

Tool Required: Advanced IP Scanner

Advanced IP Scanner is a network-scanning tool used to quickly discover all devices connected to a Local Area Network (LAN). It helps users identify devices such as computers, routers, printers, IoT devices, servers, and other network-enabled equipment by scanning a range of IP addresses.

It can be accessed from the official website: https://www.advanced-ip-scanner.com/

Features of Advanced IP Scanner

- Network Device Discovery: Detects all active devices connected to the same network.
- Displays Important Details: Shows IP address, device name, MAC address, and manufacturer information.
- Fast Scanning: Uses multi-threaded scanning to quickly check multiple IP addresses.
- Remote Access Tools: Provides options for Remote Desktop (RDP) and shared folder access.
- Portable Utility: Can run without installation for quick network checks.

Procedure:

1. Open Advanced IP Scanner on the computer. The main window appears as shown in the image (Fig.2).
2. In the IP address range box at the top, enter the network range to be scanned. In the image, the selected range is 192.168.137.0–254.
3. Click the Scan button (green icon at the top-left corner) to start searching for devices connected to the network.
4. Wait for the scanning process to complete. The software automatically checks all IP addresses within the selected range.
5. After scanning, the detected devices will appear in the Results section. In the image, several connected devices are listed, such as:

    - Arun-s-S23-FE.mshome.net
    - DeepakMaurya.mshome.net
    - LAPTOP-KH96IFD
    - Sumukh-PC.mshome.net
    - esp32-AB1150.mshome.net

6. Observe the details displayed for each device in separate columns:

    - Name: Device or host name
    - IP: Network IP address
    - Manufacturer: Device vendor/company name
    - MAC Address: Unique hardware address
    - Comments: Additional notes if available

7. Select any listed device to view its details or perform additional network actions if required.
8. Record the identified devices in the observation table for further analysis.

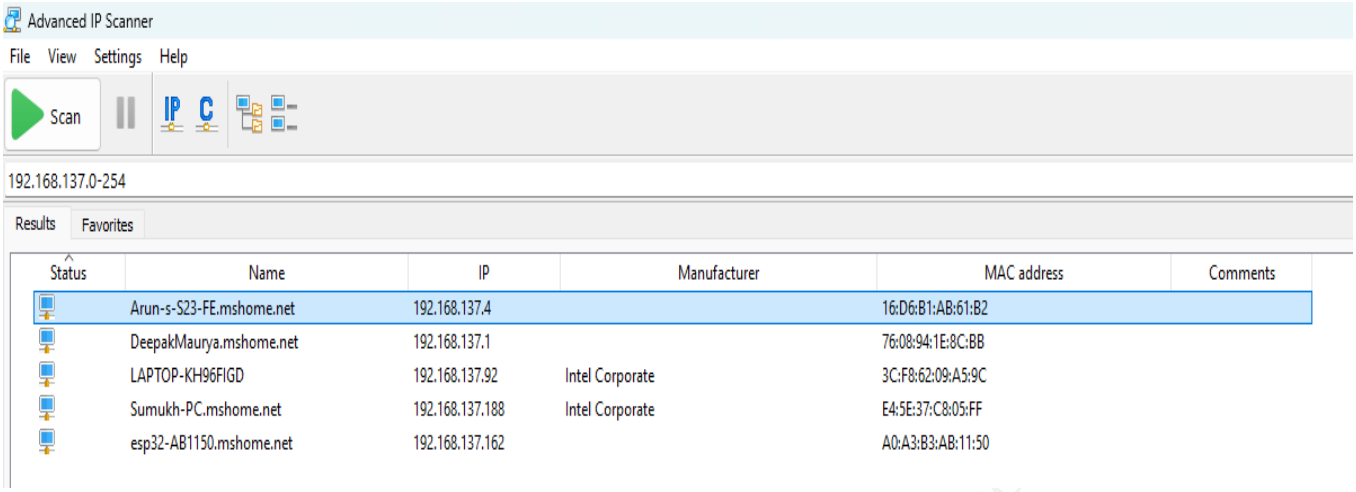


Fig.2: Advanced IP Scanner Window

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. A factory notices delays in machine coordination due to slow data transfer between devices. Which parameter should be analyzed first?
   a) Network uptime
   b) Latency
   c) Signal strength
   d) Device connectivity

2. A system is losing some data packets during transmission, causing incomplete information at the controller. Which issue is most relevant here?
   a) Jitter
   b) Packet loss
   c) Bandwidth
   d) Uptime

3. An IIOT network shows high bandwidth usage due to unnecessary repeated data transmission. What corrective step should be taken?
   a) Increase packet loss
   b) Reduce data frequency or optimize traffic
   c) Disable monitoring
   d) Increase latency

4. Wireless sensors in a plant frequently disconnect due to interference from nearby devices. Which parameter is most affected?
   a) Signal strength
   b) MTBF

c) Availability
d) OEE

5. A maintenance team uses dashboards showing green, yellow, and red indicators to monitor network condition. What is the main advantage of this approach?
   a) Complex data storage
   b) Quick visual understanding of network status
   c) Increased network traffic
   d) Reduced communication

B. Match the following

Column A	Column B
1. Latency	A. Variation in packet timing
2. Packet Loss	B. Delay in communication
3. Jitter	C. Lost data during transmission
4. Bandwidth Utilization	D. Network capacity usage
5. Network Uptime	E. Duration of active network

C. Fill in the blanks

1. Network health monitoring helps detect _____ problems before they affect operations.

2. CRC errors indicate issues in _____ transmission.

3. A network analyzer tool like Wireshark is used to study _____ flow.

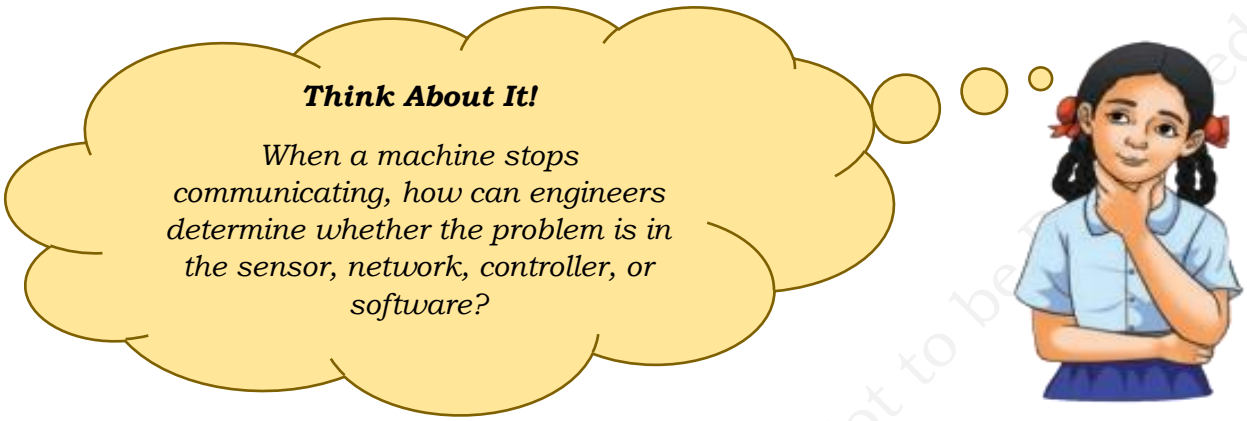
4. High jitter can disturb _____ control in industrial systems.

5. A weak signal in wireless systems may be caused by physical _____ like walls or machinery.

D. Answer the following

1. A production line starts showing delays in machine response. How would you use network parameters to identify the root cause?
2. A sensor frequently disconnects from the system in a wireless setup. What steps can be taken to improve its connectivity?
3. Engineers observe that network traffic increases significantly during peak hours. How can this issue be managed to maintain system performance?
4. Some devices in a network are not sending data to the dashboard. How can network mapping help in resolving this issue?
5. A plant wants to prevent future communication failures in its IIOT network. How can continuous monitoring and analysis support this goal?

SESSION 5: IIOT NETWORK DIAGNOSTICS, TROUBLESHOOTING AND OPTIMIZATION



3.8 IIOT Network Diagnostics

Effective diagnosis is the first and most critical step toward resolving IIOT network issues. Since IIOT systems involve numerous sensors, controllers, gateways, and cloud services, identifying the root cause of a problem requires a systematic approach.

Common IIOT network problems can be broadly categorized into:

- Physical Layer Issues: Cable faults, damaged connectors, or electrical noise.
- Data Link Layer Issues: Collisions, bandwidth overload, or switch misconfiguration.
- Network Layer Issues: IP conflicts, routing errors, or subnet mismatches.
- Application Layer Issues: Data packet loss, delayed responses, or protocol mismatches (e.g., MQTT, Modbus, OPC-UA).

3.8.1 Common Network Problems in IIOT Systems

In IIOT environments, network reliability is essential for seamless communication between sensors, controllers, gateways, and cloud platforms. However, due to the complex nature of interconnected systems, several network problems can arise, affecting data flow and overall system performance. Understanding these issues and their causes helps in timely diagnosis and resolution.

a) Intermittent Connectivity

Intermittent connectivity is one of the most common issues in IIOT networks, where devices frequently drop offline and reconnect automatically. This problem usually indicates instability in the communication path. It can be caused by faulty Ethernet cables, unstable Wi-Fi signals, or electromagnetic interference from nearby industrial equipment.

To diagnose this, engineers should start by checking physical connections and observing link lights on switches and devices. A network analyzer can be used to monitor packet drops or connection interruptions. Additionally, conducting a simple ping test helps assess latency and the stability of device communication.

b) High Latency or Delay

High latency refers to the delay in data transmission between devices such as sensors and controllers, leading to slow response times in automation systems. This typically results from network congestion, overloaded switches, or excessive data traffic within the system.

To diagnose latency problems, round-trip times can be measured using tools like ping or traceroute to identify where delays occur. Analyzing network traffic helps locate bottlenecks, while enabling Quality of Service (QoS) prioritization ensures that critical machine data is transmitted without unnecessary delay.

c) Packet Loss and Data Corruption

Packet loss occurs when some data packets fail to reach their destination, while data corruption means that transmitted data becomes unreadable or inaccurate. Both issues can cause incomplete or missing sensor data on dashboards. The root causes often include electromagnetic interference (EMI), damaged cables, or faulty nodes in the network.

Diagnosing packet loss involves checking CRC (Cyclic Redundancy Check) error logs on I/O link masters or PLCs, testing cable integrity using Ethernet testers, and isolating or replacing defective network segments to restore stable communication.

d) IP Conflicts and Addressing Errors

In IIOT networks, every device requires a unique IP address for communication. When two devices are assigned the same address, or if subnet masks are configured incorrectly, IP conflicts arise—causing devices to fail to connect or respond. Misconfigured DHCP servers can also assign incorrect addresses.
Diagnosing these issues involves scanning the network to detect duplicate IPs, verifying IP allocation policies, and ensuring that critical devices use unique static IP addresses. Correcting addressing errors ensures reliable connectivity and prevents random disconnections.

e) Protocol Mismatch or Incompatibility

IIOT networks often include devices from different manufacturers that may use varying communication protocols such as Modbus, MQTT, or OPC-UA. When incompatible protocols or outdated firmware are used, devices fail to communicate effectively.

To diagnose such issues, engineers should review device documentation, verify protocol configurations, and ensure compatibility across systems. Protocol analyzers or converters can help establish communication between different systems. Updating device firmware to the latest supported version also resolves many compatibility issues and enhances interoperability.

3.8.2 Systematic Diagnostic Approach

A systematic diagnostic approach helps technicians identify faults in a logical and efficient manner, reducing time and unnecessary troubleshooting efforts. By following step-by-step methods, problems can be detected accurately, corrected quickly, and prevented from recurring in the future (Fig.3.9).

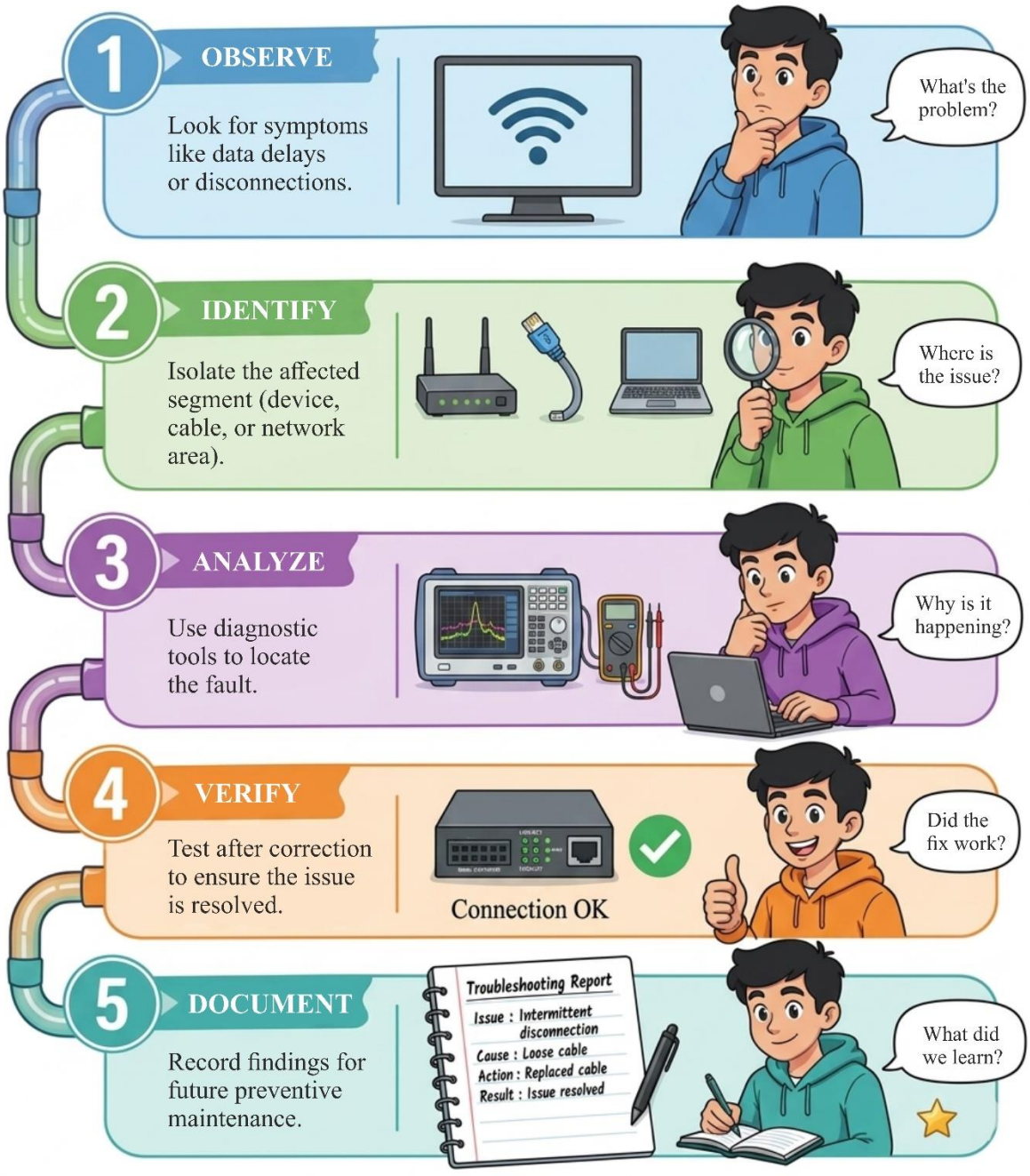


Fig.3.9: Systematic Diagnostic Approach

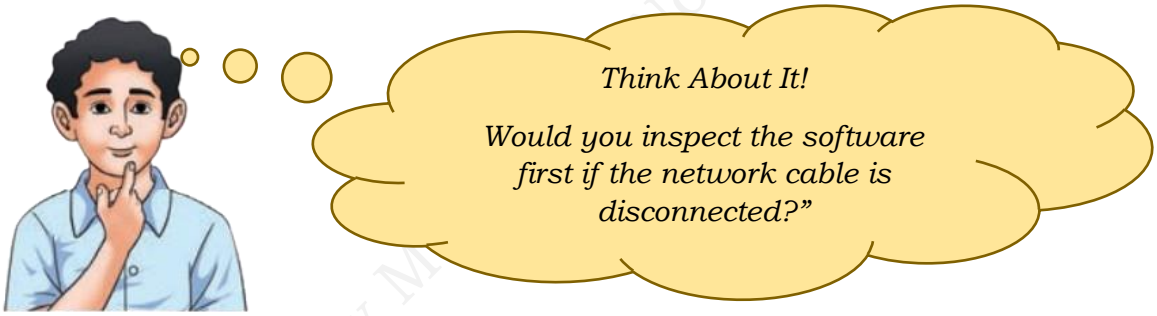
3.8.3 Diagnostic Tools and Techniques

To accurately detect and analyze problems, engineers use specialized diagnostic tools such as:

- Network Sniffers (e.g., Wireshark): For analyzing real-time traffic and packet structure.
- Protocol Analyzers: To decode and troubleshoot specific IIOT protocols.
- Network Mapping Tools: To visualize connectivity and detect unresponsive nodes.
- Performance Dashboards: To monitor latency, bandwidth, and device health metrics.

☞ Wireshark is one of the most widely used network analysis tools in both industry and cybersecurity.

3.8.4 Troubleshooting Strategies



In an IIOT system, hundreds of devices such as sensors, actuators, PLCs, gateways, and cloud servers work together through wired and wireless communication networks. When a problem occurs, it can stop data flow, reduce efficiency, or even halt production.

Therefore, troubleshooting strategies must be systematic, data-driven, and preventive. The goal is not only to fix issues quickly but also to understand why they happened and how to prevent them in the future.

i) Layer-by-Layer Troubleshooting Approach

A structured troubleshooting process starts by checking each communication layer—from hardware to software:

(a) Physical Layer

- Inspect cables, connectors, and power supplies.
- Check for broken wires, loose connections, or damaged ports.
- Ensure that network devices (routers, switches, I/O Link Masters) are powered and LEDs indicate normal operation.

(b) Data Link Layer

- Examine switch performance and port utilization.
- Verify MAC addresses and ensure no data collisions occur.
- Use I/O Link diagnostics to check link errors or signal interruptions.

(c) Network Layer

- Verify IP configurations, subnet masks, and gateway addresses.
- Identify duplicate IPs or incorrect routing paths.
- Use “ping” or “traceroute” to locate unreachable nodes.

(d) Application Layer

- Check communication between software systems (HMI, SCADA, Cloud).
- Validate data formats and protocol compatibility (MQTT, Modbus, OPC-UA, HTTP).
- Ensure security configurations or firewalls are not blocking communication.

Example: When a temperature sensor fails to update readings on the dashboard, checking power supply, network connectivity, and protocol configuration step-by-step can isolate the exact cause.

ii) Proactive vs. Reactive Troubleshooting

Reactive Troubleshooting:

Occurs after a failure is detected. Engineers respond to alarms, errors, or system downtime.

For example, replacing a burnt-out switch after communication failure.

Proactive (Preventive) Troubleshooting:

Involves continuous monitoring and diagnostics to predict failures before they happen.

For example, replacing a cable showing increasing signal loss before it causes downtime.

A mix of both ensures reliability and reduced unplanned stoppages.

iii) Diagnostic and Monitoring Tools

Using the right tools helps in faster and more accurate fault detection:

- Ping / Traceroute: Checks network reachability and latency.
- Wireshark / Packet Analyzer: Captures and analyzes communication data.
- Protocol Analyzer: Verifies correctness of IIOT protocols.
- Network Monitoring Dashboard: Visualizes bandwidth, signal strength, and device status.

- I/O Link Master Software: Monitors device communication, error counts, and sensor health.
- Thermal Camera / Multimeter: Detects overheating, loose wiring, or electrical faults.

For example, a spike in CRC errors detected by the I/O Link Master software may indicate a damaged cable or electromagnetic interference.

iv) Root Cause Analysis (RCA)

Troubleshooting should not stop at fixing the issue, but it should identify the underlying cause to prevent recurrence.

Steps for RCA:

1. Define the problem – What exactly went wrong?
2. Collect data – Gather logs, alarm reports, and user inputs.
3. Analyze cause and effect – Use "5 Whys" or Fishbone Diagram.
4. Implement corrective action – Fix the immediate issue.
5. Implement preventive action – Modify procedures or configurations to avoid recurrence.

For example, frequent sensor disconnections were traced to electromagnetic interference (EMI) near the wiring. Installing shielded cables resolved the issue and ensured stable, reliable communication.

v) Isolation and Substitution Method

When multiple devices are affected, isolate one network segment at a time to find the fault.

- Disconnect parts of the system to test smaller sections.
- Substitute components (e.g., replace one cable or sensor at a time) to identify the faulty element.

For example, if a whole sensor cluster fails, connecting each sensor individually to the I/O Link Master helps identify the defective one.

vi) Comparison with Reference Systems

If a similar line or machine is working properly, compare its network configuration with the faulty one.

- Check for firmware version differences.
- Match IP addresses, switch settings, or software parameters.
- Use configuration backups to restore normal operation.

For example, comparing a failed production line with another functional one may reveal a firmware update missing in one device.

vii) Documentation and Knowledge Sharing

Every troubleshooting activity should be recorded and documented:

- Log the date, issue description, tools used, and corrective actions.
- Maintain a digital troubleshooting logbook or knowledge base.
- Helps in faster problem-solving for similar future issues.

For example, *a* record showing repeated switch overheating in a specific area can guide future network redesign or load balancing.

viii) Continuous Learning and Predictive Strategies

Modern IIOT systems support predictive maintenance using data analytics:

- Machine learning algorithms identify unusual patterns in latency, packet loss, or device health.
- Alerts are generated before actual failure occurs.
- Engineers can schedule maintenance based on data insights instead of fixed schedules.

For example, the IIOT dashboard predicts that a network router is nearing its maximum load capacity. Engineers plan a replacement before it causes downtime.

ix) Team Collaboration and Safety

Troubleshooting is not just a technical task; it requires team coordination and safety awareness:

- Always follow electrical safety procedures while inspecting live systems.
- Collaborate with IT, maintenance, and operations teams to share data and observations.
- Use standardized communication protocols and report formats.

For example, *a* maintenance engineer collaborates with the IT team to trace an IP conflict between a PLC and HMI server.

Effective Troubleshooting Steps are summarized in Table 3.6.

Table 3.6: Troubleshooting Steps

Step	Action	Purpose
1	Observe the problem	Identify visible symptoms
2	Collect data	Gather logs, alarms, and status reports
3	Isolate the issue	Narrow down affected devices or areas
4	Analyze root cause	Find underlying reasons

5	Apply corrective action	Fix the fault
6	Verify performance	Ensure normal operation is restored
7	Document and share	Build a reference for future troubleshooting

3.9 Preventive Measures & Network Optimization

In modern industrial environments, where machines, sensors, and controllers communicate continuously, even a small network issue can disrupt production or cause costly downtime. Therefore, preventive measures and network optimization are two essential practices that ensure smooth, reliable, and efficient functioning of the IIOT ecosystem.

Preventive measures aim to identify and eliminate potential faults before they cause problems, while network optimization focuses on enhancing the performance, stability, and scalability of communication systems. Together, they improve machine uptime, safety, and productivity while reducing maintenance costs.

In IIOT, even minor communication failures can cause serious disruptions in production. Therefore, preventive measures and network optimization are key strategies to maintain continuous operation, reduce downtime, and improve overall system performance.

Preventive measures focus on early detection and correction of potential problems, while network optimization ensures that the IIOT system operates efficiently, securely, and with maximum reliability. Together, these practices help industries achieve smooth data flow, faster response times, and long-term network health.

3.9.1 Preventive Maintenance Practices

Preventive maintenance is about proactively checking and servicing network components before they fail. It includes the following:

- Regular inspections of cables, switches, and connectors help prevent unexpected disconnections.
- Periodic calibration of sensors ensures data accuracy.
- Scheduled firmware updates prevent compatibility issues and enhance performance.
- Cleaning and tightening connectors reduce the chances of data loss due to poor physical contact.

For example, a manufacturing unit reduced 30% of unplanned downtime by introducing a monthly inspection routine for all I/O link devices and network hubs.

3.9.2 Continuous Network Monitoring

Constant monitoring of network health plays an important role in the early detection of anomalies and communication issues. It helps maintain reliable data flow, improves overall system performance, and enables timely corrective actions before faults become serious.

To achieve this, network monitoring software is used to track important parameters such as latency, packet loss, bandwidth usage, and data throughput. The collected information is displayed through real-time dashboards, which provide clear visibility into device status, performance, and network connectivity. In addition, alerts and notifications automatically inform operators about abnormal conditions, allowing quick response before they develop into critical failures.

For example, a dashboard detecting rising packet errors helped engineers find a loose Ethernet connector before a complete communication breakdown occurred.

3.9.3 Firmware, Software, and Configuration Management

A properly managed and regularly updated software environment helps prevent communication failures and improves overall network reliability. Correct firmware versions, backed-up configurations, and controlled updates ensure smooth operation of IIOT devices and industrial networks.

To maintain stability, all firmware and software versions should be updated periodically to fix bugs, improve security, and enhance compatibility. Configuration files must be backed up regularly so that systems can be restored quickly after failures or device replacement. In addition, maintaining clearly labeled version history helps avoid mismatched device communication caused by incompatible software versions.

For example, automated firmware updates through the IIOT cloud reduced configuration errors and saved maintenance time by 50%.

3.9.4 Network Optimization

Network optimization focuses on improving the efficiency, speed, and reliability of data communication. It involves adjusting configurations, reducing congestion, and enhancing system responsiveness.

(a) Traffic Management and Load Balancing

Effective traffic management ensures that critical industrial data receives higher priority than non-essential traffic. This improves communication speed for important control systems and prevents delays during peak network usage.

Quality of Service (QoS) rules can be applied to prioritize control commands, alarms, and PLC communication over general data traffic. Load balancers distribute network demand evenly across devices or servers, preventing overload on a single node. Non-essential data transfers may also be scheduled during off-peak hours to maintain network stability.

Example: Prioritizing PLC control data using QoS improved robotic assembly timing accuracy by 20%.

> 👉 QoS works like a traffic police officer, giving priority to emergency vehicles over regular traffic.

(b) Network Segmentation and Topology Design

A well-designed network structure improves reliability, reduces interference, and limits faults to smaller sections of the network.

The network can be divided into logical segments such as production, monitoring, and control using VLANs. Redundant communication paths such as ring or mesh topology provide backup routes during failures. Edge computing nodes may also be used to process data locally, reducing latency and dependence on central servers.

Example: An automotive factory adopted a ring topology, ensuring that no data was lost during a switch failure.

(c) Optimizing Physical Connections

The physical layer of the network must be maintained properly to achieve stable communication and strong signal quality.

Shielded cables should be used to reduce electromagnetic interference (EMI), especially near motors and drives. Cable lengths must remain within recommended limits for better signal strength. Worn-out connectors should be inspected and replaced regularly, and network devices should be arranged systematically to reduce electrical noise.

Example: Frequent disconnections near motor drives were resolved by installing shielded twisted-pair cables, restoring reliable connectivity.

(d) Data Flow Optimization

Efficient handling of data traffic helps reduce bandwidth usage and improves network responsiveness.

Continuous data streaming can be replaced with event-based communication for non-critical signals. Data compression techniques help save bandwidth, while edge-level data caching reduces repeated requests to central servers.

Example: Event-driven reporting reduced bandwidth usage by 40% in a sensor network without affecting performance.

(e) Protocol and Communication Optimization

Selecting suitable communication protocols improves speed, reliability, and compatibility in IIOT systems.

Efficient protocols such as MQTT or OPC UA should be used for industrial applications. Redundant polling or unnecessary broadcast messages must be minimized. Refresh rates and communication cycles should be adjusted according to process requirements.

Example: Switching from HTTP to MQTT reduced network latency from 150 ms to 40 ms in a production monitoring system.

3.9.5 Security and Access Control

A secure network is also a reliable network. Protection against unauthorized access and cyber threats helps maintain uninterrupted communication.

Firewalls and VPNs should be used to secure data transmission. Multi-factor authentication (MFA) adds extra protection against unauthorized login attempts. Passwords should be updated regularly, and vulnerability scans must be performed to identify security weaknesses.

Example: VLAN-based isolation prevented malware from spreading to PLC networks.

3.9.6 Predictive Analytics and AI-Based Optimization

Modern IIOT networks use artificial intelligence and machine learning to predict failures and optimize performance automatically.

AI-based anomaly detection identifies unusual traffic patterns or abnormal device behavior. Predictive maintenance algorithms estimate when components such as switches or routers may fail. Machine learning models continuously improve network efficiency through data-driven adjustments.

Example: AI analytics predicted a router failure three days in advance based on temperature and traffic patterns, preventing downtime.

3.9.7 Redundancy and Failover Systems

High-availability IIOT networks require backup systems to ensure uninterrupted communication during faults or power failures.

Critical equipment should use dual power supplies. Redundant gateways or standby routers can automatically take over if the main system fails. Backup servers may also be maintained for both local and cloud data storage.

Example: During a power fluctuation, a redundant gateway switched over instantly, ensuring uninterrupted machine communication.

> 👉 Aircraft systems also use redundancy to maintain safety during component failures.

3.9.8 Documentation and Training

Proper documentation and trained personnel are essential for maintaining reliable network performance.

Network diagrams, IP inventories, and device manuals should be updated regularly. Maintenance teams must receive training on diagnostic tools, troubleshooting methods, and new technologies. Feedback mechanisms can also be used to improve network reliability continuously.

Example: After monthly network health training, a factory's downtime incidents dropped by 25%.

3.9.9 Sustainable Network Optimization

Modern IIOT networks should also focus on sustainability and energy efficiency to reduce operating costs and environmental impact.

Sleep modes may be enabled for sensors during idle periods. Energy-efficient switches and routers should be selected, and automatic shutdown schedules can be used during non-production hours.

Example: Optimizing power schedules reduced energy costs by 15% in an automated warehouse.

PRACTICAL ACTIVITY

ACTIVITY 1:
Identify Common IIOT Network Problems and Determine Appropriate Troubleshooting Actions

Objective: To identify common IIOT network problems and determine suitable troubleshooting methods and corrective actions for reliable communication.

Procedure:

1. Study each problem scenario related to IIOT networking.
2. Analyze the possible cause of communication failure or network issue.
3. Identify the correct troubleshooting steps.
4. Suggest the most appropriate solution.
5. Record observations in the tables below.

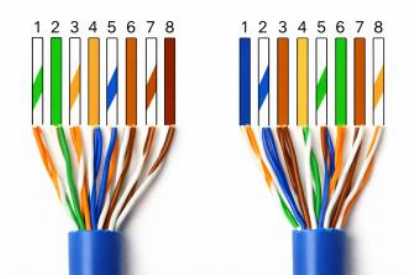
Table (a): Common IIOT Network Problems and Solutions

S.N.	Problem / Scenario	Recommended Troubleshooting Action / Solution
1	An engineer wants to establish communication between two Ethernet devices. What configuration is required?	To enable Ethernet communication, both devices must have valid IP addresses in the same subnet. For example: Device 1 → 192.168.1.10, Device 2 → 192.168.1.20, Subnet Mask → 255.255.255.0. Additionally, ensure the physical connection (Ethernet cable or network switch) is active and that firewall settings are not blocking communication.
2.	Two devices that were communicating earlier have stopped communicating.	Perform the following checks: 1) Verify physical connections (cables, switches, ports). 2) Confirm IP addresses and subnet masks have not been changed. 3) Use the ping command to test connectivity. 4) Check for IP conflicts on the network. 5) Verify firewall or security settings. 6) Restart devices if required.
3.	A controller that was previously communicating with an IO-Link master has stopped communicating.	Possible reasons include: 1) **IP address change (or DHCP configuration can change the IP after restart) or conflict** in the network. 2) Controller and IO-Link master may be in **different subnets without proper**

	Hardware is fine and both devices are connected through the organization network.	**gateway configuration**. 3) **Firewall or network security policies** in the organization network blocking communication. 4) Check if the **controller software configuration or network routing settings** have changed.
4.	Why does my system's IP address change after restarting the computer?	This usually happens when the system is configured to obtain an IP address automatically using DHCP (Dynamic Host Configuration Protocol). The router or DHCP server assigns a new available IP address each time the system reconnects to the network. To avoid this, configure a static IP address.
5.	I tried setting my system IP address to 192.168.1.257, but it was not accepted.	In IPv4 addressing, each octet must be within the range 0–255. Since 257 exceeds the allowed range, the address is invalid and cannot be assigned to any device.
6.	My system IP is 192.168.1.1 and the PLC IP is 192.198.3.6. What should I do to ping the PLC?	These two IP addresses belong to different networks. To establish communication, you can: 1) Change your PC IP to 192.198.3.x (for example, 192.198.3.10) with subnet mask 255.255.255.0, or 2) Configure a router or gateway that allows communication between the two networks.
7.	When a mobile device connects to Wi-Fi, is it configured with Static IP or DHCP by default?	By default, mobile devices use DHCP. The Wi-Fi router automatically assigns an available IP address, subnet mask, gateway, and DNS settings when the device connects to the network.
8.	What happens if two PLCs are assigned the same IP address on the same network?	Assigning the same IP address to two devices creates an IP address conflict. As a result, the network cannot correctly identify the devices, which leads to communication failures, unstable connections, or intermittent network errors. Each device in a network must have a unique IP address.

Table (b): Additional IIOT Network Problems and Corrective Actions

S.N.	Problem	Recommended Action
1	Device Offline	Check power supply, device status LEDs, and network connection. Ensure the device IP address and subnet configuration are correct and reachable from the network.
2	Gateway Communication Failure	Verify gateway configuration, protocol settings, and network parameters. Restart the gateway and check if the device is properly added and mapped in the gateway configuration.
3	Weak Wireless Signal	Ensure proper placement of the wireless gateway/router. Reduce interference sources and confirm the signal strength is within acceptable limits.

4	 Devices are not communicating even though devices and cable seem fine. Ethernet cable pin configuration may be incorrect. Above image shows the pin configuration of both sides of the Ethernet cable.	Check the **Ethernet cable pin configuration on both sides**. If the wire sequence is incorrect (as shown in the image), the devices will not communicate properly. Inspect RJ45 connector wiring on both ends. Re-crimp the cable using correct T568A or T568B standard on both sides. Incorrect wire sequence prevents communication.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. An engineer connects two Ethernet devices using a LAN cable, but they are unable to communicate. Both devices have IP addresses 192.168.1.10 and 192.168.2.20 with subnet mask 255.255.255.0. What is the most likely reason for the communication failure?
   a) Devices are in different subnets
   b) Devices need internet access
   c) Cable length is too short
   d) MAC address mismatch

2. Two devices that were communicating earlier suddenly stop communicating. The devices are powered ON and their IP configuration has not changed. What should be checked first?
   a) PLC program
   b) Physical network connections (cable, switch port)
   c) Internet speed
   d) Device firmware version

3. A controller was communicating with an IO-Link Master earlier, but communication stopped after connecting both devices to the organization network. Hardware is fine. What could be the possible issue?
   a) Network firewall blocking communication
   b) Cable insulation problem
   c) Controller memory full
   d) PLC scan cycle too high

4. An engineer notices that the PC IP address changes every time the system restarts. What is the most likely reason?

a) Static IP configuration
b) DHCP enabled on the network
c) Network cable fault
d) Gateway failure

5. While configuring the IP address manually, a technician enters 192.168.1.257 and the system does not accept it. Why is this IP address invalid?
   a) Subnet mask missing
   b) IP octet exceeds allowed range (0–255)
   c) Gateway not configured
   d) Ethernet port disabled

6. A PC has IP 192.168.1.1 and a PLC has IP 192.198.3.6. The engineer tries to ping the PLC but it fails. What is the best solution?
   a) Change the PC IP to the 192.198.3.x network
   b) Restart the PLC
   c) Replace the Ethernet cable
   d) Enable Wi-Fi on the PLC

7. A technician connects his mobile phone to Wi-Fi and notices that the device automatically receives an IP address without manual configuration. Which mechanism is responsible for this?
   a) Static IP assignment
   b) DHCP server
   c) NAT protocol
   d) MAC filtering

8. Two PLCs are accidentally configured with the same IP address on the same network. What will most likely happen?
   a) Both PLCs will communicate normally
   b) One PLC will automatically change its IP
   c) IP conflict causing unstable or failed communication
   d) Gateway will shut down

9. A device connected to the network shows Offline in the gateway software. Which of the following should be checked first?
   a) PLC program logic
   b) Device power and Ethernet connectivity
   c) Cloud server status
   d) PLC scan time

10. Two devices are connected with an Ethernet cable and both devices are working correctly. However, communication still fails. After inspection, the wire order on both ends of the cable is found to be different. What is the most likely cause?
    a) Incorrect Ethernet pin configuration
    b) Incorrect device IP address format
    c) Low internet bandwidth
    d) PLC memory overflow

B. Match the following

Column A	Column B
1. Intermittent connectivity	A. Ping test and link check
2. High latency	B. Traffic analysis & QoS
3. Packet loss	C. CRC error logs & cable testing
4. IP conflict	D. Network scan for duplicate addresses
5. Protocol mismatch	E. Verify compatibility & firmware

C. Fill in the blanks

1. Repeated disconnection of devices is known as _____ connectivity.
2. A tool like _____ is used to analyze packet-level network traffic.
3. Duplicate addressing problems in networks are called _____ conflicts.
4. Checking each layer from hardware to software is called _____troubleshooting.
5. Identifying the main reason behind a recurring issue is known as _____ analysis.

D. Answer the following

1. How would you diagnose a situation where a sensor is not updating data on the dashboard?
2. A network shows increased delay during peak production. What steps would you take to improve performance?
3. In a system with frequent communication errors, how can the isolation method help identify the faulty component?
4. Explain how comparing with a reference system can help solve configuration-related issues.
5. If electromagnetic interference affects communication, what practical corrective measures would you implement?

SESSION 6: IIOT HARDWARE TESTING, SAFETY AND MAINTENANCE PRACTICES

3.10 Line Testing on IIOT Devices

Think About It!

"If a sensor suddenly stops sending data, how to determine whether the problem is in the sensor, cable, or communication network?"

Line testing is a crucial part of hardware verification and troubleshooting in Industrial IIOT systems. It ensures that communication lines, signal connections, and data transmission paths between sensors, controllers, and network devices are functioning correctly. Through line testing, engineers can detect faults such as cable breaks, short circuits, signal loss, or interference that may disrupt communication in an industrial setup.

line testing is done to verify the integrity and reliability of physical connections in an IIOT network. It helps to:

- Confirm that each cable, connector, and interface is properly connected and transmits data.
- Identify faults like open circuits, cross connections, or signal distortion.
- Ensure that the network meets required communication standards (Ethernet, RS-485, IO-Link, etc.).
- Validate the quality of signal transmission and detect potential interference from electrical noise.

For Example, before deploying new sensors in a factory, technicians perform line testing to ensure the I/O Link cables meet required resistance and continuity parameters.

3.10.1 Types of Line Testing

Different types of tests are used depending on the network and device type.

(a) Continuity Testing: Checks if the electrical connection between two points is complete and unbroken.

- Conducted using a multimeter or continuity tester.
- Detects open circuits or disconnected wires.

Example: A sensor node not responding was traced to an open wire in the communication line using a continuity test.

(b) Insulation Resistance Testing: Measures the insulation strength between conductors to prevent current leakage.

- Conducted using an insulation tester (megger).
- Ensures safe operation in high-voltage or noisy environments.

Example: Low insulation resistance between signal and ground lines indicated cable aging, prompting preventive replacement.

(c) Signal Quality Testing: Analyzes the integrity of data signals transmitted over the line.

- Uses oscilloscopes, network analyzers, or IIOT diagnostic tools.
- Detects issues such as signal attenuation, noise interference, or reflections.

Example: High-frequency signal loss was detected on a long Ethernet line due to poor cable shielding.

Around Us

Signal quality testing is also performed in mobile networks, television systems, and internet communication.

(d) Loopback Testing: Used to verify both the transmission and reception paths of communication lines.

- The signal is sent and looped back to the source to check response accuracy.
- Commonly used for Ethernet, RS-232, and RS-485 networks.

Example: A loopback test confirmed that an I/O link master port was functional, isolating the fault to a damaged sensor cable.

(e) Power Line Testing: Checks the stability and voltage levels of power lines that feed IIOT devices.

- Detects voltage drops, surges, or grounding issues that may affect communication modules.
- Ensures consistent power supply to prevent device resets.

Example: Fluctuating sensor data was traced to a voltage dip caused by a loose power connector.

3.10.2 Tools Used for Line Testing

Various tools and instruments assist engineers in performing accurate tests as follows (refer Fig.3.10)

- Multimeter – for checking continuity, voltage, and resistance.

- Network cable tester – to verify Ethernet pin configuration and signal mapping.
- Time Domain Reflectometer (TDR) – to locate cable faults and measure line length.
- Oscilloscope – to visualize signal waveform and detect noise.
- Protocol analyzer – for testing data flow in communication networks (Modbus, IO-Link, MQTT, etc.).

Fig.3.10: Tools Used for Line Testing

For example, a TDR test identified a break in an underground Ethernet cable 27 meters from the switch cabinet (Fig.3.11).

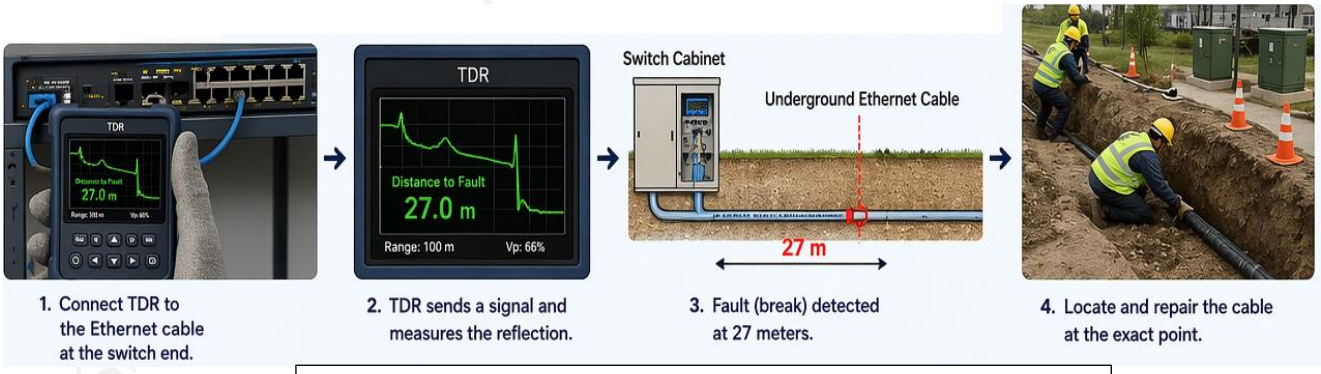


Fig.3.11: Identification of a Break in Cable Using TDR

A Time Domain Reflectometer (TDR) works similarly to radar by locating faults based on signal reflections.

3.10.3 Step-by-Step Line Testing Process (Refer Fig.3.12)

A systematic testing approach ensures accuracy and safety:

1. Isolate Power Supply – Turn off connected devices to prevent short circuits.
2. Visual Inspection – Check cables and connectors for damage or corrosion.
3. Continuity and Resistance Test – Verify electrical connections and impedance values.
4. Insulation Test – Check insulation between conductors.
5. Signal Transmission Test – Evaluate data transfer quality and strength.
6. Loopback Verification – Confirm two-way communication.
7. Document Results – Record findings for maintenance and reference.

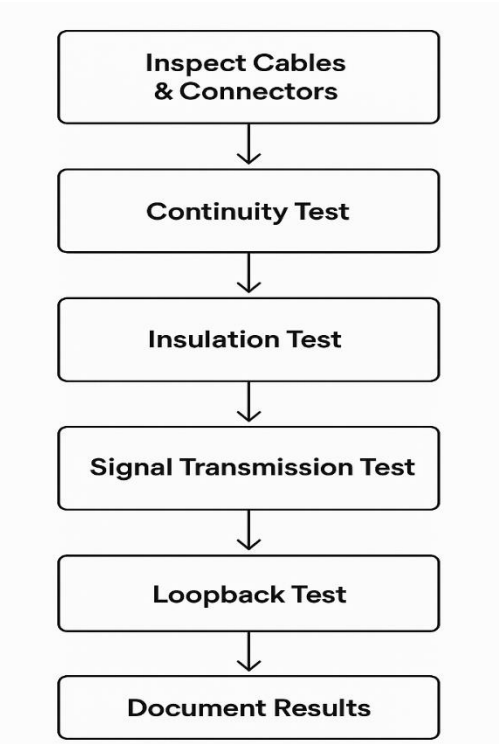


Fig.3.12: Step-by-Step Line Testing Process

For example, routine testing following this sequence detected both a mis wired connector and an underperforming Ethernet patch cable before commissioning.

3.10.4 Common Issues Detected During Line Testing

Line testing helps identify several frequent faults in IIOT installations as follows:

- Open or short cables.
- Incorrect pin assignments or cross-wiring.
- Grounding or earthing faults.
- Weak signal strength or excessive attenuation.
- Electromagnetic interference (EMI) from nearby equipment.
- Faulty connectors or improperly crimped terminations.

Example: Repeated sensor communication loss was solved by replacing a corroded RJ45 connector identified during testing.

3.10.5 Importance of Line Testing in IIOT Systems

Regular line testing is essential for maintaining reliable, safe, and efficient communication in IIOT systems. It helps detect faults in cables, connectors, and communication lines before they lead to major failures or production stoppages.

Benefits of line testing are as follows:

- Prevents communication delays, signal interruptions, and data loss.
- Extends device lifespan by reducing electrical stress caused by poor connections.
- Improves troubleshooting accuracy by quickly locating faults in the network.
- Ensures compliance with industrial standards such as IEC 61158 and ISO / IEC 11801.
- Reduces maintenance cost and unplanned downtime.
- Improves overall network reliability and production efficiency.

Example: Regular line testing at a food processing plant reduced downtime incidents by 40% over one year.

3.10.6 Integration with Digital Diagnostics

Modern IIOT systems use smart diagnostic tools that automatically perform line testing and report results through digital dashboards. These systems help maintenance teams monitor communication health continuously and take corrective action at an early stage.

Features of Digital Diagnostics are:

- IO-Link masters can perform port diagnostics to detect communication faults.
- Cloud-based platforms store and analyze test results over time.
- Predictive maintenance systems generate alerts when line quality starts degrading.
- Dashboards provide real-time visibility of cable status, signal strength, and device communication.
- Automated reports support faster maintenance planning.

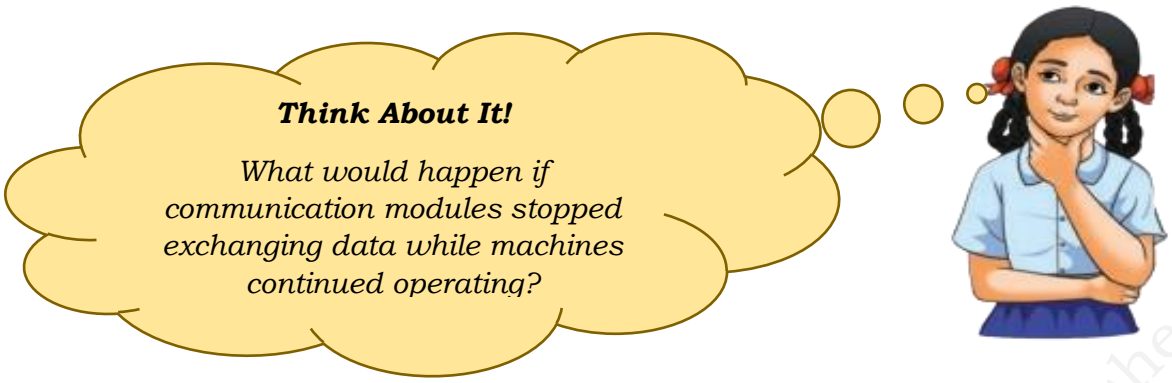
For example, an IIOT diagnostic platform automatically flagged a degrading sensor line, allowing replacement before total failure.

3.10.7 Safety Considerations

Line testing should always be conducted with safety precautions:

- De-energize circuits before performing continuity or insulation tests.
- Use proper-rated instruments for voltage and current levels.
- Follow ESD protection measures when handling sensitive devices.
- Ensure proper grounding during high-voltage insulation testing.

3.11 Communication Module Hardware Testing



Communication modules are vital components in IIOT systems that enable data transfer between sensors, controllers, gateways, and cloud platforms. They act as the link between field devices and higher-level systems using wired or wireless protocols such as Ethernet, Modbus, IO-Link, Wi-Fi, or Bluetooth.

Regular hardware testing of communication modules ensures that devices communicate accurately, without data loss, interference, or downtime. It also helps in diagnosing faults before they lead to network failure.

The objective of testing is to ensure:

- The hardware components (ports, circuits, and transceivers) are functioning correctly.
- The communication interfaces support stable and reliable data transmission.
- The signal strength and integrity meet network specifications.
- The firmware and configuration are compatible with connected IIOT devices.
- To detect hardware faults early — such as short circuits, port failures, or overheating.

Example: Before deploying a new I/O link master, engineers perform hardware testing to ensure each port transmits data correctly to sensors and the PLC.

3.11.1 Components Checked During Communication Module Hardware Testing

Communication modules include several critical components that must be tested:

1. Power Circuit:
    - Checks voltage regulation and stability of power supply.
    - Ensures the module powers up without surges or drops.
2. Communication Interfaces:
    - Ethernet, RS-485, RS-232, CAN, IO-Link, or wireless ports are tested for connectivity and speed.
    - Verifies that transmit (Tx) and receive (Rx) lines are working properly.
3. Indicators and LEDs:
    - Confirms visual status lights (Power, Link, Error, Data) operate correctly.

- Helps in quick visual diagnosis during operation.

> ☞ *Many electronic devices use LED indicators because they provide quick visual status information.*

4. Antenna or Transceiver (for wireless modules):
    - Checks antenna alignment, signal gain, and interference levels.
    - Ensures stable wireless communication range.
5. Connectors and Cables:
    - Inspects for damage, rust, or loose terminals that may cause intermittent faults.
6. Cooling and Heat Dissipation:
    - Monitors module temperature under load conditions.
    - Prevents overheating that could affect performance.

3.11.2 Common Tests Performed in Communication Module Hardware Testing

A combination of manual and automated tests ensures thorough evaluation as follows:

(a) Visual and Physical Inspection

- Check for signs of damage, burnt components, dust accumulation, or loose connections.
- Inspect connectors and solder joints for cracks or corrosion.

(b) Power-On Testing

- Apply rated voltage and check startup sequence through LED indicators.
- Measure current draw — abnormal readings may indicate internal faults.

(c) Communication Interface Testing

- Use diagnostic software or analyzers to check data transmission between module and PLC or gateway.
- Perform ping or loopback tests to confirm the path is working.
- Monitor response time and packet loss to ensure reliability.

(d) Signal Integrity and Noise Testing

- Measure signal-to-noise ratio (SNR) and detect interference.
- Use oscilloscope or protocol analyzer to observe waveform distortion or delay.

(e) Firmware Verification

- Verify that firmware is up to date and properly loaded.
- Ensure compatibility with network protocols and devices.

(f) Thermal and Stress Testing

- Operate the module under full load and elevated temperature to check stability.
- Identify overheating, which could indicate poor ventilation or component failure.

3.11.3 Tools Used for Communication Module Hardware Testing

Engineers and technicians use a variety of tools for communication module testing as follows (Fig.3.13):

- Multimeter – to measure voltage, current, and continuity.
- Ethernet/Network Tester – for port mapping and speed verification.
- Protocol Analyzer – to capture and analyze communication signals.
- Oscilloscope – to view and measure data signal quality.
- Thermal Camera or Temperature Probe – for monitoring heat output.
- Diagnostic Software – to test firmware response and data exchange.

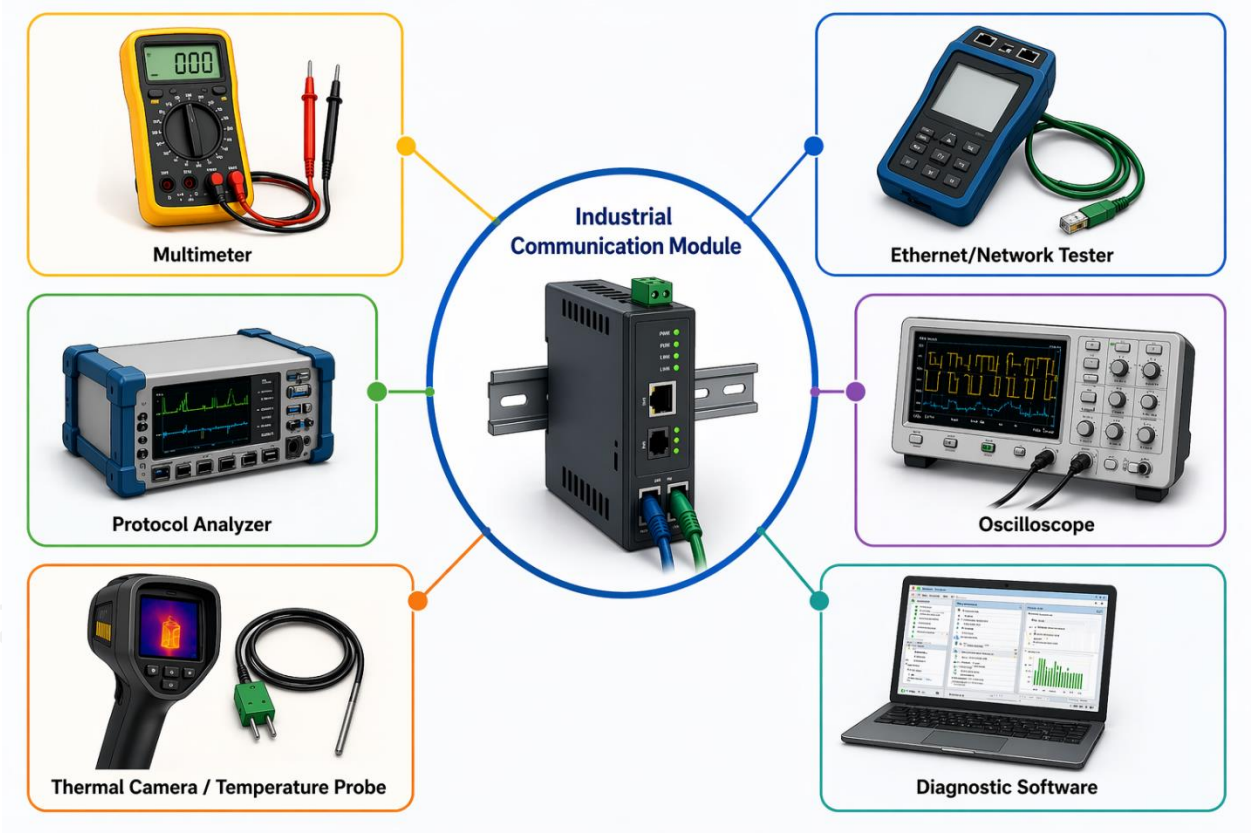


Fig.3.13: Tools Used for Communication Module Hardware Testing

Example: During maintenance, a protocol analyzer revealed high CRC errors in a Modbus module, traced to a faulty RS-485 transceiver.

3.11.4 Common Issues Detected During Hardware Testing

Hardware testing helps identify many real-world problems such as:

- Damaged connectors or pins causing intermittent signals.
- Faulty transceiver chips leading to loss of communication.
- Power supply instability due to poor voltage regulation.
- Overheating caused by poor ventilation or high current draw.
- Firmware mismatch between devices leading to data errors.
- Noise interference in communication lines from nearby electrical equipment.

3.11.5 Preventive Maintenance through Regular Testing

Performing periodic communication module testing provides several long-term benefits:

- Ensures consistent and reliable data flow across IIOT networks.
- Helps detect issues before total system failure.
- Improves equipment uptime and reduces troubleshooting time.
- Supports predictive maintenance, where potential faults are detected early.
- Reduces replacement costs and avoids unplanned downtime.

Example: In an automotive assembly line, quarterly testing of Ethernet-based communication modules reduced data loss incidents by 30%.

3.11.6 Integration with IIOT Diagnostic Systems

Modern IIOT platforms integrate self-diagnostic features for continuous hardware health monitoring as follows:

- Modules send status reports on port health, voltage levels, and communication errors.
- Dashboards visualize real-time performance metrics.
- Predictive analytics help schedule maintenance based on performance trends.

Example: An IO-Link master automatically reported a degraded signal on one port, prompting a cable replacement before communication failed.

3.11.7 Safety Guidelines During Testing

- Always de-energize circuits before handling hardware.
- Use ESD protection to prevent damage to sensitive components.

- Verify voltage ratings before applying power.
- Avoid short-circuiting terminals during measurement.
- Follow manufacturer's test procedures and safety standards (e.g., IEC 61010).

3.12 Maintenance Best Practices in IIOT Networks

In IIOT systems, reliable network performance depends on the continuous health of devices, communication links, and data pathways. Maintenance best practices in IIOT networks ensure stable connectivity, minimal downtime, and accurate data transfer between sensors, controllers, and cloud systems. A well-maintained network improves productivity, prevents unexpected breakdowns, and extends the life of connected devices.

The main goals of maintaining IIOT networks are to:

- Prevent communication failures and data loss.
- Ensure high availability and low latency in data transfer.
- Identify and correct faults before they cause downtime.
- Optimize bandwidth usage and network performance.
- Maintain cybersecurity and data integrity.

The basic maintenance best practices are as follows:

a) Regular Network Inspection

- Periodically check switches, routers, gateways, and I/O link masters for signs of damage or loose connections.
- Verify indicator LEDs for correct power, link, and data status.
- Ensure all network cables are labeled and properly routed to avoid confusion during troubleshooting.

b) Cable and Connector Management

- Replace worn-out, bent, or damaged cables immediately.
- Use shielded twisted pair (STP) cables in areas with high electrical interference.
- Inspect RJ45 and M12 connectors for corrosion or poor contact.
- Maintain proper cable bending radius and secure them to prevent mechanical stress.

c) Firmware and Software Updates

- Regularly update the firmware of IIOT devices, switches, and routers.
- Apply security patches and bug fixes provided by manufacturers.
- Backup configurations before performing updates to prevent data loss.

d) Monitoring and Diagnostics

- Use IIOT dashboards and network management tools to monitor parameters such as latency, packet loss, and device status.
- Analyze communication logs to detect anomalies or performance degradation.
- Configure alerts for disconnections or abnormal traffic patterns.

e) Preventive Maintenance Scheduling

- Establish a maintenance calendar for periodic inspection and testing of network devices.
- Clean network panels and remove dust or moisture accumulation.
- Test communication lines using line testers or network analyzers.
- Verify grounding and shielding to minimize electrical noise.

f) Data and Configuration Backup

- Regularly back up network configurations, IP tables, and device settings.
- Store backup files securely in cloud storage or local servers.
- This allows quick restoration of the network in case of failure or replacement.

g) Network Security Maintenance

- Change default passwords and use strong encryption (e.g., TLS, VPN).
- Segment critical devices using VLANs to reduce cyber risks.
- Monitor unauthorized access attempts or abnormal traffic patterns.

> 👉 *A strong password is often the first line of defense against cyberattacks.*

h) Documentation and Record Keeping

- Maintain logs of all maintenance activities, updates, and device replacements.
- Record signal strength, communication status, and test results after each inspection.
- Use CMMS (Computerized Maintenance Management Systems) for organized record keeping.

3.12.1 Predictive and Condition-Based Maintenance

With advanced IIOT analytics, network maintenance is shifting from reactive to predictive mode:

- Sensors monitor real-time data such as temperature, voltage, and signal quality.
- AI algorithms predict failures before they occur.
- Dashboards provide visual alerts and maintenance recommendations.

Example: An IIOT gateway dashboard detected increasing packet loss trends on one port. Predictive analysis indicated a potential cable degradation, which was replaced before communication failure occurred.

3.12.2 Benefits of Following Best Practices

- Improved Network Uptime and Reliability: Ensures stable communication and continuous operation of IIOT systems.
- Reduced Maintenance Cost and Unplanned Downtime: Preventive measures minimize unexpected failures and repair expenses.
- Enhanced Data Accuracy and Operational Visibility: Reliable networks provide accurate data for better monitoring and decision-making.
- Extended Device Lifespan and Energy Efficiency: Proper maintenance reduces equipment stress and improves power utilization.
- Strengthened Cybersecurity and Protection Against Data Breaches: Secure practices help safeguard systems from unauthorized access and cyber threats.

PRACTICAL ACTIVITY

ACTIVITY 1:

Analyze an IIOT Network Architecture and Identify the Data Communication Flow between Devices

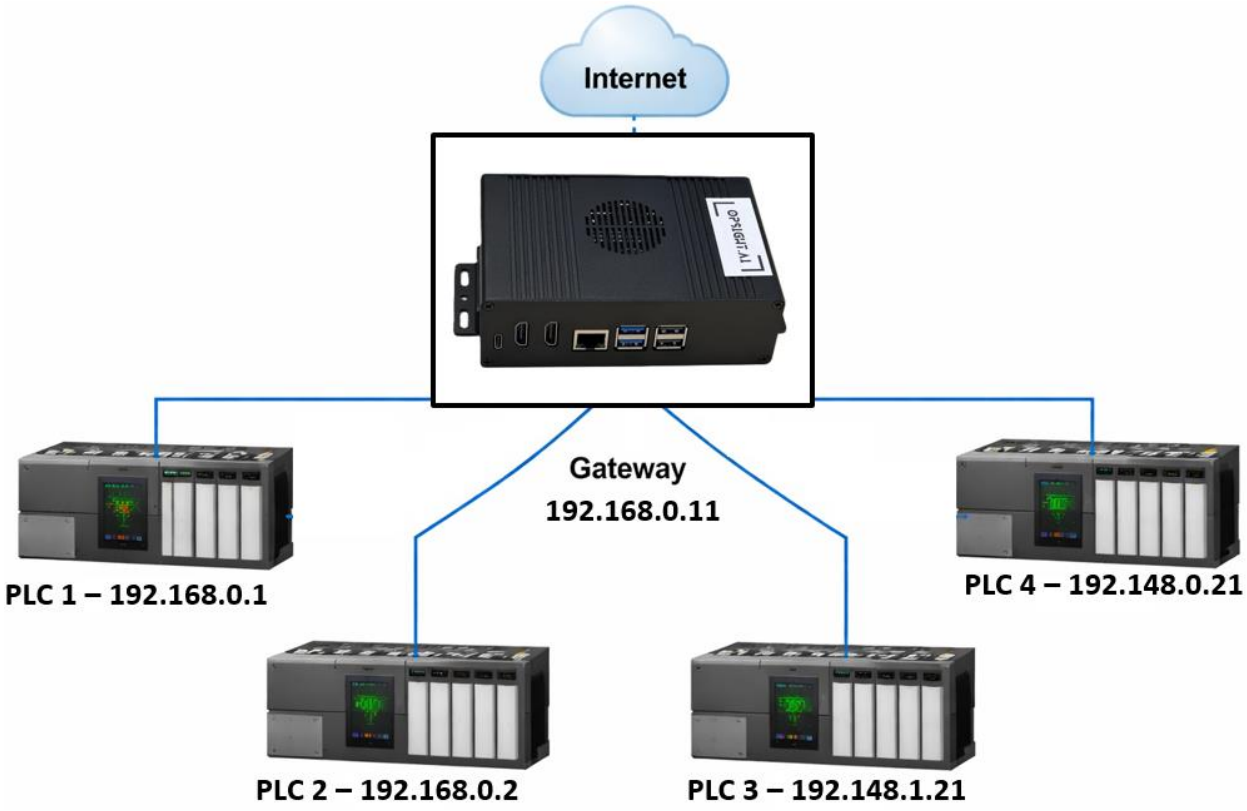


Fig.1: IIOT Network Architecture

Answer the following questions based on the architecture shown in Fig.1:

1. Based on the network architecture shown above, which device is most likely not communicating with the gateway?
   a) PLC 1
   b) PLC 2
   c) PLC 3
   d) PLC 4

2. In the given architecture, which device requires an internet connection to enable communication with the cloud?
   a) PLC 3
   b) Gateway
   c) PLC 1 and PLC 2
   d) All PLCs and the Gateway

3. In the given network architecture, what will happen if the Ethernet cable connecting PLC 1 to the Gateway is disconnected?
   a) Data from all PLCs will stop being communicated to the cloud
   b) The Gateway will stop communicating with all PLCs
   c) Only PLC 1 will lose communication with the Gateway
   d) The internet connection will go down

ACTIVITY 2:
Design a Network Architecture for an Industrial Setup and Identify All Required Hardware Components

Objective: To design a basic IIOT network architecture for an industrial setup, identify the required hardware devices, and understand their role in data communication.

Exercise: Design a basic IIOT network architecture using the devices shown in the Fig.2.

	**PLC 1**
	**IO Link Sensor**
	**IIOT Gateway**
	**PLC 2**
	**IO Link Master**
	**Ethernet Switch**

Fig.2: IIOT Network Devices

Tasks

- Draw the network architecture diagram connecting the PLCs, IO-Link Master, Ethernet Switch, and IIOT Gateway.
- Assign appropriate IP addresses to each device.
- Identify which device is responsible for sending data to the cloud.
- Explain why the Ethernet switch is required in this architecture.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. A sensor stops communicating, and continuity testing shows no signal path. What is the most likely issue?
   a) Firmware mismatch
   b) Open circuit in the cable
   c) High bandwidth usage
   d) Protocol error

2. During testing, low insulation resistance is detected between conductors. What action should be taken?
   a) Increase voltage supply
   b) Replace or repair the cable
   c) Restart controller
   d) Ignore if communication works

3. An engineer uses a loopback test and receives correct signals. What does this confirm?
   a) Power supply failure
   b) Signal noise
   c) Transmission and reception paths are working
   d) IP conflict

4. Frequent device resets are observed in an IIOT system. Which test should be prioritized?
   a) Signal quality test
   b) Power line testing
   c) Loopback test
   d) Protocol analysis

5. A TDR shows a fault at 30 meters in a cable. What should be done?
   a) Replace entire network
   b) Inspect and repair cable at that location
   c) Change protocol
   d) Increase signal strength

B. Match the following

Column A	Column B
1. Continuity Testing	A. Checks voltage stability
2. Insulation Testing	B. Identifies noise and attenuation
3. Signal Quality Testing	C. Prevents leakage current
4. Loopback Testing	D. Verifies Tx and Rx path
5. Power Line Testing	E. Detects open circuits

C. Fill in the blanks

1. A device used to locate cable faults and measure distance is called _____.
2. Testing insulation strength between conductors is performed using a _____ tester.
3. Signal distortion due to external noise is detected during _____ testing.
4. A break in communication lines can be identified using a _____ test.
5. Monitoring voltage drops in supply lines is part of _____ testing.

D. Answer the following

1. How would you systematically test a communication cable before installing a new IIOT sensor?
2. A communication module shows irregular data transmission. Which tests would you perform to identify the fault?
3. Explain how loopback testing helps in isolating faults in a communication network.
4. In a noisy industrial environment, what steps would you take to ensure signal integrity?
5. Describe how thermal testing can prevent failure in communication modules.

ANSWER KEY

MODULE 1: Automotive IIOT Applications

Session 1: Introduction to Automotive IIOT

Multiple Choice Questions	Match the following	Fill in the blanks
1. C 2. B 3. C 4. B 5. C	1-B 2-C 3-A 4-D 5-E	1. Logbooks 2. Transmission 3. IIOT 4. Theft 5. operating

Session 2: Role of IIOT in Monitoring and Material Handling in Industries

Multiple Choice Questions	Match the following	Fill in the blanks
1. B 2. C 3. B 4. C 5. B	1-C 2-A 3-D 4-B 5-E	1. Warehouse 2. identification 3. accuracy 4. dashboards 5. scheduling

Session 3: Smart Transportation and Predictive Maintenance

Multiple Choice Questions	Match the following	Fill in the blanks
1. B 2. C 3. C 4. B 5. B	1-B 2-C 3-D 4-E 5-A	1. Informatics 2. ECU 3. Condition 4. Degradation 5. Historical

MODULE 2: Remote Monitoring and controlling in IIOT Network

Session 1: Importance of Remote Monitoring and Control in IIOT Networks

Multiple Choice Questions	Match the following	Fill in the blanks
1. B 2. C	1-D	1. Modbus (or OPC-UA / WebSocket)

3. B 4. C 5. C	2-A 3-B 4-C 5-E	2. smartphones 3. red 4. Preventive 5. One

Session 2: Remote Data Acquisition Architecture

Multiple Choice Questions	Match the following	Fill in the blanks
1. B 2. C 3. C 4. B 5. B	1-B 2-A 3-C 4-D 5-E	1. Latency 2. Transmission 3. Timestamp 4. alert *(or status)* 5. electrical

Session 3: Dashboards and Data Visualization

Multiple Choice Questions	Match the following	Fill in the blanks
1. C 2. C 3. C 4. B 5. C	1-C 2-E 3-D 4-B 5-A	1. Clarity 2. Seconds 3. Laser 4. Proximity 5. leak *(or leakage)*

Session 4: Remote Control and Command Execution

Multiple Choice Questions	Match the following	Fill in the blanks
1. B 2. C 3. B 4. B 5. B	1-C 2-E 3-A 4-B 5-D	1. remote *(or distant)* 2. lightweight 3. safe *(or fail-safe)* 4. scheduled 5. encryption

Session 5: Alerts, Alarms and Proactive Analysis

Multiple Choice Questions	Match the following	Fill in the blanks
1. B 2. C 3. B 4. B 5. B	1-B 2-E 3-D 4-C 5-A	1. Safe 2. color-coded 3. historical 4. informational 5. abnormal

MODULE 3: Maintenance and Troubleshooting of I/O link Master and IIOT Network Devices

Session 1: Foundational IIOT Connectivity

Multiple Choice Questions	Match the following	Fill in the blanks
1. C 2. B 3. C 4. B 5. B	1-E 2-D 3-A 4-C 5-B	1. publish–subscribe 2. UDP 3. aggregating *(or aggregation)* 4. fast *(or real-time)* 5. interoperability

Session 2: Machine Alarm and Status Analysis

Multiple Choice Questions	Match the following	Fill in the blanks
1. C 2. C 3. A 4. D 5. C	1-B 2-A 3-D 4-C 5-E	1. Safe 2. Critical 3. Recurring*(or repeating)* 4. audible 5. severity

Session 3: Advanced Machine Performance Analytics

Multiple Choice Questions	Match the following	Fill in the blanks
1. B 2. C 3. B	1-B 2-A	1. Performance 2. Quality 3. MTBF

4. B 5. C	3-C 4-D 5-E	4. Availability 5. sensors

Session 4: IIOT Network Monitoring and Evaluation

Multiple Choice Questions	Match the following	Fill in the blanks
1. B 2. B 3. B 4. A 5. B	1-B 2-C 3-A 4-D 5-E	1. Communication 2. Data 3. data *(or packet)* 4. real-time 5. barriers

Session 5: IIOT Network Diagnostics, Troubleshooting, and Optimization

Multiple Choice Questions		Match the following	Fill in the blanks
1. A 2. B 3. A 4. B 5. B	6. A 7. B 8. C 9. B 10. A	1-A 2-B 3-C 4-D 5-E	1. Intermittent 2. Wireshark 3. IP 4. Layer-by-layer 5. Root cause

Session 6: IIOT Hardware Testing, Safety, and Maintenance Practices

Multiple Choice Questions	Match the following	Fill in the blanks
6. B 7. B 8. C 9. B 10. B	1-E 2-C 3-B 4-D 5-A	1. TDR (Time Domain Reflectometer) 2. Megger 3. signal quality 4. continuity 5. power line

GLOSSARY

Term	Definition
Actuator	A device that converts electrical signals into physical actions such as movement, switching, or control.
Alarm	A notification generated when a monitored parameter exceeds predefined limits.
Analytics	The process of examining data to identify patterns, trends, and useful information for decision-making.
Automotive IIOT	The application of Industrial Internet of Things technologies in automotive manufacturing, monitoring, and transportation systems.
Cloud Computing	The delivery of computing services such as storage, processing, and software over the internet.
Connected Mobility	A transportation system in which vehicles, infrastructure, and users exchange information through communication networks.
Dashboard	A graphical interface used to display real-time data, performance indicators, and system status.
Data Acquisition	The process of collecting data from sensors, machines, or devices for monitoring and analysis.
Data Visualization	The representation of data using charts, graphs, and visual elements to improve understanding.
Diagnostics	The process of identifying, locating, and analyzing faults in equipment or networks.
Gateway	A networking device that connects different communication networks and protocols.
Industrial Internet of Things (IIOT)	A network of interconnected industrial devices, sensors, machines, and systems that communicate and exchange data.
Industrial Network	A communication network used to connect industrial devices for data exchange and control.
Input/Output (I/O)	The communication between a system and external devices through input and output signals.
I/O-Link	A standardized industrial communication protocol used for connecting sensors and actuators to control systems.
I/O-Link Master	A device that manages communication between I/O-Link devices and higher-level control systems.
Machine Performance Analytics	The analysis of machine operational data to evaluate efficiency, productivity, and reliability.
Material Handling	The movement, storage, control, and protection of materials within manufacturing and logistics environments.
Monitoring	The continuous observation and tracking of system parameters and operational conditions.
Network Device	Hardware used to establish, manage, and maintain communication within a network.

Optimization	The process of improving system performance, efficiency, and reliability.
Predictive Maintenance	A maintenance strategy that uses data analysis to predict equipment failures before they occur.
Proactive Analysis	The examination of operational data to identify potential issues before they impact performance.
Remote Control	The ability to operate equipment or systems from a distant location through communication networks.
Remote Monitoring	The process of observing equipment and systems from a remote location using networked devices.
Sensor	A device that detects physical conditions such as temperature, pressure, or motion and converts them into signals.
Smart Transportation	The use of intelligent technologies and connected systems to improve transportation efficiency and safety.
Status Monitoring	Tracking the current operating condition of a machine, device, or process.
Troubleshooting	A systematic approach to identifying and resolving faults in systems or equipment.
Visualization Tool	Software used to present operational data in graphical formats for analysis and decision-making.

राष्ट्रीय शैक्षिक अनुसंधान और प्रशिक्षण परिषद
National Council of Educational Research and Training