Politeknik Negeri Batam

Magister Terapan Teknik Komputer

"An Integrated Electric Motor Monitoring System with Display and IoT Connectivity for Electric Motor Assessment"

Michael Rafly
7312411005

Pembimbing: **Ir. Daniel Sutopo Pamungkas, S.T., M.T., Ph.D., IPM**.

Tesis ini dibuat sebagai syarat pengajuan seminar proposal Magister Terapan Teknik Komputer
di
Politeknik Negeri Batam

July 4, 2025

Declaration

I, Michael Rafly, of the Department of Computer Science, University Batam State Polytechnic, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalized accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of UoR and public with interest in teaching, learning and research.

Michael Rafly
July 4, 2025

Abstract

Electric motors are the backbone of industrial automation and manufacturing systems, yet motor failures due to undetected mechanical imbalances, misalignment, or vibration anomalies can result in costly downtime. This thesis proposes the design and implementation of an Integrated Electric Motor Monitoring System that utilizes Condition Monitoring Sensor for real-time condition monitoring. The system features a local visual display interface for immediate on-site diagnostics and an IoT-based data transmission module for remote access and cloud-based trend analysis.

The sensor offers precise vibration measurement across multiple axes and is suitable for harsh industrial environments, enabling the system to detect early signs of motor wear or imbalance. The data acquired is processed using a microcontroller, displayed on an HMI (Human-Machine Interface), and transmitted via MQTT protocol to a cloud dashboard for remote monitoring and predictive maintenance strategies.

The objective of this research is to improve motor reliability, reduce maintenance costs, and promote preventive diagnostics by integrating sensor-based analytics with IoT infrastructure. This project aligns with the broader vision of Industry 4.0 and smart factory implementations, emphasizing predictive maintenance and system integration.

Previous studies, such as "Development of Condition Monitoring System Based on Vibration Analysis for Predictive Maintenance in Industrial Motors" (Siddique, A., Yadava, G.S., Singh, B., *IEEE Transactions on Industrial Electronics*, 2003), have highlighted the effectiveness of vibration-based monitoring in detecting motor faults. By integrating such sensors with IoT and display technologies, this research extends the conventional model to offer a more scalable, connected, and user-friendly motor health monitoring solution.

The implemented system was validated on two industrial air-compressor units driven by induction motors. On a healthy three-phase unit rated 10 HP (7.5 kW), the automatic baseline-teaching procedure, executed from the web dashboard, learned an operating baseline of B = 28.66 mm/s and derived the operating limits accordingly (off-threshold 11.46 mm/s, pre-warning 51.58 mm/s, and alarm 71.64 mm/s); over 1,767 valid samples logged across approximately 37 hours the magnitude peaked at 38.68 mm/s, that is 75.0 % of the pre-warning limit, and neither a warning nor an alarm was raised. On a 1.5 HP (1.1 kW) single-phase unit whose pneumatic circuit was leaking, so that the cut-out pressure was never reached and the motor ran continuously, teaching produced a baseline of 6.50 mm/s with an off-threshold of 2.60 mm/s, a pre-warning limit of 7.48 mm/s, and an alarm limit of 8.13 mm/s, and the system captured the complete OFF, NORMAL, WARNING, and latched-ALARM transition, raising the first WARNING at 7.52 mm/s and latching the ALARM at 8.14 mm/s while the motor temperature rose to a peak of 78 °C. Using these learned limits, the system classified the OFF, NORMAL, WARNING, and latched-ALARM states in real time, remained consistent between the local LVGL HMI and the HTML web dashboard over the WISP-mode MQTT/HTTPS link, and sustained continuous cloud updates at one dashboard read per device update, keeping operation comfortably within the free quota of the cloud service. Because no controlled fault was injected and no independent ground truth was obtained, the demonstrated capability is deviation-based condition alarming rather than validated fault detection; the provenance of every learned baseline reported in this thesis is tabulated in Section 5.2.

Keywords: Motor Monitoring, Condition Monitoring Sensor, IoT-Based Diagnostics, Predictive Maintenance, Industry 4.0 Integration

Acknowledgements

First and foremost, I would like to convey my heartfelt gratitude to Allah SWT for bestowing upon me the strength, health, and resilience needed to complete this thesis journey.

I am deeply thankful to my thesis advisor, **Ir. Daniel Sutopo Pamungkas, S.T., M.T., Ph.D., IPM.,** for his continuous guidance, encouragement, and thoughtful feedback throughout each phase of this research. His expertise and support have been crucial in directing and enhancing the quality of this study.

I would also like to express my appreciation to the Master of Applied Computer Engineering Department at Batam State Polytechnic for the support, facilities, and resources that enabled the successful execution of this project. I am especially grateful to the lab technicians and staff whose assistance was vital during the development and testing of an integrated electric motor monitoring system with display and IoT connectivity for electric motor assessment.

To my beloved family and friends, thank you for your constant support, patience, and motivation. Your unwavering belief in me has been a source of strength throughout this process.

Finally, I extend my sincere thanks to everyone who has contributed to this thesis, whether through direct assistance, encouragement, or inspiration.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Definition
API	Application Programming Interface
a-RMS	Acceleration Root Mean Square
CM	Condition Monitoring
CMS	Condition Monitoring System
CSS	Cascading Style Sheets
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
ESP32	Espressif Systems 32-bit Microcontroller
GPIO	General Purpose Input/Output
GSDML	General Station Description Markup Language
GUI	Graphical User Interface
HMI	Human Machine Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
I^2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IO-Link	Point-to-Point Serial Communication Protocol (IEC 61131-9)
IODD	IO Device Description
IoT	Internet of Things
IP	Internet Protocol
ISO	International Organization for Standardization
JS	JavaScript
JSON	JavaScript Object Notation
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LVGL	Light and Versatile Graphics Library
MAC	Media Access Control
MCU	Microcontroller Unit
MEMS	Micro Electro Mechanical System
MPB10	Multi Physics Box 10 (SICK Vibration and Temperature Sensor)
MQTT	Message Queuing Telemetry Transport

Abbreviation	Definition
NTP	Network Time Protocol
NVS	Non-Volatile Storage
PC	Personal Computer
PLC	Programmable Logic Controller
PROFINET	Process Field Network
PSRAM	Pseudo-Static Random Access Memory
REST	Representational State Transfer
RGB	Red, Green, Blue
RMS	Root Mean Square
RTDB	Realtime Database
SDK	Software Development Kit
SIG200	SICK Sensor Integration Gateway 200 (IO-Link Master)
SOPAS	SICK Open Portal for Application and Systems
SPI	Serial Peripheral Interface
SSID	Service Set Identifier
TCP/IP	Transmission Control Protocol / Internet Protocol
TFT	Thin Film Transistor
TLS	Transport Layer Security
UI	User Interface
URL	Uniform Resource Locator
v-RMS	Velocity Root Mean Square
WBS	Work Breakdown Structure
Wi-Fi	Wireless Fidelity

Chapter 1
Introduction

1.1 Background

Electric motors are fundamental components in a vast range of industries, including industrial automation, manufacturing, HVAC systems, transportation, robotics, and many other sectors (Bilgin et al., 2019). Their role as the primary drivers of mechanical operations makes them indispensable for maintaining continuous productivity, optimizing performance, and ensuring operational efficiency. Due to the critical functions they perform, any malfunction or failure of electric motors can have far-reaching consequences, such as production halts, increased operational costs, and even safety risks for personnel and equipment.

Electric motors are regularly subjected to a variety of mechanical and electrical stresses throughout their service life. Mechanical issues such as rotor imbalance, shaft misalignment, bearing deterioration, and gearbox faults commonly arise from wear and tear or improper installation. Electrical problems including insulation breakdown, short circuits, and winding faults may also develop, often exacerbated by harsh operating conditions such a high temperatures, vibrations, and humidity. These faults tend to develop gradually over time, and if left undetected, they can culminate in unexpected downtime, extensive damage to the motor and connected equipment, and costly repairs or replacements (Fofana et al., 2016).

Traditional maintenance approaches for electric motors often rely heavily on periodic preventive maintenance schedules or manual inspections. While these methods have been widely used, they possess inherent drawbacks—they tend to be labor-intensive, require substantial downtime, and frequently fail to detect incipient faults at early stages (Altun et al., 2024). This limitation means that events such as bearing wear or shaft misalignment might only be discovered after progressing to a more severe stage, thus increasing the risk of failures that disrupt production and inflate maintenance expenses.

In today's industrial landscape, the growing adoption of Industry 4.0 paradigms and smart manufacturing principles highlights the need to move beyond conventional maintenance frameworks. There is a strong demand for intelligent, automated condition monitoring systems capable of providing real-time insights into the health and performance of electric motors (Mohammed et al., 2023). These systems not only allow early detection of faults but also enable predictive maintenance strategies, where interventions are planned based on the actual condition of the equipment rather than fixed schedules. This shift leads to optimized maintenance cycles, reduced downtime, increased asset lifespan, and lower operational costs.

One of the most effective and widely adopted techniques for electric motor condition monitoring is vibration analysis (Rehman et al., 2024). Vibration-based monitoring can reveal anomalies caused by mechanical defects such as imbalance, misalignment, bearing faults, and looseness (Tiboni et al., 2022). These subtle changes in vibration signals often precede more obvious symptoms, making vibration sensors a sensitive tool for early fault detection. By integrating these sensors with Internet of Things (IoT) technology, it becomes possible to create an interconnected monitoring system that continuously collects vibration data and transmits it to centralized cloud platforms or local servers for further analysis.

Combining sensor-based data acquisition with IoT connectivity facilitates seamless remote monitoring, enabling maintenance teams to access motor health data anytime and anywhere. This connectivity allows for the application of advanced data analytics and machine learning algorithms that can automatically detect and classify faults, predict remaining useful life, and recommend maintenance actions (Abduljawwad et al., 2023). Moreover, integrating a local display interface such as a Human-Machine Interface (HMI) directly on the monitoring device supports immediate on-site diagnostics by technicians. With such a local visualization system, operators can quickly assess motor conditions without the need for specialized handheld equipment or network access, enhancing response times and reducing operational risks.

The system envisioned in this thesis integrates these cutting-edge technologies into a single cohesive platform. It employs vibration sensors like the SICK-MPB10 to measure motor vibration parameters accurately, and combines these readings with microcontroller-based real-time processing and visualization via an embedded 7-inch HMI display powered by the LVGL graphics library. Data communication is accomplished through the MQTT protocol over Wi-Fi, ensuring efficient and reliable data transfer to cloud or on-premises platforms for remote monitoring (Singh et al., 2015).

By fusing local data presentation and remote accessibility, the proposed integrated electric motor monitoring system offers comprehensive benefits. It provides continuous health assessment, early fault detection, and supports predictive maintenance approaches that drastically improve operational uptime. Furthermore, the system’s design emphasizes scalability and user-friendliness, making it suitable for deployment across varied industrial settings, from small workshops to large smart factories (Yousuf et al., 2024).

In summary, the project set forth aims not only to address significant challenges with current electric motor maintenance practices but also to contribute meaningfully to the advancement of Industry 4.0 implementations. Through leveraging sensor technology, IoT communication, and intuitive visualization, it seeks to enhance motor reliability, reduce maintenance costs, and ensure safer, smarter industrial operations.

1.2 Problem statement

Electric motors play a vital role in industrial operations, yet their unexpected failure remains a significant challenge, often leading to unplanned downtime, increased maintenance costs, and production losses. Conventional motor maintenance methods, such as periodic manual inspections, are often reactive, inefficient, and unable to detect early signs of mechanical issues like imbalance, misalignment, or bearing wear.

Furthermore, many existing monitoring systems lack real-time diagnostics, are not user-friendly, and are typically limited to local monitoring without remote access or data analysis capabilities. This limits their effectiveness in predictive maintenance and decision-making, particularly in large-scale or remote industrial facilities.

There is a critical need for a compact, reliable, and connected system that can provide continuous monitoring, on-site diagnostics through a visual interface, and remote data access for real-time assessment and long-term analysis. Therefore, this thesis aims to address the gap by designing and implementing an Integrated Electric Motor Monitoring System that combines vibration sensing, a local display, and IoT connectivity to improve motor health assessment, reduce downtime, and support smart maintenance strategies in line with Industry 4.0 standards.

1.3 Aims and objectives

Aims: The aim of this project is to design and implement an **Integrated Electric Motor Monitoring System** that utilizes real-time vibration data, local display, and IoT connectivity to assess motor health, enable predictive maintenance, and reduce unplanned downtime in industrial settings. The system is intended to support smart factory initiatives and align with Industry 4.0 principles by offering both on-site and remote monitoring capabilities.

Objectives: To achieve the stated aim, the following specific and measurable objectives will be carried out:

a) **To research and analyze** existing electric motor fault detection techniques and identify the limitations of conventional monitoring systems.
b) **To design a hardware system** that integrates a vibration sensor (such as the SICK-MPB10), a microcontroller, and an HMI (Human-Machine Interface) for real-time motor condition display.
c) **To develop an IoT communication module** using MQTT protocol to transmit motor condition data to a cloud-based dashboard for remote access and trend analysis.
d) **To implement software algorithms** that process the measured vibration magnitude and automatically learn machine-specific thresholds, so that a statistically significant departure from the operating level the machine itself has taught — a deviation of the kind commonly associated with developing mechanical problems — is detected and reported in real time.
e) **To test and evaluate** the performance of the system in a simulated or real industrial environment, measuring its accuracy, reliability, and responsiveness.
f) **To assess the effectiveness** of the proposed system in reducing maintenance costs and improving motor reliability compared to traditional maintenance practices.

1.4 Scope and limitations

The scope of this thesis is bounded as follows. The system is developed and evaluated as a single-sensor, single-asset monitoring module: one MPB10 condition-monitoring sensor is mounted on one machine at a time, and the resultant velocity-RMS magnitude together with the sensor temperature are the only quantities acquired, classified, and transmitted. No spectral analysis, order tracking, or machine-learning classification is performed on the device, and no attempt is made to identify the type of a developing defect.

The classification limits are derived from a baseline that the system learns from the monitored machine itself. This makes the thresholds adaptive, but it also makes the meaning of every reported state dependent on the condition of the machine at the moment of teaching. In the sessions reported in Chapter 5, teaching was executed while the machine was already running at its operating level; the learned baseline therefore represents that operating condition rather than an independently verified healthy condition. Where a machine was already behaving abnormally when it was taught, the session demonstrates the adaptive mechanism and the alarming behaviour of the system, but it does not by itself prove a fault-detection capability.

Each baseline reported in this thesis was obtained from a single teaching cycle of ten samples over approximately five seconds. The repeatability of the teaching procedure — that is, the spread of the learned baseline over repeated cycles at an identical operating point — was not measured, so no standard deviation is reported for any baseline, and no confidence interval can be attached to the derived limits.

No controlled fault was injected into either monitored machine, and no independent ground truth (disassembly, calibrated vibration meter, or graded unbalance) was obtained. The evaluation is therefore presented as deviation-based condition alarming — the detection of a significant departure from the operating level the machine itself has taught — and not as a validated fault-detection or fault-diagnosis capability; consequently no detection rate and no false-alarm rate are claimed. Finally, the absolute scale of the magnitude reported by the sensor through the IODD process-data mapping was not verified against a calibrated reference instrument, so all conclusions are drawn from relative changes rather than from absolute vibration severity.

1.5 Solution approach

To address the challenges associated with electric motor failure and the limitations of traditional maintenance methods, this project adopts a technology-driven solution that combines sensor-based condition monitoring, embedded system design, and IoT integration.

The proposed system will utilize a vibration sensor, to continuously monitor the mechanical condition of electric motors. The sensor data will be collected and processed in real-time using a microcontroller unit (MCU), which will analyze vibration patterns to detect early signs of motor faults such as imbalance, misalignment, or bearing wear.

A Interface display will be integrated into the system to provide on-site technicians with a visual display of the motor’s health status, allowing for immediate diagnostics without the need for external diagnostic tools.

To enable remote access and cloud-based analytics, the system will feature IoT connectivity using the MQTT protocol, allowing the processed data to be transmitted to a cloud dashboard. This platform will support historical data logging, trend analysis, and alerts for predictive maintenance decision-making.

The methodology for this project includes:

a) **Requirement Analysis and System Design** – Identifying functional and technical requirements; designing system architecture and hardware components.

b) **Sensor Integration and Microcontroller Programming** – Interfacing the vibration sensor with a suitable microcontroller (e.g., ESP32 or Arduino) and developing firmware for real-time data processing.

c) **HMI Development** – Creating a user-friendly local display interface to visualize motor status and alerts.

d) **IoT Communication Implementation** – Establishing secure and reliable MQTT-based communication between the device and cloud dashboard.

e) **Cloud Dashboard Setup** – Developing a custom HTML/JavaScript web dashboard backed by a cloud database (Cloud Firestore) for remote monitoring, statistics accumulation, and remote teaching commands, and deploying it to a public static-hosting service (Netlify) so that it is reachable over HTTPS from any browser.

f) **Testing and Evaluation** – Validating system performance through real-time testing under controlled or operational conditions, and evaluating accuracy, responsiveness, and usability.

This approach ensures that the final system is scalable, efficient, and aligned with the goals of smart manufacturing and predictive maintenance under the Industry 4.0 framework.

1.6 Summary of contributions and achievements

This thesis presents a comprehensive design and implementation of an integrated electric motor monitoring system with display and IoT connectivity for electric motor assessment. The contributions and achievements of this work are summarized as follows:

1. Development of an Integrated Monitoring System:
   Successfully designed and implemented a comprehensive electric motor condition monitoring system that integrates vibration-based sensor data acquisition with real-time processing capabilities.

2. Local Visualization via Human-Machine Interface (HMI):
   Developed an intuitive local display interface that provides instant, on-site visualization of motor health parameters, enabling technicians to perform quick diagnostics without additional equipment.

3. IoT Connectivity using MQTT Protocol:
   Integrated MQTT-based IoT communication to facilitate remote monitoring by transmitting sensor data to a cloud platform, allowing for continuous condition monitoring and real-time alerts.

4. Automatic Threshold Teaching and Deviation-Based Condition Alarming:
   Implemented an automatic baseline-teaching mechanism that learns the operating vibration-magnitude level of the monitored machine and derives ISO-anchored operating limits (off, pre-warning, and alarm) from it. Each measured magnitude is classified in real time into OFF, NORMAL, WARNING, or latched ALARM states, providing an early indication that the machine has departed from its own learned operating level and thereby supporting condition-based maintenance decisions. The evidence gathered in this thesis supports deviation-based alarming; establishing a validated fault-detection capability would require controlled fault injection and independent ground truth, as set out in Section 1.4.

5. Enhancement of Industrial Operational Efficiency:
   Contributed to optimizing maintenance schedules and improving reliability of electric motors in industrial environments by providing scalable, user-friendly monitoring technology aligned with Industry 4.0 standards.

6. Quota-Efficient and Robust IoT Design:
   Introduced a single-latest-document cloud strategy in which one device update corresponds to exactly one dashboard read, combined with server-timestamp-based offline detection and automatic unsubscription on the dashboard, enabling sustained continuous operation within the free quota of the cloud service.

1.7 Organization of the report

This thesis is organized into six chapters, each focusing on critical aspects of the research, design, development, and evaluation of the integrated electric motor monitoring system with display and IoT connectivity:

Chapter 1: Introduction

This chapter introduces the context and motivation for the research, discusses the importance of electric motors in industrial environments, outlines common faults and maintenance challenges, and defines the research objectives, scope, and significance of the study.

Chapter 2: Literature Review

This chapter reviews existing studies and technologies related to electric motor condition monitoring, vibration-based sensing, IoT integration, local display interfaces, and predictive maintenance, and identifies the gaps that the proposed system addresses.

Chapter 3: Methodology

This chapter details the research design and the overall development methodology, including the system concept, architectural layers, hardware selection, software development approach, data flow, and the testing and validation plan.

Chapter 4: System Design and Implementation

This chapter presents the realized system in detail: the layered architecture, the network topology with the WISP-mode wireless link, the panel-mounted hardware assembly, the sensor configuration through the SIG200 gateway, the ESP32-S3 firmware, the automatic ISO-anchored threshold-teaching mechanism, the LVGL local monitor, and the globally deployed web dashboard.

Chapter 5: Testing, Results, and Discussion

This chapter reports the end-to-end functional verification of the system, the baseline-teaching and classification results, the continuous-operation and consistency evaluation between the local HMI and the web dashboard, the motor-testing arrangement, and a discussion of the findings and their limitations.

Chapter 6: Conclusion and Recommendations

This chapter draws the conclusions of the work against the objectives stated in Section 1.3, states plainly what the measured evidence does and does not establish, and sets out the recommendations that follow from the limitations identified in Sections 1.4 and 5.8.

Chapter 2
Literature Review

2.1 Overview

Electric motors are widely used in industrial, commercial, and domestic applications due to their reliability and efficiency. However, mechanical faults such as misalignment, unbalance, looseness, and bearing failures remain common issues that can lead to performance degradation or motor breakdown (Zakharov et al., 2018). As a result, condition monitoring has emerged as a crucial aspect of modern maintenance strategies.

Condition monitoring (CM) involves tracking motor health parameters in real-time to detect early signs of failure. Among the various CM techniques, vibration analysis is considered one of the most effective methods for detecting mechanical anomalies. According to Siddique et al. (2003), vibration-based diagnostics can provide early warnings for a wide range of faults before they become critical, allowing for proactive maintenance and increased machine uptime.

In the past decade, technological advances have introduced smart sensors and embedded systems capable of acquiring and analyzing vibration signals (Mohd Ghazali et al., 2021). Vibration sensors like the SICK-MPB10 offer high-resolution multi-axis measurement and are suitable for deployment in harsh environments, making them ideal for industrial motor monitoring applications.

Additionally, the emergence of the Internet of Things (IoT) has revolutionized how condition monitoring data is managed. IoT protocols such as MQTT (Message Queuing Telemetry Transport) enable lightweight and efficient transmission of data to cloud servers, allowing for remote monitoring, real-time alerts, and predictive maintenance analytics. These developments align with Industry 4.0, which advocates for smart, connected, and autonomous systems (Hassanalieragh et al., 2015).

In fault diagnosis, Sahu et al. (2020) state that "motor problems can be identified and diagnosed using the time and frequency domain features extracted from the vibration signals", highlighting vibration as a key indicator of mechanical and electrical faults. Furthermore, Setiawan et al. (2023) emphasize that "a hybrid strategy that combines temperature analysis and vibration provides a reliable way to identify BLDC motor problems early on", confirming that integrating these parameters leads to higher fault detection accuracy and supports predictive maintenance strategies.
Example of "risk" of unintentional plagiarism

Several researchers and companies have explored vibration-based motor monitoring systems integrated with microcontrollers and communication modules:

- **Siddique, et al. (2003)** emphasized the effectiveness of vibration analysis in detecting motor faults and presented the foundation for predictive maintenance based on condition data.
- **Kumari et al. (2021)** developed a motor vibration monitoring system using an accelerometer and Arduino, displaying data via LCD and transmitting it to a PC.

However, the system lacked cloud integration and IoT connectivity.

- **Islam et al. (2024)** proposed a smart monitoring system using ESP8266 for temperature and vibration sensing in motors with a cloud dashboard. While it showed IoT functionality, it didn't offer a local visual interface for on-site diagnostics.
- **Siemens, ABB, and Schneider Electric** have developed industrial-grade monitoring systems, such as Motor Condition Monitoring Solutions (CMS), but these are often complex, expensive, and require high-level integration with proprietary systems.

These studies and commercial products highlight the progression from offline diagnostic methods to semi-smart and fully connected systems, but there is still room for improvement in terms of affordability, scalability, and user-friendliness.

2.2 Critique of the review

The reviewed literature confirms that vibration analysis, when combined with embedded and IoT technologies, offers a viable path for real-time electric motor condition monitoring. The existing systems, while effective, often present trade-offs:

- **Lack of on-site visual diagnostics** in low-cost systems.
- **High cost and complexity** in industrial-grade solutions.
- **Limited adaptability** for small or medium-sized enterprises.

The proposed project aims to fill these gaps by integrating three essential features:

- **Reliable vibration sensing** using industrial-grade sensors.
- **Real-time local display** via an HMI for quick diagnostics.
- **IoT-based data transmission** for cloud analytics and remote monitoring.

This integration enables both field-level technicians and remote supervisors to make timely, informed decisions regarding motor maintenance.

Existing research has laid a strong foundation for motor condition monitoring, several limitations remain in terms of usability, accessibility, and modularity:

- Most systems are either local-only or cloud-only, lacking a hybrid architecture that supports both.
- Commercial solutions are cost-prohibitive for small industries or educational use.
- Few implementations offer flexible sensor integration with open-source development environments.

The proposed thesis contributes to the field by:

- Combining local HMI and IoT connectivity into a single, unified solution.
- Using open-source hardware and lightweight protocols (e.g., MQTT) for flexibility and scalability.
- Providing a low-cost yet reliable alternative to high-end commercial motor monitoring systems.

By addressing these gaps, the project not only advances the technical field but also aligns with the practical needs of industries aiming to adopt smart maintenance systems under Industry 4.0.

2.3 Feature Comparison with Existing Projects

To clarify the position and advantages of the proposed system design, the following presents a comparison of the main features between this research and several similar projects that have been previously developed, covering aspects of sensors, interfaces, communications, analytics, system resilience, industrial integration, and the standards applied, as summarized in Table 2.1:

Table 1.1 Feature Comparison with Existing Projects

Aspect	MPB10 + ESP32/LVGL + MQTT + Server	ESP32 + PZEM/ADXL + MQTT	IJARSCT 2025 (ESP32, multi-sensor)	iCAST-ES 2023 (ESP32 + WTVB01 + MQTT)
Core Sensor	SICK MPB10 (3-axis vibration, v-RMS, a-RMS, kurtosis/crest factor, shock, temperature)	ADXL345 (vibration), DS18B20, PZEM	MPU-9250 (vibration), PT1000, ACS712, ZMPT101B, encoder	WTVB01 (multi-axis vibration RS-485), temp
Sensor Grade	Industrial grade, IP68, IO-Link v1.1, 5 ms	Hobby/low-cost	Hobby/research	Semi-industrial (WTVB01 IP67)
Local UI	Interactive LVGL display (graph, status, menu)	LCD 16×2 (very limited)	PC/web UI (no embedded HMI)	E-ink local + web dashboard
Communication	IO-Link → SIG200 (REST API) → ESP32-S3; MQTT/HTTPS to the cloud dashboard through a WISP-mode wireless link.	MQTT (Pub/Sub)	Wi-Fi (to PC/web), partial control	MQTT (Mosquitto), Modbus RS-485
Analytics	Edge basic (threshold/ISO 10816-3) + server (trending/alerts); potential ML	Fuzzy logic (hobby grade)	Filter/FFT in ESP32; analysis on PC	Threshold, alarms; ISO 10816-1 reference
Offline Independence	High: data/status still visible on LVGL screen even if broker is down	Low (small LCD; relies on broker)	Medium (PC/dashboard)	Medium–High (handheld mode; local broker)
Industrial Integration	IO-Link compatible via SICK SIG200 IO-Link master, exposing sensor data through a	No	No	Modbus RTU (common in industry)

	REST API over Ethernet to the ESP32-S3.			
Standard/ Reference	ISO 10816-3 via MPB10 indicators; shock temp	None specified	General FFT	ISO 10816-1 as threshold

2.4 Summary

This chapter reviews the development and current state of electric motor condition monitoring systems, emphasizing the importance of vibration analysis as a key technique for detecting early mechanical faults. Research shows that real-time condition monitoring using smart sensors and embedded systems can significantly reduce downtime and improve maintenance planning. Recent technological advancements, including IoT integration and MQTT communication protocols, enable efficient remote monitoring and predictive analytics, aligning with Industry 4.0 principles. However, existing systems often lack a combination of local visual diagnostics and cloud-based access, or they are too expensive and complex for smaller industrial users.

The proposed system aims to overcome these challenges by offering a hybrid solution that integrates:

- Industrial-grade vibration sensors,
- A local HMI display, and
- IoT connectivity for remote monitoring.

Compared to existing literature and commercially available products, the proposed work distinguishes itself by offering a cost-effective, modular, and scalable system that addresses many of the limitations found in current condition monitoring solutions. Its design prioritizes affordability without compromising on performance, making it an attractive option for a broader spectrum of users, particularly targeting small-to-medium enterprises that often find sophisticated systems financially and operationally inaccessible. By leveraging modular components, the system allows for easy customization and future upgrades, ensuring flexibility to adapt to varying user requirements and evolving industrial standards. Furthermore, its scalable architecture enables seamless expansion from basic monitoring setups to more comprehensive and integrated smart maintenance solutions, aligning with the dynamic needs of modern industry.

This inclusive design approach not only democratizes access to advanced condition monitoring technologies but also supports the strategic goals of Industry 4.0 by facilitating smarter, data-driven decision-making processes. By integrating robust communication protocols and combining local and remote diagnostic capabilities, the system enhances operational efficiency, minimizes downtime, and optimizes maintenance schedules. In doing so, it contributes to a smarter, more connected industrial ecosystem, driving innovation while supporting sustainable and cost-efficient maintenance practices. Ultimately, this work bridges the gap between cutting-edge technology and practical usability, paving the way for widespread adoption of smart maintenance systems across diverse industrial environments.

Chapter 3
Methodology

3.1 Research Design

This study adopts a Research and Development (R&D) approach, combining embedded systems design, real-time communication protocols, and web-based dashboard development. The goal is to build an integrated IoT system capable of monitoring critical electric motor condition parameters using the SICK-MPB10 vibration sensor, capturing motor health data and visualizing them both locally via a display and remotely through a web interface.

3.2 System Overview

The proposed system consists of:

- **Sensor Layer:** The SICK MPB10 condition-monitoring sensor captures the machine vibration and reports the resultant velocity-RMS magnitude together with a temperature reading over IO-Link.
- **Gateway Layer:** A SICK SIG200 sensor-integration gateway acts as the IO-Link master and exposes the sensor process data through its REST API on the local network.
- **Controller Layer:** An ESP32-S3 microcontroller retrieves the process data through periodic REST (HTTP) requests, performs threshold-based classification, controls the Human–Machine Interface (HMI), and manages IoT communication.
- **Display Layer:** A 7-inch LVGL-based HMI provides real-time on-site visualization of the motor condition.
- **Communication Layer:** Wi-Fi connectivity through a WISP-mode wireless router enables data transmission to the cloud.
- **Dashboard Layer:** A web-based dashboard developed in HTML/JavaScript, backed by a cloud database and deployed to public hosting, displays real-time and accumulated motor-condition data for remote monitoring and hosts the remote teaching controls.

The overall arrangement of these six layers is illustrated in Figure 3.1.

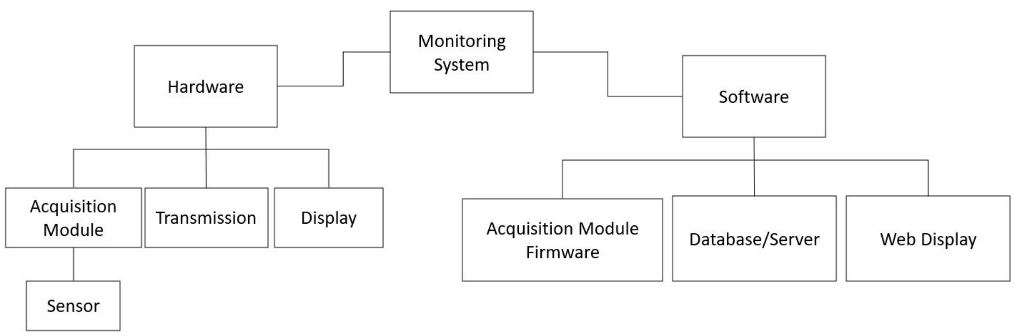

Figure 3.1 Feature Comparison with Existing Projects

3.3 System Architecture

The architecture of this system includes:

- **Input:** SICK MPB10 vibration sensor providing the resultant vibration magnitude (v-RMS) indicative of imbalance, misalignment, or bearing issues, together with an integrated temperature reading as a secondary health indicator; both quantities are exposed by the SIG200 gateway through its REST API.

- **Processing:** ESP32-S3 microcontroller acquires sensor data, conducts preliminary processing, manages local display updates, and handles MQTT communication.

- **Local Visualization:** Real-time motor health metrics are displayed via a 7-inch touchscreen using the LVGL graphics library.

- **Communication Protocol:** MQTT over Wi-Fi ensures low-latency, efficient data exchange between the monitoring device and the remote server.

The resulting system architecture and network topology are illustrated in Figure 3.2 and Figure 3.3, respectively.

Figure 3.2 System Architecture

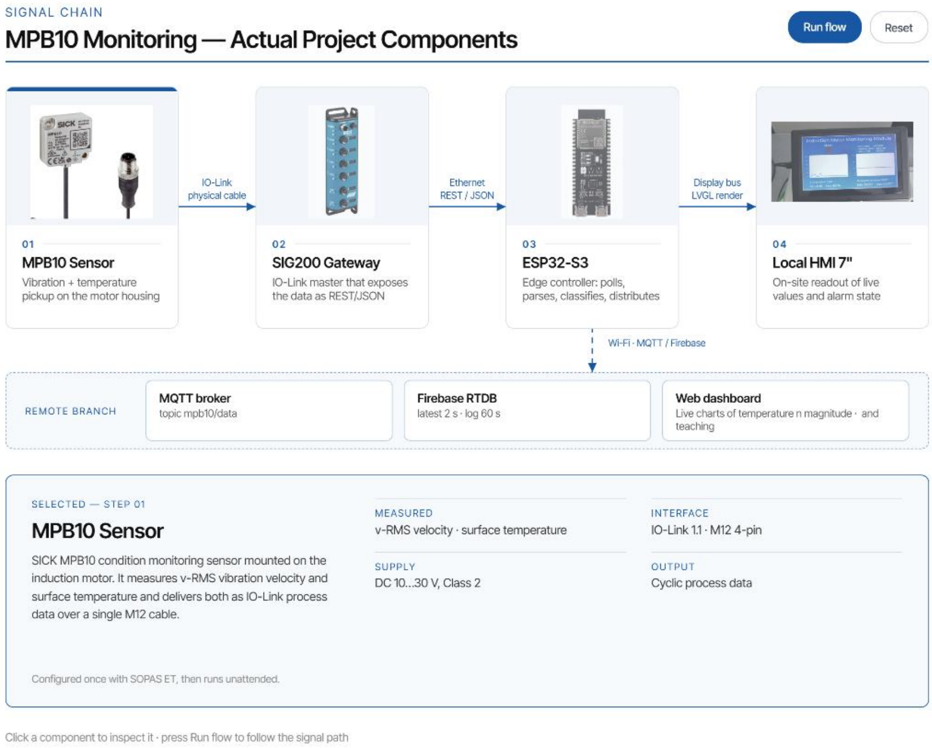

Figure 3.3 System Topology

3.4 Hardware Components

- **SICK MPB10 Condition-Monitoring Sensor:**
An industrial-grade MEMS-based sensor that reports the resultant vibration magnitude (v-RMS) and temperature; it communicates over IO-Link and is connected through a shielded M12 cable.

- **SICK SIG200 Sensor-Integration Gateway:**
The IO-Link master that parameterizes the sensor (via its IODD) and exposes the process data through a REST API over Ethernet.

- **ESP32-S3 Microcontroller with 7-inch RGB Display:**
A dual-core controller with integrated Wi-Fi and on-board PSRAM that acquires the data via REST, executes the monitoring logic, and renders the LVGL HMI on the attached 7-inch panel.

- **Panel Assembly:**
An industrial enclosure with a DIN rail carrying a 24 V DC switching power supply and a miniature circuit breaker, housing the gateway and power distribution as a single deployable module.

- **Wireless Router (WISP mode):**
Provides the local subnet for the monitoring devices and the wireless uplink to the internet.

3.5 Software Development

- **Embedded Firmware:**

Developed using the Arduino core for the ESP32-S3. Responsible for periodic acquisition of the vibration magnitude and temperature from the SIG200 REST API (500 ms interval), automatic threshold teaching with persistent (NVS/flash) storage of the learned limits, real-time condition classification with a latched alarm, updating the local HMI, and publishing JSON-formatted data to the cloud.

- **HMI Interface with LVGL**:

Designed with EEZ Studio and rendered by the LVGL graphics library; displays real-time readings, trend charts, running min/max records, the learned limits, and an alarm indicator.

- **MQTT Communication:**

Implements MQTT v3.1 or v5 for TCP/IP-based data transfer over Wi-Fi, ensuring reliable remote data publishing and subscription.

3.6 Web Dashboard Development

- **Purpose:**

To allow remote users to monitor electric motor health by accessing real-time and historical data via an interactive dashboard.

- **Technology Stack Options:**

A. JavaScript Stack:

Backend with Node.js using MQTT.js client; frontend employs HTML/CSS with Chart.js for visualization; real-time updates via WebSocket or MQTT-over-WebSocket.

Advantages*:* Lightweight, cross-platform, suitable for cloud deployment.

B. C# Stack:

ASP.NET Core MVC backend with MQTTnet library; frontend using Razor pages or Blazor with LiveCharts or Syncfusion for charting; real-time via SignalR.

Advantages*:* Strong typing, Windows server compatibility, robust architecture.

- **Dashboard Features:**

Real-time vibration data graphs, status indicators with warning levels, historical trends visualization, CSV data export, and responsive design compatible with mobile and tablet devices.

Based on these considerations, the implemented system adopts the JavaScript approach in a serverless form: a single-page HTML/CSS/JavaScript dashboard with Chart.js for visualization, communicating directly with Cloud Firestore through its REST interface for the latest data, statistics, and remote commands (teach, alarm reset, and parameter reset). The dashboard is deployed to a public static-hosting service (Netlify), making it reachable over HTTPS from any browser without a dedicated backend server.

3.7 Data Flow

- The SICK MPB10 sensor reports the resultant vibration magnitude and temperature to the SIG200 gateway over IO-Link.
- The ESP32-S3 polls the SIG200 REST API every 500 ms, classifies each magnitude sample

against the learned limits, and updates the local HMI in real time.

- At a longer interval (approximately every 10 seconds), the controller writes the latest values, limits, and condition status to a single cloud document that is repeatedly overwritten, and the same payload can be published over MQTT for streaming consumers.
- The web dashboard polls only this latest document (one read per device update), accumulates the received values in browser memory for its charts, log, and statistics, and applies a staleness check that reports the sensor as offline and stops polling when no fresh data arrives.

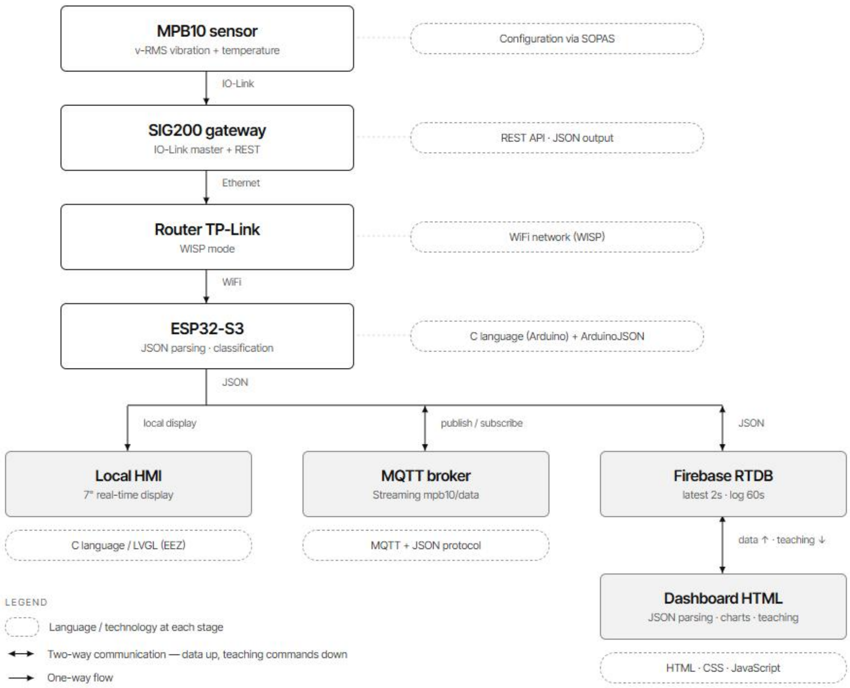

Figure 3.4 Data flow of the proposed system

3.8 Testing and Validation

- **Unit Testing:** Verify the individual modules — REST-based sensor acquisition, threshold teaching and classification, NVS persistence, Wi-Fi/cloud communication, and the LVGL GUI.
- **Functional (System) Verification:** Exercise the complete chain end-to-end under controlled, manually induced excitation to verify teaching, limit derivation, state classification, alarm latching, and HMI–dashboard consistency.
- **Motor Testing:** Apply the verified system to the induction-motor spindle setup, re-executing the teaching procedure with the sensor mounted on the running motor.
- **Continuous-Operation Testing:** Confirm stability over long-duration operation, including sustained cloud updates within the service quota and correct offline detection on the dashboard.

3.9. Work Breakdown Structure (WBS)

The Work Breakdown Structure (WBS) of this system divides the overall project into smaller, manageable components called work packages. This hierarchical approach helps organize and control tasks to ensure successful project completion.

Literature Review and Requirements Analysis
Conduct a comprehensive literature review on:
- IoT-based electric motor monitoring systems.
- Features and specifications of SICK-MPB10 sensor.
- Human-Machine Interface (HMI) technologies using LVGL.

System Design
- Design the overall system architecture including sensor integration, microcontroller control, HMI interface, IoT communication, and remote dashboard.
- Design electronic circuitry for connecting SICK-MPB10 sensor with ESP32-S3 microcontroller.
- Design the HMI layout using LVGL on a 7-inch display.

Hardware Procurement and Preparation
- Procure and test the SICK-MPB10 vibration sensor.
- Acquire ESP32-S3 microcontroller and supporting hardware components (display unit, Wi-Fi modules, power supply).

Embedded Software Development
Implement ESP32-S3 firmware for:
- Periodic vibration data acquisition from the SICK-MPB10 sensor.
- Data processing and filtering algorithms for raw sensor data.
- Real-time display updates on LVGL-based HMI.

Web Dashboard Development
- Develop backend MQTT client and data handling services using Node.js or ASP.NET Core.
- Create frontend dashboard with real-time and historical vibration data visualization using Chart.js or LiveCharts.
- Implement alerting, notification systems, and data export features (e.g., CSV download).

System Integration and Functional Testing
- Integrate hardware and software components into a working prototype.
- Conduct end-to-end testing of data flow from sensor to dashboard.
- Calibrate sensor readings and validate accuracy against reference devices.

System Evaluation and Analysis
- Evaluate system performance for detecting common motor faults (e.g., imbalance, bearing degradation, overheating).
- Analyze real-time system reliability, data latency, and fault detection sensitivity.

Documentation and Final Report
- Prepare technical documentation covering hardware design, embedded firmware, communication protocols, and user manuals.
- Write the final report including literature review, methodology, experimental results, analysis, conclusions, and future work recommendations.

Presentation and Demonstration

- Prepare presentation materials for seminars or thesis defense.
- Conduct live demonstrations highlighting local display functionality and IoT remote monitoring capabilities.

Table 3.1 Data flow of the proposed system

No.	Work Package	Description	Duration	Time Frame
1	Literature Review & Requirements Analysis	Review studies, sensor specs, HMI tech, MQTT protocols, try to identify parameters and system needs	2 weeks	Week 1 – Week 2
2	System Design	Design overall system architecture, all electronic circuit for sensor integration, HMI UI, MQTT format	2 weeks	Week 3 – Week 4
3	Hardware Procurement & Preparation	Purchase and test SICK-MPB10 sensor, ESP32-S3, display, and any other peripherals	2 weeks	Week 5 – Week 6
4	Embedded Software Development	Develop firmware for data reading, processing, MQTT communication, and HMI interface	4 weeks	Week 7 – Week 10
5	Web Dashboard Development	Backend or frontend development for real-time data display, alerts, and responsive dashboard	3 weeks	Week 11 – Week 13
6	System Integration & Testing	Integrate all components, conduct functional and stress testing, ensure stability	3 weeks	Week 14 – Week 16

7	System Evaluation & Analysis	Evaluate all of the system performance, fault detection, and user experience	2 weeks	Week 17 – Week 18
8	Documentation & Final Report	Write technical documents, manuals, and the final thesis or project report	2 weeks	Week 19 – Week 20
9	Presentation & Demonstration	Prepare all of presentation materials and conduct live system demo	1 week	Week 21

Chapter 4 System Design and Implementation

4.1 Design Overview and Functional Layers

This chapter describes the detailed design and implementation of the proposed integrated induction-motor condition monitoring system. Following the methodology established in Chapter 3, the system is realized as four cooperating functional layers: sensing, processing, visualization, and communication. At the sensing layer, a SICK MPB10 condition-monitoring sensor is mounted on the induction motor and configured to output the resultant vibration magnitude directly as a single scalar value, together with a temperature reading as a secondary health indicator. In contrast to an earlier design iteration, in which the raw per-axis acceleration components (X, Y, and Z) were transmitted and recombined into a resultant magnitude on the controller, the direct-magnitude configuration reduces controller-side computation, lowers the transmitted data volume, and improves system responsiveness.

The MPB10 is connected to a SICK SIG200 sensor-integration gateway through IO-Link, and the SIG200 exposes the magnitude and temperature values through its REST API. The processing layer is implemented on an ESP32-S3 microcontroller, which retrieves the process data through periodic REST (HTTP) requests, performs threshold-based classification, drives the local display, and manages IoT publishing. The visualization layer is split into two complementary parts: an LVGL-based Human–Machine Interface (HMI) rendered by the ESP32-S3, which serves purely as a local monitor for on-site personnel, and an HTML web dashboard that provides global monitoring and hosts the baseline-teaching feature. Finally, the communication layer transmits the processed magnitude and status using the MQTT protocol over Wi-Fi through a wireless router operating in WISP mode.

4.2 System Architecture

The architecture follows a layered design in which each layer has a clearly delimited responsibility. At the sensing layer, the MPB10 captures the vibration of the motor and reports the resultant magnitude, which is made available by the SIG200 gateway through its REST API. At the processing layer, the ESP32-S3 retrieves the magnitude via HTTP requests, executes ISO-based threshold classification, and coordinates the local and remote outputs. At the visualization layer, the LVGL HMI presents the live magnitude and condition state directly at the machine, while the HTML web dashboard provides global visualization and the teaching interface. At the communication layer, the ESP32-S3 publishes the data via MQTT over the WISP-mode wireless link to the broker and the web dashboard, enabling real-time streaming, historical storage, and remote access. The complete architecture and data flow are shown in Figure 4.1.

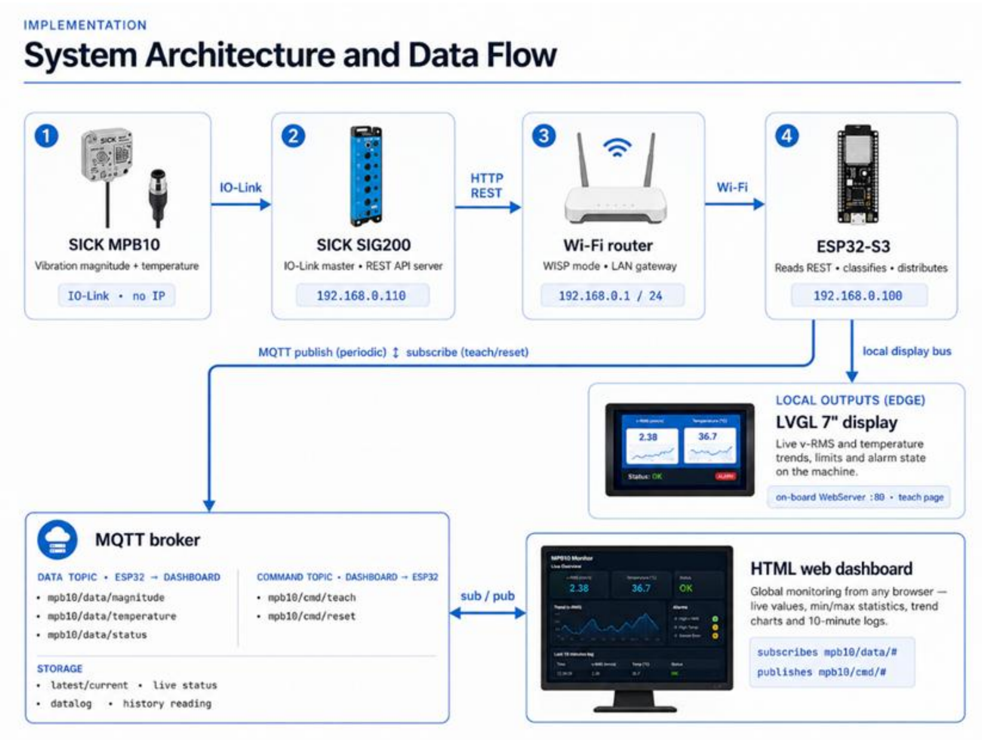

Figure 4.1 Architecture and data flow of the proposed system.

4.3 Network Topology and Configuration

The network topology describes how the devices are interconnected. The MPB10 communicates with the SIG200 over IO-Link, and the SIG200 makes the process data available through its REST API on the local network. The ESP32-S3 obtains the magnitude via HTTP requests and connects wirelessly to a router operating in WISP mode, which acts as a wireless client to an upstream access point while providing an isolated local subnet. A key property of this arrangement is that the local subnet remains stable regardless of the upstream internet source: the addresses of the SIG200, the ESP32-S3, and the gateway are fixed within the router’s LAN, so the monitoring system continues to operate consistently even if the upstream connection changes.

Within this subnet, the ESP32-S3 is assigned a static IP address of

192.168.0.100 with the router as the gateway at 192.168.0.1, and the SIG200 is configured at 192.168.0.110. Because a static IP configuration bypasses automatic DNS assignment, public DNS servers are configured explicitly in the firmware so that cloud hostnames can still be resolved. The resulting network topology is illustrated in Figure 4.2, and the addressing and protocol assignments are summarized in Table 4.1.

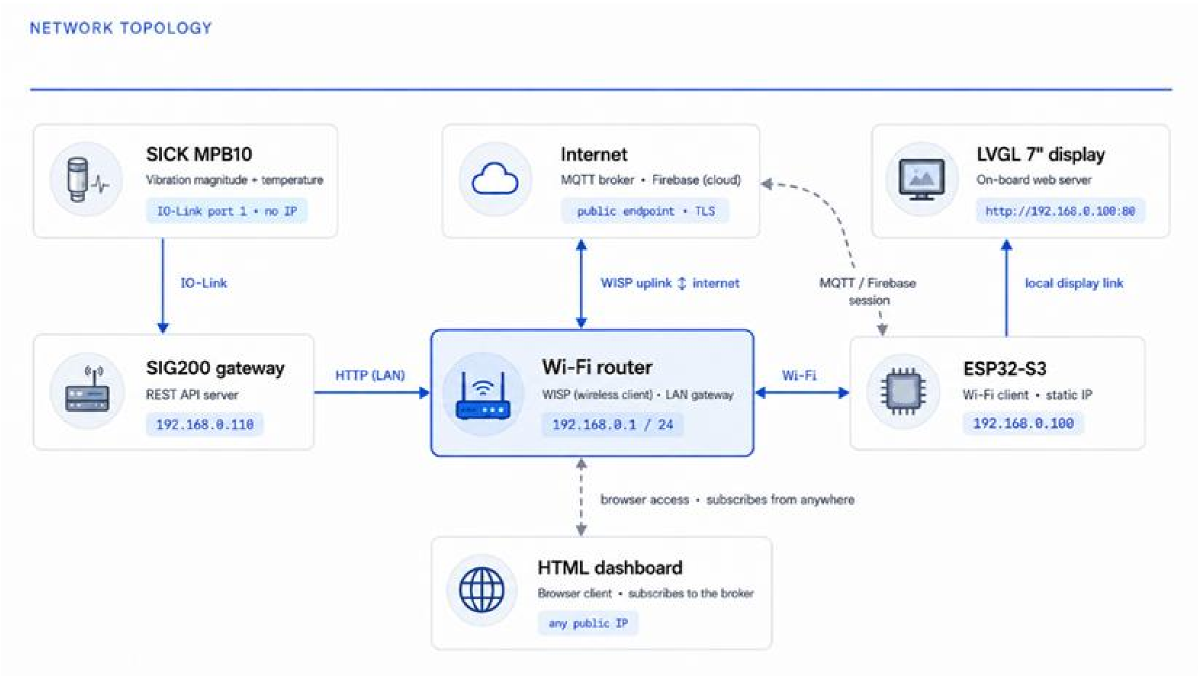

Figure 4.2 System (network) topology with IP addressing and the WISP-mode path

Table 4.1 Network configuration and protocol assignment.

Device / Segment	Parameter	Value
MPB10 ↔ SIG200	Sensor interface	IO-Link
SIG200 ↔ ESP32-S3	Gateway interface	REST API (HTTP)
SIG200	IP address	192.168.0.110
ESP32-S3	Static IP address	192.168.0.100
ESP32-S3	Subnet mask	255.255.255.0
ESP32-S3	Gateway	192.168.0.1
Wireless router	Operating mode	WISP (wireless client)
Wireless router	LAN gateway IP	192.168.0.1
ESP32-S3 ↔ broker	IoT protocol / port	MQTT / 1883

4.4 Hardware Implementation

From the hardware perspective, the system integrates the SICK MPB10 condition-monitoring sensor, the SICK SIG200 sensor-integration gateway, and an

ESP32-S3 microcontroller with an attached 7-inch RGB display panel used as the local HMI. The MPB10 is mounted on the induction motor and configured to report the resultant vibration magnitude. The SIG200 acts as the IO-Link master and exposes the sensor data through its REST API server. The ESP32-S3, equipped with 8 MB of PSRAM required for the display frame buffer, retrieves the magnitude via HTTP requests, executes the monitoring logic, drives the display, and connects to the WISP-mode router for IoT communication. The main hardware components are listed in Table 4.2, and the hardware topology is shown in Figure 4.3.

Table 4.2 Main hardware components.

Component	Function	Interface
SICK MPB10	Vibration condition-monitoring sensor; outputs resultant magnitude and temperature	IO-Link
SICK SIG200	Sensor-integration gateway (IO-Link master / REST API server)	IO-Link ↔ REST API (Ethernet)
ESP32-S3	Controller: acquisition, threshold logic, local HMI, MQTT	REST API (HTTP) / Wi-Fi
7-inch display panel	Local monitor rendered with LVGL	RGB parallel, driven by ESP32-S3
Wireless router	Network uplink to broker/cloud	Wi-Fi (WISP mode)
Induction motor	Monitored asset (vibration source)	—

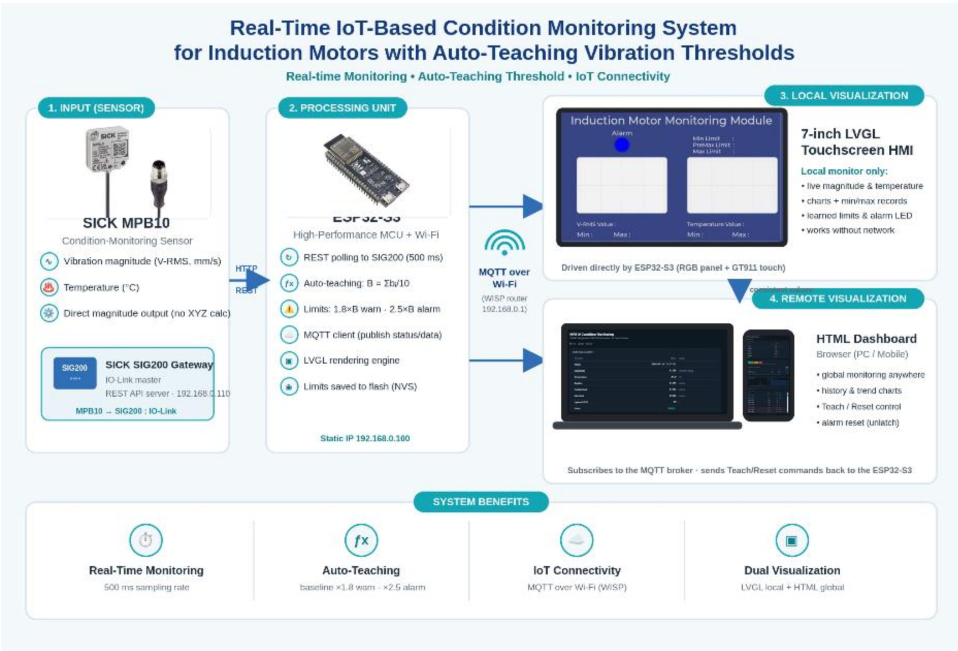

Figure 4.3 Hardware topology of the monitoring system.

4.5 Sensor Configuration and Data Acquisition

The MPB10 is parameterized through the SIG200 gateway using the SOPAS engineering environment. The sensor's IO Device Description (IODD) file is first uploaded to the gateway and assigned to the corresponding IO-Link port, which allows the gateway to interpret the sensor's process data with meaningful labels instead of raw bytes. The process-data output of the MPB10 is then configured so that the first value field carries the resultant velocity-RMS magnitude (v-RMS) and the second value field carries the temperature. Restricting the payload to these two quantities keeps the acquisition path lean and reduces both the controller-side parsing effort and the volume of data forwarded to the cloud.

For acquisition, the port ownership on the SIG200 is assigned to the REST interface, and the ESP32-S3 issues a periodic HTTP POST request to the gateway's readDevice endpoint every 500 ms. The gateway responds with the parsed process data in JSON format, from which the firmware extracts the magnitude and temperature fields. The 500 ms polling interval is chosen as a balance between responsiveness of the condition assessment and the processing load on the controller, which must concurrently render the local HMI.

4.6 Firmware Implementation on the ESP32-S3

The embedded firmware is developed with the Arduino core for the ESP32-S3 and is organized around a non-blocking main loop that schedules four recurring activities: (i) magnitude and temperature acquisition from the SIG200 every 500 ms; (ii) threshold-based classification of each new magnitude sample; (iii) rendering of the LVGL HMI; and (iv) IoT publishing. The classification result drives the HMI state, an alarm LED indicator, and the status field transmitted to the dashboard.

Two implementation aspects are essential for robust field operation. First, the learned baseline and derived limits are stored in the ESP32-S3 non-volatile storage (NVS/flash) immediately after a successful teaching cycle; at start-up the firmware reloads these values automatically, so the monitoring configuration survives power cycling and unexpected resets without requiring the operator to repeat the teaching procedure. Second, the alarm state is implemented as a latched condition: once the magnitude has exceeded the maximum limit, the alarm remains active even if the vibration subsequently decreases, and it can only be cleared through an explicit reset command. This latching behavior follows common industrial-alarm practice and guarantees that transient fault events are not silently missed by the operator.

For IoT communication, the firmware formats the magnitude, temperature, learned limits, alarm flag, and condition status as a lightweight JSON payload and publishes it over MQTT to a predefined topic. In parallel, the latest record is written to a single cloud document that is repeatedly overwritten rather than appended, a quota-efficient strategy discussed further in Section 4.9.

4.7 Automatic Threshold Teaching Implementation

To avoid manual configuration and to adapt the limits to each specific motor installation, the system implements an automatic threshold-teaching procedure based on the vibration magnitude reported by the MPB10. Teaching is triggered from the HTML web dashboard (or the on-board local web page) while the motor operates in a known-healthy condition. During this phase the firmware records ten consecutive magnitude samples b_1, b_2, …, b_{10} at the 500 ms sampling interval, so that one teaching cycle takes approximately five seconds. The healthy baseline B is obtained as the arithmetic mean of these samples, as expressed in Equation (4.1) and, in general form, in Equation (4.2):

$$B = (b_1 + b_2 + b_3 + \cdots + b_{10}) / 10 \quad (4.1)$$

$$B = (1/N) \sum b_i\,, \;\; i = 1 \ldots N\,, \;\; N = 10 \quad (4.2)$$

From the learned baseline B, the operating limits are derived as fixed multiples of the baseline, anchored to the ISO vibration-severity evaluation zones (ISO 10816 / ISO 20816 define severity zones A–D for rotating machinery, where zones A/B denote good-to-acceptable operation, zone C denotes an unsatisfactory condition suitable for a warning, and zone D denotes an unacceptable condition suitable for an alarm). A pre-warning limit and an alarm limit are computed from B, and an additional off-threshold is used to detect a stopped motor, as given in Equations (4.3)–(4.5):

$$L\_off = 0.4 \times B \quad (motor\text{-}off\ detection) \quad (4.3)$$

$$L\_pre = 1.8 \times B \quad (pre\text{-}warning) \quad (4.4)$$

$$L\_max = 2.5 \times B \quad (alarm) \quad (4.5)$$

The scaling factors are fixed in firmware (OFF_RATIO = 0.4, FACTOR_WARNING = 1.8, FACTOR_ALARM = 2.5) and are chosen so that the resulting limits approximate the ISO severity-zone boundaries for the applicable motor class; they can be re-tuned for other motor ratings or mounting classes. Once the limits are established, each measured magnitude M is classified against them in real time, as summarized in Table 4.3. Before the first teaching has been performed, no limits exist and the system reports a not-taught state in which no alarm can be triggered.

Table 4.3 Condition classification rule based on the measured magnitude M.

Condition	**Rule**
OFF (motor stopped)	$M < L\_off$ (= 0.4 × B)
NORMAL	$L\_off \leq M < L\_pre$
WARNING (pre-limit exceeded)	$L\_pre \leq M < L\_max$
ALARM (latched until reset)	$M \geq L\_max$ (= 2.5 × B)

A worked example of the teaching procedure is given in Table 4.4 using the ten healthy-state samples recorded on the 7.5 kW three-phase test motor. The mean of the ten samples yields a baseline of B = 28.66 mm/s; applying the firmware factors gives L_off = 11.46 mm/s, L_pre = 51.58 mm/s, and L_max = 71.64 mm/s. Table 4.5 then illustrates how representative magnitude readings are evaluated against these derived limits. The complete teaching procedure and the resulting classification bands are shown in Figure 4.4 and Figure 4.5, respectively.

Table 4.4 Baseline teaching samples recorded on the 7.5 kW three-phase test motor.

Sample	**Magnitude (mm/s)**	**Note**
b1	28.40	Threshold reference
b2	28.90	Threshold reference
b3	28.50	Threshold reference
b4	29.00	Threshold reference
b5	28.60	Threshold reference
b6	28.80	Threshold reference
b7	28.40	Threshold reference
b8	28.90	Threshold reference
b9	28.70	Threshold reference
b10	28.40	Threshold reference
Baseline B (mean)	28.66	$= \sum b_i / 10$

Table 4.5 Worked example of the condition-evaluation rule using the learned limits of the reference compressor. The magnitudes in this table are illustrative inputs chosen to exercise every band; all measured results are reported in Chapter 5.

Actual magnitude M (mm/s)	**Compared with limits**	**Classified condition**

Actual magnitude M (mm/s)	Compared with limits	Classified condition
0.20	M < 11.46	OFF (motor stopped)
28.66	11.46 ≤ M < 51.58	NORMAL
37.45	11.46 ≤ M < 51.58	NORMAL
55.20	51.58 ≤ M < 71.64	WARNING
72.50	M ≥ 71.64	ALARM (latched)

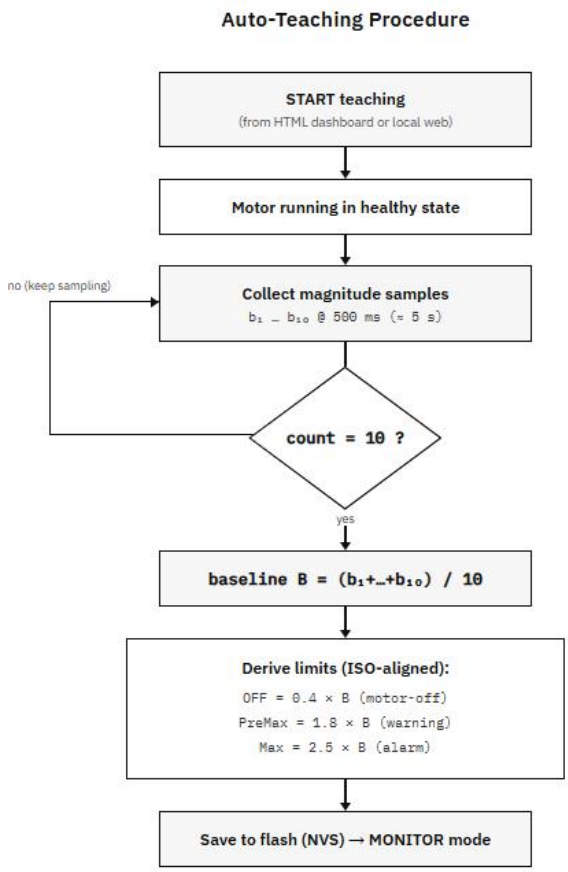

Figure 4.4 Auto-teaching procedure executed on the ESP32-S3.

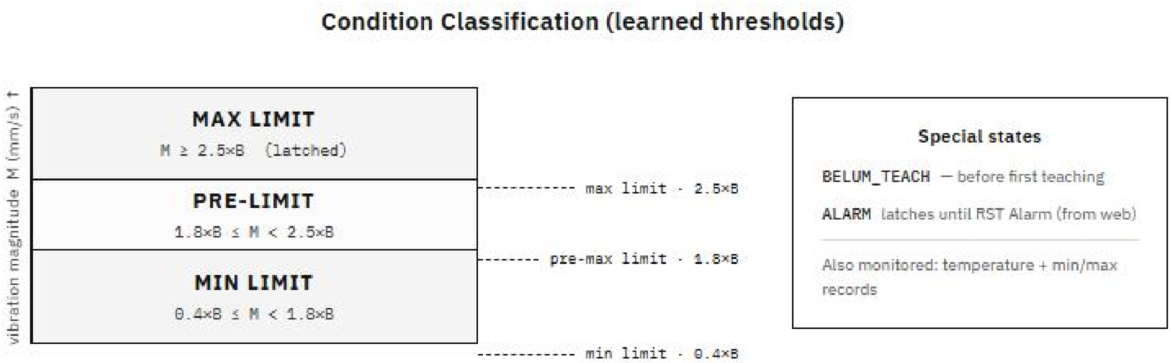

Figure 4.5 Condition classification bands derived from the learned baseline.

4.8 Local HMI Implementation (LVGL)

The local HMI is designed with EEZ Studio and rendered by the LVGL graphics library on the 7-inch display attached to the ESP32-S3. The interface serves purely as a local monitor: it displays the live v-RMS magnitude and temperature on meters and trend charts, the running minimum and maximum records, the learned Min/PreMax/Max limits, and an alarm indicator, but it does not host configuration functions. Because the display panel requires a large frame buffer, the buffer is allocated in the on-board PSRAM. A representative layout of the local monitor is shown in Figure 4.6.

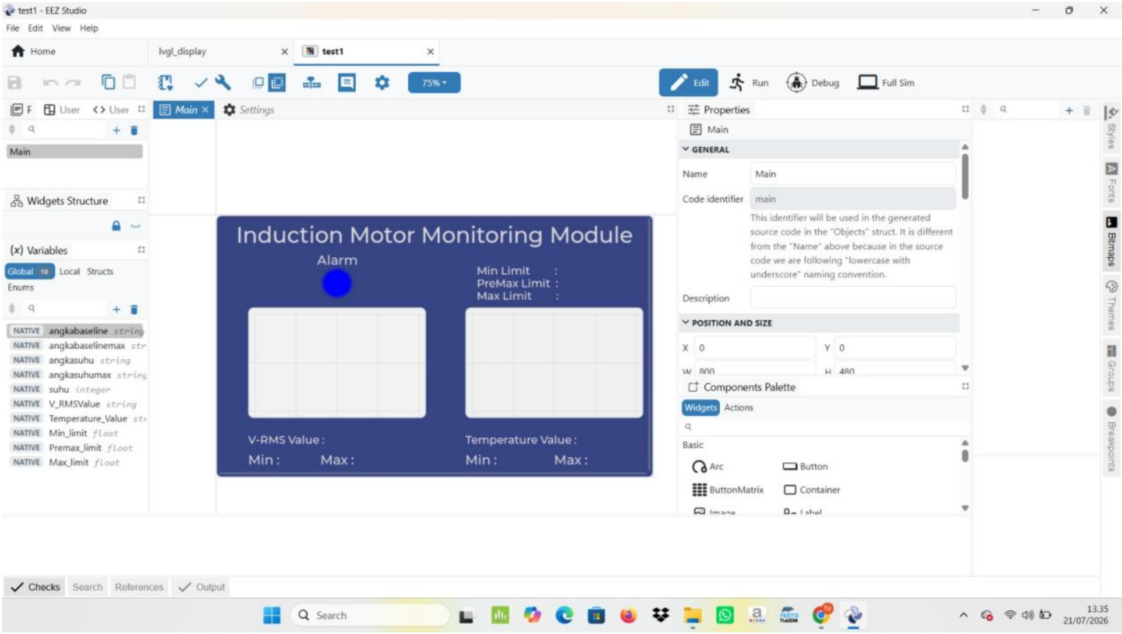

Figure 4.6 LVGL local-monitor layout (EEZ Studio): v-RMS magnitude and temperature meters/charts, min/max records, learned Min/PreMax/Max limits, and an alarm indicator.

4.9 Web Dashboard Implementation

Global monitoring and the teaching feature are implemented in an HTML web dashboard built with HTML, CSS, and JavaScript, with charts rendered by a JavaScript charting library. The dashboard presents the real-time magnitude and temperature, the learned limits, the condition status, accumulated minimum/maximum statistics, and short-term trend charts. Operator controls are provided for triggering the teaching procedure remotely, resetting the latched alarm, and resetting the learned parameters.

Two design decisions improve the robustness and efficiency of the dashboard. First, to conserve the cloud read quota, the dashboard does not query a growing collection of historical log documents; instead, the device overwrites a single "latest" document, and the dashboard polls only this document, accumulating the received values in browser memory to build its trend charts and statistics. With this scheme one device update corresponds to exactly one dashboard read. Second, to prevent misleading "stale" readings when the sensing device is powered off, the dashboard applies a staleness check based on the server-side update timestamp: if the latest document has not changed within a defined time window, the interface reports the sensor as offline and automatically unsubscribes (stops polling), rather than continuing to display outdated values as if they were live. A manual subscribe/unsubscribe control is also provided in the interface.

4.10 Summary

This chapter presented the realized design of the monitoring system: a direct-magnitude sensing chain (MPB10 → SIG200 REST API → ESP32-S3), a non-blocking firmware with automatic ISO-anchored threshold teaching, persistent (NVS) storage of learned limits, a latched alarm mechanism, an LVGL local monitor, and a quota-efficient web dashboard with offline detection, all interconnected through MQTT over a WISP-mode wireless link. The next chapter evaluates this implementation on two operating induction-motor-driven air compressors.

Chapter 5 Testing, Results, and Discussion

5.1 Testing Objectives and Scope

The evaluation verifies the end-to-end behavior of the implemented system on two operating machines. Specifically, the tests confirm: (i) correct magnitude and temperature acquisition from the MPB10 through the SIG200 REST API; (ii) execution of the automatic baseline-teaching procedure from the web dashboard; (iii) derivation of the operating limits according to the ISO-anchored rules of Section 4.7; (iv) correct real-time classification of the operating condition, including the latched alarm behavior; (v) consistency between the local LVGL HMI and the HTML web dashboard; and (vi) the reliability of MQTT-based data transmission over the WISP-mode link. The scope of what these tests can and cannot establish is bounded by Section 1.4: they verify the acquisition, teaching, classification, and communication chain, and they demonstrate deviation-based condition alarming, but they do not constitute a validated fault-detection study, because no controlled fault was injected and no independent ground truth was obtained for either machine.

5.2 Experimental Setup

The experimental evaluation was conducted in two separate campaigns on two industrial air compressors, each driven by an induction motor. The first campaign used a healthy reciprocating compressor rated 10 HP (7.5 kW) with a 520 litre receiver and a 10 bar cut-out setting, supplied from a three-phase 50 Hz industrial mains; this unit provided the reference data set. Its motor operates in an automatic load/unload duty cycle, running until the receiver reaches the cut-out pressure and then stopping until the pressure falls to the cut-in level, which produces the characteristic on/off pattern visible in the recorded magnitude trends. The second campaign used a smaller oil-less reciprocating compressor rated 1.5 HP (1.1 kW) with a 30 litre receiver, supplied from a single-phase mains. This second unit had a known abnormality: a leak in its pneumatic circuit prevented the receiver from ever reaching the cut-out pressure, so the motor was unable to unload and ran continuously without its normal rest intervals. This condition is documented from the operating behaviour of the machine and is the only ground truth available for the abnormal session; the internal mechanical state of the motor was not inspected. In both campaigns the MPB10 sensor is mounted on the compressor motor housing and the complete monitoring chain of Chapter 4 is deployed. The assembled monitoring module used in both campaigns is shown in Figure 5.1, and every test session together with the baseline it produced is recapitulated in Table 5.1.

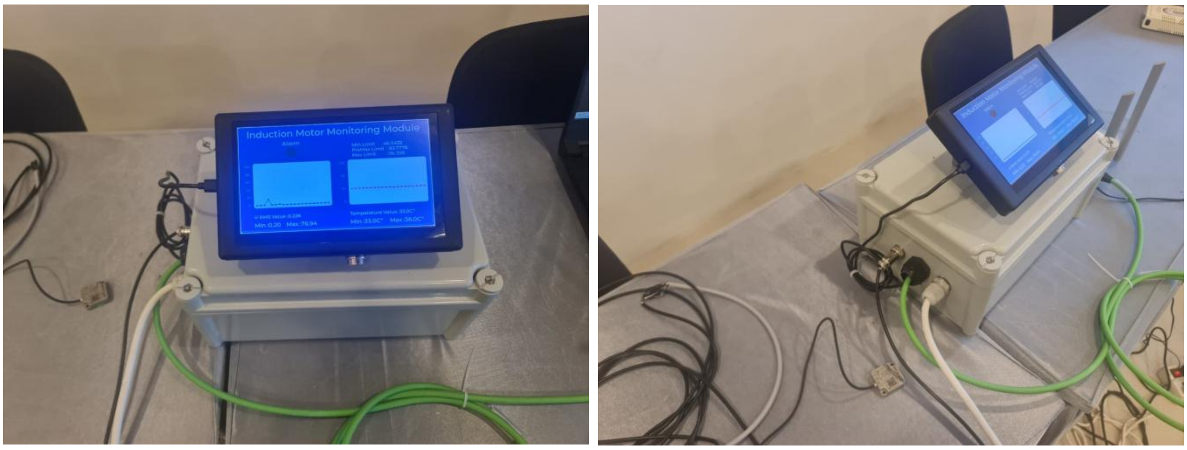

Figure 5.1 The assembled monitoring module: (left) front view of the enclosure with the 7-inch LVGL HMI; (right) the module with the MPB10 sensor and the IO-Link and Ethernet cabling connected. The limits displayed on the screen (B = 46.54 mm/s) belong to the bench commissioning session listed as Session A in Table 5.1 and are not one of the two reported test sessions.

Three teaching sessions were executed in the course of this work, and each one produced its own baseline. To remove any ambiguity about where a given baseline comes from, all three are recapitulated in Table 5.1 together with the date, the test object, the operating point, the sensor mounting position, the scaling factors in force, and the three limits derived from the baseline. Session A is a bench functional check of the module and is included only because its limits are visible on the HMI in Figure 5.1; the results reported in the remainder of this chapter come from Sessions B and C.

Table 5.1 Recapitulation of the teaching sessions and the baselines they produced.

Parameter	**Session A — bench commissioning**	**Session B — reference (healthy)**	**Session C — abnormal condition**
Date / time	Bench trial preceding the logged campaigns (not logged)	24 Jul 2026 10:00 – 25 Jul 2026 22:53	29 Jul 2026 13:11:22 – 14:18:14
Test object	Monitoring module only; sensor not mounted on a machine	Reciprocating air compressor, 10 HP (7.5 kW), 520 L, 10 bar, three-phase	Oil-less reciprocating air compressor, 1.5 HP (1.1 kW), 30 L, single-phase
Operating point	Not applicable; manually induced excitation	Automatic load/unload duty cycle against the 10 bar cut-out setting	Continuous run; cut-out pressure never reached because of an air leak
Sensor mounting point	Sensor free on the bench	Compressor motor housing	Compressor motor housing (same arrangement)
Teaching factors (off / pre / alarm)	0.4 / 1.8 / 2.5	0.4 / 1.8 / 2.5	0.4 / 1.15 / 1.25
Learned baseline B (mm/s)	46.54	28.66	6.50
L_off (mm/s)	18.62	11.46	2.60
L_pre (mm/s)	83.78	51.58	7.48

Parameter	Session A — bench commissioning	Session B — reference (healthy)	Session C — abnormal condition
L_max (mm/s)	116.36	71.64	8.13
Valid samples	Not logged	1,767 (full log); 348 (reference window)	67
Highest magnitude (mm/s)	76.94	38.68	17.34
Temperature range (°C)	34 – 36	37 – 42	31 – 78
Highest state reached	ALARM (manual excitation)	NORMAL	ALARM (latched)
Reported in	Figure 5.1 (module photograph only)	Figures 5.2, 5.3, 5.7; Table 5.2	Figures 5.5, 5.6

5.3 Baseline Teaching Results

The system was evaluated by analyzing the resultant vibration magnitude reported directly by the MPB10 under two operating conditions of the reference compressor: with the motor stopped and with the motor running. Because the sensor outputs the resultant magnitude as a single scalar retrieved through the SIG200 REST API, the assessment is based on the magnitude value itself rather than on a per-axis recombination performed on the controller, which simplifies the data path.

With the compressor stopped, the magnitude remained low and stable. The teaching procedure was then triggered from the web dashboard while the 10 HP three-phase compressor ran at its normal operating level, collecting ten magnitude samples and producing a baseline of B = 28.66 mm/s, from which the limits L_off = 11.46 mm/s, L_pre = 51.58 mm/s, and L_max = 71.64 mm/s were derived automatically. The learned values were stored in flash and reloaded at start-up, confirming the persistence mechanism of Section 4.6. It should be noted, in line with Section 1.4, that this baseline reflects the operating level of the machine at the moment of teaching; the mechanical health of the compressor was not independently verified beforehand, and the baseline was obtained from a single teaching cycle, so no spread is available for it. In addition to the magnitude, the temperature reported by the MPB10 was displayed on the local HMI (meter, trend chart, and running minimum/maximum) and published alongside the magnitude for remote monitoring. Figure 5.2 presents the baseline-teaching result together with the derived limits over a reference session of 348 valid samples recorded on 24 July 2026 between 10:00:59

and 16:03:54, in which the magnitude peaked at 37.45 mm/s and the temperature at 42 °C.

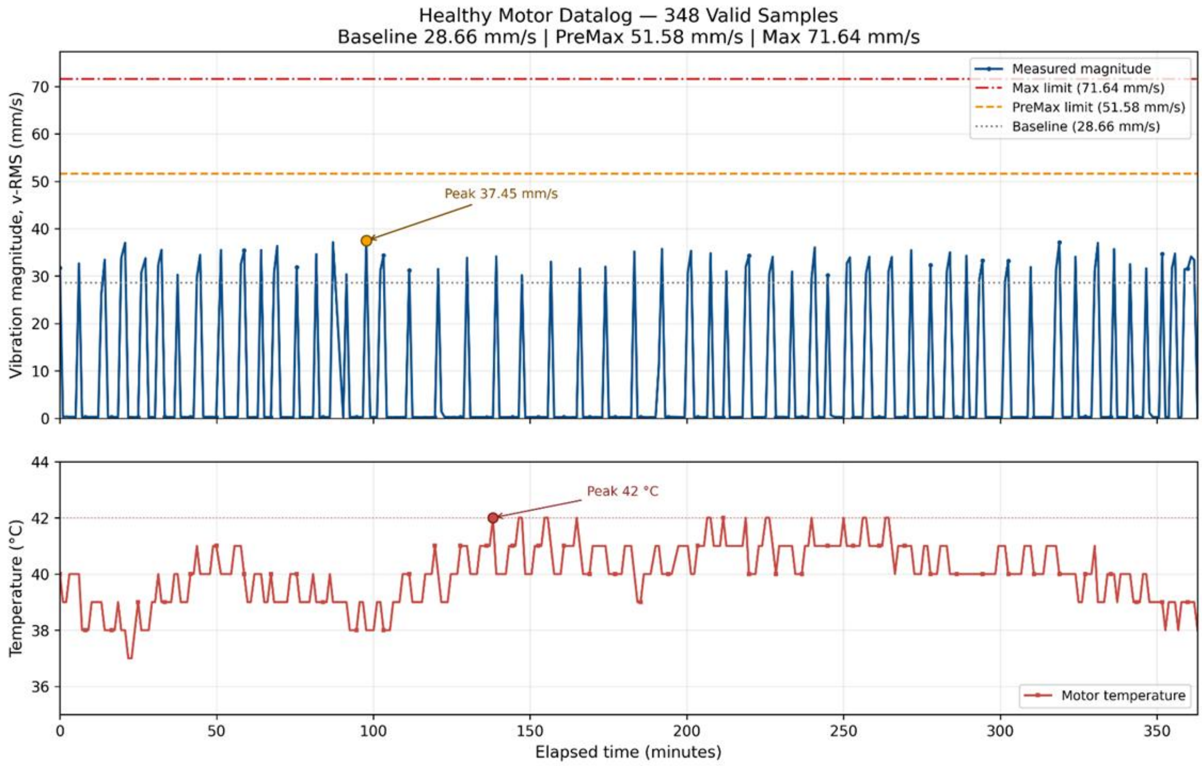

Figure 5.2 Baseline teaching result and derived limits (L_off, L_pre, L_max) on the healthy 10 HP (7.5 kW) three-phase compressor — 348 valid samples.

5.4 Condition Classification Results

With the compressor running, the magnitude increased but remained below the pre-warning limit throughout the observation period, and the system accordingly reported a normal condition. Across the 348 samples of the reference session the magnitude peaked at 37.45 mm/s at 97.7 minutes, that is 72.6 % of the pre-warning limit of 51.58 mm/s, and neither a WARNING nor an ALARM was raised. The magnitude followed the load/unload duty cycle of the compressor, falling below L_off = 11.46 mm/s whenever the unit stopped at the cut-out pressure and returning into the learned band once it restarted, without sudden spikes or abnormal fluctuations, indicating stable operation and the absence of critical anomalies during the test. Figure 5.3 shows the measured magnitude and temperature trend after teaching: prior to teaching, no baseline or warning/alarm limits exist, so no alarm can be triggered; once the teach routine completes, the pre-max (1.8×) and max (2.5×) limits are derived and the magnitude is continuously evaluated against them.

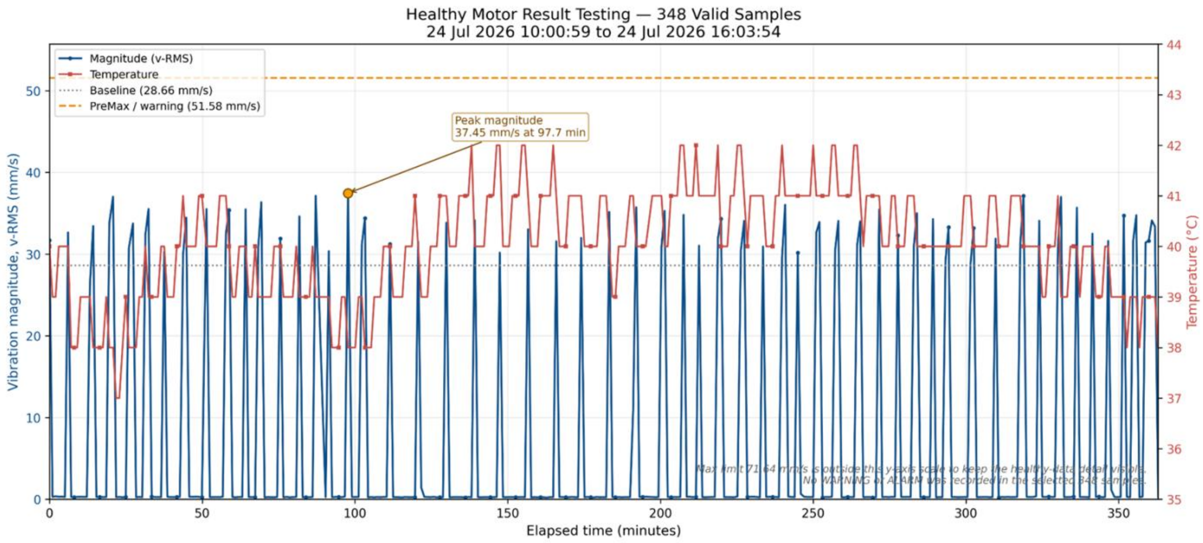

Figure 5.3 Measured vibration-magnitude and temperature trend of the healthy 10 HP (7.5 kW) three-phase compressor after auto-teaching (348 valid samples, 24 July 2026 10:00:59 to 16:03:54).

Representative classification results for the reference session are summarized in Table 5.2. The rule set correctly identified the stopped-machine state and the normal operating band throughout the session; the WARNING and latched-ALARM states of the same rule set were exercised on the second machine and are reported in Section 5.6.2.

Table 5.2 Condition-classification results for the healthy 10 HP (7.5 kW) three-phase compressor (B = 28.66 mm/s; L_off = 11.46 mm/s, L_pre = 51.58 mm/s, L_max = 71.64 mm/s).

Stage	Magnitude (mm/s)	Band	State
Before start	0.20	< L_off	OFF (motor stopped)
After start	28.60	L_off–L_pre	NORMAL
Session peak (348 samples)	37.45	L_off–L_pre	NORMAL
Full-datalog peak (1,767 samples)	38.68	L_off–L_pre	NORMAL (75.0 % of L_pre)

5.5 Consistency between the Local HMI and the Web Dashboard

Consistency between the local LVGL HMI and the HTML web dashboard was verified during the tests. The magnitude and condition state shown on the local monitor matched the values received by the web dashboard over MQTT, confirming reliable end-to-end data transmission through the WISP-mode link. The teaching procedure, executed from the web interface, correctly produced the baseline and limits that were subsequently displayed on both the local monitor and the dashboard. A representative view of the web dashboard is shown in Figure 5.4.

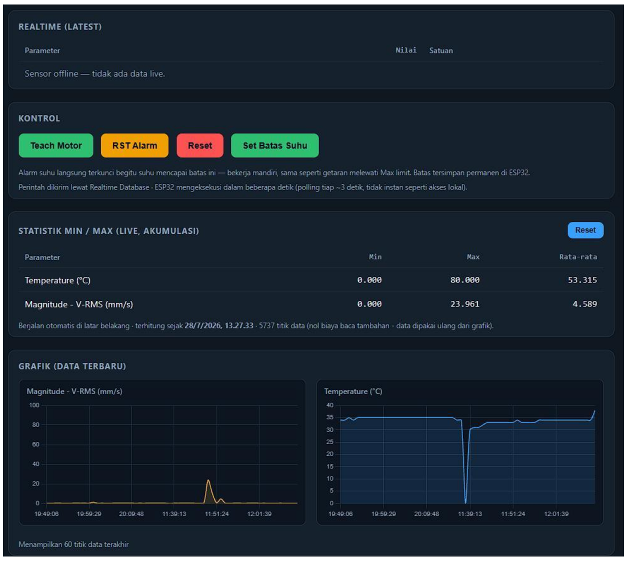

Figure 5.4 HTML web dashboard for global monitoring and teaching (representative).

5.6 IoT Communication and Data-Efficiency Evaluation

The IoT path was evaluated with respect to both reliability and resource efficiency. The MQTT publications arrived at the broker and dashboard at the expected cadence, and the dashboard's staleness detection behaved as designed: when the sensing device was powered off, the interface reported the sensor as offline within the configured time window and automatically stopped polling, instead of continuing to present outdated values as if they were live. The single-latest-document strategy kept the cloud read cost at exactly one read per device update, in contrast to a log-collection query approach in which every dashboard refresh would read tens of historical documents; this reduction is what keeps the system comfortably within the free quota of the cloud service during continuous operation.

5.6.1 Test Setup and Reference Condition

The MPB10 sensor remained mounted on the machine in the same configuration as in the earlier tests, and data were logged continuously to the cloud database over an extended period. For the healthy 10 HP three-phase compressor the full datalog spans 24 July 2026 10:00 to 25 July 2026 22:53, that is approximately 37 hours and 1,767 valid samples. Over this period the magnitude alternated between the OFF band

and the learned operating band in step with the load/unload duty cycle of the unit, and reached a maximum of 38.68 mm/s at 17:21 on 24 July 2026, that is 75.0 % of the pre-warning limit, while the motor temperature stayed between 37 °C and 42 °C. No WARNING and no ALARM was recorded during the entire period — that is, no false alarm occurred in 1,767 consecutive samples — consistent with the baseline established in Section 5.3. The complete datalog is summarised in Figure 5.7.

For the abnormal-condition test a different machine was used: the 1.5 HP (1.1 kW) single-phase oil-less compressor described in Section 5.2, whose pneumatic circuit was leaking. Because the leak prevented the receiver from reaching the cut-out pressure, the pressure switch never released the motor, and the unit ran continuously instead of following its normal load/unload cycle. The automatic teaching procedure was executed on this machine so that the operating limits reflected its own running level, and the scaling factors were tightened from 1.8 and 2.5 to 1.15 (pre-warning) and 1.25 (alarm), as foreseen in Section 4.7, so that a developing deviation could be resolved within the much narrower operating band of this smaller unit. The teaching produced a baseline of 6.50 mm/s, from which an off-threshold of 2.60 mm/s, a pre-max limit of 7.48 mm/s, and an alarm limit of 8.13 mm/s were derived automatically. Two qualifications apply to this baseline and are carried into Section 5.8: it was learned while the machine was already operating abnormally, so it represents the running level of a leaking system rather than a verified healthy condition, and it was obtained from a single teaching cycle.

5.6.2 Measured Fault Behaviour

The measured vibration magnitude and temperature during the abnormal-condition test are shown in Figure 5.5. The session was recorded on 29 July 2026 between 13:11:22 and 14:18:14 and comprises 67 valid samples over approximately 67 minutes. For the first 4.5 minutes the compressor was stopped and the magnitude remained at about 0.2 mm/s, below the off-threshold of 2.60 mm/s, so the system reported OFF. At start-up the magnitude rose immediately to about 6.3 mm/s and was classified as NORMAL. Because the leak prevented the cut-out pressure from being reached, the motor never unloaded and both parameters then rose together in a signature clearly led by temperature: the motor temperature climbed steadily from 31 °C to a peak of 78 °C, while the vibration magnitude increased more gradually from 6.3 mm/s to a peak of 17.34 mm/s, that is 2.7 times the learned baseline, as the machine ran without its normal rest intervals under sustained thermal load.

The correlated behaviour of both parameters is shown in Figure 5.5. The first WARNING was raised at 13:26:43, when the magnitude reached 7.52 mm/s and exceeded the pre-max limit of 7.48 mm/s at a motor temperature of 49 °C. Two minutes and thirty-seven seconds later, at 13:29:20, the magnitude reached 8.14 mm/s and crossed the alarm limit of 8.13 mm/s at a temperature of 54 °C, raising the latched ALARM, which remained active for the remainder of the session even though the magnitude continued to rise well beyond the limit. Ten minutes therefore elapsed between the start of the machine and the first WARNING, and a further two and a half minutes to the latched ALARM. It should be stated explicitly that the abnormality demonstrated here is a documented operating fault of the pneumatic system — a leak causing continuous duty — and not a mechanical defect of the motor that was independently confirmed; the temperature reported by the sensor is the physically leading indicator, while the classification rule of Table 4.3 responds to the accompanying rise in vibration magnitude. The complete transition from OFF through NORMAL and WARNING to the latched ALARM is shown in Figure 5.6, and the aggregated datalog of the healthy reference session is summarised in Figure 5.7.

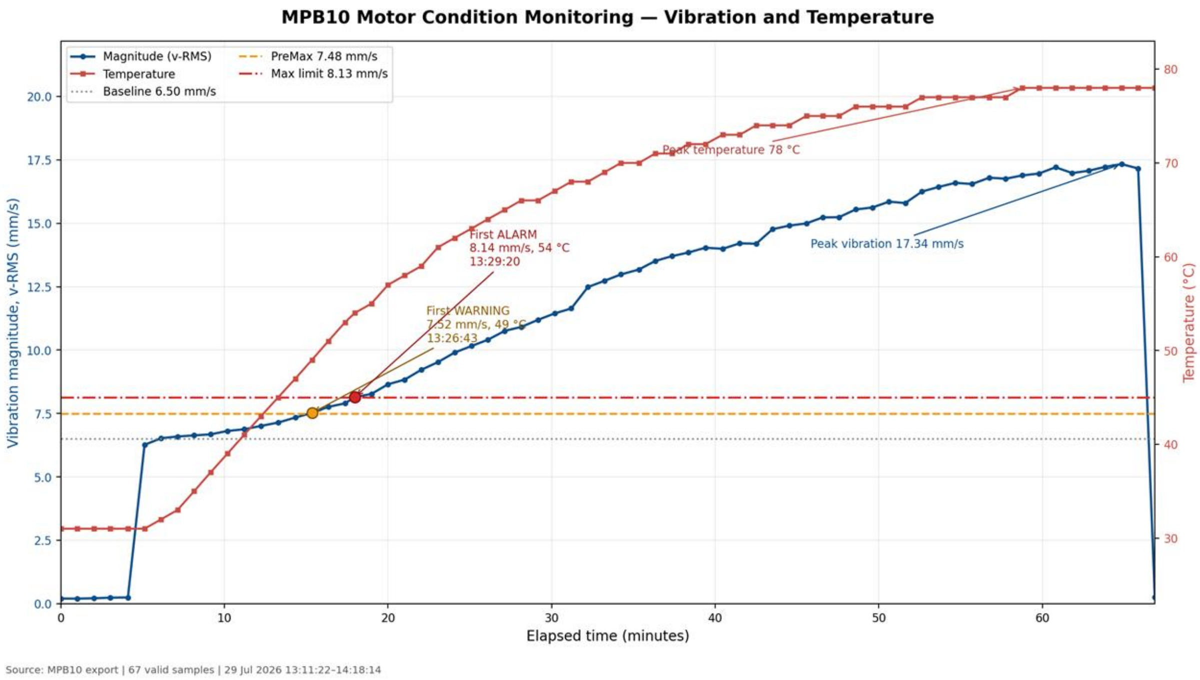

Figure 5.5 Correlated rise of vibration and temperature on the abnormal 1.5 HP (1.1 kW) single-phase compressor running continuously because of an air leak (67 valid samples, 29 July 2026 13:11:22 to 14:18:14).

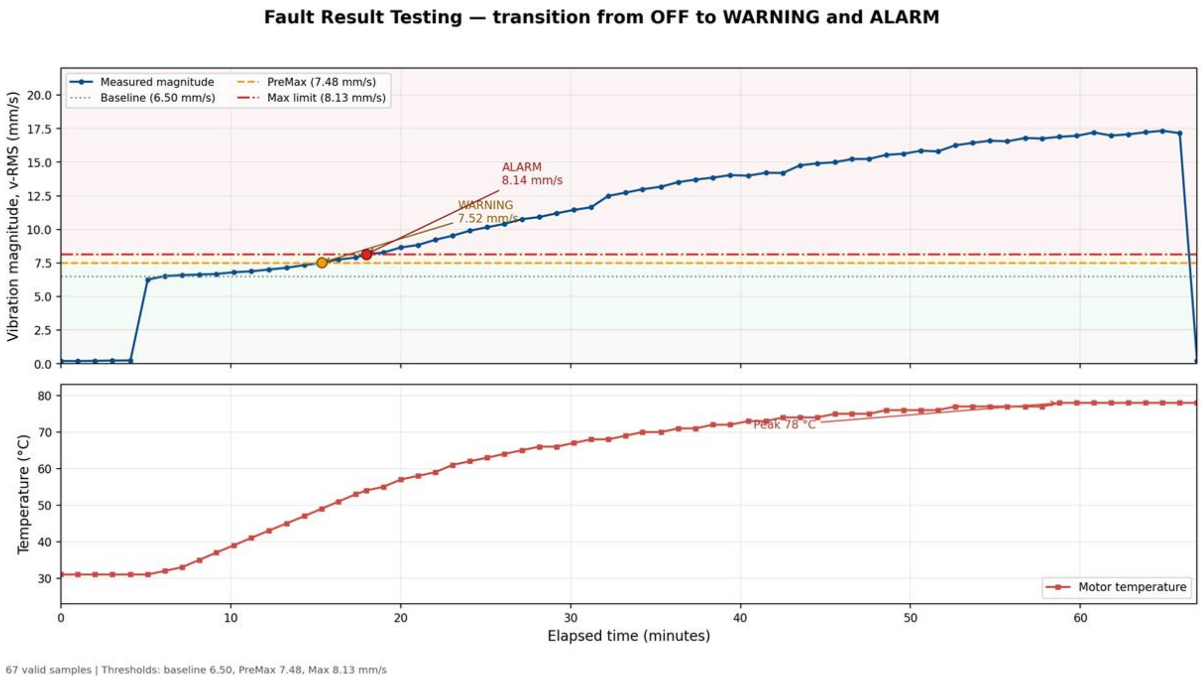

Figure 5.6 Transition from OFF to WARNING and latched ALARM on the abnormal 1.5 HP (1.1 kW) single-phase

compressor (baseline 6.50 mm/s, PreMax 7.48 mm/s, Max 8.13 mm/s).

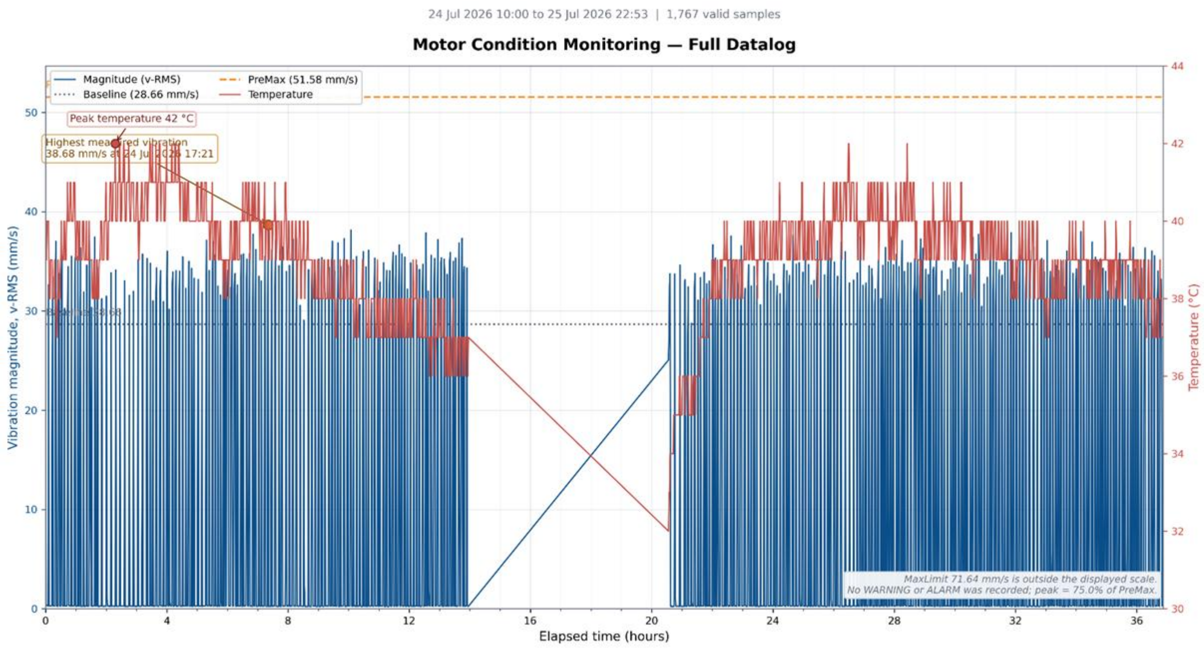

Figure 5.7 Full datalog summary from the sensor — healthy 10 HP (7.5 kW) three-phase compressor, 1,767 valid samples, 24 July 2026 10:00 to 25 July 2026 22:53.

5.7 IoT Communication and Data-Efficiency Evaluation

The IoT path was evaluated with respect to both reliability and resource efficiency. The MQTT publications arrived at the broker and dashboard at the expected cadence, and the dashboard's staleness detection behaved as designed: when the sensing device was powered off, the interface reported the sensor as offline within the configured time window and automatically stopped polling, instead of continuing to present outdated values as if they were live. The single-latest-node strategy kept each update to a small, fixed-size transfer, in contrast to a log-collection query approach in which every dashboard refresh would download tens of historical entries; this reduction is what keeps the system comfortably within the free bandwidth quota of the cloud service during continuous operation.

5.8 Discussion

Overall, the results demonstrate that the proposed system reliably captures the vibration magnitude in real time, learns a machine-specific baseline, and classifies the operating condition against limits derived from that baseline. Three claims are supported by the measured data and are stated here explicitly. First, the classification chain is stable under continuous operation: across 1,767 consecutive samples spanning approximately 37 hours on the reference compressor, the peak magnitude reached 75.0 % of the pre-warning limit and no false WARNING or false ALARM was produced. Second, the alarming chain responds to a real departure from the learned operating level: on the leaking compressor, which ran continuously because its cut-out pressure was never reached, the system raised the WARNING at 7.52 mm/s against a 7.48 mm/s limit and latched the ALARM at 8.14 mm/s against an 8.13 mm/s limit, ten and twelve and a half minutes respectively after the machine started. Third, the design decisions of Chapter 4 hold in the field: the learned limits survived power cycling through the NVS store, the latched alarm retained the excursion until it was explicitly reset, and the single-latest-document strategy held the cloud cost to one dashboard read per device update, approximately 8,640 reads per day at the ten-second publishing interval, well inside the free daily quota of the cloud service. The direct-magnitude approach reduces controller-side computation and data volume, and the combination of local visualization and IoT connectivity provides both on-site and remote condition awareness.

Several limitations qualify these findings and bound the claims that can be drawn from them. First, the measurement scale has not been verified. The magnitudes reported through the IODD process-data mapping have not been compared against a calibrated hand-held vibration meter, and the values recorded here are high relative to the severity zones that ISO 10816/20816 define for machines of this size; until such a comparison is made at several operating points, the results should be read as relative changes rather than as absolute vibration severity. Second, no controlled fault was injected and no independent ground truth was obtained: the abnormality of the second machine is documented from its operating behaviour (a pneumatic leak causing continuous duty) and not from disassembly or from a graded, repeatable defect, so the demonstrated capability is deviation-based condition alarming rather than validated fault detection, and no detection rate or false-alarm rate can be reported. Third, every baseline in this thesis comes from a single teaching cycle of ten samples over five seconds, so the repeatability of the teaching procedure and the spread of the learned baseline are unknown, and the teaching window is short compared with the duty cycle of the machines tested. Fourth, teaching was executed while each machine was already running at its operating level rather than in an independently verified healthy condition, which for the second machine means the baseline describes a leaking system; that session therefore demonstrates the adaptive mechanism and the alarming behaviour of the system, but does not by itself prove a fault-detection capability. Fifth, the scaling factors are not universal: they had to be tightened from (0.4, 1.8, 2.5) to (0.4, 1.15, 1.25) for the smaller machine, and they require re-tuning for other ratings, mounting classes, or operating-speed ranges. Sixth, the classification rule evaluates the vibration magnitude only, whereas the abnormal session was led physically by temperature; no learned temperature limit exists. These aspects define the boundary between the demonstrated functionality and broader generalization, and each of them is addressed by a specific recommendation in Section 6.2.

5.9 Summary

This chapter validated the implemented system on two operating air compressors. On the healthy 10 HP (7.5 kW) three-phase unit the automatic teaching produced a baseline of 28.66 mm/s and limits of 11.46, 51.58, and 71.64 mm/s, and over 1,767 valid samples spanning approximately 37 hours the magnitude peaked at 38.68 mm/s, that is 75.0 % of the pre-warning limit, with no false warning and no false alarm. On the abnormal 1.5 HP (1.1 kW) single-phase unit, which ran continuously because a pneumatic leak prevented its cut-out pressure from being reached, the teaching produced a baseline of 6.50 mm/s with limits of 2.60, 7.48, and 8.13 mm/s, and the system captured the full sequence from OFF through NORMAL and WARNING to the latched ALARM, raising the first WARNING at 7.52 mm/s and 49 °C ten minutes after start-up and latching the ALARM at 8.14 mm/s and 54 °C two and a half minutes later, while the temperature rose to a peak of 78 °C. The classification rule therefore reported the OFF, NORMAL, WARNING, and latched ALARM states correctly on both machines; the local HMI and the web dashboard remained consistent through the MQTT/WISP path; and the quota-efficient dashboard design with offline detection operated as intended. The provenance of every baseline used in this chapter is recorded in Table 5.1. These findings confirm the effectiveness of the system for real-time, deviation-based condition monitoring of induction-motor-driven machinery and support its applicability for condition-based maintenance, subject to the limitations set out in Section 5.8.

Chapter 6 Conclusion and Recommendations

6.1 Conclusion

This thesis set out to design, implement, and evaluate an integrated electric-motor monitoring system combining an industrial condition-monitoring sensor, a local HMI, and IoT connectivity. The objectives stated in Section 1.3 were addressed through the design of Chapter 4 and the evaluation of Chapter 5, and the following conclusions are drawn.

First, the sensing and processing chain works end to end. The MPB10 delivers the resultant velocity-RMS magnitude and the sensor temperature through the SIG200 IO-Link gateway, and the ESP32-S3 acquires both quantities over the REST interface every 500 ms, classifies each sample, drives the LVGL HMI, and publishes the result to the cloud without interrupting the display refresh. The chain ran without loss of data across a continuous logging period of approximately 37 hours and 1,767 valid samples.

Second, the automatic teaching mechanism produces usable, machine-specific limits without manual configuration. On the 10 HP three-phase compressor the procedure learned a baseline of 28.66 mm/s and derived limits of 11.46, 51.58, and 71.64 mm/s; on the 1.5 HP single-phase compressor it learned 6.50 mm/s and derived 2.60, 7.48, and 8.13 mm/s. The learned values survived power cycling through the NVS store, confirming the persistence design of Section 4.6.

Third, the classification and alarming logic behaves as specified. The reference session remained within the OFF and NORMAL bands throughout, with a peak of 38.68 mm/s corresponding to 75.0 % of the pre-warning limit and no false warning or false alarm in 1,767 consecutive samples. The abnormal session reproduced the complete OFF, NORMAL, WARNING, and latched-ALARM sequence, with the WARNING raised at 7.52 mm/s against a 7.48 mm/s limit and the ALARM latched at 8.14 mm/s against an 8.13 mm/s limit, and the alarm remained latched until it was explicitly reset from the dashboard.

Fourth, the local and remote views stayed consistent, and the single-latest-document cloud strategy held the cost of continuous operation to one dashboard read per device update. At the ten-second publishing interval this amounts to approximately 8,640 reads per day, comfortably inside the free daily quota of the cloud service.

Finally, and of equal importance, the boundary of what has been demonstrated must be stated plainly. What this work establishes is deviation-based condition alarming: the system reliably detects and reports a departure from the operating level that the machine itself has taught it, and it does so on two machines of very different rating. Because no controlled fault was injected, because the abnormality of the second machine is documented only from its operating behaviour, and because the absolute vibration scale has not been checked against a calibrated instrument, this work does not establish a validated fault-detection or fault-diagnosis capability. Raising the claim to that level is the purpose of the recommendations that follow.

6.2 Recommendations for Further Work

The following recommendations follow directly from the limitations identified in Sections 1.4 and 5.8 and define the next stage of this work.

Verify the measurement scale. The magnitude reported through the IODD process-data mapping should be compared against a calibrated hand-held vibration meter at a minimum of three distinct operating points, and the deviation reported. Until this comparison exists, the absolute values in Chapter 5 must be read as relative indicators only, and any reference to ISO severity zones remains provisional.

Impose a teaching precondition in firmware. Teaching should be permitted only on a machine whose healthy condition has been independently verified, and re-teaching should be disabled for the duration of any anomaly investigation, so that an abnormal operating level cannot be silently adopted as the new baseline. This is an interlock in the firmware, not merely an operating procedure.

Record the operating point with every baseline. Each teaching cycle should store the conditions under which it was executed — speed, load, discharge pressure, and the sensor mounting position — alongside the learned baseline, so that limits are compared only between sessions taken at the same operating point.

Measure the repeatability of the teaching procedure. Ten consecutive teaching cycles should be executed at an identical operating point and the mean and standard deviation of the learned baseline reported. This quantifies the confidence interval of every derived limit and is a prerequisite for defending any threshold as a decision boundary.

Lengthen the teaching window. The present cycle averages ten samples over five seconds, which is short relative to the duty cycle of the machines tested. A window spanning at least one complete load/unload cycle, or a percentile-based estimator in place of the arithmetic mean, would yield a baseline that is far less sensitive to the instant at which teaching happens to be triggered.

Establish ground truth through graded fault injection. Controlled unbalance should be introduced in defined increments on a test rig, the detection outcome recorded at each level, and the detection rate and false-alarm rate reported. Only with this evidence can the claim be raised from deviation-based alarming to validated fault detection.

Extend the classification rule to temperature. The abnormal session was led physically by temperature rather than by vibration, yet the rule of Table 4.3 evaluates the magnitude alone. A learned temperature limit derived by the same teaching mechanism would allow the system to report thermal deviations directly instead of leaving them to operator interpretation.

References

Abduljawwad, M., Khaleel, M., Ogedengbe, T. S., & Abraheem, S. (2023). Sensors for daily utilization. *Int. J. Electr. Eng. and Sustain.*, 106-119.

Altun, M. (2024). *Data-Driven and Knowledge-Assisted Model-Based Frameworks for Supporting Facility Maintenance* (Doctoral dissertation, Middle East Technical University (Turkey)).

Bilgin, B., & Emadi, A. (2019). Electric motor industry and switched reluctance machines. In *Switched Reluctance Motor Drives* (pp. 1-33). CRC Press.

Espressif Systems. (2023). ESP-IDF Programming Guide. Retrieved from https://docs.espressif.com/projects/esp-idf/en/latest/esp32s3/index.html

Espressif Systems. (2023). ESP32-S3 Technical Reference Manual (Version 1.0). Retrieved from https://www.espressif.com/en/support/documents/technical-documents

Fofana, I., & Hadjadj, Y. (2016). Electrical-based diagnostic techniques for assessing insulation condition in aged transformers. *Energies*, *9*(9), 679. https://doi.org/10.3390/en9090679

Hassanalieragh, M., Page, A., Soyata, T., Sharma, G., Aktas, M., Mateos, G., ... & Andreescu, S. (2015, June). Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges. In *2015 IEEE international conference on services computing* (pp. 285-292). IEEE. https://doi.org/10.1109/SCC.2015.47

Islam, M., Zaman, A., Nessa, M., & Margaret, N. (2024). Smart monitoring: IOT enhanced fault detection techniques for single-phase induction motor. http://hdl.handle.net/10361/25861

Kumari, S., Raj, R., & Komati, R. (2021). A Thing Speak IoT Based Vibration Measurement and Monitoring System Using an Accelerometer sensor. *Int. J. Res. Appl. Sci. Eng. Technol*, *9*, 1249-1258. https://doi.org/10.22214/ijraset.2021.37599

LVGL. (2023). LVGL Documentation - Graphics Library for Embedded Systems. Retrieved from https://docs.lvgl.io

Mohd Ghazali, M. H., & Rahiman, W. (2021). Vibration analysis for machine monitoring and diagnosis: A systematic review. *Shock and Vibration*, *2021*(1), 9469318. https://doi.org/10.1155/2021/9469318

Mohammed, N. A., Abdulateef, O. F., & Hamad, A. H. (2023). An IoT and machine learning-based predictive maintenance system for electrical motors. *Journal Européen des Systèmes Automatisés*, *56*(4), 651. https://doi.org/10.18280/jesa.560414

Nextion. (2022). 7.0" HMI Display User Manual. Retrieved from https://nextion.tech

Rehman, S. U., Usman, M., Toor, M. H. Y., & Hussaini, Q. A. (2024). Advancing structural health monitoring: A vibration-based IoT approach for remote real-time systems. *Sensors and Actuators A: Physical*, *365*, 114863. https://doi.org/10.1016/j.sna.2023.114863

Sahu, P. K., Bhattacharjee, A. K., Sahu, A. K., & Choudhury, B. (2020). Fault diagnosis of electric motors using vibration signal analysis. *International Journal of Applied Power Engineering (IJAPE),* *9*(3), 199-206. https://doi.org/10.11591/ijape.v9.i3.pp199-206

Setiawan, A., Utomo, D. S., & Wibowo, A. T. (2023). A Literature Review: Bearing Fault in BLDC Motor Based on Vibration and Temperature Analysis. *Journal of Electrical, Electronics, and Informatics Communication Technology (JEEICT), 2*(2), 57-63. https://doi.org/10.20961/jeeict.v2i2.100165

Siddique, A., Yadava, G. S., & Singh, B. (2003, August). Applications of artificial intelligence techniques for induction machine stator fault diagnostics. In *4th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives, 2003. SDEMPED 2003.* (pp. 29-34). IEEE. https://doi.org/10.1109/DEMPED.2003.1234543

Singh, K. M., & Sumathi, P. (2015). Vibration parameter estimation methods for ultrasonic measurement systems–a review. *IET Science, Measurement & Technology*, *9*(4), 492-504. https://doi.org/10.1049/iet-smt.2014.0176

Tiboni, M., Remino, C., Bussola, R., & Amici, C. (2022). A review on vibration-based condition monitoring of rotating machinery. *Applied Sciences*, *12*(3), 972. https://doi.org/10.3390/app12030972

Yousuf, M., Alsuwian, T., Amin, A. A., Fareed, S., & Hamza, M. (2024). IoT-based health monitoring and fault detection of industrial AC induction motor for efficient predictive maintenance. *Measurement and Control*, *57*(8), 1146-1160. https://doi.org/10.1177/00202940241231473

Zakharov, A., Kobelev, A., Makarov, L., & Skitovich, S. (2018, October). Research, development, manufacturing and application of energy efficient electric motors. In *2018 X International Conference on Electrical Power Drive Systems (ICEPDS)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICEPDS.2018.8571692