Teach To India Publication

ITI Trade

Industrial Internet of Things (IIoT) Technician
इंडस्ट्रियल इंटरनेट ऑफ थिंग्स (IIoT) टेक्नीशियन

(1st Year)

Second Edition

CTS | NSQF-Level 4

TATA-Sponsored Trade

Dual Language: English | हिंदी

TRADE THEORY + MCQs

All-in-One:

- Trade Theory
- Employability Skills
- Exam Mock Test

For ITI Students Across India,
Based on the DGT/NCVT Syllabus and NIMI Exam Pattern

Teach To India
Publication

Industrial Internet of Things (IIoT) Technician - First Year
इंडस्ट्रियल इंटरनेट ऑफ थिंग्स (IIoT) टेक्नीशियन - प्रथम वर्ष

A Comprehensive Textbook with MCQ Practice and Detailed Solutions
Under the Craftsmen Training Scheme (CTS) | NSQF Level 4

Designed for:

ITI students across all states. This book is prepared as per the latest syllabus prescribed by DGT / NCVT and follows the NIMI examination pattern.

Key Features of the Book:

Dual Language Format: English | हिंदी

Detailed Trade Theory: Structured according to Learning Outcomes

Comprehensive MCQ Practice: Topic-wise Multiple-Choice Questions with Detailed Solutions

Complete Coverage of ITI Examination Sections:

- **Trade Theory**
- **Employability Skills**

Question Bank: Includes 2 Full-Length Mock Tests with Complete Solutions.

Also Useful For:

This book is also useful for **CITS** and for preparing for various **technical recruitment examinations** conducted by the **Railways, PSUs, SSC, DRDO, ISRO, state government departments, metro projects, and other government organizations.**

Title: Industrial Internet of Things (IIoT) Technician - First Year
Subtitle: A Comprehensive Textbook with MCQ Practice and Detailed Solutions
Dual-Language Edition: English | हिंदी

Editor-in-Chief: Dr. Parvendra Kumar
Editorial and Technical Support: Teach To India Technical Team
Computer Graphics & Layout: Teach To India Design Team

Authors:
Samender Singh
Dept. of CSE, ITS Engineering College, Gr. Noida, U. P.
Dr. Parvendra Kumar
B.Tech (UPTU), PG Diploma (C-DAC Hyderabad), M.Tech (IIT Roorkee), Ph.D
Sagar Panwar
Dept. of CSE, Shobhit University, Saharanpur, U.P.

Reviewer:
Ashish Singh
Instructor, Government ITI, Siddiqpur, Jaunpur, U.P.

Publisher:
Teach To India Publication
Adarsh Colony, Saharanpur, U.P. – 247001
Mobile: +91 9084496877
Email: info@teachtoindia.com | Website: www.teachtoindia.com

Printed at: Shree Education and Publication Private Limited, Ajmer, Rajasthan
Edition: Second Edition, 2026
ISBN: 978-81-69424-47-9

Copyright © Teach To India Publication. All rights reserved.

Legal Note:
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means — electronic, mechanical, photocopying, recording, or otherwise — without prior written permission of the publisher. While every effort has been made to ensure accuracy, the publisher assumes no responsibility for errors. Feedback and suggestions for improvement are always welcome.

Colophon:
This book is printed on environmentally responsible paper. The layout, typesetting, and graphics have been optimized for dual-language (English-Hindi) clarity and accessibility, suitable for technical and vocational training.

Printed in India

Price: ₹745/-

Preface | प्रस्तावना

This book, **Industrial Internet of Things (IIoT) Technician**, has been specially designed to help students succeed in both academic examinations and career-oriented preparation.

यह पुस्तक, **इंडस्ट्रियल इंटरनेट ऑफ थिंग्स (IIoT) टेक्नीशियन**, विद्यार्थियों को शैक्षणिक परीक्षाओं तथा करियर-केंद्रित तैयारी दोनों में सफलता दिलाने के उद्देश्य से विशेष रूप से तैयार की गई है।

It includes detailed Trade Theory, Employability Skills, and a question bank in mock test format based on the NIMI exam pattern.

इसमें विस्तृत ट्रेड थ्योरी, एम्प्लॉयबिलिटी स्किल्स तथा निमी परीक्षा पैटर्न पर आधारित मॉक टेस्ट प्रारूप में प्रश्न बैंक सम्मिलित किया गया है।

This book follows the latest syllabus prescribed by **DGT/NCVT** and is aligned with the latest **NIMI** examination pattern. It is structured for easy understanding and practical application.

यह पुस्तक **DGT/NCVT** द्वारा निर्धारित नवीनतम पाठ्यक्रम का पालन करती है तथा नवीनतम **NIMI** परीक्षा पैटर्न के अनुरूप तैयार की गई है। इसे सरल समझ और व्यावहारिक उपयोग को ध्यान में रखते हुए संरचित किया गया है।

The MCQs in this book have been designed at multiple levels—**Remembering, Understanding, Application, and Analysis**—in a dual-language format to enhance conceptual clarity and examination readiness.

इस पुस्तक में दिए गए **MCQs** को बहु-स्तरीय स्तरों—**स्मरण, समझ, अनुप्रयोग, और विश्लेषण**—पर द्विभाषी प्रारूप में तैयार किया गया है, ताकि संकल्पनात्मक स्पष्टता तथा परीक्षा-तत्परता को सुदृढ़ किया जा सके।

Our goal is not only to help students excel in **ITI courses and NCVT examinations**, but also to prepare them for competitive employment opportunities in both the **government and private sectors**.

हमारा उद्देश्य केवल विद्यार्थियों को **ITI पाठ्यक्रमों** एवं **NCVT परीक्षाओं** में उत्कृष्ट प्रदर्शन के लिए सक्षम बनाना ही नहीं, बल्कि उन्हें **सरकारी** तथा **निजी** दोनों क्षेत्रों में प्रतिस्पर्धी रोजगार अवसरों के लिए भी तैयार करना है।

How to Study This Book | इस पुस्तक का अध्ययन कैसे करें

The Trade Theory section is covered in detail. Students are advised to study this section thoroughly and carefully, and to develop a clear conceptual understanding with the help of detailed explanations, diagrams, and a flow-based presentation.

ट्रेड थ्योरी अनुभाग को विस्तृत रूप से प्रस्तुत किया गया है। विद्यार्थियों को सलाह दी जाती है कि वे इस अनुभाग का गहन एवं सावधानीपूर्वक अध्ययन करें तथा विस्तृत व्याख्याओं, आरेखों और क्रमबद्ध प्रस्तुतीकरण की सहायता से अपनी अवधारणाओं को स्पष्ट एवं सुदृढ़ करें।

Except for the Trade Theory section, the other sections contain important summaries. These summaries are sufficient in accordance with the weightage of the respective sections.

ट्रेड थ्योरी अनुभाग को छोड़कर अन्य सभी अनुभागों में महत्वपूर्ण सारांश दिए गए हैं। ये सारांश संबंधित अनुभागों के वेटेज के अनुसार पर्याप्त हैं।

Practice the MCQs only after completing the theory part of the module.

थ्योरी भाग पूर्ण करने के बाद ही संबंधित बहुविकल्पीय प्रश्नों (MCQs) का अभ्यास करें।

Students are advised to study this book in only one language, either Hindi or English. They should not compare the Hindi version with the English version during study.

विद्यार्थियों को सलाह दी जाती है कि वे इस पुस्तक का अध्ययन केवल एक ही भाषा—हिंदी अथवा अंग्रेज़ी—में करें। अध्ययन के समय हिंदी और अंग्रेज़ी संस्करणों की आपस में तुलना न करें।

In case of any discrepancy in technical terminology, translation, or conceptual interpretation, the English version shall be considered authoritative.

तकनीकी शब्दावली, अनुवाद या अवधारणात्मक व्याख्या में किसी भी असंगति की स्थिति में अंग्रेज़ी संस्करण को प्रामाणिक माना जाएगा।

At the end of the book, practice sets based on the NIMI exam pattern have been provided. Students are strongly advised to practice these questions at least twice before appearing for the examination.

पुस्तक के अंत में NIMI परीक्षा पैटर्न पर आधारित अभ्यास सेट प्रदान किए गए हैं। विद्यार्थियों को दृढ़तापूर्वक सलाह दी जाती है कि वे परीक्षा में सम्मिलित होने से पूर्व इन प्रश्नों का कम से कम दो बार अभ्यास अवश्य करें।

To practice the question bank in a computer-based mock test format, scan the QR code provided in the last part of the book.

प्रश्न बैंक का अभ्यास कंप्यूटर-आधारित मॉक टेस्ट प्रारूप में करने के लिए, पुस्तक के अंतिम भाग में दिए गए QR कोड को स्कैन करें।

Acknowledgment | आभार

The content of this book has been developed with reference to the official ITI syllabus and the guidelines issued by the Directorate General of Training (DGT) and the National Instructional Media Institute (NIMI). It has been prepared using the prescribed syllabus documents and standard training resources for educational purposes.

इस पुस्तक की सामग्री का विकास आधिकारिक आईटीआई पाठ्यक्रम तथा प्रशिक्षण महानिदेशालय (DGT) और राष्ट्रीय अनुदेशात्मक मीडिया संस्थान (NIMI) द्वारा जारी दिशा-निर्देशों के संदर्भ में किया गया है। इसे शैक्षिक उद्देश्यों के लिए निर्धारित पाठ्यक्रम दस्तावेजों एवं मानक प्रशिक्षण संसाधनों के आधार पर तैयार किया गया है।

The publishers gratefully acknowledge the contribution of these institutions to curriculum development and the promotion of vocational education in India.

प्रकाशक भारत में पाठ्यक्रम विकास तथा व्यावसायिक शिक्षा के प्रोत्साहन में इन संस्थानों के योगदान के प्रति कृतज्ञतापूर्वक आभार व्यक्त करते हैं।

Syllabus	
Learning Outcomes	**Trade Theory**
LO-1	Handle and maintain various electronic instruments like voltmeters, ammeters, etc. and identify the basic electronic and electrical components.
LO-2	Demonstrate characteristics of various electrical, electronic components, and explore digital electronic systems.
LO-3	Identify basic electronics components & PCB.
LO-4	Demonstrate the use of different types of digital sensors and analog sensors for various applications
LO-5	Demonstrate basic C programming instructions required for IIoT application.
LO-6	Familiarization with microcontroller, its architecture, Microcontrollers viz. Arduino boards and explore the different commands used in Microcontrollers platform.
LO-7	Demonstrate the application of single board computers (SBCs).
LO-8	Interface microcontrollers with sensors, actuators, and communication modules for connecting IIoT systems.
LO-9	Apply the knowledge of network protocols, data transmission methods, open source/ cloud integration and API for IIoT applications.
LO-10	Demonstrate different types of open-source databases and explore multiple operations associated with data used for building IIoT applications.
LO-11	Explore the technicalities like edge devices, gateways, open-source cloud platforms and Wi-Fi module for establishing connectivity between IIoT devices and open-source cloud platforms.
LO-12	Demonstrate basics of python programming for IIoT applications.
LO-13	Program Raspberry Pi using Python editor and interface with peripherals for IIoT applications.
LO-14	Develop and deploy mobile and Web application for IIoT using open-source platforms.
LO-15	Program and interface devices to build various IIoT applications.
LO-16	Explore security and privacy challenges associated with IIoT systems.
Modules	**Employability Skills**
Introduction to Employability Skills	Outline the importance of Employability Skills for the current job market and future of work. List different learning and employability related GOI and private portals and their usage. Research and prepare a note on different industries, trends, required skills and the available opportunities
Constitutional values - Citizenship	Explain the constitutional values, including civic rights and duties, citizenship, responsibility towards society etc. that are required to be followed to become a responsible citizen. Discuss the role of personal values and ethics such as honesty, integrity, caring and respecting others, etc. in personal and social development.
Becoming a Professional in the 21st Century	Discuss relevant 21st century skills required for employment. Highlight the importance of practicing 21st century skills like Self-Awareness, Behavior Skills, time management, critical and adaptive thinking, problem-solving, creative thinking, social and cultural awareness, emotional awareness, learning to learn etc. in personal or professional life. Create a pathway for adopting a continuous learning mindset for personal and professional development.
Basic English Skills	Use appropriate grammar and sentences while interacting with others. Read English text with appropriate articulation. Role play a situation on how to talk appropriately to a customer in English, over the phone or in person. Write a brief note/paragraph / letter/e -mail using correct English.
Career Development & Goal Setting	Create a career development plan. Identify well-defined short- and long-term goals
Communication Skills	Demonstrate how to communicate effectively using verbal and nonverbal communication etiquette. Write a brief note/paragraph on a familiar topic. Explain the importance of communication etiquette including active listening for effective

	communication. Role play a situation on how to work collaboratively with others in a team.
Diversity and Inclusion	Exhibit how to behave, communicate, and conduct oneself appropriately with all genders and PwD
Financial and Legal Literacy	Discuss various financial institutions, products, and services. Demonstrate how to conduct offline and online financial transactions, safely and securely and check passbook/statement. Explain the common components of salary such as Basic, PF, Allowances (HRA, TA, DA, etc.), tax deductions. Calculate income and expenditure for budgeting. Discuss the legal rights, laws, and aids.
Essential Digital Skills	Describe the role of digital technology in day-to-day life and the workplace. Demonstrate how to operate digital devices and use the associated applications and features, safely and securely. Demonstrate how to connect devices securely to internet using different means. Follow the dos and don’ts of cyber security to protect against cybercrimes. Discuss the significance of displaying responsible online behavior while using various social media platforms. Create an e-mail id and follow e- mail etiquette to exchange e - mails. Show how to create documents, spreadsheets and presentations using appropriate applications utilize virtual collaboration tools to work effectively.
Entrepreneurship	Describe the types of entrepreneurship and enterprises. Discuss the process of identifying opportunities for potential business and relevant regulatory and statutory requirements. Describe the 4Ps of Marketing-Product, Price, Place and Promotion and apply them as per requirement. Create a sample business plan, for the selected business opportunity. Discuss various sources of funding and identify associated financial and legal risks with its mitigation plan.
Customer Service Duration	Describe different types of customers. Role play a situation on how to identify customer needs and respond to them in a professional manner. Explain various tools used to collect customer feedback. Discuss the significance of maintaining hygiene and dressing appropriately.
Getting ready for apprenticeship & Jobs	Draft a professional Curriculum Vitae (CV). Use various offline and online job search sources such as employment exchanges, recruitment agencies, and job portals respectively. Demonstrate how to apply to identified job openings using offline /online methods as per requirement. Discuss how to prepare for an interview. Role play a mock interview. List the steps for searching and registering for apprenticeship opportunities.
Introduction to Artificial Intelligence (AI)	Understanding AI. How does AI work? Types of AI. What can AI do? Impact of AI on Jobs & Industries. Exploring Careers with AI. Learning with AI. Using AI Responsibly.

Table of Contents

Part – 1: Trade Theory | ट्रेड थ्योरी 1

Learning Outcome | अधिगम परिणाम – 1 2

1.1 Safe Working Environment and Electrical Safety Practices | सुरक्षित कार्य वातावरण और विद्युत सुरक्षा व्यवहार 2

1.2 Handling and Maintenance of Electrical Measuring Instruments|विद्युत मापन उपकरणों का संचालन एवं रखरखाव 3

1.3 Use and Maintenance of Pliers in Electrical and Electronics Work|सुरक्षित कार्य वातावरण एवं विद्युत सुरक्षा व्यवहार 4

1.4 Use and Maintenance of Clamps, Drills and Testers|क्लैम्प, ड्रिल और टेस्टर का उपयोग एवं रखरखाव .. 6

1.5 Use and Maintenance of Soldering Tools|सोल्डरिंग उपकरणों का उपयोग एवं अनुरक्षण 7

1.6 Classification of Materials as Conductors, Semiconductors, Insulators and Superconductors|सामग्रियों का चालक, अर्धचालक, कुचालक तथा अतिचालक के रूप में वर्गीकरण 8

1.7 Classification of Components into Active, Passive and Electromechanical Components | घटकों का सक्रिय, निष्क्रिय तथा विद्युत-यांत्रिक घटकों में वर्गीकरण 9

1.8 Types of Basic Electronic Components: Resistor, Capacitor, Diode and Transistor | मूलभूत इलेक्ट्रॉनिक अवयवों के प्रकार: रेसिस्टर, कैपेसिटर, डायोड और ट्रांजिस्टर 10

1.9 Working Principle of Basic Electronic Components | मूलभूत इलेक्ट्रॉनिक अवयवों का कार्य सिद्धांत 12

MCQ's | बहुविकल्पीय प्रश्न 13

Learning Outcome | अधिगम परिणाम – 2 15

2.1 Safe Working Environment in Electrical and Digital Electronics Laboratory|विद्युत एवं डिजिटल इलेक्ट्रॉनिक्स प्रयोगशाला में सुरक्षित कार्य परिवेश 15

2.2 Voltage, Current, Resistance and Capacitance in Electrical Circuits|विद्युत परिपथों में वोल्टेज, धारा, प्रतिरोध तथा धारिता 16

2.3 Ohm's Law, Current–Voltage Analysis and Ohmic & Non-Ohmic Devices|ओम का नियम, धारा–वोल्टेज विश्लेषण तथा ओमिक एवं नॉन-ओमिक युक्तियाँ..... 17

2.4 Electrical Power in Circuits and Effect of Voltage, Current and Resistance|परिपथों में विद्युत शक्ति तथा वोल्टेज, धारा और प्रतिरोध का प्रभाव 19

2.5 Reading Component Datasheets and Understanding Power Rating|अवयव डाटा शीट का अध्ययन तथा पावर रेटिंग की समझ 20

2.6 Functionality of Transistors and Integrated Circuits (ICs)|ट्रांजिस्टर तथा समेकित परिपथों (ICs) की कार्यप्रणाल 21

2.7 Boolean Algebra and Logic Design for Digital Functions|डिजिटल कार्यों के लिए बूलियन बीजगणित और लॉजिक डिज़ाइन 23

MCQ's | बहुविकल्पीय प्रश्न 25

Learning Outcome | अधिगम परिणाम – 3 27

3.1 Safe Working Environment in Electronics and PCB Work|इलेक्ट्रॉनिक्स एवं पीसीबी कार्य में सुरक्षित कार्य वातावरण 27

3.2 Measurement of AC and DC Voltage and Current|ए.सी. और डी.सी. वोल्टेज तथा धारा का मापन... 28

3.3 Construction and Testing of Electronic Circuit for Basic Applications|मूलभूत अनुप्रयोगों हेतु इलेक्ट्रॉनिक परिपथ का निर्माण एवं परीक्षण 29

3.4 Measurement and Calculation of Power in RLC Circuit|RLC सर्किट में शक्ति का मापन और गणना 31

MCQ's | बहुविकल्पीय प्रश्न 33

Learning Outcome | अधिगम परिणाम – 4 35

4.1 Safe Working Environment While Handling Sensors and Electronic Circuits|सेंसरों और इलेक्ट्रॉनिक परिपथों को संभालते समय सुरक्षित कार्य वातावरण 35

4.2 Selection of Sensors from Datasheet or Manual for Required Projects|आवश्यक परियोजना परिपथों हेतु डाटाशीट या मैनुअल से सेंसरों का चयन..... 36

4.3 Adjustment of Sensor Sensitivity for Proper Operation|उचित संचालन हेतु सेंसर संवेदनशीलता का समायोजन 38

4.4 Measurement of Output of PIR Sensor, IR Sensor and Similar Sensors |PIR सेंसर, IR सेंसर तथा समान सेंसरों के आउटपुट का मापन..... 39

MCQ's | बहुविकल्पीय प्रश्न 41

Learning Outcome | अधिगम परिणाम – 5 43

5.1 Safe Working Practices in Embedded Programming and IIoT Laboratory|एम्बेडेड प्रोग्रामिंग एवं IIoT प्रयोगशाला में सुरक्षित कार्य पद्धतियाँ 43

5.2 Variables, Data Types, Identifiers, Constants, Keywords and Operators in C Programming| सी प्रोग्रामिंग में वेरिएबल, डेटा टाइप, आइडेंटिफायर, कॉन्स्टेंट, कीवर्ड तथा ऑपरेटर 44

5.3 Loops, Control Statements and Functions in C Programming|C प्रोग्रामिंग में लूप्स, नियंत्रण कथन एवं फंक्शन्स 48

5.4 Concept of Embedded System and Open-Source Software Tools|एम्बेडेड सिस्टम तथा ओपन-सोर्स सॉफ्टवेयर टूल्स की संकल्पना 52

MCQ's | बहुविकल्पीय प्रश्न 54

Learning Outcome | अधिगम परिणाम – 6 56

6.1 Safe Working Practices in Microcontroller and Arduino Laboratory| माइक्रोकंट्रोलर और Arduino प्रयोगशाला में सुरक्षित कार्य प्रथाएँ 56

6.2 Classification of Microcontroller Families|माइक्रोकंट्रोलर परिवारों का वर्गीकरण 57

6.3 Parts of Microcontroller: CPU, ALU and GPIO| माइक्रोकंट्रोलर के भाग: CPU, ALU और GPIO 59

6.4 Classification of Microcontroller Families and Timer/Counter Units| माइक्रोकंट्रोलर परिवारों और टाइमर/काउंटर इकाइयों का वर्गीकरण 60

6.5 Serial Interface of Microcontroller|माइक्रोकंट्रोलर का सीरियल इंटरफ़ेस 62

6.6 Classification of Microcontroller Families and Interrupt System in Microcontroller|माइक्रोकंट्रोलर परिवारों का वर्गीकरण तथा माइक्रोकंट्रोलर में इंटरप्ट प्रणाली . 64

6.7 Memory Unit in Microcontroller|माइक्रोकंट्रोलर में मेमोरी यूनिट 65

6.8 ADC and DAC in Microcontroller| माइक्रोकंट्रोलर में ADC और DAC 67

6.9 Classification of Microcontroller Families| माइक्रोकंट्रोलर फैमिली का वर्गीकरण 68

MCQ's | बहुविकल्पीय प्रश्न 70

Learning Outcome | अधिगम परिणाम – 7 72

7.1 Safe Working Practices While Using Single Board Computers| सिंगल बोर्ड कंप्यूटर का उपयोग करते समय सुरक्षित कार्य प्रथाएँ 72

7.2 Introduction, Demonstration and Basic Setup of Single Board Computers| सिंगल बोर्ड कंप्यूटर का परिचय, प्रदर्शन और मूलभूत सेटअप 73

7.3 Introduction, Demonstration and Basic Setup of Single Board Computers – Installation of Open-Source Operating System on Raspberry Pi|सिंगल बोर्ड कंप्यूटर का परिचय, प्रदर्शन एवं मूल सेटअप – रास्पबेरी पाई कंप्यूटर पर ओपन-सोर्स ऑपरेटिंग सिस्टम का इंस्टॉलेशन 74

7.4 Use of SBC as a Web Server for IoT Applications|आईओटी अनुप्रयोगों के लिए वेब सर्वर के रूप में एसबीसी का उपयोग 76

7.5 Introduction, Demonstration and Basic Setup of Single Board Computers – Application of SBCs in Robotics Using Servo Motor|सिंगल बोर्ड कंप्यूटर का परिचय, प्रदर्शन और मूल सेटअप – सर्वो मोटर का उपयोग करते हुए रोबोटिक्स में SBCs का अनुप्रयोग 77

MCQ's | बहुविकल्पीय प्रश्न 79

Learning Outcome | अधिगम परिणाम – 8 81

8.1 Safe Working Practices in Microcontroller Interfacing and IIoT Systems|माइक्रोकंट्रोलर इंटरफेसिंग और IIoT सिस्टम्स में सुरक्षित कार्य प्रथाएँ 81

8.2 Safe Working Practices in Microcontroller Interfacing and IIoT Systems – Interfacing of Digital| माइक्रोकंट्रोलर इंटरफेसिंग और IIoT सिस्टम्स में सुरक्षित कार्य प्रथाएँ – डिजिटल सिस्टम्स का इंटरफेसिंग..... 82

8.3 Safe Working Practices in Microcontroller Interfacing and IIoT Systems – Interfacing of Analog Sensors with Microcontroller|माइक्रोकंट्रोलर इंटरफेसिंग और IIoT सिस्टम्स में सुरक्षित कार्य प्रथाएँ – माइक्रोकंट्रोलर के साथ एनालॉग सेंसर का इंटरफेसिंग 84

8.4 Safe Working Practices in Microcontroller Interfacing and IIoT Systems – Interfacing of Actuators, Relay and Output Devices|माइक्रोकंट्रोलर इंटरफेसिंग और IIoT सिस्टम्स में सुरक्षित कार्य प्रथाएँ – एक्ट्यूएटर्स, रिले और आउटपुट डिवाइस का इंटरफेसिंग 85

8.5 Safe Working Practices in Microcontroller Interfacing and IIoT Systems – Interfacing of Communication Modules|माइक्रोकंट्रोलर इंटरफेसिंग और IIoT सिस्टम्स में सुरक्षित कार्य प्रथाएँ – संचार मॉड्यूल्स का इंटरफेसिंग 87

MCQ's | बहुविकल्पीय प्रश्न 89

Learning Outcome | अधिगम परिणाम – 9 92

9.1 Safe Working Practices in IIoT Networking and Cloud-Based Applications|IIoT नेटवर्किंग और क्लाउड-आधारित अनुप्रयोगों में सुरक्षित कार्य पद्धतियाँ 92

9.2 Concept and Fundamentals of Industrial Internet of Things (IIoT)| औद्योगिक इंटरनेट ऑफ थिंग्स (IIoT) की अवधारणा और मूल सिद्धांत (IIoT) एप्लिकेशन्स . 93

9.3 Safe Working Practices in IIoT Networking and Cloud-Based Applications – Four-Layer Architecture of IIoT Systems|आईआईओटी नेटवर्किंग और क्लाउड-आधारित अनुप्रयोगों में सुरक्षित कार्य प्रथाएँ – आईआईओटी सिस्टम की चार-स्तरीय आर्किटेक्चर 94

9.4 Safe Working Practices in IIoT Networking and Cloud-Based Applications – HTTP Methods Using Postman| IIoT नेटवर्किंग और क्लाउड-आधारित एप्लिकेशन्स में सुरक्षित कार्य प्रथाएँ – पोस्टमैन का उपयोग करके HTTP मेथड्स 96

9.5 Real-Time Applications of Internet of Things (IoT)| इंटरनेट ऑफ थिंग्स (IoT) के रियल-टाइम अनुप्रयोग97

9.6 Safe Working Practices in IIoT Networking and Cloud-Based Applications – Network Topologies Used in IIoT Communication Systems| आईआईओटी नेटवर्किंग और क्लाउड-आधारित अनुप्रयोगों में सुरक्षित कार्य पद्धतियाँ – आईआईओटी संचार प्रणालियों में प्रयुक्त नेटवर्क टोपोलॉजी..... 98

MCQ's | बहुविकल्पीय प्रश्न 100

Learning Outcome | अधिगम परिणाम – 10 102

10.1 Safe Working Practices in IIoT Data Handling and Computer Laboratory|आईआईओटी डेटा हैंडलिंग और कंप्यूटर प्रयोगशाला में सुरक्षित कार्य पद्धतियाँ..... 102

10.2 Safe Working Practices in IIoT Data Handling and Computer Laboratory – Introduction to IIoT and Data Management|आईआईओटी डेटा हैंडलिंग और कंप्यूटर प्रयोगशाला में सुरक्षित कार्य पद्धतियाँ – आईआईओटी और डेटा प्रबंधन प्रयोगशाला का परिचय..... 103

10.3 Safe Working Practices in IIoT Data Handling and Computer Laboratory – Overview of Open-Source Database Types Used in IIoT| आईआईओटी डेटा हैंडलिंग और कंप्यूटर प्रयोगशाला में सुरक्षित कार्य पद्धतियाँ – आईआईओटी में उपयोग किए जाने वाले ओपन-सोर्स डेटाबेस प्रकारों का अवलोकन 104

10.4 Safe Working Practices in IIoT Data Handling and Computer Laboratory – Database Characteristics and Suitability for IIoT Applications| आईआईओटी डेटा हैंडलिंग और कंप्यूटर प्रयोगशाला में सुरक्षित कार्य पद्धतियाँ – आईआईओटी अनुप्रयोगों के लिए डेटाबेस की विशेषताएँ और उपयुक्तता..... 106

10.5 Safe Working Practices in IIoT Data Handling and Computer Laboratory – Ethical Considerations in IIoT Data Management| आईआईओटी डेटा हैंडलिंग और कंप्यूटर प्रयोगशाला में सुरक्षित कार्य पद्धतियाँ – आईआईओटी डेटा प्रबंधन में नैतिक विचार 107

MCQ's | बहुविकल्पीय प्रश्न 109

Learning Outcome | अधिगम परिणाम – 11 111

11.1 Safe Working Practices in IIoT Connectivity and Cloud-Integrated Systems| IIoT कनेक्टिविटी और क्लाउड-इंटीग्रेटेड सिस्टम में सुरक्षित कार्य पद्धतियाँ..... 111

11.2 High Power, High Range and High Bandwidth Communication Technologies| उच्च शक्ति, उच्च परास और उच्च बैंडविड्थ संचार प्रौद्योगिकियाँ..... 112

11.3 Safe Working Practices in IIoT Connectivity and Cloud-Integrated Systems – Low Power, Low Range and High Bandwidth Communication Technologies|IIoT कनेक्टिविटी और क्लाउड-एकीकृत प्रणालियों में सुरक्षित कार्य प्रथाएँ – कम शक्ति, कम रेंज और उच्च बैंडविड्थ संचार प्रौद्योगिकियाँ..... 113

11.4 High Power, High Range and High Bandwidth Communication Technologies & Low Power, High Range and Low Bandwidth Communication Technologies| उच्च शक्ति, उच्च रेंज तथा उच्च बैंडविड्थ संचार प्रौद्योगिकियाँ एवं निम्न शक्ति, उच्च रेंज तथा निम्न बैंडविड्थ संचार प्रौद्योगिकियाँ..... 115

11.5 High Power, High Range and High Bandwidth Communication Technologies – Wi-Fi Connectivity Between Hardware and Cloud|उच्च शक्ति, उच्च रेंज और उच्च बैंडविड्थ संचार प्रौद्योगिकियाँ – हार्डवेयर और क्लाउड के बीच वाई-फाई कनेक्टिविटी..... 116

11.6 High Power, High Range and High Bandwidth Communication Technologies – Data Collection, Storage and Data Management in IIoT Systems| उच्च शक्ति, उच्च रेंज और उच्च बैंडविड्थ संचार प्रौद्योगिकियाँ – IIoT प्रणालियों में डेटा संग्रह, भंडारण और डेटा प्रबंधन..... 117

11.7 High Power, High Range and High Bandwidth Communication Technologies – Connecting Wi-Fi Module to Router Through Programming| उच्च शक्ति, उच्च रेंज और उच्च बैंडविड्थ संचार प्रौद्योगिकियाँ – प्रोग्रामिंग के माध्यम से वाई-फाई मॉड्यूल को राउटर से जोड़ना 118

11.8 High Power, High Range and High Bandwidth Communication Technologies – Interfacing Sensor with Wi-Fi Module|उच्च शक्ति, उच्च रेंज और उच्च बैंडविड्थ संचार प्रौद्योगिकियाँ – सेंसर को वाई-फाई मॉड्यूल के साथ इंटरफेसिंग 119

MCQ's | बहुविकल्पीय प्रश्न 121

Learning Outcome | अधिगम परिणाम – 12 123

12.1 Safe Working Practices in Python Programming and IIoT Laboratory|पायथन प्रोग्रामिंग और IIoT प्रयोगशाला में सुरक्षित कार्य प्रथाएँ..... 123

12.2 Fundamentals of Python Programming for IIoT Applications|आईआईओटी अनुप्रयोगों के लिए पाइथन प्रोग्रामिंग के मूल सिद्धांत..... 124

12.3 Object-Oriented Programming in Python|पायथन में ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग..... 126

12.4 Code Efficiency and Optimization in Python Programming| पायथन प्रोग्रामिंग में कोड दक्षता और अनुकूलन 127

12.5 Fundamentals of Python Programming for IIoT Applications Problem-Solving and Algorithm Design in Python| IIoT अनुप्रयोगों के लिए पाइथन प्रोग्रामिंग के मूलभूत सिद्धांत – पाइथन में समस्या समाधान और एल्गोरिथ्म डिजाइ 129

12.6 Object-Oriented Programming in Python – Code Readability and Style in Python

Programming|पायथन में ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग – पायथन प्रोग्रामिंग में कोड पठनीयता और शैली 130

12.7 Safe Working Practices in Python Programming and IIoT Laboratory: Error Handling and Debugging in Python| पायथन प्रोग्रामिंग और IIoT प्रयोगशाला में सुरक्षित कार्य पद्धतियाँ: पायथन में त्रुटि प्रबंधन और डिबगिंग..... 131

MCQ's | बहुविकल्पीय प्रश्न 133

Learning Outcome | अधिगम परिणाम – 13 135

13.1 Safe Working Practices in Raspberry Pi and Peripheral Interfacing| रास्पबेरी पाई और परिधीय इंटरफेसिंग में सुरक्षित कार्यप्रणाली..... 135

13.2 Raspberry Pi Processor and System Overview| रास्पबेरी पाई प्रोसेसर और सिस्टम अवलोकन 136

13.3 PIN Configuration, Memory, Graphics, Ethernet and USB Ports of Raspberry Pi| रास्पबेरी पाई के पिन कॉन्फ़िगरेशन, मेमोरी, ग्राफिक्स, ईथरनेट और USB पोर्ट 138

13.4 Safe Working Practices in Raspberry Pi and Peripheral Interfacing: Installation of Python Editors and Supporting Software on Raspberry Pi| रास्पबेरी पाई और पेरिफेरल इंटरफेसिंग में सुरक्षित कार्य प्रथाएँ: रास्पबेरी पाई पर पायथन एडिटर्स और सहायक सॉफ़्टवेयर की स्थापना..... 139

MCQ's | बहुविकल्पीय प्रश्न 141

Learning Outcome | अधिगम परिणाम – 14 143

14.1 Safe Working Practices in Android, Web and IIoT Application Development| एंड्रॉइड, वेब और IIoT एप्लिकेशन विकास में सुरक्षित कार्य अभ्यास 143

14.2 Safe Working Practices in Android, Web and IIoT Application Development: User Interface Design in Web and Android Applications| एंड्रॉइड, वेब और IIoT अनुप्रयोग विकास में सुरक्षित कार्य अभ्यास: वेब और एंड्रॉइड अनुप्रयोगों में उपयोगकर्ता इंटरफेस डिज़ाइ..... 144

14.3 Raspberry Pi Processor and System Overview Use of Palette, Viewer and Component Properties in App Development|रास्पबेरी पाई प्रोसेसर और सिस्टम अवलोकन एप्लिकेशन विकास में पैलेट, व्यूअर और कंपोनेंट प्रॉपर्टीज़ का उपयोग 146

14.4 Raspberry Pi Processor and System Overview Logic Flow Design for Android and Web Application|रास्पबेरी पाई प्रोसेसर और सिस्टम अवलोकन – एंड्रॉइड और वेब एप्लिकेशन के लिए लॉजिक फ्लो डिज़ाइन. 147

14.5 Backend Development Using Block Coding| ब्लॉक कोडिंग का उपयोग करके बैकएंड विकास..... 148

MCQ's | बहुविकल्पीय प्रश्न 151

15.1 Safe Working Practices in Programming and Interfacing of IIoT Devices|IIoT डिवाइसों के प्रोग्रामिंग और इंटरफेसिंग में सुरक्षित कार्य प्रथाए..... 153

15.2 Selection of Sensors for Industrial Automation Based on Problem Statement| संपूर्ण स्वचालन के लिए समस्या विवरण के आधार पर सेंसर का चयन 154

15.3 Implementation of Different Sensors and Output Devices in IIoT Applications| IIoT अनुप्रयोगों में विभिन्न सेंसर और आउटपुट डिवाइस का कार्यान्वयन 155

15.4 Measurement and Monitoring of Sensor Values Using Tools and Serial Monitor| उपकरणों और सीरियल मॉनिटर का उपयोग करके सेंसर मानों का मापन और निगरानी 157

15.5 Uploading IIoT Data to Cloud or Virtual Database and Testing the System| IIoT डेटा को क्लाउड या वर्चुअल डेटाबेस में अपलोड करना और सिस्टम का परीक्षण..... 158

15.6 Interfacing Output Devices Controlled by Cloud or Remote Access| क्लाउड या रिमोट एक्सेस द्वारा नियंत्रित आउटपुट डिवाइस का इंटरफेसिंग 160

MCQ's | बहुविकल्पीय प्रश्न 162

Learning Outcome | अधिगम परिणाम – 16 164

16.1 Safe Working Practices in Secure IIoT Systems and Network Environment| सुरक्षित IIoT सिस्टम और नेटवर्क वातावरण में सुरक्षित कार्य प्रथाएँ..... 164

16.2 IoT Communication Protocols: MQTT, CoAP, HTTP and HTTPS| आईओटी संचार प्रोटोकॉल: MQTT, CoAP, HTTP और HTTPS..... 165

16.3 Machine-to-Machine Communication and Internet of Things| मशीन-टू-मशीन संचार और इंटरनेट ऑफ थिंग्स 166

16.4 Cloud Services in IIoT: SaaS, PaaS and IaaS|IIoT में क्लाउड सेवाएँ: SaaS, PaaS और IaaS.... 168

16.5 Types of Attacks in IoT Systems| आईओटी सिस्टम में हमलों के प्रकार 169

16.6 IoT Security Threats and Vulnerabilities|आईओटी सुरक्षा खतरे और कमजोरियां . 170

Part – 2: Employability Skills | रोजगार योग्य कौशल 174

1. Introduction to Employability Skills | रोज़गारयोग्यता कौशल का परिचय..... 175

MCQ's | बहुविकल्पीय प्रश्न 178

2. Constitutional values – Citizenship | संवैधानिक मूल्य - नागरिकता 184

MCQ's | बहुविकल्पीय प्रश्न 186

3. Becoming a Professional in the 21st Century | 21वीं सदी में एक पेशेवर बनना..... 193

MCQ's | बहुविकल्पीय प्रश्न 195

4. Basic English Skills | मूल अंग्रेज़ी कौशल 202

MCQ's | बहुविकल्पीय प्रश्न 204

5. Career Development & Goal Setting | कैरियर विकास और लक्ष्य निर्धारण 210

MCQ's | बहुविकल्पीय प्रश्न 212

6. Communication Skills | संचार कौशल 219

MCQ's | बहुविकल्पीय प्रश्न 221

7. Diversity and Inclusion | विविधता और समावेशन 227

MCQ's | बहुविकल्पीय प्रश्न 229

8. Financial and Legal Literacy | वित्तीय और कानूनी साक्षरता 236

MCQ's | बहुविकल्पीय प्रश्न 238

9. Essential Digital Skills | आवश्यक डिजिटल कौशल 244

MCQ's | बहुविकल्पीय प्रश्न 246

10. Entrepreneurship | उद्यमिता 253

MCQ's | बहुविकल्पीय प्रश्न 255

11. Customer Service | ग्राहक सेवा 262

MCQ's | बहुविकल्पीय प्रश्न 264

12. Getting ready for apprenticeship & Jobs | प्रशिक्षुता और नौकरियों के लिए तैयारी 271

MCQ's | बहुविकल्पीय प्रश्न 273

13. Introduction to Artificial Intelligence (AI) | कृत्रिम बुद्धिमता का परिचय 280

MCQ's | बहुविकल्पीय प्रश्न 284

Part – 3: Mock Tests | मॉक टेस्ट्स 291

Mock Tests | मॉक टेस्ट - 1 292

Mock Tests | मॉक टेस्ट - 2 301

"Every Concept Powers You One Step Closer to Success."

Part – 1: Trade Theory | ट्रेड थ्योरी

Learning Outcome | अधिगम परिणाम – 5

5.1 Safe Working Practices in Embedded Programming and IIoT Laboratory|एम्बेडेड प्रोग्रामिंग एवं IIoT प्रयोगशाला में सुरक्षित कार्य पद्धतियाँ

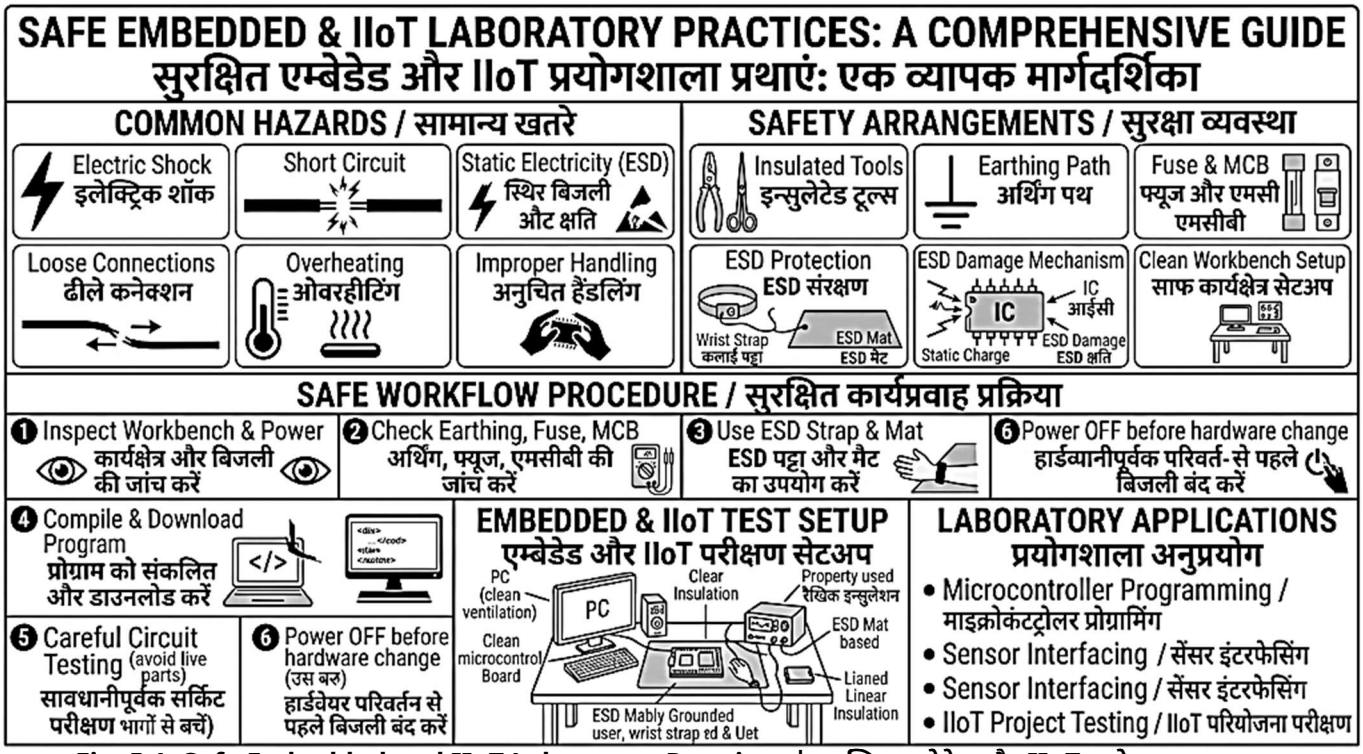


Fig. 5.1: Safe Embedded and IIoT Laboratory Practices | सुरक्षित एम्बेडेड और IIoT प्रयोगशाला अभ्यास

Definition (Fig. 5.1)

A safe working environment in an embedded programming and IIoT laboratory means maintaining proper conditions, safe equipment, and correct work practices so that programming, hardware interfacing, downloading, and testing can be carried out without injury, device damage, or data loss. It includes clean workbench, proper ventilation, correct wiring, and safe handling of microcontroller boards, computers, and power supply units.

Importance of Safety

Safety is important while working with computers, microcontroller boards, power supply, and interfacing circuits because IIoT work combines software and hardware. Proper safety practice prevents electric shock, short circuit, overheating, and damage to sensitive electronic devices. It also improves reliability, accuracy, and confidence during practical work.

Types of Common Hazards

Common hazards are **electric shock**, **short circuit**, **static electricity**, **loose connections**, **overheating**, and **improper handling of devices**. Static electricity may damage microcontrollers and ICs. Loose wiring may cause wrong output or failure. Overheating may occur due to excess current or poor ventilation.

परिभाषा (Fig. 5.1)

एम्बेडेड प्रोग्रामिंग और IIoT प्रयोगशाला में सुरक्षित कार्य वातावरण का अर्थ है उचित परिस्थितियों, सुरक्षित उपकरणों, और सही कार्य पद्धतियों को बनाए रखना, ताकि प्रोग्रामिंग, हार्डवेयर इंटरफेसिंग, डाउनलोडिंग, और परीक्षण बिना किसी चोट, उपकरण क्षति, या डेटा हानि के किए जा सकें। इसमें स्वच्छ कार्यबेंच, उचित वेंटिलेशन, सही वायरिंग, तथा माइक्रोकंट्रोलर बोर्ड, कंप्यूटर, और पावर सप्लाई इकाइयों का सुरक्षित संचालन सम्मिलित है।

सुरक्षा का महत्व

कंप्यूटर, माइक्रोकंट्रोलर बोर्ड, पावर सप्लाई, और इंटरफेसिंग परिपथों के साथ कार्य करते समय सुरक्षा महत्वपूर्ण है क्योंकि IIoT कार्य में सॉफ्टवेयर और हार्डवेयर दोनों सम्मिलित होते हैं। उचित सुरक्षा अभ्यास विद्युत आघात, शॉर्ट सर्किट, अत्यधिक ताप, और संवेदनशील इलेक्ट्रॉनिक युक्तियों की क्षति को रोकता है। यह व्यावहारिक कार्य के दौरान विश्वसनीयता, शुद्धता, और आत्मविश्वास में भी सुधार करता है।

सामान्य खतरों के प्रकार

सामान्य खतरे विद्युत आघात, शॉर्ट सर्किट, स्थैतिक विद्युत, ढीले कनेक्शन, अधिक ताप, और उपकरणों का अनुचित हैंडलिंग हैं। स्थैतिक विद्युत माइक्रोकंट्रोलर और ICs को क्षति पहुंचा सकती है। ढीली वायरिंग गलत आउटपुट या विफलता का कारण बन सकती है। अधिक धारा या खराब वेंटिलेशन के कारण अधिक ताप उत्पन्न हो सकता है।

Constructional Features of Safety Arrangements
Important safety arrangements include insulated tools, earthing, ESD protection, fuse, MCB, and proper workbench setup. Insulated tools reduce shock risk. Earthing provides a safe path for leakage current. ESD wrist strap and mat protect electronic components from static charge. Fuse and MCB protect the circuit from overload and fault.

Working Principle
First, inspect the workbench, supply, and connections. Use ESD protection and insulated tools. Check earthing, fuse, and MCB before powering the circuit. Compile the program correctly, then download it to the board using proper cable connection. Test the circuit carefully, avoid touching live parts, and switch off the supply before changing any hardware connection.

Merits and Demerits
Following safety procedures prevents accidents, protects equipment, and improves project success. Ignoring safety may cause electric shock, board damage, data loss, wrong output, and equipment failure.

Applications
These practices are used in microcontroller programming, sensor interfacing, and IIoT project testing.

सुरक्षा व्यवस्थाओं की संरचनात्मक विशेषताएँ
महत्वपूर्ण सुरक्षा व्यवस्थाओं में इंसुलेटेड टूल्स, अर्थिंग, ESD प्रोटेक्शन, फ्यूज, MCB, और उचित वर्कबेंच सेटअप शामिल हैं। इंसुलेटेड टूल्स झटके के जोखिम को कम करते हैं। अर्थिंग लीकेज करंट के लिए सुरक्षित मार्ग प्रदान करती है। ESD रिस्ट स्ट्रैप और मैट इलेक्ट्रॉनिक कंपोनेंट्स को स्थैतिक आवेश से बचाते हैं। फ्यूज और MCB ओवरलोड और फॉल्ट से परिपथ की सुरक्षा करते हैं।

कार्य सिद्धांत
सबसे पहले, कार्य मेज, आपूर्ति और संयोजनों का निरीक्षण करें। ESD सुरक्षा और इन्सुलेटेड उपकरणों का उपयोग करें। परिपथ को चालू करने से पहले अर्थिंग, फ्यूज और MCB की जाँच करें। प्रोग्राम को सही तरीके से संकलित करें, फिर उचित केबल संयोजन का उपयोग करके इसे बोर्ड में डाउनलोड करें। परिपथ का सावधानीपूर्वक परीक्षण करें, जीवित भागों को छूने से बचें, और किसी भी हार्डवेयर संयोजन को बदलने से पहले आपूर्ति को बंद करें।

लाभ और हानियाँ
सुरक्षा प्रक्रियाओं का पालन करने से दुर्घटनाओं की रोकथाम होती है, उपकरणों की सुरक्षा होती है, और परियोजना की सफलता में सुधार होता है। सुरक्षा की उपेक्षा करने से विद्युत आघात, बोर्ड क्षति, डेटा हानि, गलत आउटपुट, और उपकरण विफलता हो सकती है।

अनुप्रयोग
इन प्रथाओं का उपयोग माइक्रोकंट्रोलर प्रोग्रामिंग, सेंसर इंटरफेसिंग, और IIoT परियोजना परीक्षण में किया जाता है।

5.2 Variables, Data Types, Identifiers, Constants, Keywords and Operators in C Programming| सी प्रोग्रामिंग में वेरिएबल, डेटा टाइप, आइडेंटिफायर, कॉन्स्टेंट, कीवर्ड तथा ऑपरेटर

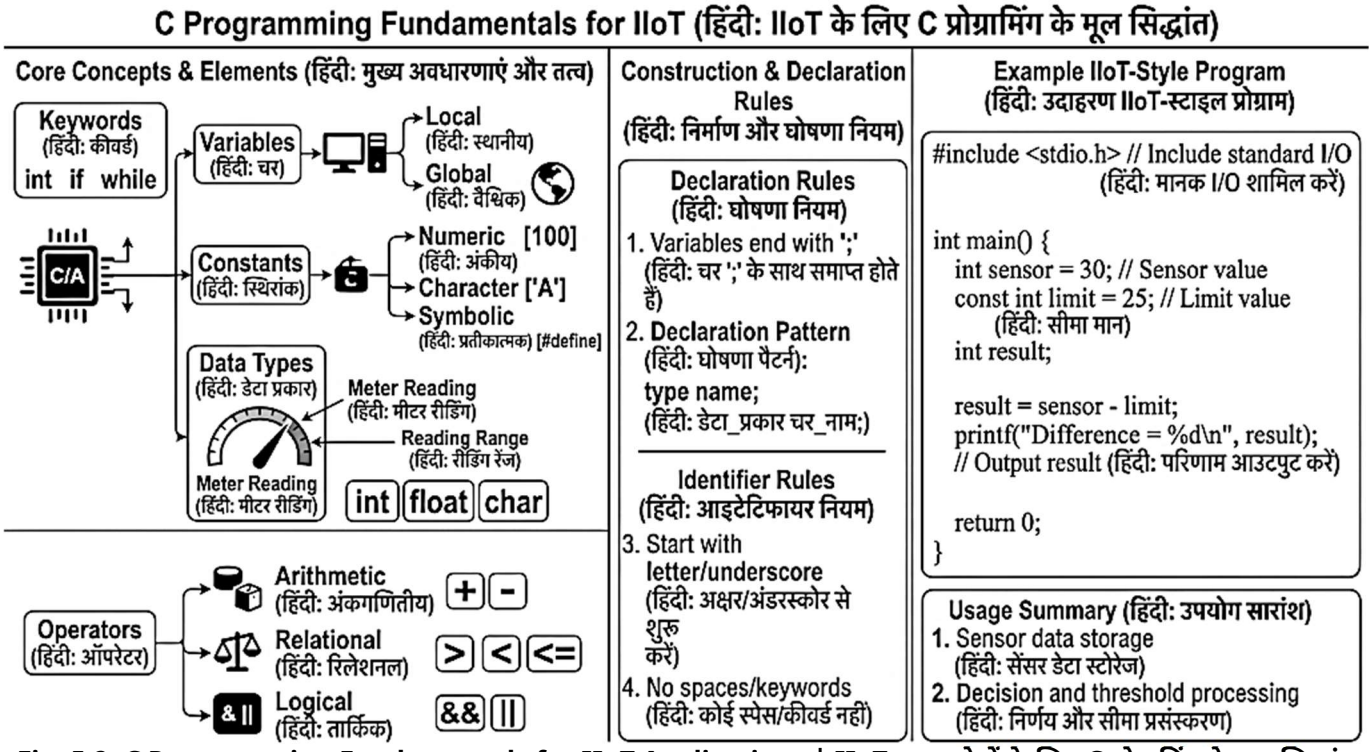


Fig. 5.2: C Programming Fundamentals for IIoT Applications | IIoT अनुप्रयोगों के लिए C प्रोग्रामिंग के मूल सिद्धांत

Definition (Fig. 5.2)
In C language, a **variable** is a named memory location used to store data. A **data type** defines the kind of value stored, such as integer or character. An **identifier** is the name given to variables, functions, or arrays. A **constant** is a fixed value that does not change during program execution. **Keywords** are reserved words such as int, float, if, and while. **Operators** are symbols used to perform calculations and comparisons.

Importance in IIoT Programming
These basic elements are important because every IIoT program uses variables to store sensor data, constants for fixed values, data types for memory control, and operators for decision-making and calculations. Proper use improves program accuracy, readability, and execution.

Classification
Variables may be local or global. Constants may be numeric, character, or symbolic. Basic data types are int, float, char, and double. Derived data types include arrays, pointers, structures, and unions. Operators are of arithmetic, relational, logical, assignment, and increment/decrement types.

Constructional Features
A C statement ends with a semicolon. Variable declaration follows the format: data_type variable_name; Example: int temp;. Identifiers must begin with a letter or underscore and should not use spaces or keywords. Operator symbols include +, -, *, /, >, <, &&, ||, =, ++, and --.

Working Principle
First, declare variables with suitable data types. Then assign values using the assignment operator. Use arithmetic and logical operators in expressions to process data. Finally, display or use the output in control or sensing action.

Examples
These elements are used in sensor value storage, threshold comparison, and arithmetic processing in IIoT programs.

Examples
These elements are used in **sensor value storage, threshold comparison, arithmetic processing, and decision-making** in IIoT programs.

Example 1: Simple Variable Program

```
#include <stdio.h>

int main() {
   int temperature = 28;
   printf("Temperature = %d", temperature);
   return 0;
```

परिभाषा (Fig. 5.2)
C भाषा में, वेरिएबल एक नामित मेमोरी स्थान होता है जिसका उपयोग डेटा को संग्रहित करने के लिए किया जाता है। डेटा टाइप संग्रहित मान के प्रकार को परिभाषित करता है, जैसे पूर्णांक या अक्षर। आइडेंटिफ़ायर वह नाम है जो वेरिएबल, फ़ंक्शन, या ऐरे को दिया जाता है। कॉन्स्टेंट एक स्थिर मान होता है जो प्रोग्राम के निष्पादन के दौरान परिवर्तित नहीं होता। कीवर्ड आरक्षित शब्द होते हैं, जैसे int, float, if, और while। ऑपरेटर वे प्रतीक होते हैं जिनका उपयोग गणना और तुलना करने के लिए किया जाता है।

IIoT प्रोग्रामिंग में महत्व
ये मूल तत्व महत्वपूर्ण हैं क्योंकि प्रत्येक IIoT प्रोग्राम सेंसर डेटा को संग्रहीत करने के लिए वेरिएबल्स, स्थिर मानों के लिए कॉन्स्टैंट्स, मेमोरी नियंत्रण के लिए डेटा टाइप्स, तथा निर्णय-निर्धारण और गणनाओं के लिए ऑपरेटर्स का उपयोग करता है। इनका उचित उपयोग प्रोग्राम की शुद्धता, पठनीयता, और निष्पादन में सुधार करता है।

वर्गीकरण
वेरिएबल्स स्थानीय (लोकल) या वैश्विक (ग्लोबल) हो सकते हैं। कॉन्स्टेंट्स संख्यात्मक, वर्णात्मक, या प्रतीकात्मक हो सकते हैं। बेसिक डेटा टाइप्स int, float, char, और double हैं। व्युत्पन्न (डेराइव्ड) डेटा टाइप्स में arrays, pointers, structures, और unions शामिल हैं। ऑपरेटर्स अंकगणितीय, रिलेशनल, लॉजिकल, असाइनमेंट, और इन्क्रीमेंट/डिक्रीमेंट प्रकार के होते हैं।

संरचनात्मक विशेषताएँ
एक C स्टेटमेंट सेमीकोलन से समाप्त होता है। वेरिएबल डिक्लेरेशन का प्रारूप होता है: data_type variable_name; उदाहरण: int temp;. आइडेंटिफायर का प्रारंभ अक्षर या अंडरस्कोर से होना चाहिए और इसमें स्पेस या कीवर्ड का उपयोग नहीं होना चाहिए। ऑपरेटर प्रतीकों में +, -, *, /, >, <, &&, ||, =, ++, और -- शामिल हैं।

कार्य सिद्धांत
सबसे पहले, उपयुक्त डेटा प्रकारों के साथ चर घोषित करें। फिर असाइनमेंट ऑपरेटर का उपयोग करके मान निर्दिष्ट करें। डेटा को संसाधित करने के लिए अभिव्यक्तियों में अंकगणितीय और लॉजिकल ऑपरेटरों का उपयोग करें। अंत में, आउटपुट को प्रदर्शित करें या उसे नियंत्रण या संवेदन क्रिया में उपयोग करें।

उदाहरण
ये तत्व IIoT प्रोग्रामों में सेंसर मान संग्रहण, थ्रेशोल्ड तुलना, तथा अंकगणितीय प्रसंस्करण में उपयोग किए जाते हैं।

उदाहरण
इन तत्त्वों का उपयोग IIoT प्रोग्रामों में **सेंसर मान (Sensor Value) का संग्रहण, थ्रेशोल्ड (सीमा) की तुलना, अंकगणितीय गणना तथा निर्णय लेने (Decision Making)** के लिए किया जाता है।

उदाहरण 1: सरल वेरिएबल प्रोग्राम

```
#include <stdio.h>

int main() {
   int temperature = 28;
   printf("Temperature = %d", temperature);
   return 0;
```

```
}
```

Output: Temperature = 28

Explanation:

The variable temperature stores an integer value, which is displayed using printf().

Example 2: Using Different Data Types

```
#include <stdio.h>

int main() {
   int count = 10;
   float voltage = 12.5;
   char status = 'A';

   printf("Count = %d\n", count);
   printf("Voltage = %.1f V\n", voltage);
   printf("Status = %c\n", status);

   return 0;
}
```

Output: Count = 10
Voltage = 12.5 V
Status = A

Explanation:

This program demonstrates three basic data types: int, float, and char.

Example 3: Using Arithmetic Operators

```
#include <stdio.h>

int main() {
   int a = 20, b = 5;

   printf("Addition = %d\n", a + b);
   printf("Subtraction = %d\n", a - b);
   printf("Multiplication = %d\n", a * b);
   printf("Division = %d\n", a / b);

   return 0;
}
```

Output: Addition = 25
Subtraction = 15
Multiplication = 100
Division = 4

Explanation:

Arithmetic operators (+, -, *, /) perform mathematical calculations.

Example 4: Simple C Program Using Variables, Constants, and Operators

```
#include <stdio.h>

int main() {
   int sensor = 30;
   const int limit = 25;
}
```

आउटपुट: Temperature = 28

व्याख्या:

इस प्रोग्राम में **temperature** नामक वेरिएबल (चर) में पूर्णांक (Integer) मान **28** संग्रहीत किया गया है, जिसे printf() फ़ंक्शन की सहायता से प्रदर्शित किया गया है।

उदाहरण 2: विभिन्न डेटा प्रकारों का उपयोग

```
#include <stdio.h>

int main() {
   int count = 10;
   float voltage = 12.5;
   char status = 'A';

   printf("Count = %d\n", count);
   printf("Voltage = %.1f V\n", voltage);
   printf("Status = %c\n", status);

   return 0;
}
```

आउटपुट: Count = 10
Voltage = 12.5 V
Status = A

व्याख्या:

यह प्रोग्राम C भाषा के तीन मूलभूत डेटा प्रकारों **int (पूर्णांक)**, **float (दशमलव संख्या)** तथा **char (वर्ण)** के उपयोग को प्रदर्शित करता है।

उदाहरण 3: अंकगणितीय ऑपरेटरों का उपयोग

```
#include <stdio.h>

int main() {
   int a = 20, b = 5;

   printf("Addition = %d\n", a + b);
   printf("Subtraction = %d\n", a - b);
   printf("Multiplication = %d\n", a * b);
   printf("Division = %d\n", a / b);

   return 0;
}
```

आउटपुट: Addition = 25
Subtraction = 15
Multiplication = 100
Division = 4

व्याख्या:

इस प्रोग्राम में अंकगणितीय ऑपरेटर **(+, −, ×, ÷)** अर्थात **+**, **-**, ***** तथा **/** का उपयोग करके गणितीय संक्रियाएँ की गई हैं।

उदाहरण 4: वेरिएबल, कॉन्स्टेंट तथा ऑपरेटर का उपयोग

```
#include <stdio.h>

int main() {
   int sensor = 30;
   const int limit = 25;
```

```
    int result = sensor - limit;

    printf("Sensor Value = %d\n", sensor);
    printf("Limit = %d\n", limit);
    printf("Difference = %d\n", result);

    return 0;
}
```

Output: Sensor Value = 30
Limit = 25
Difference = 5

Explanation:
The variable sensor stores the measured value, while limit is declared as a constant because it remains unchanged. The subtraction operator (-) calculates the difference between the sensor value and the fixed limit.

Example 5: Using Relational Operator

```
#include <stdio.h>

int main() {
    int sensor = 30;

    if (sensor > 25)
        printf("Sensor value is above the limit.");
    else
        printf("Sensor value is within the limit.");

    return 0;
}
```

Output: Sensor value is above the limit.

Explanation:
The relational operator (>) compares two values and is widely used in IIoT applications for threshold detection, alarms, and automatic control.

```
    int result = sensor - limit;

    printf("Sensor Value = %d\n", sensor);
    printf("Limit = %d\n", limit);
    printf("Difference = %d\n", result);

    return 0;
}
```

आउटपुट: Sensor Value = 30
Limit = 25
Difference = 5

व्याख्या:
इस प्रोग्राम में **sensor** एक वेरिएबल है, जिसमें मापा गया मान संग्रहीत किया जाता है, जबकि **limit** को **कॉन्स्टेंट (स्थिरांक)** घोषित किया गया है क्योंकि इसका मान पूरे प्रोग्राम के दौरान अपरिवर्तित रहता है। **घटाव (-) ऑपरेटर** का उपयोग करके सेंसर मान और निर्धारित सीमा के बीच का अंतर ज्ञात किया गया है।

उदाहरण 5: रिलेशनल ऑपरेटर का उपयोग

```
#include <stdio.h>

int main() {
    int sensor = 30;

    if (sensor > 25)
        printf("Sensor value is above the limit.");
    else
        printf("Sensor value is within the limit.");

    return 0;
}
```

आउटपुट: Sensor value is above the limit.

व्याख्या:
इस प्रोग्राम में **रिलेशनल (तुलनात्मक) ऑपरेटर >** का उपयोग करके दो मानों की तुलना की गई है। ऐसे ऑपरेटरों का व्यापक उपयोग **औद्योगिक इंटरनेट ऑफ थिंग्स (IIoT)** अनुप्रयोगों में **सीमा (Threshold) का पता लगाने, अलार्म उत्पन्न करने तथा स्वचालित नियंत्रण (Automatic Control)** के लिए किया जाता है।

5.3 Loops, Control Statements and Functions in C Programming|C प्रोग्रामिंग में लूप्स, नियंत्रण कथन एवं फंक्शन्स

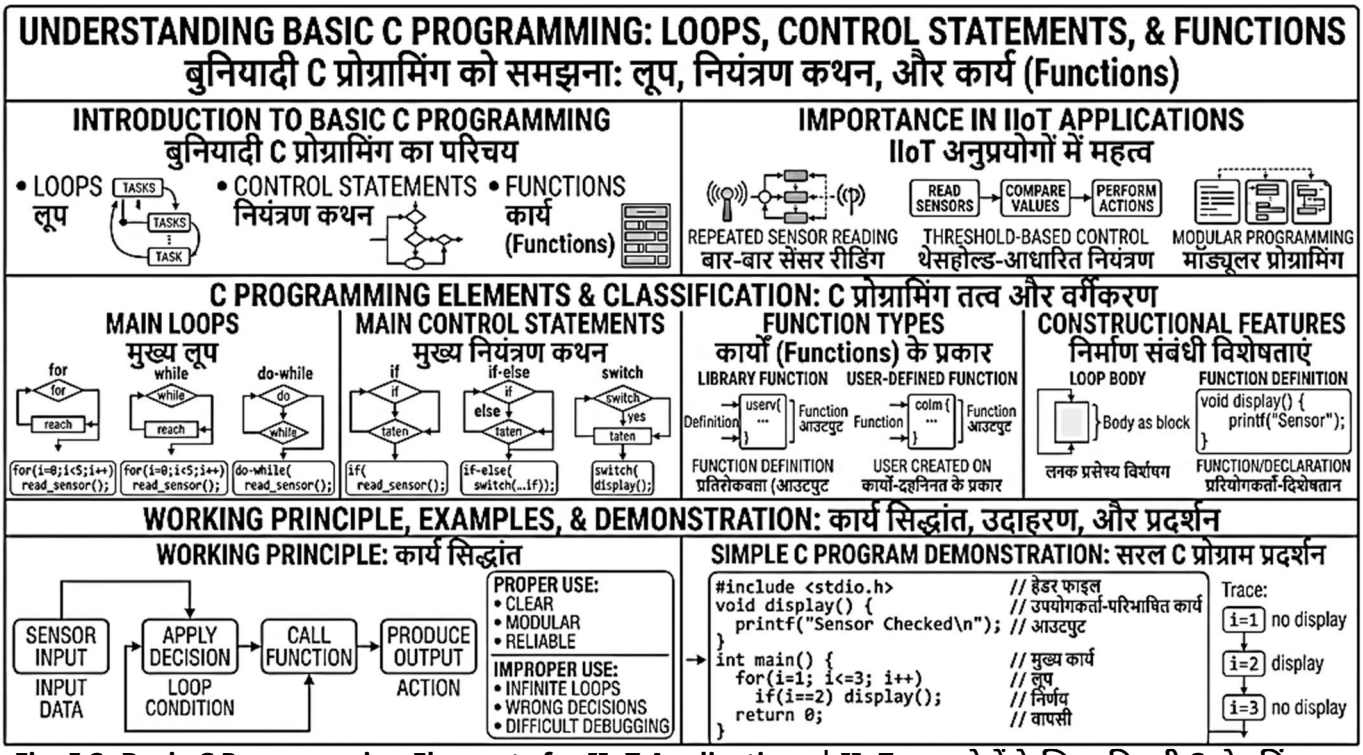


Fig. 5.3: Basic C Programming Elements for IIoT Applications | IIoT अनुप्रयोगों के लिए बुनियादी C प्रोग्रामिंग तत्व

Definition (Fig. 5.3)

In the first-year IIoT Technician curriculum, this topic is included under basic C programming as "use of loops, control statements and types of function in programming." In C, loops repeat a task, control statements make decisions or alter flow, and functions organize a program into small reusable blocks.

Importance in Embedded and IIoT Applications

These programming elements are important because IIoT devices often read sensors repeatedly, compare values with conditions, and perform output actions in a structured manner. Proper control logic makes programs simpler, faster, and easier to test in embedded applications.

Classification

The main loops are **for, while, and do-while**. The main control statements are **if, if-else, switch, break, and continue**. Functions are of two types: **library functions** and **user-defined functions**. In C, while checks the condition before repetition, do-while checks it after one execution, and for uses start, test, and advance expressions together. break exits the nearest loop or switch, while continue jumps to the next loop cycle.

Constructional Features

A loop body is usually written as a block. Conditional statements use expressions inside parentheses. A function definition has return type, function name,

परिभाषा (Fig. 5.3)

प्रथम वर्ष के IIoT तकनीशियन पाठ्यक्रम में यह विषय मूल C प्रोग्रामिंग के अंतर्गत "प्रोग्रामिंग में लूप्स, नियंत्रण कथनों और फंक्शनों के प्रकारों का उपयोग" के रूप में सम्मिलित है। C में, लूप्स किसी कार्य को दोहराते हैं, नियंत्रण कथन निर्णय लेते हैं या प्रवाह को परिवर्तित करते हैं, और फंक्शन्स किसी प्रोग्राम को छोटे पुनः प्रयोज्य खंडों में व्यवस्थित करते हैं।

एम्बेडेड तथा IIoT अनुप्रयोगों में महत्व

ये प्रोग्रामिंग तत्व महत्वपूर्ण हैं क्योंकि IIoT युक्तियाँ प्रायः सेंसरों को बार-बार पढ़ती हैं, मानों की शर्तों के साथ तुलना करती हैं, और एक संरचित तरीके से आउटपुट क्रियाएँ निष्पादित करती हैं। उचित नियंत्रण तर्क प्रोग्रामों को एम्बेडेड अनुप्रयोगों में अधिक सरल, तेज, और परीक्षण के लिए आसान बनाता है।

वर्गीकरण

मुख्य लूप for, while, और do-while होते हैं। मुख्य नियंत्रण स्टेटमेंट if, if-else, switch, break, और continue होते हैं। फंक्शन दो प्रकार के होते हैं: लाइब्रेरी फंक्शन और यूज़र-डिफाइंड फंक्शन। C में, while पुनरावृत्ति से पहले कंडीशन की जाँच करता है, do-while एक बार निष्पादन के बाद कंडीशन की जाँच करता है, और for प्रारंभ, परीक्षण, और अग्रिम अभिव्यक्तियों का एक साथ उपयोग करता है। break निकटतम लूप या switch से बाहर निकलता है, जबकि continue अगले लूप चक्र पर जाता है।

संरचनात्मक विशेषताएँ

एक लूप बॉडी सामान्यतः एक ब्लॉक के रूप में लिखी जाती है। कंडीशनल स्टेटमेंट्स में कोष्ठकों के अंदर अभिव्यक्तियों का उपयोग किया जाता है। एक फंक्शन डेफिनिशन में रिटर्न टाइप, फंक्शन का

parameter list, and body, while a declaration ends with a semicolon.

Working Principle

First, take input or sensor data. Then apply a decision statement or loop condition. Call the required function to process the task and produce the output. Proper use gives clear, modular, and reliable programs, while improper use may cause infinite loops, wrong decisions, and difficult debugging. These are used in repeated sensor reading, threshold-based control, and modular IIoT programs.

for(i=0;i<5;i++) read_sensor(); if(temp>limit) alarm();

Example 1: for Loop

Print numbers from 1 to 5.

```
#include <stdio.h>
int main()
{
   int i;
   for(i=1;i<=5;i++)
      printf("%d ",i);
   return 0;
}
```

Output: 1 2 3 4 5

Example 2: while Loop

Read a sensor value five times.

```
#include <stdio.h>
int main()
{
   int count=1;
   while(count<=5)
   {
      printf("Reading Sensor %d\n",count);
      count++;
   }
   return 0;
}
```

Example 3: do-while Loop

Display a message at least once.

```
#include <stdio.h>
int main()
{
   int i=1;
   do
   {
      printf("Checking Device\n");
      i++;
   }while(i<=3);

   return 0;
}
```

Example 4: if Statement

```
#include <stdio.h>
int main()
{
```

नाम, पैरामीटर सूची, और बॉडी होती है, जबकि एक डिक्लेरेशन सेमीकोलन पर समाप्त होती है।

कार्य सिद्धांत

सबसे पहले, इनपुट या सेंसर डेटा लें। फिर निर्णय कथन या लूप स्थिति लागू करें। कार्य को संसाधित करने और आउटपुट उत्पन्न करने के लिए आवश्यक फ़ंक्शन को कॉल करें। उचित उपयोग से स्पष्ट, मॉड्यूलर और विश्वसनीय प्रोग्राम प्राप्त होते हैं, जबकि अनुचित उपयोग से अनंत लूप, गलत निर्णय और कठिन डीबगिंग हो सकती है। इनका उपयोग बार-बार सेंसर रीडिंग, थ्रेशोल्ड-आधारित नियंत्रण और मॉड्यूलर IIoT प्रोग्रामों में किया जाता है।

for(i=0;i<5;i++) read_sensor(); if(temp>limit) alarm();

उदाहरण 1: for लूप

1 से 5 तक की संख्याएँ प्रदर्शित करें।

```
#include <stdio.h>
int main()
{
   int i;
   for(i=1;i<=5;i++)
      printf("%d ",i);
   return 0;
}
```

आउटपुट: 1 2 3 4 5

उदाहरण 2: while लूप

सेंसर के मान का पाँच बार पठन करें।

```
#include <stdio.h>
int main()
{
   int count=1;
   while(count<=5)
   {
      printf("Reading Sensor %d\n",count);
      count++;
   }
   return 0;
}
```

उदाहरण 3: do-while लूप

संदेश को कम-से-कम एक बार प्रदर्शित करें।

```
#include <stdio.h>
int main()
{
   int i=1;
   do
   {
      printf("Checking Device\n");
      i++;
   }while(i<=3);

   return 0;
}
```

उदाहरण 4: if स्टेटमेंट

```
#include <stdio.h>
int main()
{
```

```
   int temperature=40;

   if(temperature>35)
      printf("High Temperature");

   return 0;
}
```

Example 5: if-else Statement

```
#include <stdio.h>
int main()
{
   int sensor=0;

   if(sensor==1)
      printf("Object Detected");
   else
      printf("No Object");

   return 0;
}
```

Example 6: switch Statement

```
#include <stdio.h>
int main()
{
   int choice=2;

   switch(choice)
   {
      case 1: printf("LED ON");
            break;
      case 2: printf("Motor ON");
            break;
      default: printf("Invalid Choice");
   }

   return 0;
}
```

Example 7: break Statement

```
#include <stdio.h>
int main()
{
   int i;

   for(i=1;i<=5;i++)
   {
      if(i==3)
         break;
      printf("%d ",i);
   }

   return 0;
}
```

Output: 1 2

Example 8: continue Statement

```
#include <stdio.h>
```

```
   int temperature=40;

   if(temperature>35)
      printf("High Temperature");

   return 0;
}
```

उदाहरण 5: if-else स्टेटमेंट

```
#include <stdio.h>
int main()
{
   int sensor=0;

   if(sensor==1)
      printf("Object Detected");
   else
      printf("No Object");

   return 0;
}
```

उदाहरण 6: switch स्टेटमेंट

```
#include <stdio.h>
int main()
{
   int choice=2;

   switch(choice)
   {
      case 1: printf("LED ON");
            break;
      case 2: printf("Motor ON");
            break;
      default: printf("Invalid Choice");
   }

   return 0;
}
```

उदाहरण 7: break स्टेटमेंट

```
#include <stdio.h>
int main()
{
   int i;

   for(i=1;i<=5;i++)
   {
      if(i==3)
         break;
      printf("%d ",i);
   }

   return 0;
}
```

आउटपुट: 1 2

उदाहरण 8: continue स्टेटमेंट

```
#include <stdio.h>
```

```
int main()
{
   int i;

   for(i=1;i<=5;i++)
   {
      if(i==3)
         continue;
      printf("%d ",i);
   }

   return 0;
}
```

Output: 1 2 4 5

Example 9: Function without Arguments

```
#include <stdio.h>

void welcome()
{
   printf("Welcome to IIoT Programming\n");
}

int main()
{
   welcome();
   return 0;
}
```

Example 10: Function with Arguments

```
#include <stdio.h>

void displayValue(int value)
{
   printf("Sensor Value = %d",value);
}

int main()
{
   displayValue(25);
   return 0;
}
```

Example 11: Function Returning a Value

```
#include <stdio.h>

int square(int n)
{
   return n*n;
}

int main()
{
   int result;
   result=square(6);
   printf("%d",result);

   return 0;
```

```
int main()
{
   int i;

   for(i=1;i<=5;i++)
   {
      if(i==3)
         continue;
      printf("%d ",i);
   }

   return 0;
}
```

आउटपुट: 1 2 4 5

उदाहरण 9: बिना आर्ग्युमेंट वाला फ़ंक्शन

```
#include <stdio.h>

void welcome()
{
   printf("Welcome to IIoT Programming\n");
}

int main()
{
   welcome();
   return 0;
}
```

उदाहरण 10: आर्ग्युमेंट्स (Arguments) वाले फ़ंक्शन

```
#include <stdio.h>

void displayValue(int value)
{
   printf("Sensor Value = %d",value);
}

int main()
{
   displayValue(25);
   return 0;
}
```

उदाहरण 11: मान लौटाने वाला फ़ंक्शन

```
#include <stdio.h>

int square(int n)
{
   return n*n;
}

int main()
{
   int result;
   result=square(6);
   printf("%d",result);

   return 0;
```

```
}
```

Output: 36

Example 12: Simple IIoT Program using Loop, Decision Statement, and Function

```
#include <stdio.h>

void display()
{
   printf("Sensor Checked\n");
}

int main()
{
   int i;

   for(i=1;i<=3;i++)
   {
      if(i==2)
         display();
   }

   return 0;
}
```

Output: Sensor Checked

```
}
```

आउटपुट: 36

उदाहरण 12: लूप, निर्णय कथन एवं फ़ंक्शन का उपयोग करके सरल IIoT प्रोग्राम

```
#include <stdio.h>

void display()
{
   printf("Sensor Checked\n");
}

int main()
{
   int i;

   for(i=1;i<=3;i++)
   {
      if(i==2)
         display();
   }

   return 0;
}
```

आउटपुट: सेंसर सत्यापित किया गया।

5.4 Concept of Embedded System and Open-Source Software Tools|एम्बेडेड सिस्टम तथा ओपन-सोर्स सॉफ्टवेयर टूल्स की संकल्पना

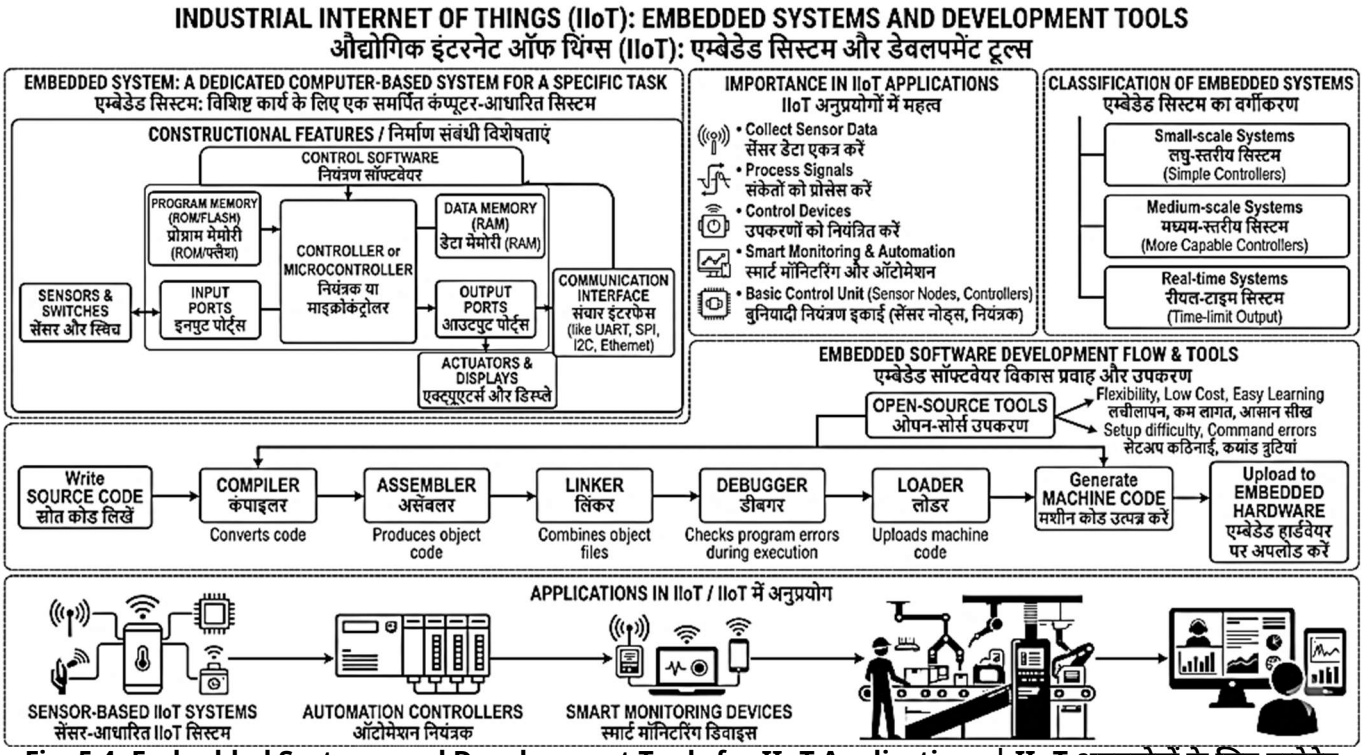


Fig. 5.4: Embedded Systems and Development Tools for IIoT Applications | IIoT अनुप्रयोगों के लिए एम्बेडेड सिस्टम और विकास उपकरण

Definition (Fig. 5.4)
In the IIoT Technician first-year curriculum, this topic is included as the concept of embedded systems and open-source software tools such as debugger, compiler, assembler, and interpreter. An embedded system is a dedicated computer-based system built for a specific task using hardware, memory, input/output interface, and control software.

Importance in IIoT Applications
Embedded systems are important in IIoT because they collect sensor data, process signals, control devices, and support smart monitoring and automation in real time. They form the basic control unit in sensor nodes, automation controllers, and smart industrial devices.

Classification
Embedded systems may be small-scale, medium-scale, or real-time embedded systems. Small-scale systems use simple controllers, medium-scale systems use more capable controllers, and real-time systems must produce output within a defined time limit. Open-source development tools include **compiler, assembler, interpreter, linker, loader, and debugger**. The compiler converts source code, the assembler produces object code, the linker combines object files, and the debugger checks program errors during execution.

Constructional Features
The main constructional features of an embedded system are controller or microcontroller, program memory, RAM, input/output ports, communication interface, and embedded software. These parts work together to sense input, process data, and control output devices.

Working Principle
The program development flow is: write source code, compile it, assemble and link it, debug errors, generate machine code, and upload it to the embedded hardware. Proper use of open-source tools gives flexibility, low cost, and easy learning, but improper use may cause setup difficulty, command errors, and debugging problems.

Applications
These concepts are used in sensor-based IIoT systems, automation controllers, and smart monitoring devices. **Simple workflow:** Source Code → Compiler / Assembler → Debugger → Machine Code → Embedded Hardware.

परिभाषा (Fig. 5.4)
IIoT तकनीशियन प्रथम वर्ष के पाठ्यक्रम में यह विषय एम्बेडेड सिस्टम तथा ओपन-सोर्स सॉफ्टवेयर टूल्स जैसे डिबगर, कम्पाइलर, असेंबलर, और इंटरप्रेटर की संकल्पना के रूप में सम्मिलित है। एम्बेडेड सिस्टम एक समर्पित कंप्यूटर-आधारित प्रणाली है, जिसे हार्डवेयर, मेमोरी, इनपुट/आउटपुट इंटरफेस, और नियंत्रण सॉफ्टवेयर का उपयोग करके किसी विशिष्ट कार्य के लिए निर्मित किया जाता है।

IIoT अनुप्रयोगों में महत्व
एम्बेडेड सिस्टम IIoT में महत्वपूर्ण हैं क्योंकि वे सेंसर डेटा एकत्र करते हैं, सिग्नलों का प्रसंस्करण करते हैं, उपकरणों को नियंत्रित करते हैं, तथा रीयल-टाइम में स्मार्ट मॉनिटरिंग और स्वचालन का समर्थन करते हैं। वे सेंसर नोड्स, ऑटोमेशन कंट्रोलर्स, और स्मार्ट औद्योगिक उपकरणों में मूल नियंत्रण इकाई का निर्माण करते हैं।

वर्गीकरण
एम्बेडेड सिस्टम छोटे-स्तर, मध्यम-स्तर, या रियल-टाइम एम्बेडेड सिस्टम हो सकते हैं। छोटे-स्तर के सिस्टम सरल नियंत्रकों का उपयोग करते हैं, मध्यम-स्तर के सिस्टम अधिक सक्षम नियंत्रकों का उपयोग करते हैं, और रियल-टाइम सिस्टम को निर्धारित समय सीमा के भीतर आउटपुट देना आवश्यक होता है। ओपन-सोर्स डेवलपमेंट टूल्स में कंपाइलर, असेम्बलर, इंटरप्रेटर, लिंकर, लोडर, और डिबगर शामिल होते हैं। कंपाइलर स्रोत कोड को परिवर्तित करता है, असेम्बलर ऑब्जेक्ट कोड तैयार करता है, लिंकर ऑब्जेक्ट फाइलों को संयोजित करता है, और डिबगर निष्पादन के दौरान प्रोग्राम त्रुटियों की जांच करता है।

संरचनात्मक विशेषताएँ
एक एम्बेडेड सिस्टम की मुख्य संरचनात्मक विशेषताएँ कंट्रोलर या माइक्रोकंट्रोलर, प्रोग्राम मेमोरी, RAM, इनपुट/आउटपुट पोर्ट, कम्युनिकेशन इंटरफेस, और एम्बेडेड सॉफ्टवेयर हैं। ये भाग मिलकर इनपुट को सेंस करते हैं, डेटा को प्रोसेस करते हैं, और आउटपुट डिवाइस को नियंत्रित करते हैं।

कार्य सिद्धांत
प्रोग्राम विकास प्रवाह इस प्रकार है: स्रोत कोड लिखें, उसे संकलित करें, उसे असेम्बल और लिंक करें, त्रुटियों का डिबग करें, मशीन कोड उत्पन्न करें, और उसे एम्बेडेड हार्डवेयर में अपलोड करें। ओपन-सोर्स उपकरणों का उचित उपयोग लचीलापन, कम लागत, और आसान अधिगम प्रदान करता है, लेकिन अनुचित उपयोग से सेटअप में कठिनाई, कमांड त्रुटियाँ, और डिबगिंग समस्याएँ हो सकती हैं।

अनुप्रयोग
ये अवधारणाएँ सेंसर-आधारित IIoT प्रणालियों, स्वचालन नियंत्रकों, तथा स्मार्ट मॉनिटरिंग उपकरणों में उपयोग की जाती हैं। सरल **कार्यप्रवाह:** स्रोत कोड → कम्पाइलर / असेम्बलर → डिबगर → मशीन कोड → एम्बेडेड हार्डवेयर।

MCQ's | बहुविकल्पीय प्रश्न

Q1. What is the main purpose of a compiler in C programming? / सी प्रोग्रामिंग में एक कंपाइलर का मुख्य उद्देश्य क्या है?
(a) Writing the program / प्रोग्राम लिखना
(b) Editing the program / प्रोग्राम को संपादित करना
(c) Converting code into machine language / कोड को मशीन भाषा में बदलना
(d) Debugging the program / प्रोग्राम को डिबग करना
Ans. c | Sol. : Compiler translates C code into machine language for execution by the computer. / कंपाइलर सी कोड को मशीन भाषा में अनुवादित करता है ताकि कंप्यूटर उसे निष्पादित कर सके।

Q2. Which symbol is used to indicate the end of a statement in C? / सी में कथन के अंत को दर्शाने के लिए किस प्रतीक का उपयोग किया जाता है?
(a) : / कोलन
(b) ; / सेमिकोलन
(c) . / पूर्ण विराम
(d) , / कॉमा
Ans. b | Sol. : Every statement in C must end with a semicolon (;). / सी में हर कथन को सेमिकोलन (;) के साथ समाप्त करना आवश्यक है।

Q3. What is a variable in C language? / सी भाषा में वेरिएबल क्या है?
(a) A fixed number / एक स्थिर संख्या
(b) A memory location to store data / डेटा संग्रहीत करने के लिए एक मेमोरी स्थान
(c) A special symbol / एक विशेष प्रतीक
(d) A hardware device / एक हार्डवेयर डिवाइस
Ans. b | Sol. : A variable stores data that can change during program execution. / वेरिएबल डेटा को संग्रहीत करता है जो प्रोग्राम निष्पादन के दौरान बदल सकता है।

Q4. In which loop does the condition check happen after executing the loop body once? / किस लूप में लूप बॉडी को एक बार निष्पादित करने के बाद शर्त की जांच होती है?
(a) For loop / फॉर लूप
(b) While loop / व्हाइल लूप
(c) Do-while loop / डू-व्हाइल लूप
(d) Infinite loop / अनंत लूप
Ans. c | Sol. : Do-while loop executes at least once before checking the condition. / डू-व्हाइल लूप शर्त की जांच करने से पहले कम से कम एक बार निष्पादित होता है।

Q5. How does a compiler help in C programming? / सी प्रोग्रामिंग में एक कंपाइलर किस प्रकार मदद करता है?
(a) It writes code automatically / यह स्वतः कोड लिखता है
(b) It checks for logical mistakes only / यह केवल तर्क त्रुटियों की जांच करता है
(c) It translates code into machine language / यह कोड को मशीन भाषा में अनुवादित करता है
(d) It stores the program / यह प्रोग्राम को संग्रहीत करता है
Ans. c | Sol. : Compiler converts high-level C code into machine-readable binary form for execution. / कंपाइलर उच्च स्तरीय सी कोड को मशीन-पठनीय बाइनरी रूप में अनुवादित करता है ताकि उसे निष्पादित किया जा सके।

Q6. What role does a variable play in a C program? / सी प्रोग्राम में वेरिएबल क्या भूमिका निभाता है?
(a) It stores program name / यह प्रोग्राम का नाम संग्रहीत करता है
(b) It stores data values / यह डेटा मान संग्रहीत करता है
(c) It controls loops / यह लूप नियंत्रित करता है
(d) It compiles the code / यह कोड को कंपाइल करता है
Ans. b | Sol. : Variables temporarily hold data during the execution of a program. / वेरिएबल प्रोग्राम के निष्पादन के दौरान अस्थायी रूप से डेटा संग्रहीत करते हैं।

Q7. How do logical operators assist in decision-making? / निर्णय लेने में लॉजिकल ऑपरेटर किस प्रकार सहायता करते हैं?
(a) By combining multiple conditions / कई शर्तों को जोड़कर
(b) By compiling faster / तेजी से कंपाइलिंग करके
(c) By storing results / परिणामों को संग्रहीत करके
(d) By deleting data / डेटा हटाकर
Ans. a | Sol. : Logical operators help evaluate combined conditions as true or false. / लॉजिकल ऑपरेटर कई शर्तों का मूल्यांकन सही या गलत के रूप में करने में मदद करते हैं।

Q8. While testing an embedded circuit in an IIoT setup as shown in the image, why is ESD grounding (wrist strap and mat) necessary? / चित्र में दर्शाए गए IIoT सिस्टम परीक्षण सेटअप में एम्बेडेड सर्किट की जाँच करते समय ESD ग्राउंडिंग (रिस्ट स्ट्रैप और मैट) क्यों आवश्यक है?

(a) To increase circuit voltage / परिपथ का वोल्टेज बढ़ाने के लिए
(b) To protect components from static electricity damage / स्थैतिक विद्युत से घटकों को नुकसान से बचाने के लिए
(c) To reduce power consumption / ऊर्जा खपत कम करने के लिए
(d) To speed up data processing / डाटा प्रोसेसिंग तेज करने के लिए
Ans. b | Sol. : ESD (Electrostatic Discharge) grounding prevents static charges from damaging sensitive electronic components during testing. Wrist straps and mats safely discharge static electricity. / ESD (इलेक्ट्रोस्टैटिक डिस्चार्ज) ग्राउंडिंग संवेदनशील इलेक्ट्रॉनिक घटकों को स्थैतिक आवेश से होने वाले नुकसान से बचाती है। रिस्ट स्ट्रैप और मैट स्थैतिक विद्युत को सुरक्षित रूप से पृथ्वी में प्रवाहित कर देते हैं।

Q9. Which C language keyword is used to declare an integer variable? / C भाषा में पूर्णांक वेरिएबल घोषित करने के लिए किस कुंजीशब्द का उपयोग किया जाता है?
(a) int / इंट
(b) float / फ्लोट
(c) char / चार

(d) double / डबल

Ans. a | Sol. : The keyword int is used to declare integer variables in C programming. / C प्रोग्रामिंग में पूर्णांक वेरिएबल घोषित करने के लिए int कुंजीशब्द का उपयोग किया जाता है।

Q10. Which of the following is a correct identifier in C? / निम्न में से कौन सा C में एक सही आइडेंटिफायर है?

(a) 2value / 2वैल्यू

(b) _value2 / _वैल्यू2

(c) value-2 / वैल्यू-2

(d) value 2 / वैल्यू 2

Ans. b | Sol. : Identifiers can start with an underscore or a letter but not a number or space. / आइडेंटिफायर एक अंडरस्कोर या अक्षर से शुरू हो सकता है, लेकिन संख्या या स्पेस से नहीं।

Q11. Which loop is guaranteed to execute at least once in C? / C में कौन सा लूप कम से कम एक बार निष्पादित होना सुनिश्चित होता है?

(a) for loop / फॉर लूप

(b) while loop / वाइल लूप

(c) do-while loop / डू-वाइल लूप

(d) switch case / स्विच केस

Ans. c | Sol. : The do-while loop executes the body first and checks the condition later. / do-while लूप पहले बॉडी को निष्पादित करता है और बाद में शर्त की जांच करता है।

Q12. Which editor is commonly used for writing C programs? / C प्रोग्राम लिखने के लिए सामान्यतः कौन सा संपादक उपयोग किया जाता है?

(a) MS Word / एमएस वर्ड

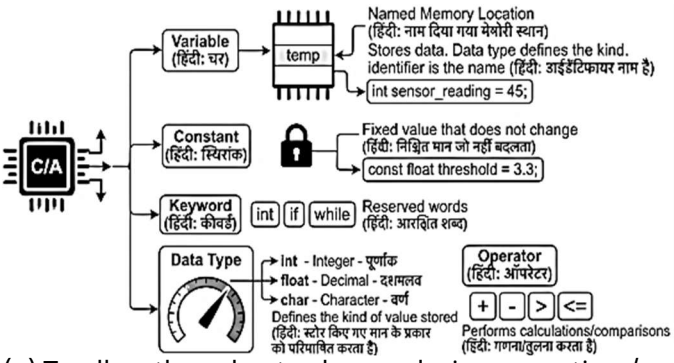
(b) Turbo C Editor / टर्बो सी संपादक

(c) Paint / पेंट

(d) Excel / एक्सेल

Ans. b | Sol. : Turbo C is one of the editors/IDEs commonly used for learning and writing C programs, especially in educational environments. However, many other editors and IDEs are also used today. / टर्बो C C प्रोग्राम लिखने और सीखने के लिए उपयोग किए जाने वाले सामान्य एडिटर/IDE में से एक है, विशेषकर शैक्षणिक वातावरण में। हालांकि आज कई अन्य एडिटर और IDE भी उपयोग किए जाते हैं।

Q13. In an embedded system program, a developer writes const float threshold = 3.3; as shown in the image. What is the purpose of using const here? / एम्बेडेड सिस्टम प्रोग्राम में डेवलपर const float threshold = 3.3; लिखता है जैसा कि चित्र में दिखाया गया है। यहाँ const का उपयोग करने का उद्देश्य क्या है?



(a) To allow the value to change during execution / निष्पादन के दौरान मान को बदलने की अनुमति देना

(b) To store multiple values in the variable / एक ही चर में अनेक मान संग्रहित करना

(c) To keep the value fixed and prevent modification / मान को स्थिर रखना और परिवर्तन से बचाना

(d) To increase processing speed automatically / प्रोसेसिंग गति को स्वतः बढ़ाना

Ans. c | Sol. : The keyword const is used to declare a constant variable whose value cannot be changed during program execution. It ensures data safety and prevents accidental modification. / const कीवर्ड का उपयोग स्थिर (constant) चर घोषित करने के लिए किया जाता है, जिसका मान प्रोग्राम के निष्पादन के दौरान बदला नहीं जा सकता। यह डेटा की सुरक्षा सुनिश्चित करता है और अनजाने परिवर्तन से बचाता है।

Q14. Which analysis correctly explains why for loops are preferred for known iterations? / किस विश्लेषण से स्पष्ट होता है कि ज्ञात पुनरावृत्तियों के लिए for लूप पसंद किए जाते हैं?

(a) They are faster / वे तेज़ हैं

(b) They are easier to write and understand / लिखना और समझना आसान है

(c) They consume less memory / कम मेमोरी का उपयोग करते हैं

(d) They support recursion / पुनरावृत्ति का समर्थन करते हैं

Ans. b | Sol. : for loops clearly show initialization, condition, and increment in one line, making iteration management simple. / for लूप एक ही पंक्ति में इनिशियलाइज़ेशन, शर्त और वृद्धि को दर्शाते हैं जिससे पुनरावृत्ति प्रबंधन सरल हो जाता है।

Q15. Which analysis best describes the importance of switch-case in complex C programs? / जटिल C प्रोग्रामों में स्विच-केस के महत्व का सबसे अच्छा विश्लेषण कौन सा है?

(a) Makes code longer / कोड को लंबा बनाता है

(b) Simplifies multiple condition handling / कई शर्तों के प्रबंधन को सरल बनाता है

(c) Reduces memory size / मेमोरी आकार को कम करता है

(d) Speeds up variable declaration / वेरिएबल घोषणा को तेज करता है

Ans. b | Sol. : switch cases organize multiple choices in a cleaner and readable format compared to multiple if-else statements. / switch केस कई विकल्पों को अधिक स्वच्छ और पठनीय प्रारूप में व्यवस्थित करते हैं, जो कई if-else कथनों की तुलना में बेहतर होता है।

Q16. Analyze why embedded systems require highly optimized C code. / विश्लेषण करें कि एंबेडेड सिस्टम को अत्यधिक अनुकूलित C कोड की आवश्यकता क्यों होती है।

(a) For stylish code / स्टाइलिश कोड के लिए

(b) For faster debugging / तेज़ डिबगिंग के लिए

(c) Due to limited hardware resources / सीमित हार्डवेयर संसाधनों के कारण

(d) To support multiple users / कई उपयोगकर्ताओं का समर्थन करने के लिए

Ans. c | Sol. : Embedded devices often have limited memory and processing power, requiring efficient code. / एंबेडेड डिवाइसेज़ में अक्सर सीमित मेमोरी और प्रोसेसिंग पावर होती है, जिसके लिए कुशल कोड की आवश्यकता होती है।

Liked this sample? Get the complete book with all modules, MCQs, and practice questions.

How to Purchase This Book

Click the Link or **Scan the QR** code below to get the complete book at a special discount. Order directly from Publication Website: https://teachtoindia.com/product/industrial-internet-of-things-iiot-technician-first-year/

Browse All ITI Trade Books at Special Discounted Prices

View the full collection at: https://teachtoindia.com/iti-books/

Also available on Flipkart, Amazon, and Meesho.

Trusted by ITI Students, Trainees, and Instructors Across India.

For any queries related to our books or institutional purchases, please contact us:

WhatsApp/Mobile: +91 9084496877
Email: teachtoindia1@gmail.com
Website: www.teachtoindia.com