happiest minds
AI FIRST. AGILE ALWAYS.

AI-Powered HMI Test Automation **for Edge Energy Utilities**

Contents

01 Introduction 03

02 Limitations of Conventional HMI Automation Approaches 03

03 Screen-as-Contract Validation Model 04

04 Validation System Architecture Overview 04

4.1. HMI Navigation and Input Control 05

4.2. Screen Capture Layer 05

4.3. AI-Based Screen Data Extraction 06

4.4. Data parsing and structuring 07

4.5. Validation Framework 07

05 Conclusion 07

01
Introduction

Edge energy utilities embed intelligence to make real operational decisions at the edge. Human-Machine Interfaces (HMIs) serve as the primary interaction points for field operators and become the source of truth in that moment. Yet, HMI validation lags the rest of the system. It is often manual or partially simulated. The result is a growing gap between the rigor in the backend and how the operator-facing HMI is tested.

Modern energy meters, protection relays, inverters, and controllers expose operational state, alarms, configuration parameters, and diagnostics directly through local displays. In many deployment scenarios, the HMI is the primary reference during commissioning, troubleshooting, and fault recovery. Errors in HMI presentation—incorrect values, misleading units, or inconsistent navigation—can lead to operational delays, incorrect decisions, or safety risks. These challenges must be addressed through a screen-based validation architecture augmented by AI-driven data extraction.

This article presents a point of view for an AI-powered approach to HMI test automation that treats the rendered screen as the validation contract. By combining controlled screen capture, AI-based text extraction, deterministic parsing, and rule-based validation, the proposed architecture enables scalable, hardware-agnostic, and operator-centric HMI verification for edge energy devices.

02
Limitations of Conventional HMI Automation Approaches

HMIs are typically built using embedded front-end frameworks where firmware supplies data and display drivers render directly to the screen buffer. Unlike web-based systems, there is limited control or visibility into internal UI structures once the application runs on real hardware. While development-time hooks such as screen casting or frame-buffer access are theoretically possible, they are rarely available or practical in production firmware.

AREA	LIMITATIONS
UI Access	No access to internal UI metadata or screen buffers in production builds
Execution Environment	Heavy dependence on simulators that do not accurately reflect real hardware behaviour
Rendering Pipeline	Screen content is rendered through display drivers and frame buffers with no support for interception or access
Production Constraints	Final production firmware builds exclude any hooks for remote screen access
Validation Surface	Automation cannot reliably observe the final operator-visible output on physical devices

03

Screen-as-Contract Validation Model

The proposed approach takes a practical view of the HMI by treating the rendered screen as the contract. Display content on the HMI interface produced by the firmware becomes the device under test. Validation is entirely based on what is visible to the operator, without any dependence on UI frameworks, image rendering pipelines, or other internal firmware details. The core principles of this approach are outlined below:

HMI navigation through physical or electrical interaction mechanisms	**Screen content captured directly from the deployed device**	**Validation based on data extracted from the rendered screen**	**Backend correctness assessed only through operator-visible behaviour**

This model aligns validation with real-world usage and minimizes dependency on internal UI architecture, enabling stable automation across firmware revisions and hardware variants.

04

Validation System Architecture Overview

HMI Navigation and Input Control

- Screen Navigation
- Physical / Electrical Interaction
- Relay, Solenoid, Button Pusher
- Operator-equivalent Inputs

Screen Capture

- Fixed Industrial Camera
- Controlled Lightning
- High-resolution Image Capture

AI-based Screen Data Extraction

- AI Models Selection & Training
- Layout-aware Extraction
- Multi-language Support
- Screen Order Preservation

Data Validation

- Label Validation
- Static & Dynamic Text Check
- Data Tolerence Checks
- Unit Consistency
- Backend Correlation for Data Validation

Reporting

- Pass/Fail Results
- Screen Evidence
- Logs and Audit Data

4.1
HMI Navigation and Input Control

Navigation on the HMI screen requires physical interactions. For an effective test automation strategy, there should be a clear, programmatic way to navigate. This architecture supports multiple navigation strategies, including electrical triggering through relays, mechanical actuation using solenoids or dedicated button pushers. Navigation mode selection criteria emphasize reliability, repeatability, and minimal mechanical complexity for long-duration regression testing environments.

4.2
Screen Capture Layer

Screen capture is one of the most important aspects and has a direct impact on the accuracy of subsequent steps in data extraction and validation. The architecture suggests the use of fixed-focus industrial or machine-vision cameras mounted rigidly to capture the HMI screen after navigation to a specified screen. Another important aspect is controlled lighting during screen capture. One of the strategies is to create a fixture for this purpose, with the key considerations including:

- Stable camera positioning to eliminate perspective distortion
- Controlled lighting to minimize glare and reflections
- Resolution sufficient to capture fine numeric detail
- Synchronization with display refresh where applicable

Experience indicates that investment in capture stability yields greater returns than excessive OCR model tuning.

4.3
AI-Based Screen Data Extraction

AI is employed exclusively for perception tasks—converting captured screen data into textual information. Several OCR models, namely LayoutLM, EasyOCR, TrOCR or PaddleOCR, can be considered for this purpose. Model selection is driven by:

- Numeric recognition accuracy
- Preservation of reading order
- Robustness to font and contrast variation
- Multi-language support

Receive Screen Image

The layer receives a screen image as input.

AI Model Selection and Training

Appropriate AI model is selected and Trained using the fonts used by the firmware.

Extract Textual Data

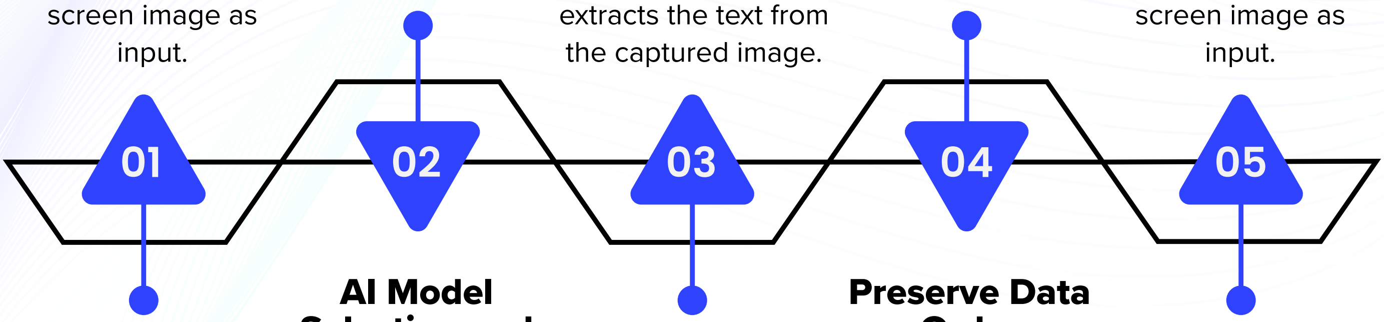
Trained OCR model extracts the text from the captured image.

Preserve Data Order

Layout Aware Models maintain the correct reading and the extracted order of the screen data.

Reveive Screen Image

The layer receives a screen image as input.



Out-of-the-box OCR models are rarely sufficient for production HMI validation, particularly in edge energy systems where fonts are proprietary, contrast is inconsistent, and numeric accuracy is non-negotiable. Generic models normally produce acceptable text recognition but an unacceptable rate of false positives, especially for closely spaced digits and for characters that resemble each other, such as “u” and “v”, etc. To address this, the OCR models are selectively fine-tuned using product-specific screen captures and the actual font files used by the HMI. This targeted fine-tuning of the selected AI model significantly improves robustness and reduces false positives.

4.4
Data Parsing and Structuring

The data parsing layer transforms raw OCR model output into a structured screen representation suitable for validation. The OCR model should also retain the order. This data parsing and structuring is very important for validating the positional information on the screens, such as the order in which the titles, labels, values, and units are displayed. Responsibilities include:

- Identifying semantic roles (titles, labels, values)
- Preserving relative ordering
- XX.XX Normalizing numeric formats and units

This structured representation serves as the basis for deterministic validation.

4.5
Validation Framework

Extracted and structured data is then passed to the validation logic, which is rule-based and explicit, avoiding probabilistic decision-making. The Test Framework shall include the required reference data for all screens and match it against the AI-extracted data. Typical validation checks include:

- Label correctness against specification
- XX.✔ Numeric value validation within tolerance
- XX.XX Unit presence and consistency
- Screen availability and navigation integrity

This separation ensures that AI uncertainty does not affect validation outcomes.

In addition to textual validation extracted from HMI screens, the framework should also consider validating non-textual visual elements that convey critical operator state.

05
Conclusion

AI-powered HMI test automation enables a shift from implementation-centric testing to operator-centric validation. By treating the screen as the contract and applying AI selectively for perception, organizations can achieve scalable, hardware-agnostic, and maintainable HMI validation for edge energy systems. This approach complements the traditional verification methods by addressing a critical gap in real-world system validation.

ABOUT THE AUTHORS

RAVI NAGA KUMAR

Ravinagakumar K is a Principal Test Architect with 20 years of experience in test automation, embedded systems, and Industrial Automation / IIoT. He has led the design and development of scalable test automation systems across multiple industries, bringing a strong end-to-end system perspective spanning devices, protocols, applications, and platforms. His work also focusses on AI-assisted and intelligent test automation, automation frameworks, simulators, and data-driven test strategies for complex industrial systems.

SHAHBAZ KHAN

Test Module Lead with 10+ years of experience in validating complex embedded, automotive, and industrial IoT systems. Specialized in designing scalable test automation frameworks, protocol-level validation, and simulators, with strong hands-on expertise in Python-based tooling, CI/CD pipelines, and AI-driven validation techniques for media and system-level quality analysis. Focused on delivering reliable, high-quality, and future-ready testing solutions across hardware–software ecosystems.

About Happiest Minds Technologies

Happiest Minds Technologies Limited (BSE, NSE: HAPPSTMNDS) is an AI First, customer-centric digital engineering company committed to delivering 'Happiest People . Happiest Customers'. With an integrated approach that spans from chip to cloud, Happiest Minds delivers secure and scalable solutions across product engineering, cybersecurity, analytics , and automation platforms. Happiest Minds brings purpose and precision to every engagement, helping enterprises solve complex business challenges and fast-track their digital evolution across industry sectors such as Banking, Financial Services & Insurance (BFSI), EdTech, Healthcare & Life Sciences, Hi-Tech and Media & Entertainment, Industrial, Manufacturing, Energy & Utilities, and Retail, CPG & Logistics.

Happiest Minds has been honored by both the Golden Peacock Awards and the Institute of Company Secretaries of India (ICSI) for its exemplary Corporate Governance practices. Guided by its mission of 'Happiest People . Happiest Customers' and consistently recognized as a great place to work, Happiest Minds is headquartered in Bengaluru, India, with a global presence across the Americas, UK, Europe, Australia, the Middle East, Africa, and Asia.

For more information, write to us at
business@happiestminds.com

happiest minds

AI FIRST. AGILE ALWAYS.