Interoperable data retrieval solutions for smart manufacturing applications

Tadigotla Narendra Reddy[1,*], Prasad Prasannan[2], Seema B. Hegde[3], Prakash Vinod[4], Mervin A. Herbert[5], Shrikantha S. Rao[6]

[1,2,4]*Central Manufacturing Technology Institute, Bangalore, India*
[3]*Siddaganga Institute of Technology, Tumkur, India*
[5,6]*National Institute of Technology Karnataka, Surathkal, India*

KEYWORDS

Data Retrieval,
Open-Source Software,
PLC Data Acquisition,
Industry 4.0,
Open-Source Frameworks,
Industrial Automation,
IIoT.

ABSTRACT

Industry 4.0/Digital Transformation technologies benefit industries by enhancing productivity, quality improvements, ease of maintenance and flexibility in manufacturing operations. One major challenge in digitalisation projects is acquiring data from industrial machinery and equipment due to proprietary industrial protocols. Integrating the "Industrial Internet of Things (IIoT)" has spurred digitalisation in manufacturing, enhancing productivity and reliability while lowering labour costs. IIoT links industrial machinery like PLCs, CNCs, and robots via a robust network, generating crucial data for automation. However, due to cost constraints, acquiring data from legacy machines, which use diverse communication protocols, poses a significant challenge, particularly for MSMEs. This paper explores leveraging open-source frameworks for real-time data acquisition from industrial machines in Industry 4.0. Through several test cases, the paper demonstrates the feasibility and benefits of using open-source solutions over vendor-specific licensed software. Additionally, it validates "Wireshark" as a versatile open-source tool for data acquisition across various vendor-specific machines, utilising standard communication protocols. This research offers a pioneering approach to digitalising MSME manufacturing industries while minimising costs efficiently.

1. Introduction

Industry 4.0 represents a drastic change in manufacturing, introducing a digital revolution with networking, IoT, and data mining technologies. This transition simplifies production processes, improving productivity and cutting costs. The crucial function of industrial data is critical to driving automation and optimisation efforts in this progression.

Programmable Logic Controllers (PLCs), Computer Numeric Controllers (CNCs), and Robots are essential control systems in industrial processes (Langmann & Stiller, 2019; Lins et al., 2017; Goel & Gupta, 2020). These systems produce large quantities of data for monitoring, predictive analysis, and quality control. Historically, industrial OEMs offered proprietary software that allowed data recovery but often resulted in high maintenance expenses.

*Corresponding author E-mail: narendra@cmti.res.in

https://doi.org/10.58368/MTT.22.11-12.2023.1-19

The study recommends using open-source data-gathering software as a cost-efficient option compatible with various communication protocols for PLCs and CNCs. Our goals involve investigating open-source software options, testing their effectiveness on older computers, and recording technical findings for future use in manufacturing.

The following sections explore open-source data acquisition tools for real-time industrial data collection from PLCs, CNCs, and Collaborative robots. We also present validation results, guiding cost-effective and enduring digitalisation.

2. Cross Communication Protocols Used in Industrial Automation

Communication protocols are essential for enabling the transmission of data and compatibility across industrial automation systems. These protocols are standardised rules and conventions that regulate data transmission among various components of an industrial system.

Industrial automation utilises various communication protocols designed to meet specific applications and requirements.

Fieldbus Protocols: Fieldbus protocols facilitate communication within localised industrial networks by operating at the physical layer.

- Modbus: Renowned for simplicity and versatility, Modbus is a serial communication protocol widely used for connecting diverse industrial devices (Dias et al., 2018).
- Profibus: Offering high-speed communication and robustness, Profibus finds typical applications in manufacturing and process automation industries (Aminaie & Aminaie, 2020).
- DeviceNet: Tailored to connect industrial devices in manufacturing processes, DeviceNet stands out for its simplicity and ease of implementation.

Ethernet - based Protocols: Ethernet-based protocols extend communication capabilities over broader network infrastructures.

- Profinet: As a real-time Industrial Ethernet protocol, Profinet ensures fast and reliable communication, making it suitable for modern industrial applications (Dias et al., 2018).
- EtherNet/IP: With features like real-time data exchange and remote monitoring, EtherNet/IP is a prevalent choice in various industrial automation systems (Decotignie, 2005).

Understanding these physical layer protocols is pivotal for constructing efficient and interoperable communication networks in industrial settings. Each protocol caters to industrial needs, fostering seamless data exchange and control.

Industrial IoT (IIoT) protocols: Industrial IoT (IIoT) protocols are specifically designed for communication in Industrial Internet of Things (IIoT) applications, facilitating efficient data exchange and control in interconnected industrial systems. These protocols prioritise scalability, interoperability, and security.

- OPC UA (Unified Architecture) is a prevalent IIoT protocol renowned for its resilience, scalability, and ability to work seamlessly with other systems. It offers a standardised architecture for ensuring secure and dependable communication between industrial devices and software applications. OPC UA enables seamless integration across

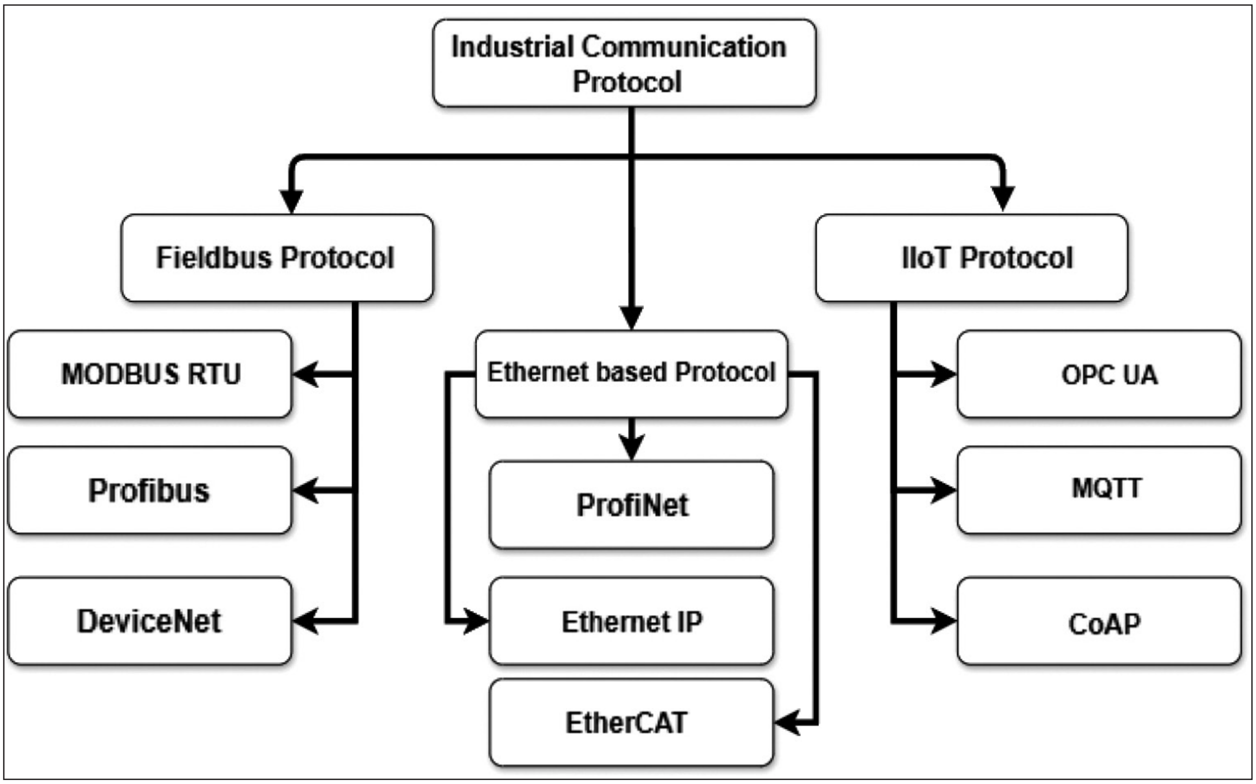


Fig. 1. Different types of industrial communication protocols.

various industrial systems, allowing quick data interchange while assuring data integrity and authentication (Martins et al., 2023).

- MQTT, also known as Message Queuing Telemetry Transport, is a messaging protocol designed for IIoT applications that often face limited bandwidth and high latency limitations. It facilitates effective communication across sensors, devices, and backend systems, enabling the transfer of real-time data and interactions based on events. MQTT's uncomplicated yet adaptable architecture renders it highly suitable for establishing connections between several devices in scattered IIoT scenarios (Hunkeler et al., 2008).
- CoAP, or Constrained Application Protocol, is a lightweight and efficient communication protocol designed for constrained devices in IoT and IIoT applications. It operates over UDP, making it suitable for low-power devices and networks with limited resources such as bandwidth and memory. CoAP enables devices to exchange data and control commands using a RESTful architecture, similar to HTTP, but with optimisations for constrained environments. It supports methods like GET, POST, PUT, and DELETE for resource manipulation and employs UDP's simplicity while offering reliability through confirmable messages and retransmissions. Organisations often use CoAP when devices require efficient communication over constrained networks, such as sensor networks, smart cities, and industrial automation systems.

A thorough grasp of these data connections and physical layer protocols is essential for ensuring dependable data transfer, facilitating compatibility across different systems, and maximising the efficiency of industrial automation systems. These protocols collectively enhance the effectiveness and efficiency of communication networks, hence boosting productivity in industrial activities.

3. Challenges in Data Retrieval from Industrial Automation Devices

Difficulties in extracting data from industrial automation devices are significant obstacles that industrial companies must overcome to utilise their data fully for optimisation and decision-making. The obstacles stem from characteristics in industrial contexts, and the numerous technologies utilised in industrial automation. Here is a detailed explanation of some of the main obstacles:

Industrial automation devices such as PLCs, CNCs, and sensors use various protocols for communication. The protocols can range from outdated proprietary ones to current standardised ones. Overseeing the compatibility and information sharing among devices utilising varying protocols presents a significant obstacle for industrial data retrieval systems.

Legacy Systems: Numerous industrial facilities depend on older technologies with outdated communication protocols or without standardised interfaces. Integrating outdated systems with present data retrieval solutions necessitates thorough planning and typically leads to increased complexity and costs.

OEM Lock-In: Industrial automation solutions are usually offered by particular suppliers, causing organisations to rely on exclusive technologies and solutions. This dependence hinders interoperability with systems from many vendors and constrains the adaptability of data retrieval solutions.

Scalability: It is crucial when industrial activities grow, leading to a substantial rise in data produced by automation devices. Scaling data retrieval systems to handle increasing data volumes while ensuring performance and reliability can be difficult, particularly for firms with complicated and remote industrial infrastructures.

Cost Constraint: Introducing and upholding data retrieval technology in industrial environments might lead to significant costs. This includes the initial capital outlay for hardware, software, and infrastructure and continuous expenses for maintenance, updates, and licensing fees for proprietary software solutions. Smaller firms may find these costs a barrier, which could impede the implementation of advanced data retrieval technology.

A blend of technological advancement, cooperation within the sector, and careful strategic planning is necessary to overcome these obstacles. Open-source solutions, standardised communication protocols like OPC UA and MQTT, and initiatives like Industry 4.0 and Industrial IoT Consortium are essential for overcoming challenges and enabling smooth data retrieval in industrial automation settings. Organisations need to assess their needs, think about future growth and adaptability, and focus on investing in data retrieval solutions that match their strategic goals and financial limitations.

4. Case Study on Real-Time Data Retrieval from Industrial PLCs Using Experimental Analysis

The PLCs are rugged and robust digital computers designed for automation of the industrial process through programming using Ladder, FBD, SFC, or ST method for complex control processes such as turning on the machine or subunits, regulating the machine parameters such as temperature, mixing ratio or regulating the speed of conveyer belt and so on. Industries use it in batch processing operations like controlling chillers and heaters and monitoring energy consumption. It is a vital component of any industrial automation and consists of three main blocks: analogue/digital input, output, and processing unit. The PLCs play a significant role in collecting and storing real-time data, which can be further given for data analysis through a TCP/IP, serial, or PPI communication, as shown in Figure 2.

The PLCs can be connected over a network to communicate with other devices or over IoT for remote monitoring and control through various communication protocols. There are around twelve OEMs globally who design and develop PLCs ranging from simple control to complex automation systems, and prominent vendors among them are Siemens, Mitsubishi, Allan Bradley/ Rockwell Automation, Messung, Delta, ABB, Schneider Electric and so on., These commercial PLCs will be supporting a specific set of communication protocols of the OEMs. The OEMs with proprietary software can acquire data from these protocols.

Users, along with PLCs, should purchase these software platforms, which will usually be costly and uninteroperable, and cross-platform usage will not be possible. Even though OPC UA servers Martins et al. (2023) are available nowadays, with machines that are interoperable with any communication protocol, legacy machines do not have this OPC server. As these machines are costly and in good health, they cannot be replaced, and hence, there is a need for open-source software that can be used with such machines without maintenance cost and is capable of being interoperable.

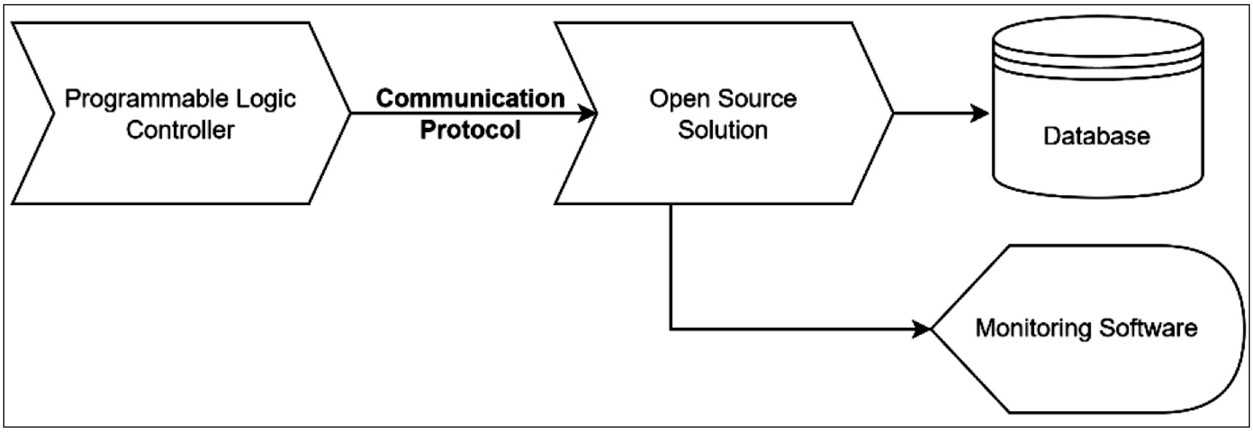


Fig. 2. Industrial data retrieval setup.

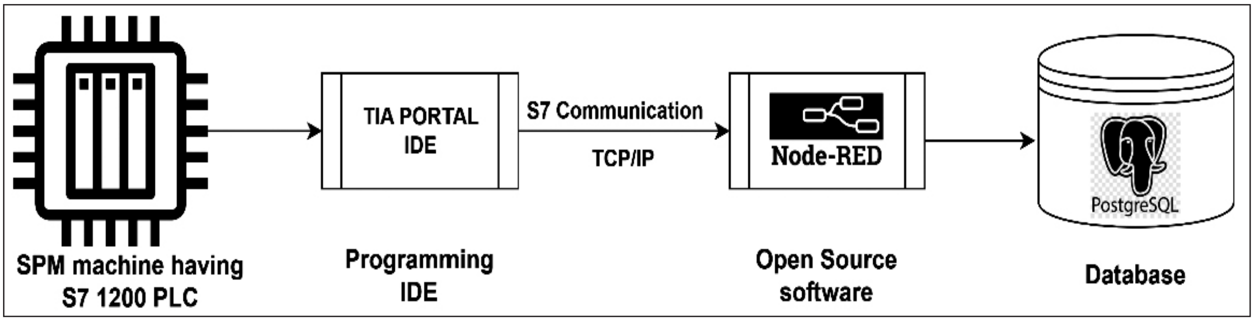


Fig. 3. S7 1200 PLC validation setup with node - red.

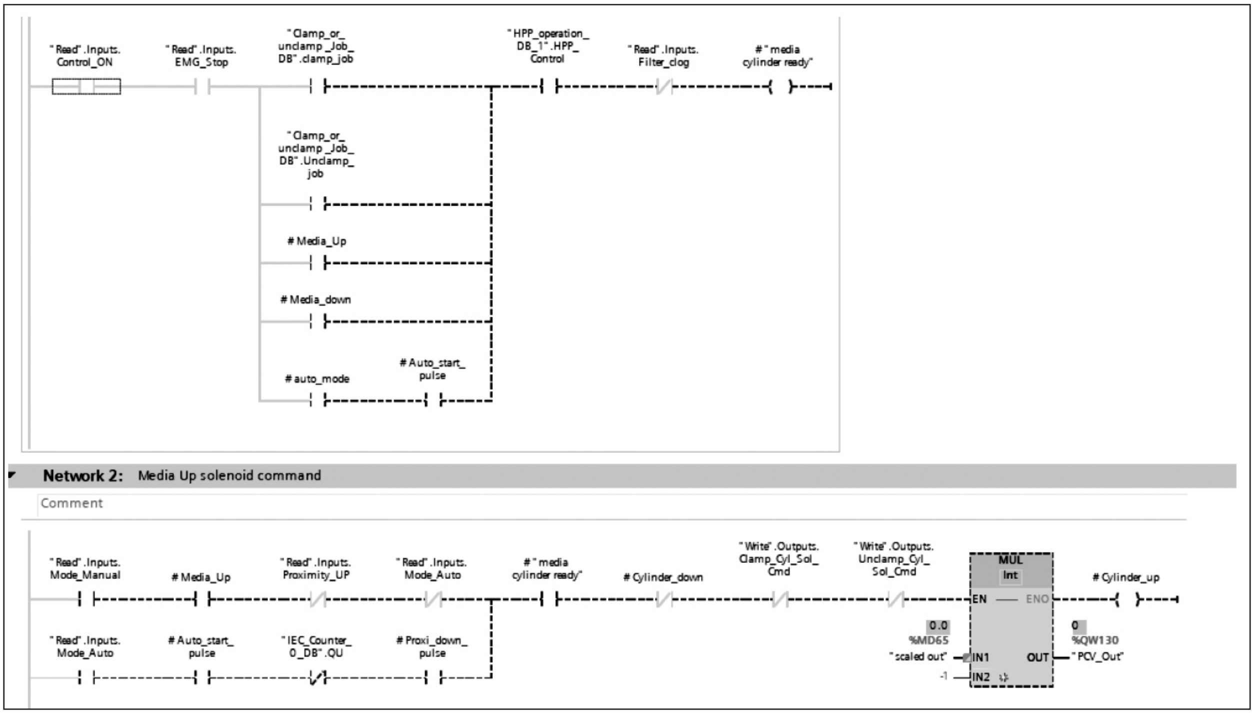


Fig. 4. Ladder programming for micro AFFM machine.

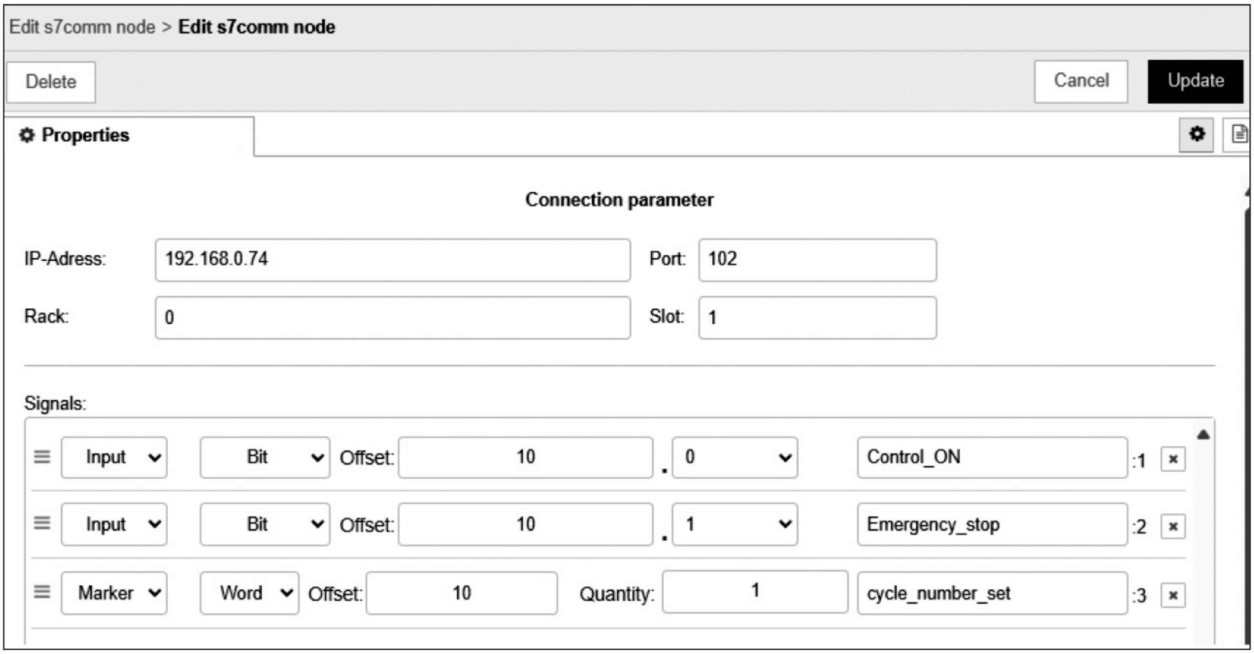


Fig. 5. S7 communication connection parameters in node-red.

4.1. Siemens PLC testing and validation (S7 1200 without OPC UA)

It is an advanced compact PC-based programmable logic controller from Siemens whose hardware configuration and programming can easily be done using a Totally Integrated Automation (TIA) portal. The TIA software platform writes a program over Siemens PLC and communicates, diagnoses, and collects data over one single interface.

- **Data retrieval using node-red open-source software**

The test bed setup for validation of Open-source Node-Red Open-source tool is shown in Fig 3.

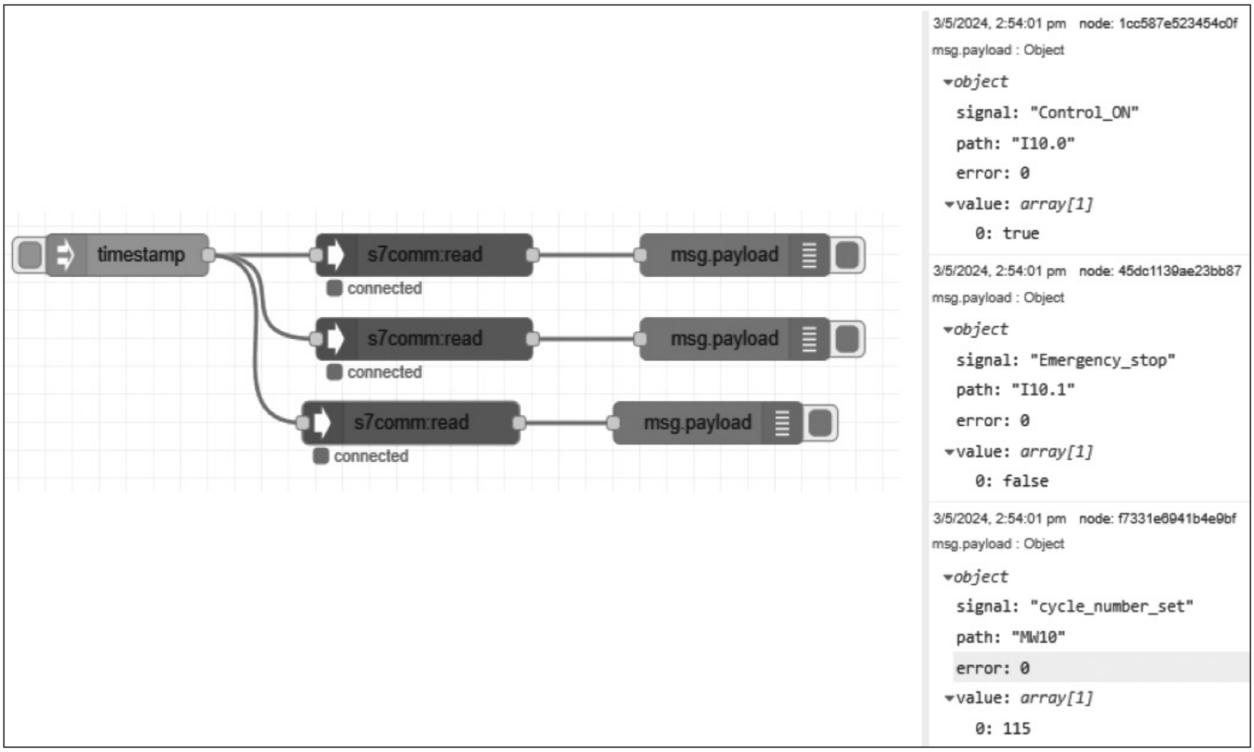


Fig. 6. Flow diagram of the node-red for S7 or TCP/IP communication.

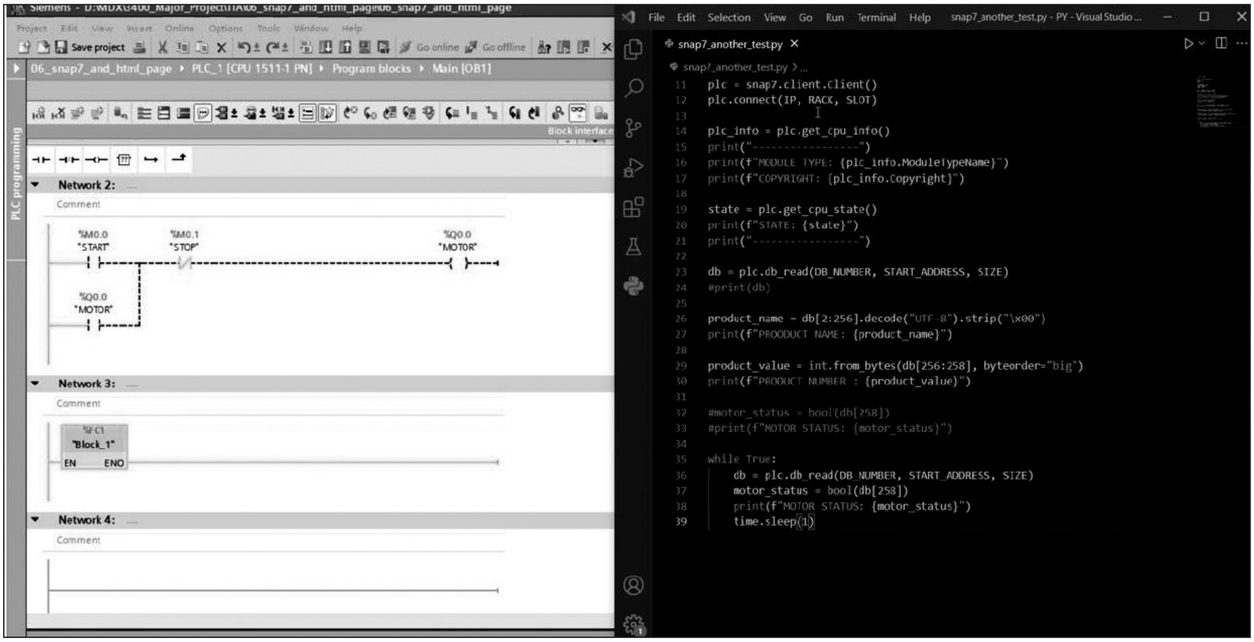


Fig. 7. The data block retrieval using the snap 7 library.

In this setup, the PLC is programmed using ladder programming to get the various status data such as control switch, chiller, emergence stop, and so on from the Special Purpose machine through the TIA portal after the appropriate device configurations are done, as shown in Figure 4.

The obtained data tags are placed in the data block of the TIA portal. They should be obtained to retrieve the data using the S7 / TCP/IP communication from the TIA portal to the external interface remote partner, i.e., either Node Red or Snap 7 and then connect Node-Red data tool by initiating and configure a flow graph for the S7 communication protocol with the data block of the TIA portal with the IP address of the Host PLC as shown in Fig. 5 and Fig 6.

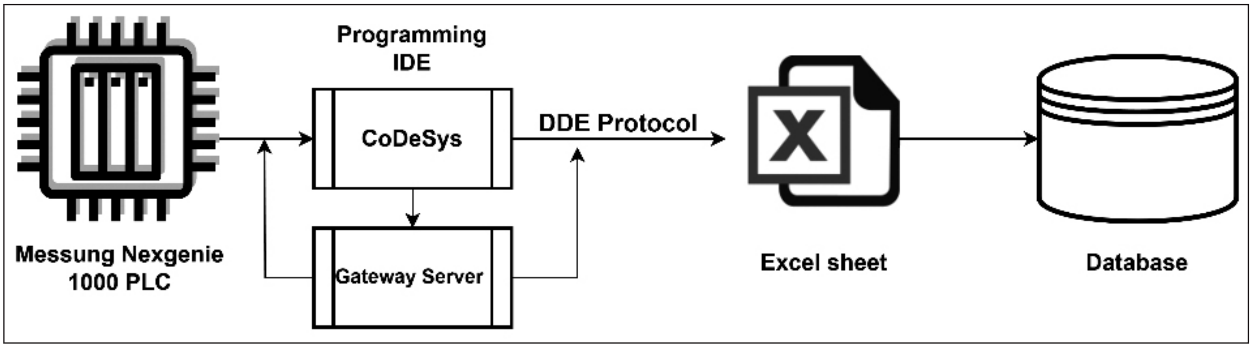


Fig. 8. Messung nexgenie 1000 PLC data retrieval setup.

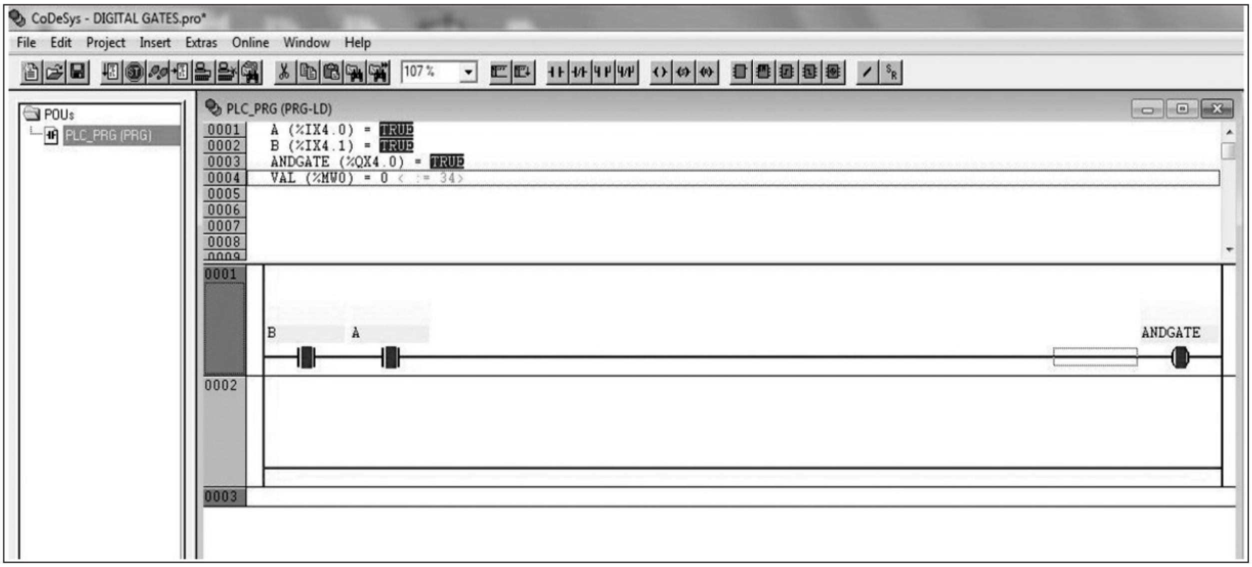


Fig. 9. Controlled data environment system (CoDeSys) IDE.

The output status output can be seen in the console, as shown in Figure 7, and can be transferred to the database for further analysis.

- **Data retrieval using SNAP 7 open-source software**

Another feasible way to retrieve data from the legacy Siemens PLC is using the open-source Python IDE library package Snap 7. Snap 7 is a programmatic data retrieval method in which the program is developed using sockets, as shown in Figure 7.

Using this library, the data blocks can be accessed from the TIA portal using the socket connection of the PLC, as shown in Figure 8. However, the complexity here is knowing the size of each data block, which needs proper wrapping and can be solved using a higher level of programming logic.

All the S7 series Siemens PLC will have S7 communication protocol inbuilt, and hence, this validates that from any S7 series Siemens PLC using the generic Open-source software specifically using Node-Red (simulation method) and snap 7 (Programming Method), the data of the connected machine can be retrieved without using any proprietary software. We have explored the data retrieval in the S7 1200 & S7 1500 series of Siemens PLC.

4.2. Messung PLC testing and validation (Nexgenie 1000)

The Nexgenie 1000 PLC is a compact ultrafast PLC from Mitsubishi Electric India for hassle-free functionalities for various industrial applications with a global standard IEC 61131-3 programming languages support.

The messung PLCs are set up to retrieve data using the open-source programming IDE called Controlled Data Environment System (CODESYS), which contains an integrated visualisation tool by IEC 61131-3 standard using the DDE protocol, as shown in figure 8.

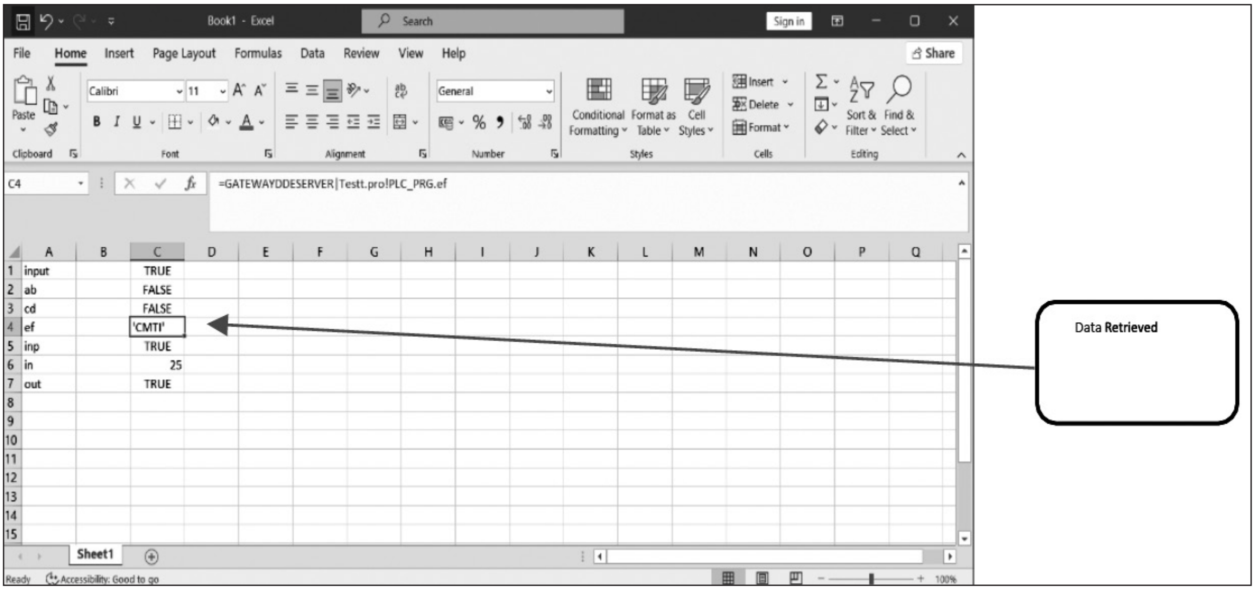


Fig. 10. The real-time data retrieved from the PLC to excel.

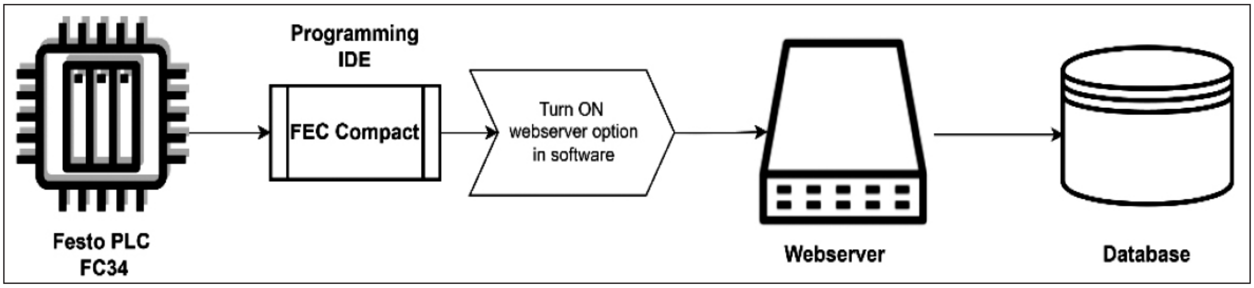


Fig. 11. Festo FC PLC setup for data retrieval.

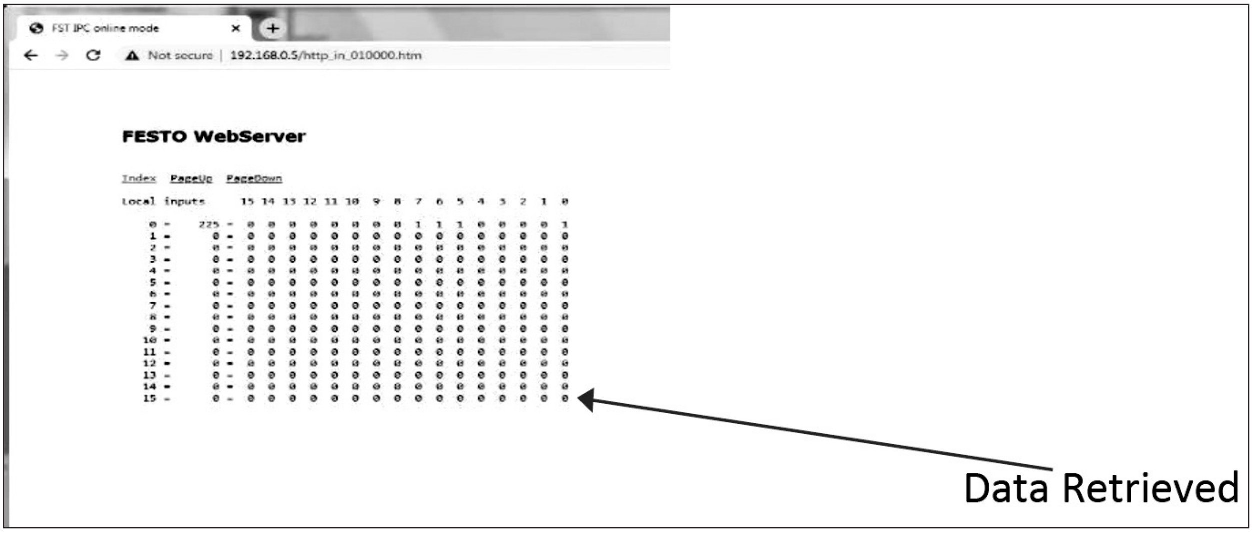


Fig. 12. The data retrieved from the PLC store on the web page.

Programming is written on the CoDeSys IDE to retrieve the data from the PLC ladder. It taps the input and output memory variables with the status of the variable, as shown in Figure 8 when the IDE goes online.

The dynamic data exchange protocol is instantiated using the serial communication interface for accessing the data from the CoDeSys IDE using the DDE gateway, and the real-time data generated at the PLC will be stored in the Excel database, as shown in Figure 9 for further data analysis.

Hence the Messung Nexgenie series PLC uses an open-source IDE CoDeSys and uses DDE protocol without any proprietary software.

4.3. FESTO PLC testing and validation (FC34 PLC)

Festo is a compact, low-cost PLC used in many small to medium-control industrial applications. It can also be programmed using structured text programming, a command-based, user-friendly language with English-like statements.

The PLC is programmed on the FST FEC compact editor, configuring the driver configuration and communication preferences to TCP/IP. The Festo web server is connected to the PLC using its IP address through TCP/IP communication.

Once the web server is connected to the PLC, then, by pinging the IP address of the PLC, the data from the PLC can be retrieved from the PLC and stored in the local pages of the server, as shown in Figure 12.

Hence, from the Festo FC series PLCs, the data can be collected in a web server through TCP/IP communication, which can be retrieved whenever required, anytime, anywhere by connecting to the server.

5. Case Study on Real-Time Data Retrieval from Industrial CNCs Using Experimental Analysis

The CNC is the computer-based system used to control and operate machinery in manufacturing processes. CNC systems are commonly used in machine tools, such as mills, lathes, routers, and grinders. The CNC system receives instructions as a program, typically written in a specific language called G-code and M-code, which contains commands for tool movements, speeds, and other parameters. The CNC system interprets these instructions and precisely controls the movement and operation of the machine tool to produce the desired output. CNC technology is primarily used for tasks that involve cutting, shaping, drilling, or milling materials with high precision and accuracy. Therefore, CNC is focused on controlling the movement and operation of machine tools, while PLC is designed to monitor and control various industrial processes and machinery. CNC is typically used in applications requiring precise positioning and machining.

The different types of CNC machines will be available on the market depending on their specific functionality. However, the data retrieval from the machine will happen through a communication protocol through the interface connected to the communication module. The communication modules will be connected to a serial or Ethernet interface. Modbus, TCP/IP, MT Connect, Profinet, and Profibus are widely used communication protocols.

Table 1
Industrial CNC OEMs and their data retrieval methods with limitations.

Sl No.	CNC OEM	Data Retrieval Method	Limitations
1	FANUC CNC	FOCAS API	Requires advanced programming skills
2	Heidenhain CNC	PyLSV2 package	Limited access to parameters

5.1. Fanuc CNC testing and validation (Oi MF Series)

Fanuc Focas API is a programming interface provided by Fanuc, a leading manufacturer of industrial robots and CNC (Computer Numerical Control) systems. The Focas API allows developers to interact with and control Fanuc CNC systems using various programming languages. It provides a set of functions and commands that enable applications to retrieve information from CNC controllers, monitor machine status, send commands, and perform data exchanges with the CNC system. The Focas API is widely used in manufacturing, automation, and robotics industries to develop custom software solutions for Fanuc CNC machines.

The Above CNC machine is used for milling the workpieces for which monitoring parameters such as Modes of operation, Status, and the spindle speed for the precise machining and the healthy spindle is necessary. The CNC machine and the client used for data retrieval are connected through the LAN/WAN network using their IP address obtained from the

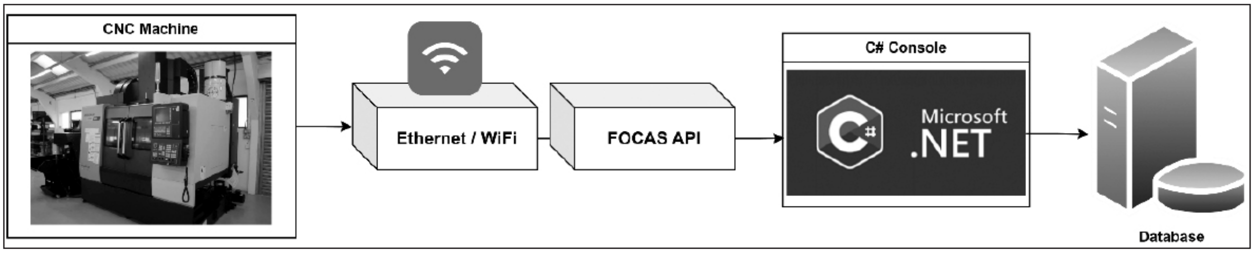


Fig. 13. Fanuc CNC setup for data retrieval.

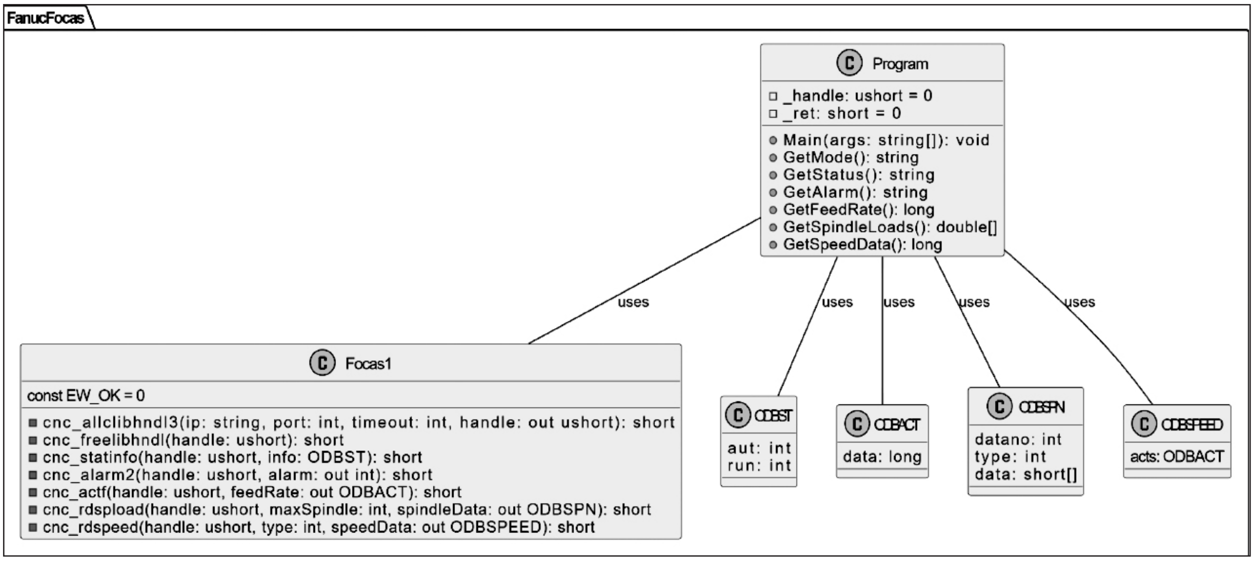


Fig. 14. Flow chart showing the data retrieval method in FANUC CNC.

embedded or ethernet communication setting of the CNC machine.

Data communication is done using the ethernet or WI-FI connection, which has a TCP/IP communication protocol through the Open CNC Focas API, a data window library. Specifically, fwlib.DLL proxy function files of suitable versions are used to retrieve data using the open-source C# client console. The sample code for this is attached to the appendix part. The flowchart of the program code is shown in Fig 14.

The Fanuc open CNC document gives the standard function calls with a function Focas1. cnc_freelibhndl () that need to be used to obtain the various parameters of the CNC machine using which the program needs to be written, as shown in Figure 15. The machine's mode, status, spindle speed, and feed rate are retrieved from the CNC machine in this sample.

Likewise, data can be retrieved from the FOCAS API using any open-source client console for any Fanuc controller-based CNC machine. Hence, the proprietary MT Connect protocol-based MT-LINKi can be avoided.

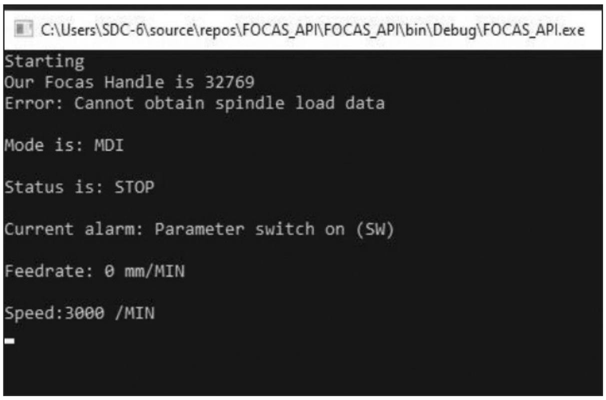


Fig. 15. Real data retrieved from the CNC machine.

5.2. Heidenhain CNC testing and validation (iTNC530E)

This CNC machine is a top-tier device for single-axis milling and turning operations. It is equipped with advanced intelligent connectivity technology and includes a proprietary software called workstation software, which allows for easy retrieval of controller data. The CNC controller and client workstation use the LSV2 communication protocol over an ethernet connection. Figure 16 illustrates the configuration for retrieving data from the controller using

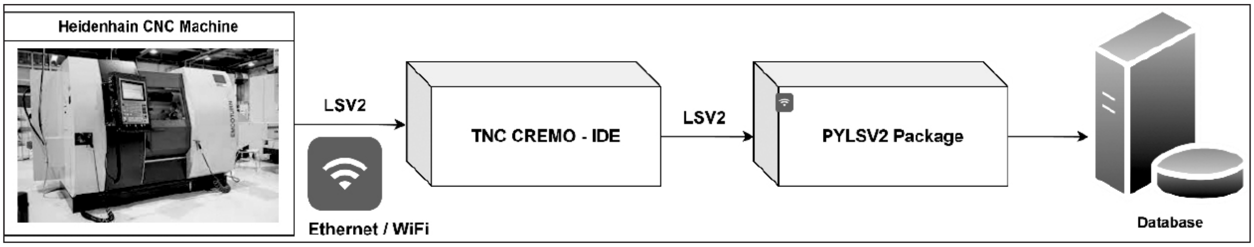


Fig. 16. Setup for data retrieval from heidenhain CNC machine.

```
import pyLSV2

# Connect to the Heidenhain machine

ip_address = "192.168.56.101"  #  IP address of your machine

with pyLSV2.LSV2(ip_address) as con:

    con.connect()

    # Check if connected to an iTNC control

    if con.is_itnc():

        # Read software version using data path (only for iTNC controls)

        software_version = con.read_data_path("/PLC/memory/K/1")

        print(f"Software version: {software_version}")

        # Read PLC data using data path

        plc_value = con.read_data_path("/PLC/memory/D/12")  # Adjust address and data type as needed

        print(f"PLC value (D12): {plc_value}")

    else:

        # Read software version using control attribute (works for all controls)

        software_version = con.versions.control

        print(f"Software version: {software_version}")

        # Read PLC memory using memory address (works for all controls)

        plc_data = con.read_plc_memory(100, pyLSV2.MemoryType.DWORD, 1)  # Adjust address, memory type, and count as needed

        print(f"PLC data (D100): {plc_data[0]}")

# Disconnect from the machine

con.disconnect()
```

TNCremo and the open-source Python library file PyLSV2 (Balduzzi et al., 2023).

LSV2 is a fast and efficient serial communication protocol designed for transferring files and exchanging data between CNCs and client devices, often computer workstations, employing a DNC-max client-server strategy. The TNCremo software package, which is open-source, is utilised to retrieve data from the PLC and transmit it to the client workstation via the ethernet interface. The pyLSV2 Python library package is designed to retrieve control data from the CNC machine using LSV2 communication. This is achieved by invoking specified calls for every desired function, as long as the data is in unencrypted format. The sample code with package pyLSV2 is given.

Similarly, the Python library package pyLSV2 allows for data extraction from any Heidenhain CNC machine. This communication occurs through LSV2 communication between the CNC and TNCremo software, eliminating the need for the proprietary floor machine plan program.

6. Case Study on Real-Time Data Retrieval from Collaborative Robots Using Experimental Analysis

The case study empirically analyses real-time data extraction from cooperative robots. Cobots are developed for collaborative work. Real-time data is essential for performance monitoring, safety, and productivity. Trials will be run with collaborative robots performing various tasks while collecting data. The data may include joint and tool positions, sensor readings (force and torque), operating statuses (standby or movement), and other vital factors. Then, experimental methods are used to examine the data to understand better the robot's behaviour, performance patterns, efficiency, safety, and operational difficulties. The analytics may include statistical analysis, visualisation, correlation research, and comparison with established models or benchmarks. The case study uses real-time data retrieval and experimental analysis to conclude, make recommendations, or suggest improvements. These insights are essential for improving robot operations, identifying areas for improvement, improving human-robot collaboration, and assuring system dependability and performance in collaborative robot environments.

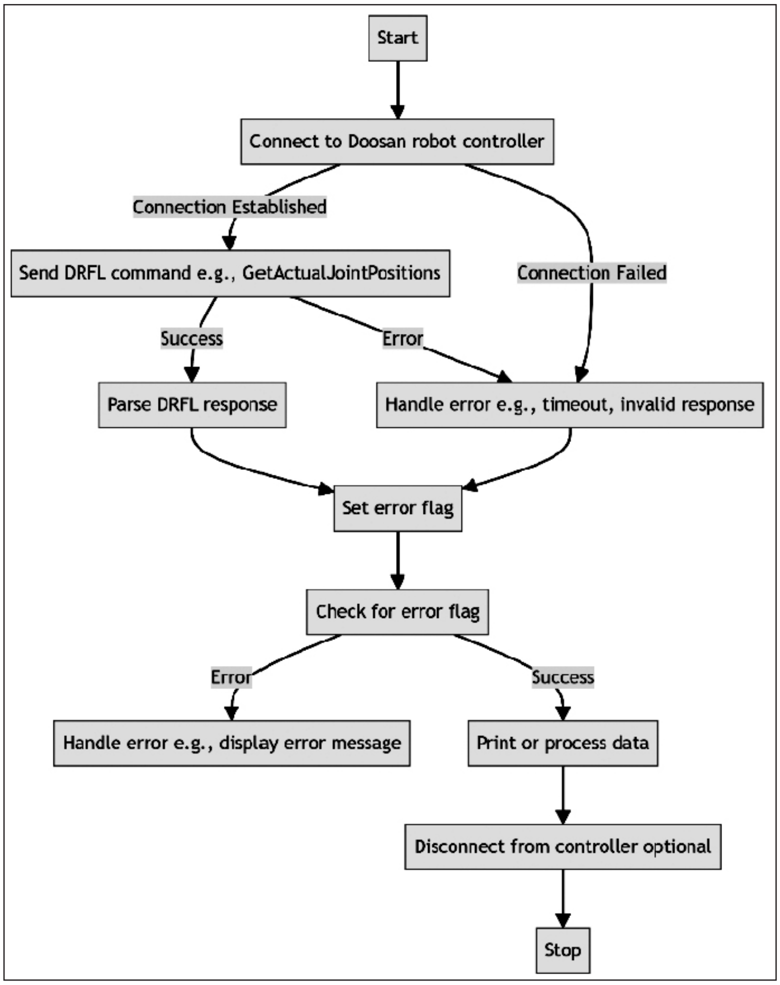


Fig. 17. Flow chart showing the data retrieval method in doosan cobot using DRFL.

```
#include <iostream>

#include “../../include/DRFLEx.h” // Include the DRFL library header file

int main() {

  DRAFramework::CDRFLEx Drfl;    // Create an instance of the CDRFLEx class for communication with
  the robot

  if (!Drfl.open_connection(“192.168.1.1”))

  {

    std::cerr << “Failed to connect to the robot controller!” << std::endl;

    return 1; // Return error code

  }

  float jointPos[6];    // Get the current joint positions of the robot

  if (Drfl.get_joint_pos(jointPos)) {

    std::cout << “Joint Positions:” << std::endl;        // Print the retrieved joint positions

    for (int i = 0; i < 6; ++i) {

      std::cout << “Joint “ << i + 1 << “: “ << jointPos[i] << “ degrees” << std::endl;

    }

  } else {

    std::cerr << “Failed to retrieve joint positions!” << std::endl;

  } // Close the connection to the robot controller

  Drfl.CloseConnection();

  return 0;

}
```

6.1. Doosan cobot testing and validation (Doosan A0509s)

To retrieve data from a Doosan robot using the DRFL (Doosan Robot Framework Library) package, you need to connect to the robot controller and use the methods given by the DRFL package to retrieve different data. This encompasses a range of data, such as system information, monitoring the state of the robot, sensor data, input/output data, data related to program execution, settings for configuring the robot, and handling/logging of errors. The data retrieval process starts by establishing a connection, diverges into several paths to retrieve specific data types, and then converges at a central location for processing and usage. Error handling systems guarantee the dependability of data retrieval processes. In summary, the DRFL package facilitates organised and efficient extraction of data from Doosan robots, effectively assisting in monitoring, control, and analysis activities.

Sample code to get the current position of the robot is given above.

7. Proposal of Wireshark as a Universal Open - Source Solution for Dataretrieval from Industrial Machines Using Experimental Analysis

Wire shark is an open-source network protocol sniffer tool that is easy to use. It is used for collecting, diagnosing, and analysing network

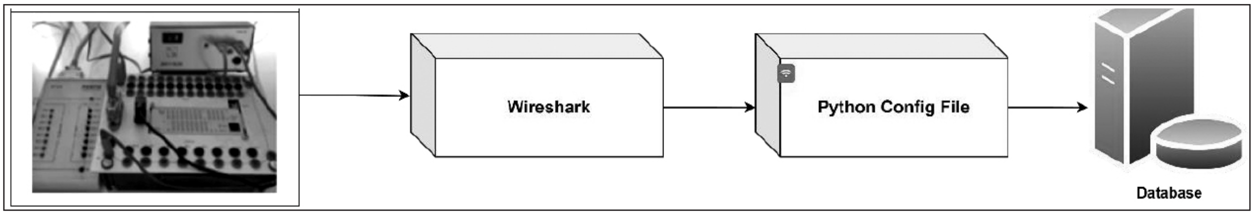
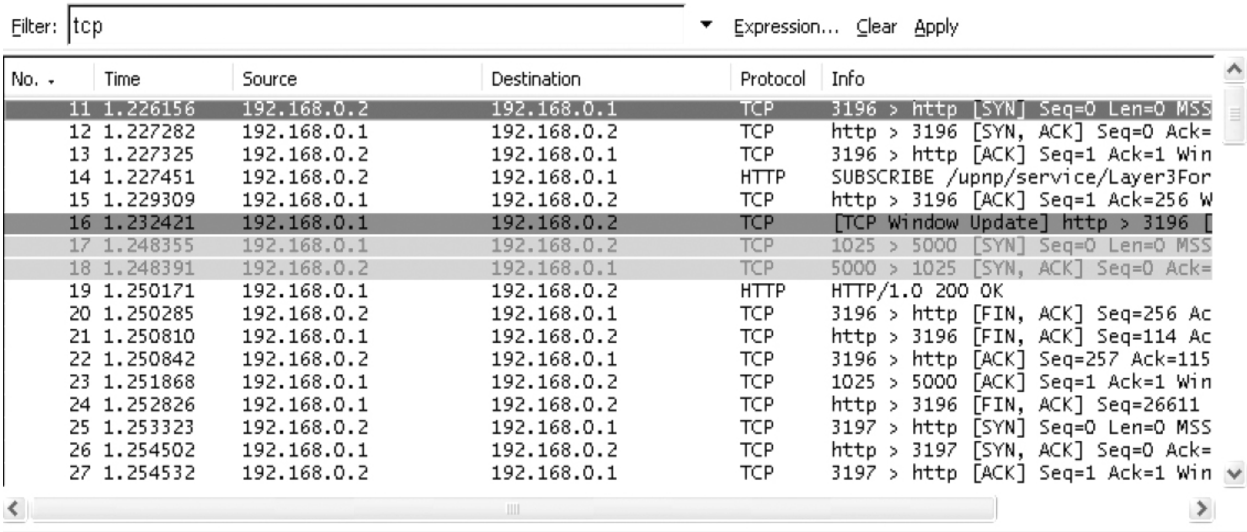


Fig. 18. Setup to retrieve data through the wireshark tool from PLC.



No.	Time	Source	Destination	Protocol	Info
11	1.226156	192.168.0.2	192.168.0.1	TCP	3196 > http [SYN] Seq=0 Len=0 MSS
12	1.227282	192.168.0.1	192.168.0.2	TCP	http > 3196 [SYN, ACK] Seq=0 Ack=
13	1.227325	192.168.0.2	192.168.0.1	TCP	3196 > http [ACK] Seq=1 Ack=1 Win
14	1.227451	192.168.0.2	192.168.0.1	HTTP	SUBSCRIBE /upnp/service/Layer3For
15	1.229309	192.168.0.1	192.168.0.2	TCP	http > 3196 [ACK] Seq=1 Ack=256 W
16	1.232421	192.168.0.1	192.168.0.2	TCP	[TCP Window Update] http > 3196 [
17	1.248355	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [SYN] Seq=0 Len=0 MSS
18	1.248391	192.168.0.2	192.168.0.1	TCP	5000 > 1025 [SYN, ACK] Seq=0 Ack=
19	1.250171	192.168.0.1	192.168.0.2	HTTP	HTTP/1.0 200 OK
20	1.250285	192.168.0.2	192.168.0.1	TCP	3196 > http [FIN, ACK] Seq=256 Ac
21	1.250810	192.168.0.1	192.168.0.2	TCP	http > 3196 [FIN, ACK] Seq=114 Ac
22	1.250842	192.168.0.2	192.168.0.1	TCP	3196 > http [ACK] Seq=257 Ack=115
23	1.251868	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [ACK] Seq=1 Ack=1 Win
24	1.252826	192.168.0.1	192.168.0.2	TCP	http > 3196 [FIN, ACK] Seq=26611
25	1.253323	192.168.0.2	192.168.0.1	TCP	3197 > http [SYN] Seq=0 Len=0 MSS
26	1.254502	192.168.0.1	192.168.0.2	TCP	http > 3197 [SYN, ACK] Seq=0 Ack=
27	1.254532	192.168.0.2	192.168.0.1	TCP	3197 > http [ACK] Seq=1 Ack=1 Win

Fig. 19. PCAP file captured from festo PLC through ethernet interface.

traffic statistics. It can handle a wide range of data, from general packet-captured files to specialised communication protocol connections. The system can capture real-time data transmitted over the network and selectively analyse the recorded packets using either a packet filter or a data filter to obtain the specific information needed. The wire shark tool was discovered as an innovative alternative during the investigation into open-source software options for extracting data from PLCs and CNCs. By utilising a packet capture file (PCAP), this tool can intercept the communication between the PLC and the client. It then applies a filter in the specified format to extract the content. Data can only be taken and examined from the device in an unencrypted format. This strategy is ineffective for encrypted data found in state-of-the-art Programmable Logic Controllers (PLCs) that have OPC UA capability. Nevertheless, it is possible to quickly extract unencrypted raw data from legacy PLCs and CNCs using the Wireshark protocol analyser tool, as depicted in Figure 18.

This procedure is verified by employing a FESTO PLC linked to Wireshark over the Ethernet interface via the TCP/IP communication protocol, as depicted in Figure 18. The Festo FC series PLC captures all packet communications and sends them to a PCAP file, including live data. A filter is then applied to isolate traffic between the PLC's IP address and the client's IP address. The captured data can be further filtered using the syntax:

ip.addr==<source address> && ip.addr== <destination address>

The file is exported with designated packets in cap format for subsequent extraction. The configuration file is constructed using an open-source Python module. It is used to selectively process the original Payload data of the Captured TCP packets. The original payload data is retrieved from the TCP payload's original attribute (pct [TCP]. payload. original) and appended to the tcp data list. The TCP primary data is retrieved and subsequently stored in an Excel file, as depicted in figure 20.

Additionally, this data can be easily visualised into a format using the input, output, status, and memory variables, as depicted in Figure 21, following the Festo PLC programming syntax

Therefore, by utilising the open-source Wireshark network protocol analyser program, it is possible

UDP Payload Data
b'\x14'
b'\x14\x00FESTO IPC V2.26'
b'L?'
b'L?\x00FEC0,$71B,TESTT,B:\\PROJECT.RUN,5166:6B46'
b'LB'
b'LB\x00=1685678739,Fri Jun 02 09:35:39 2023'
b'L?'
b'L?\x00FEC0,$71B,TESTT,B:\\PROJECT.RUN,5166:6B46'
b'FP'
b'FP\x00AB'
b'FDA:\\PROJECT.ZIP'
b'FDA:\\PROJECT.ZIP\x00\x07\r\nACCESS ERROR'
b'FDA:\\PROJECT.ZIP'
b'FDA:\\PROJECT.ZIP\x00\x07\r\nACCESS ERROR'
b'FDB:\\PROJECT.ZIP'
b'FDB:\\PROJECT.ZIP\x00\x07\r\nACCESS ERROR'
b'FDB:\\PROJECT.ZIP'
b'FDB:\\PROJECT.ZIP\x00\x07\r\nACCESS ERROR'
b'FDA:\\PROJECT.SAV'
b'FDA:\\PROJECT.SAV\x00\x07\r\nACCESS ERROR'
b'FDA:\\PROJECT.SAV'
b'FDA:\\PROJECT.SAV\x00\x07\r\nACCESS ERROR'
b'FDB:\\PROJECT.SAV'
b'FDB:\\PROJECT.SAV\x00\x07\r\nACCESS ERROR'
b'FDB:\\PROJECT.SAV'
b'FDB:\\PROJECT.SAV\x00\x07\r\nACCESS ERROR'
b'FDA:\\PROJECT.RUN'
b'FDA:\\PROJECT.RUN\x00\x07\r\nACCESS ERROR'
b'FDA:\\PROJECT.RUN'
b'FDA:\\PROJECT.RUN\x00\x07\r\nACCESS ERROR'
b'FDB:\\PROJECT.RUN'
b'FDB:\\PROJECT.RUN\x00'
b'FTA:\\FSTPCR22.EXE'
b'FTA:\\FSTPCR22.EXE\x00=35883,16.1.2004,18:31:50'

b'DA0.0'
b'DA0.0\x00=0'
b'DE0.0'
b'DE0.0\x00=0'
b'DA0.1'
b'DA0.1\x00=0'
b'DZO'
b'DZO\x00=1'
b'DPO'
b'DPO\x00=1,0,0,0,0,0'
b'BRL'
b'BRL\x00=0'
b'DF'
b'DF\x00=0,0,0'
b'DA0.0'
b'DA0.0\x00=0'
b'DE0.0'
b'DE0.0\x00=0'
b'DA0.1'
b'DA0.1\x00=0'
b'DZO'
b'DZO\x00=1'
b'DPO'
b'DPO\x00=1,0,0,0,0,0'
b'BRL'
b'BRL\x00=0'
b'DF'
b'DF\x00=0,0,0'
b'DA0.0'
b'DA0.0\x00=0'
b'DE0.0'

	A	B	C
334	b'BRL'		
335	b'BRL\x00=0'		
336	b'DF'		
337	b'DF\x00=0,0,0'		
338	b'DA0.0'		
339	b'DA0.0\x00=0'		
340	b'DE0.0'		
341	b'DE0.0\x00=0'		
342	b'DA0.1'		
343	b'DA0.1\x00=1'		
344	b'DZO'		
345	b'DZO\x00=1'		
346	b'SP0'		
347	b'SP0\x00'		
348	b'DP0'		
349	b'DP0\x00=1,0,0,0,0,0'		
350	b'BRL'		
351	b'BRL\x00=0'		
352	b'DF'		
353	b'DF\x00=0,0,0'		
354	b'DA0.0'		
355	b'DA0.0\x00=0'		
356	b'DE0.0'		
357	b'DE0.0\x00=0'		
358	b'DA0.1'		
359	b'DA0.1\x00=1'		
360	b'DZO'		
361	b'DZO\x00=1'		
362	b'DP0'		
363	b'DP0\x00=1,0,0,0,0,0'		
364	b'BRL'		
365	b'BRL\x00=0'		
366	b'DF'		
367	b'DF\x00=0,0,0'		
368	b'DA0.0'		

Fig. 20. The real data of the PLC extracted from the excel sheet.

Transactions	Program Status	Digital Input 1	Digital Output 1	Digital Output 2	Counter
	DP0	DE0.0	DA0.0	DA0.1	DZO
1	Inactive	OFF	OFF	OFF	ON
2	Inactive	OFF	OFF	OFF	ON
3	Inactive	OFF	OFF	OFF	ON
RUN PROGRAM 0 (RP0)					
4	Active	OFF	OFF	ON	ON
5	Active	OFF	OFF	ON	ON
6	Active	ON	ON	ON	ON
7	Active	OFF	OFF	ON	ON
8	Active	OFF	OFF	ON	ON
9	Active	OFF	OFF	ON	ON
10	Active	OFF	OFF	ON	ON
11	Active	OFF	OFF	ON	ON
12	Active	OFF	OFF	ON	ON
13	Active	OFF	OFF	ON	ON
14	Active	ON	ON	ON	ON
15	Active	ON	ON	ON	ON
16	Active	OFF	OFF	ON	ON
17	Active	OFF	OFF	ON	ON
18	Active	OFF	OFF	ON	ON
19	Active	OFF	OFF	ON	ON
STOP PROGRAM 0 (SP0)					
20	Inactive	OFF	OFF	ON	ON
21	Inactive	OFF	OFF	ON	ON
22	Inactive	OFF	OFF	ON	ON
23	Inactive	OFF	OFF	ON	ON
24	Inactive	OFF	OFF	ON	ON

Fig. 21. Visualisation of the retrieved data from PLC through wireshark.

to extract the RAW data from outdated PLCs and CNCs and store it in a database through an ethernet or USB interface. If the system only has a serial interface, it is possible to establish a connection by utilising a serial-to-USB converter and setting up a virtual interface. This will allow for the easy retrieval of data from the legacy machines.

8. Conclusion

The study examines industrial automation, explicitly acquiring real-time data using open-source frameworks. Obtaining data in real-time from industrial control systems like PLCs and CNCs is crucial for raising productivity, cutting costs, and improving operational efficiency in manufacturing operations. Several original equipment manufacturers (OEMs) provide exclusive frameworks for controlling and exchanging data, which can be financially expensive, particularly for micro, small, and medium enterprises (MSMEs). In addition, older controllers frequently need to meet the compatibility requirements of Industry 4.0 standards, which restricts their ability to analyse and monitor data.

The research promotes using open-source software and technologies that may be used for CNCs, PLCs, and robots. In this paper, we have conducted experiments on multiple hardware involving different Original Equipment Manufacturers (OEMs) that use various communication protocols. Our objective was to confirm the practicality of collecting data using open-source frameworks such as Node-RED, CoDeSys, Python libraries, and new techniques like the Wireshark network analyser.

Node-RED provides remarkable adaptability and variety with its visual programming interface and massive collection of pre-built nodes. However, it is essential to acknowledge that other open-source tools also possess their unique advantages. Python modules provide versatility and the ability to tailor to specific needs, whereas Wireshark offers robust network analysis capabilities.

It is crucial to recognise that every instrument possesses its benefits and constraints. Although Node-RED's intuitive interface streamlines intricate procedures, there may be better options for highly specialised applications. Similarly, while some tools may demonstrate exceptional performance in specific domains, they may provide a different degree of user-friendliness.

The experimental study results establish the foundation for considerable progress in Industrial 4.0 IIoT technology, facilitating real-time data retrieval and improving efficiency in industrial processes. To fully embrace the Industry 4.0 revolution, it is crucial to focus on advancing open-source solutions for communication across control systems, mainly when dealing with older machinery. The architecture extends to machines installed across the enterprise network. This research showcases the feasibility of retrieving data from widely used OEMs, PLCs, CNCs, and robots without relying on proprietary software. The case study focuses on demonstrating this possibility and may expand to include other particular or customised machines through thorough analysis.

References

Aminaie, P., & Aminaie, P. (2020). Profinet vs Profibus. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2011.14167

Balduzzi, M., Sortino, F., Castello, F., & Pierguidi, L. (2023). A security analysis of CNC machines in industry 4.0. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 13959 LNCS.* https://doi.org/10.1007/978-3-031-35504-2_7

Bozdal, M., Samie, M., & Jennions, I. (2018). A Survey on CAN bus protocol: Attacks, Challenges, and potential solutions. *Proceedings - 2018 International Conference on Computing, Electronics and Communications Engineering, ICCECE 2018, 201-205.* https://doi.org/10.1109/iCCECOME.2018.8658720

Decotignie, J. D. (2005). Ethernet-based real-time and industrial communications. *Proceedings of the IEEE,* 93(6), 1102-1117. https://doi.org/10.1109/jproc.2005.849721

Dias, A. L., Sestito, G. S., Turcato, A. C., & Brandao, D. (2018). Panorama, challenges and opportunities in PROFINET protocol research. *2018 13th IEEE International Conference on Industry Applications, INDUSCON 2018 - Proceedings.* https://doi.org/10.1109/INDUSCON.2018.8627173

Doosan Robotics. (2023). API−DRFL: A Python Package for Data Analysis. Retrieved from https://github.com/doosan−robotics/API−DRFL. License:MITLicense.

Gaitan, V. G., & Zagan, I. (2022). Modbus protocol performance analysis in a variable configuration of the physical fieldbus architecture. *IEEE Access, 10,* 123942-123955. https://doi.org/10.1109/ACCESS.2022.3224720

Goel, R., & Gupta, P. (2020). *Robotics and Industry 4.0.* In: Nayyar, A., & Kumar, A. (eds) A Roadmap to Industry 4.0: smart production, sharp business, and sustainable development. *Advances in Science, Technology & Innovation.* Springer. https://doi.org/10.1007/978-3-030-14544-6_9

Herath, H. M. K. K. M. B., Ariyathunge, S. V. A. S. H., & Priyankara, H. D. N. S. (2020). Development of a data acquisition and monitoring system based on MODBUS RTU communication protocol. *International Journal of Innovative Science and Research Technology, 5(6)*. https://doi.org/10.38124/ijisrt20jun479

Hunkeler, U., Truong, H. L., & Stanford-Clark, A. J. (2008). MQTT-S — A publish/subscribe protocol for wireless sensor networks. *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08),* 791-798. DOI:10.1109/COMSWA.2008.4554519

Kim, I., Kim, T., Sung, M., Tisserant, E., Bessard, L., & Choi, C. (2013). An open-source development environment for industrial automation with EtherCAT and PLC open motion control. *IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA),* Cagliari, Italy, 1-4. doi: 10.1109/ETFA.2013.6648122

Langmann, R., & Stiller, M. (2019). The PLC as a smart service in industry 4.0 production systems. *Applied Sciences, 9(18),* 3815. https://doi.org/10.3390/app9183815

Li, Y., Wang, Y., & Ma, C. (2015). Design of communication system in intelligent instrument based on HART protocol. *2015 IEEE International Conference on Mechatronics and Automation, ICMA 2015,* 351-356. https://doi.org/10.1109/ICMA.2015.7237510

Lin, Z., & Pearson, S. (2018). An inside look at industrial ethernet communication protocols strategic marketing manager texas instruments strategic marketing manager texas instruments. *Texas Instruments, White Paper.*

Lins, R. G., Guerreiro, B., Schmitt, R., Sun, J., Corazzim, M., & Silva, F. R. (2017). A novel methodology for retrofitting CNC machines based on the context of industry 4.0. *IEEE International Symposium on Systems Engineering, ISSE - Proceedings.* https://doi.org/10.1109/SysEng.2017.8088293

Mageshkumar, G., Kasthuri, N., Tamilselvan, K. S., Suthagar, S., & Sharmila, A. (2020). Design of industrial data monitoring device using IoT through modbus protocol. *International Journal of Scientific and Technology Research, 9(1),* 1392-1396.

Martins, A., Lucas, J., Costelha, H., & Neves, C. (2023). CNC machines integration in smart factories using OPC UA. *Journal of Industrial Information Integration, 34.* https://doi.org/10.1016/j.jii.2023.100482

Nast, M., Butzin, B., Golatowski, F., & Timmermann, D. (2019). Performance analysis of a secured BACnet/IP network. *2019 15th IEE International workshop on factory communication systems (WFCS),* 1-8.

Quintã, A. F., Santos, J. P., & Cardeira, C. (2005). Performance assessment of DDE versus activex in manufacturing environments. CLME'2005-ICEM, 1011-1020. Maputo. Mozambique.

Rockwell Automation. (2022). *DeviceNet network configuration: DNET- UM004E-EN-P.* https://literature.rockwellautomation.com/idc/groups/literature/documents/um/dnet-um004_-en-p.pdf

Sharpe, R, Warnicke, E., Lamping, U. (2004). Wireshark User's Guide Version 4.3.0. Wireshark foundation.

Siemens, PROFINET System Description. (Accessed: 15.05.2021). [Online]. Available: https://cache.industry.siemens.com/dl/files/127/19292127/ att 69558/v1/profinet system description en-US en-US.pdf

APPENDIX

1. Sample code for retrieving Fanuc CNC Data using FOCAS API (only including function part of program)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace FanucFocas
{
  class Program
  {  static ushort _handle = 0;
    static short _ret = 0;

    static void Main(string[] args)
    {
      Console.WriteLine("Starting");

      _ret = Focas1·cnc_allclibhndl3("172·18·30·147", 8193, 6, out _handle);

      if (_ret != Focas1.EW_OK)
      {
        Console.WriteLine("Cannot Connect");
        Console.Read();
      }
      else
      {
        Console.WriteLine($"Our Focas Handle is {_handle}");
        string mode = GetMode();
        string status = GetStatus();
        string alarm = GetAlarm();
        long feedRate = GetFeedRate();
        double[] spindleLoads = GetSpindleLoads();
        long speedData = GetSpeedData();
        Console.WriteLine($"\nMode is: {mode}");
        Console.WriteLine($"\nStatus is: {status}");
        Console.WriteLine($"\nCurrent alarm: {alarm}");
        Console.WriteLine($"\nFeedrate: {feedRate} mm/MIN");
        Console.WriteLine($"\nSpeed:{speedData} /MIN");
        for (int i = 0; i < spindleLoads.Length; i++)
        {
          Console.WriteLine($"Spindle {i + 1}: {spindleLoads[i]}");
        }
        Console.Read();
      }
      Focas1·cnc_freelibhndl(_handle);
    }
```

Tadigotla Narendra Reddy is a Scientist specialising in the areas of system developments, machine developments, solution developments related to Smart Manufacturing & Industry 4.0, Ultra Precision Machine and subsystem Developments & Nanotechnology Equipment's. He is currently working on Smart Manufacturing Solutions development related to Real-time error correction, Intelligent machining, Digital Manufacturing, Smart Metrology, and IIoT. He is pursuing research work at NITK in Smart manufacturing. He has developed indigenous products ranging from TRL6 – TRL8, such as Intelligent Ultra Precision Turning Machines, Ultra Stiff Ultra Precision Turning Machines, Portable Scanning Tunnelling Microscopes (STM), Closed Loop Nano Positioning Stage, Nano-Motion Systems (Hydrostatic Slides & Aerostatic Spindle)

Prasad Prasannan is a Project Fellow II at the Central Manufacturing Technology Institute (CMTI) in Bangalore and hails from Kerala, India. He holds a B.Tech in Electrical and Electronics Engineering and specialises in Industrial Automation. Prasad has conducted data retrieval validation for various industrial devices and has two years of experience developing IoT solutions and contributing to research development projects. His expertise lies in bridging technology and industry needs, reflecting a dedication to excellence in engineering and technology, focusing on automation and innovation.
(E-mail: iamprasadpattingal@gmail.com)

Seema B. Hegde is an Assistant Professor in the Department of Electronics and Communication Engineering at Siddaganga Institute of Technology Tumkuru. She has both academic and research experience of 20 years and has published more than 25 research articles in reputed journals and conferences. Her area of interest is the Internet of Things, Embedded systems, Computer networks, and Artificial Intelligence. She was also a technical committee member and session chair at international conferences and symposiums.
(E-mail: seemab_hegde@sit.ac.in)

Prakash Vinod is a senior scientist with rich experience in planning, executing and managing multidisciplinary applied research and Development works in advanced manufacturing, resulting in industry-ready products and solutions. Currently, he is heading "The Centre for Smart Manufacturing, Precision Machine Tools and Aggregates". Over the past 33 years at CMTI, he has been working for Technology and Machine/ Product development activities (TRL3 to TRL8) and developed more than 30 machines/products in the areas of Ultra-precision machine tools, and its Aggregates, Smart machines and equipment, Smart Manufacturing and Industry 4.0 solutions to Manufacturing industries, Special purpose machines (SPMs), Development of machines, mechanisms and processes for Nano finishing, Micromachining & fabrication, Nano Metrology & Characterization, Inspection, Calibration & Testing of machine tools, Vibration & Noise Analysis and condition monitoring of machines and rotating equipments. (E-mail: prakashv@cmti.res.in)

Mervin A. Herbert is working as Associate Professor in Mechanical Engineering Department at National Institute of Technology NITK, Surathkal. His research interests are Friction Stir Welding, Semi solid processing of composites, applications of Artificial Neural Network. He has 6 years of Industrial Experience and 27 years of diverse experience in teaching and also contributed for setting up the research labs in the department. He has supervised 12+ Ph.D aspirants.
(E-mail: merhertoma@gmail.com)

Shrikantha S. Rao is working as Professor in Mechanical Engineering Department at National Institute of Technology NITK, Surathkal. He is presently Dean (Alumni and Corporate Relations) of NITK. He has 32+ years of diverse experience in teaching and also established Hands on experience labs like CAD/CAM, CNC Lab and Microcontroller Labs at National Institute of Technology NITK. His application interests are Artificial Intelligence and its Applications in Manufacturing. He has been involved in various research projects namely Earth Volume Estimation software for MRPL project, Scrap Assessment software for Chiramith Watch Works, Mangalore etc., He has supervised 16 Ph.D aspirants and published over 75 papers in international journals.
(E-mail: ssrcsr@gmail.com)