Industrial Automation System - Functional Description

Document Version: 1.0
Date: November 10, 2025
System Version: 0.1.0

Executive Summary

The Industrial Automation System is a comprehensive, enterprise-grade automation platform designed for industrial environments. Built on a microservices architecture, it provides visual flow-based automation, multi-protocol communication support (including specialized radio systems), real-time monitoring, and centralized analytics across multiple facility locations.

System Overview

Purpose

To provide a unified automation platform that enables industrial facilities to:

- Create and manage automation workflows through an intuitive visual interface
- Integrate diverse communication protocols and hardware interfaces
- Monitor and analyze system operations in real-time
- Aggregate data and insights across multiple geographical locations
- Leverage AI assistance for workflow creation and optimization

Target Platform

- **Primary Deployment:** UP Boards running Ubuntu 22.04+
- **Architecture:** Distributed edge computing with central aggregation
- **Deployment Model:** Containerized microservices (Docker)

Core Functional Components

1. Flow Engine

Purpose: Execute user-defined automation workflows using a node-based visual programming paradigm.

Key Features:

- **Visual Flow Creation:** Drag-and-drop interface for building automation logic
- **Node-Based Architecture:** Modular, reusable components for different operations
- **Real-Time Execution:** Event-driven processing with configurable triggers
- **Flow Validation:** Pre-execution checks to ensure workflow integrity
- **Hot Deployment:** Update flows without system restart
- **State Management:** Persistent flow state across restarts

Supported Node Types:

- **Inject Nodes:** Trigger flows based on time schedules or external events
- **Debug Nodes:** Output debugging information and intermediate values
- **Function Nodes:** Execute custom JavaScript logic within flows
- **MQTT Nodes:** Publish and subscribe to MQTT topics

- **Interface Nodes:** Communicate with hardware interfaces (radio, serial, network)
- **Logic Nodes:** Boolean operations, conditionals, and routing
- **Transform Nodes:** Data manipulation and format conversion

Flow Operations:

- Create, update, delete, and deploy flows via REST API
- Start/stop individual flows
- Monitor flow execution status
- View flow execution history and outcomes

---

2. Communication Interfaces

Purpose: Provide unified abstraction layer for diverse industrial communication protocols and hardware.

2.1 Radio Communication

Ritron DTX Radio Interface

- **Function:** Two-way radio communication for voice and data
- **Protocol:** Serial/USB communication
- **Capabilities:**
  - PTT (Push-to-Talk) control
  - Channel selection
  - Audio routing
  - Status monitoring
  - Emergency alerts

Motorola DLR Radio Interface

- **Function:** Digital license-free radio communication
- **Protocol:** Serial/USB communication
- **Capabilities:**
  - Digital voice communication
  - Text messaging
  - Group calling
  - Channel management
  - Signal strength monitoring

2.2 Wireless Protocols

Zigbee Interface

- **Function:** Low-power mesh networking for sensors, actuators, and wireless buttons
- **Supported Devices:** Temperature sensors, motion detectors, smart switches, wireless buttons
- **Supported Coordinators:** ConBee II, CC2531, CC2652, SONOFF Zigbee Bridge
- **Capabilities:**
  - Device discovery and pairing
  - Mesh network management
  - Battery status monitoring
  - Custom cluster commands
  - Button event detection
  - Multi-click pattern recognition

Zigbee Button Support:

- **Supported Button Devices:**
  - IKEA TRADFRI (5-button remote)
  - Philips Hue Tap (4-button, battery-free)
  - Xiaomi/Aqara Button (multi-click, hold)
  - Tuya Smart Button
  - SONOFF SNZB-01
- **Button Event Types:**
  - Single press/click
  - Double press (500ms window)
  - Triple press
  - Hold (long press)
  - Release (after hold)
  - Device-specific actions (brightness, scenes)
- **Button Features:**
  - Multi-click detection with configurable timing
  - Confidence-based event filtering
  - Battery level monitoring
  - Automatic binding and configuration
  - Per-device and per-action filtering
  - Rich event metadata (device type, endpoint, cluster)
- **Use Cases:**
  - Scene triggering and control
  - Emergency alerts (triple-click pattern)
  - Equipment start/stop controls
  - Lighting and HVAC control
  - Mobile worker check-in buttons
  - Safety zone acknowledgment

Bluetooth/BLE Interface

- **Function:** Short-range wireless communication
- **Use Cases:** Mobile device integration, beacon tracking, proximity sensing
- **Capabilities:**
  - Device scanning and discovery
  - GATT service interaction
  - Characteristic read/write
  - Notification handling

2.3 Network Protocols

Modbus TCP/RTU

- **Function:** Industrial protocol for PLC and sensor communication
- **Capabilities:**
  - Read/write holding registers
  - Read input registers
  - Read coils and discrete inputs
  - Function code support

OPC UA (Planned)

- **Function:** Platform-independent industrial interoperability
- **Capabilities:**
  - Secure client/server communication
  - Subscription-based data updates
  - Historical data access

TCP/IP Sockets

- **Function:** Direct network communication
- **Modes:** Client and server
- **Protocols:** TCP and UDP

2.4 Camera Integration

IP Camera Interface

- **Function:** Integrate with IP cameras for video analytics events and triggers
- **Supported Manufacturers:** Hikvision, Dahua, Axis, and generic ONVIF-compatible cameras
- **Connection:** HTTP/HTTPS API integration with event notification callbacks
- **Capabilities:**
  - Motion detection triggers
  - Dwelling/loitering detection
  - Line crossing detection (virtual tripwire)
  - Intrusion detection (zone-based)
  - Face detection events
  - Tamper detection (camera obstruction/defocus)
  - Object abandoned detection
  - Object missing/removal detection
  - Snapshot capture on demand
  - Configurable sensitivity and confidence thresholds

Event Types:

- **Motion Detection:** Triggered when movement is detected in defined zones
  - Configurable sensitivity (0-100%)
  - Region-based detection
  - Minimum confidence threshold
- **Dwelling Detection:** Triggered when person/object remains in area for specified duration
  - Configurable dwell time (seconds)
  - Use cases: Queue monitoring, unauthorized loitering, safety zone violations
- **Line Crossing:** Triggered when object crosses virtual line
  - Directional detection (A→B, B→A, or bidirectional)
  - Use cases: People counting, access control, perimeter security
- **Intrusion Detection:** Triggered when object enters prohibited zone
  - Polygon-based zone definition
  - Use cases: Restricted area access, safety zone monitoring
- **Face Detection:** Triggered when human face is detected

  - Optional face recognition integration
  - Use cases: Access control, attendance tracking
- **Tamper Detection:** Triggered when camera is blocked, moved, or defocused
  - Use cases: Vandalism prevention, maintenance alerts
- **Object Abandoned/Missing:** Triggered when object left behind or removed
  - Use cases: Unattended baggage detection, theft prevention

Integration Features:

- Real-time event streaming via HTTP callbacks
- Snapshot capture on event trigger
- Multi-camera support (up to 64 cameras per interface)
- Event filtering by confidence level
- Automatic camera discovery (ONVIF)
- Camera health monitoring
- Bandwidth-optimized event notifications

---

3. Web-Based User Interface

Purpose: Provide intuitive, responsive web interface for system interaction and monitoring.

3.1 Dashboard

- **System Overview:** Real-time status of all subsystems
- **Active Flows:** Currently running automation workflows
- **System Metrics:** CPU, memory, uptime, and performance indicators
- **Recent Activity:** Latest flow executions and system events
- **Quick Actions:** Start/stop flows, view alerts, access documentation

3.2 Flow Editor

- **Visual Canvas:** Drag-and-drop flow design workspace
- **Node Palette:** Categorized library of available nodes
- **Connection Management:** Wire nodes together to define data flow
- **Node Configuration:** Property panels for each node type
- **Validation Feedback:** Real-time error checking and suggestions
- **Deployment Controls:** Test, validate, and deploy flows

3.3 Analytics Dashboard

- **Flow Execution Metrics:** Success rates, execution times, error counts
- **Time-Series Graphs:** Historical trends and patterns
- **Interface Statistics:** Communication success/failure rates per interface
- **Custom Reports:** Configurable data visualizations
- **Export Capabilities:** CSV, JSON, and PDF export

3.4 Interface Configuration

- **Interface Management:** Add, configure, and remove communication interfaces
- **Connection Testing:** Verify interface connectivity
- **Parameter Configuration:** Set communication parameters (baud rate, IP, etc.)
- **Status Monitoring:** Real-time interface health indicators

3.5 Settings

- **System Configuration:** Global system parameters
- **User Management:** Access control and authentication
- **Site Information:** Location details and identification
- **Integration Settings:** MQTT broker, database, external APIs
- **Logging Configuration:** Log levels and retention policies

---

4. Logging and Analytics

Purpose: Comprehensive data capture and analysis for operational intelligence.

4.1 Local Logging

- **Flow Execution Logs:** Detailed record of every flow run
- **System Logs:** Application events, errors, and warnings
- **Interface Logs:** Communication attempts, successes, and failures
- **Audit Logs:** User actions and configuration changes
- **Performance Logs:** Resource utilization metrics

Log Levels:

- Error: Critical failures requiring immediate attention
- Warn: Potential issues that don't halt operation
- Info: Normal operational events
- Debug: Detailed troubleshooting information

Storage:

- In-memory (development): Fast but non-persistent
- SQLite (production): Local persistent storage
- File-based: Structured JSON log files with rotation

4.2 Local Analytics

- **Real-Time Dashboards:** Live system performance visualization
- **Historical Analysis:** Trend identification over time
- **Statistical Summaries:** Aggregated metrics (mean, median, percentiles)
- **Anomaly Detection:** Identify unusual patterns or behaviors
- **Custom Queries:** SQL-based data exploration

4.3 Central Log Aggregation (Planned)

- **Multi-Site Collection:** Aggregate logs from all deployed systems
- **Centralized Storage:** Time-series database (TimescaleDB/PostgreSQL)
- **Cross-Site Analytics:** Compare and correlate data across locations
- **Enterprise Dashboards:** Organization-wide operational visibility
- **Long-Term Retention:** Configurable archival and compliance storage

---

5. Real-Time Communication

Purpose: Enable live updates and bidirectional communication between backend and frontend.

WebSocket Server:

- **Port:** 3001
- **Protocol:** Socket.io
- **Features:**
  - Flow execution status updates

    - System event notifications
    - Live log streaming
    - Interface state changes
    - Multi-client broadcast

Events:

- `flow:started` - Flow execution begins
- `flow:completed` - Flow execution finishes
- `flow:error` - Flow execution fails
- `interface:connected` - Interface comes online
- `interface:disconnected` - Interface goes offline
- `system:alert` - Critical system notification

API Endpoints

REST API (Port 3000)

Base URL: `http://localhost:3000/api`

Flow Management

- `GET /api/flows` - List all flows
- `POST /api/flows` - Create new flow
- `GET /api/flows/:id` - Get flow details
- `PUT /api/flows/:id` - Update flow
- `DELETE /api/flows/:id` - Delete flow
- `POST /api/flows/:id/deploy` - Deploy flow
- `POST /api/flows/:id/start` - Start flow execution
- `POST /api/flows/:id/stop` - Stop flow execution

Analytics

- `GET /api/analytics/flows` - Flow execution statistics
- `GET /api/analytics/flows/:id` - Specific flow analytics
- `GET /api/analytics/interfaces` - Interface performance metrics
- `GET /api/analytics/system` - System-wide statistics

Interface Management

- `GET /api/interfaces` - List all interfaces
- `POST /api/interfaces` - Add new interface
- `GET /api/interfaces/:id` - Get interface details
- `PUT /api/interfaces/:id` - Update interface configuration
- `DELETE /api/interfaces/:id` - Remove interface
- `POST /api/interfaces/:id/test` - Test interface connectivity

System

- `GET /health` - System health check
- `GET /` - API information and available endpoints

Data Flow Architecture

Flow Execution Lifecycle

1. **Design Phase**
   - User creates flow in visual editor
   - Frontend sends flow definition to backend API
   - Backend validates flow structure and node configuration
2. **Deployment Phase**
   - Flow Engine receives validated flow
   - Nodes are instantiated with their configurations
   - Connections between nodes are established
   - Flow is marked as "deployed" but not running
3. **Execution Phase**
   - Trigger event occurs (inject node, external event, API call)
   - Flow Engine starts execution from trigger node
   - Messages propagate through connected nodes
   - Each node processes input and produces output
   - Debug nodes log intermediate values
   - Interface nodes communicate with hardware
4. **Completion Phase**
   - Flow execution reaches terminal nodes
   - Execution metadata is logged to database
   - Analytics are updated
   - WebSocket clients receive completion notification

Message Passing

Messages passed between nodes contain:

- `payload` : Primary data being processed
- `topic` : Message categorization/routing
- `timestamp` : Message creation time
- `metadata` : Additional contextual information

Security Features

Current Implementation

- **CORS Protection:** Configurable allowed origins
- **Rate Limiting:** API request throttling (100 requests per 15 minutes)
- **Helmet.js:** HTTP header security
- **Input Validation:** Request parameter sanitization

Planned Enhancements

- **Authentication:** User login with JWT tokens
- **Authorization:** Role-based access control (RBAC)
- **TLS/SSL:** Encrypted communication
- **API Keys:** Service-to-service authentication

- **Audit Logging:** Track all user actions

Deployment Architecture

Single-Site Deployment

```
┌─────────────────────────────────────┐
│        UP Board (Ubuntu)            │
│                                     │
│  ┌───────────────────────────────┐  │
│  │   Docker Compose              │  │
│  │                               │  │
│  │  ┌────────┐  ┌────────────┐   │  │
│  │  │Backend │  │  Frontend  │   │  │
│  │  │(Node.js│  │   (React)  │   │  │
│  │  │ :3000) │  │  (Nginx)   │   │  │
│  │  └────────┘  └────────────┘   │  │
│  │                               │  │
│  │  ┌────────┐  ┌────────────┐   │  │
│  │  │ MQTT   │  │  Database  │   │  │
│  │  │Broker  │  │  (SQLite)  │   │  │
│  │  └────────┘  └────────────┘   │  │
│  └───────────────────────────────┘  │
│                                     │
│  Hardware Interfaces:               │
│  - USB (Radio, Serial)              │
│  - Zigbee Coordinator               │
│  - Bluetooth Adapter                │
│  - Ethernet (Modbus, OPC UA)        │
└─────────────────────────────────────┘
```

Multi-Site Deployment (Planned)

```
Site A (UP Board)          Site B (UP Board)          Site C (UP Board)
     │                          │                          │
     │                          │                          │
     └──────────────────────────┼──────────────────────────┘
                                │
                                │
                                ▼
                   ┌─────────────────────────┐
                   │  Central Aggregation    │
                   │        Server           │
                   │                         │
                   │  - PostgreSQL +         │
                   │    TimescaleDB          │
                   │  - Grafana Dashboards   │
                   │  - Analytics Engine     │
                   │  - Alert Management     │
                   └─────────────────────────┘
```

Use Cases

Manufacturing Floor Automation

- Monitor production line sensors via Modbus
- Control equipment based on sensor thresholds
- Send radio alerts to maintenance personnel
- Log all production events for quality analysis

Warehouse Operations

- Track inventory with Zigbee tags
- Automate lighting based on motion detection
- Coordinate forklift operations via radio
- Generate real-time inventory reports

Facility Management

- Monitor HVAC systems via BACnet/Modbus
- Control access points with Bluetooth credentials
- Alert security via radio on anomalies
- Analyze energy consumption patterns

Emergency Response

- Broadcast evacuation alerts via radio systems
- Monitor fire/smoke sensors in real-time
- Coordinate response teams with location tracking
- Log all emergency events for compliance

Security and Surveillance

- **Perimeter Monitoring:** Camera line crossing detection triggers alerts when perimeter breached
- **Dwelling Detection:** Alert security when person loiters in restricted area for >5 seconds
- **Intrusion Alerts:** Automatically notify guards via radio when camera detects intrusion in prohibited zones
- **Tamper Detection:** Trigger alarm and snapshot when camera is obscured or vandalized
- **Integration Example:**

```
[Camera Event: Line Crossing] → [Function: Assess Threat] → [Radio Alert: Security Team]
                              ↓
                         [Snapshot Capture] → [Log to Database]
```

Retail Loss Prevention

- **Abandoned Object Detection:** Alert staff when bag/package left unattended for >2 minutes
- **Queue Management:** Count people crossing into queue area, alert manager when threshold exceeded
- **After-Hours Monitoring:** Motion detection after closing time triggers security response
- **Example Flow:**

```
[Camera: Dwelling in Closed Area] → [Check Time] → [After Hours?] → [Radio Alert + Snapshot]
```

Industrial Safety

- **Restricted Zone Monitoring:** Camera intrusion detection in hazardous areas triggers evacuation alarm
- **PPE Compliance:** Face detection without hard hat triggers supervisor alert
- **Equipment Tampering:** Missing object detection on critical equipment triggers maintenance alert
- **Safety Integration:**

```
[Camera: Intrusion in Danger Zone] → [Emergency Stop Equipment] → [Radio Broadcast]
                                    ↓
                            [Log Incident] → [Compliance Report]
```

Warehouse and Logistics

- **Loading Dock Monitoring:** Line crossing detection counts trucks entering/exiting
- **Dwell Time Tracking:** Monitor how long vehicles remain at loading bays
- **Unauthorized Access:** After-hours motion detection in restricted inventory areas
- **Workflow Example:**

```
[Camera: Line Crossing at Gate] → [Increment Counter] → [Update Dashboard]
                                 ↓
                            [Over Capacity?] → [Alert Supervisor]
```

Worker Safety and Compliance

- **Check-In Buttons:** Workers press Zigbee button when entering hazardous zone
- **Emergency Stop:** Triple-click Zigbee button triggers emergency equipment shutdown
- **Supervisor Alert:** Hold button for 5 seconds to request immediate supervisor assistance
- **Workflow Example:**

```
[Zigbee Button: Triple Click] → [Emergency Stop Equipment] → [Radio Broadcast: All Clear Zone]
                              ↓
                         [Log Safety Event] → [Notify Management]
```

Smart Building Control

- **Scene Control:** IKEA TRADFRI 5-button remote controls lighting scenes across warehouse
- **HVAC Adjustment:** Button press cycles through temperature presets
- **After-Hours Shutdown:** Single button press at exit triggers facility-wide shutdown sequence
- **Multi-Button Integration:**

```
[Zigbee Button: Button 1] → [Scene: Full Lighting]
[Zigbee Button: Button 2] → [Scene: Reduced Lighting]
[Zigbee Button: Button 3] → [Scene: Emergency Lighting]
[Zigbee Button: Button 4] → [Scene: All Off]
```

Mobile Worker Coordination

- **Location Check-In:** Worker presses button at checkpoint, logs location and timestamp
- **Task Completion:** Double-click confirms task completion, updates workflow
- **Help Request:** Hold button triggers alert to nearest supervisor with GPS location
- **Integration Flow:**

```
[Zigbee Button: Single Press] → [Log Check-In] → [Update Worker Location]
                              ↓
                         [Camera Snapshot] → [Verify Presence]
```

System Requirements

Hardware (Per UP Board)

- **Processor:** Intel Atom x5-Z8350 or better
- **RAM:** 4GB minimum, 8GB recommended
- **Storage:** 32GB minimum, 64GB+ recommended
- **Network:** Gigabit Ethernet
- **USB Ports:** 3+ (for radio/serial interfaces)
- **Optional:** Zigbee/Bluetooth dongles

Software

- **OS:** Ubuntu 22.04 LTS or newer
- **Runtime:** Node.js 20+, npm 11+
- **Containerization:** Docker 24+, Docker Compose 2+
- **Database:** SQLite 3 (local), PostgreSQL 15+ (central)
- **MQTT Broker:** Mosquitto 2.0+ (optional)

Network

- **Bandwidth:** 10 Mbps minimum per site
- **Latency:** <100ms to central server (for multi-site)
- **Ports Required:**
  - 3000 (Backend API)
  - 5173/80 (Frontend)
  - 1883 (MQTT, optional)
  - Custom ports for industrial protocols

Future Enhancements

AI-Assisted Flow Creation

- **Natural Language Processing:** "Create a flow that monitors temperature and sends alerts"
- **Flow Recommendations:** Suggest optimizations based on usage patterns
- **Anomaly Detection:** Automatically identify unusual system behavior
- **Predictive Maintenance:** Forecast equipment failures based on historical data

Advanced Analytics

- **Machine Learning Models:** Pattern recognition in operational data
- **Predictive Analytics:** Forecast future trends and issues
- **Correlation Analysis:** Identify relationships between variables
- **Custom Dashboards:** User-configurable visualization layouts

Extended Protocol Support

- **BACnet:** Building automation integration
- **Profibus/Profinet:** Siemens PLC communication

- **EtherCAT:** Real-time industrial Ethernet
- **CAN Bus:** Automotive and industrial vehicle communication

Mobile Application

- **iOS/Android Apps:** Native mobile monitoring and control
- **Push Notifications:** Real-time alerts on mobile devices
- **Offline Mode:** Limited functionality without network
- **QR Code Configuration:** Quick device provisioning

Support and Maintenance

Logging and Diagnostics

- **Log Levels:** Configurable verbosity (error, warn, info, debug)
- **Log Rotation:** Automatic archival and cleanup
- **Remote Logging:** Forward logs to syslog/central server
- **Health Endpoints:** `/health` API for monitoring tools

Backup and Recovery

- **Database Backups:** Automatic scheduled backups
- **Configuration Export:** JSON export of all system settings
- **Flow Versioning:** Track changes to automation workflows
- **Disaster Recovery:** Restore from backup procedures

Updates and Upgrades

- **Docker-Based Updates:** Pull new container images
- **Rolling Updates:** Zero-downtime deployment
- **Database Migrations:** Automatic schema updates
- **Backward Compatibility:** Maintain flow compatibility across versions

---

Conclusion

The Industrial Automation System provides a robust, scalable platform for industrial automation needs. Its visual flow-based approach lowers the barrier to automation while maintaining the flexibility required for complex industrial environments. With comprehensive logging, analytics, and planned AI enhancements, the system grows with organizational needs from single-site deployments to enterprise-wide operations.

Document Control

Version	Date	Author	Changes
1.0	Nov 10, 2025	System Documentation	Initial functional description

For Technical Support:

- Documentation: See `/docs` folder
- API Reference: `http://localhost:3000/`
- Issue Tracking: Project repository