DuoCore™ IoT Platform

The IoT Platform Built for Reliability

DuoCore Platform Device

Dual ESP32 Architecture	**Factory Configuration**
Zero-Touch Deployment	**Full-Stack Ownership**
MQTT Bidirectional	**Complete Data Privacy**

Version 2.0
November 2025

Proprietary and Confidential

Executive Summary

This platform is a sophisticated dual-microcontroller IoT system built around two ESP32 chips on a single PCB, designed for flexible deployment across multiple product applications. The architecture separates concerns between internet connectivity and local sensing, enabling powerful build-once-deploy-anywhere manufacturing with factory-based project assignment and over-the-air (OTA) updates.

Key Innovation: Devices are manufactured with generic boot firmware, then configured in the factory with specific application software based on a server-side demand database. Factory WiFi credentials are erased before shipping, and customers simply power on the device and provision it to their WiFi network via a mobile-friendly web interface.

System Architecture

Hardware Overview

The PCB contains two ESP32 microcontrollers with distinct roles:

• **Processor ESP32** - Primary Role: Internet connectivity and cloud communication

- WiFi for HTTP/MQTT connectivity
- Serial UART to Scanner ESP32
- MQTT bidirectional communication with cloud
- HTTP API interactions and project assignment coordination

• **Scanner ESP32** - Primary Role: Local sensing and scanning operations

- WiFi radio for promiscuous monitoring
- BLE scanning and ESP-NOW for external sensor communication
- Serial UART to Processor ESP32
- WiFi device detection and local sensor data collection

Why Two ESP32s?

- Radio Isolation: Dedicated radios prevent interference between scanning and connectivity
- Fixed Channel Operation: Scanner maintains fixed channel for ESP-NOW, eliminating channel-hopping issues
- Processing Separation: CPU-intensive scanning doesn't impact cloud communication
- Reliability: If scanning operations crash, internet connectivity remains stable (and vice versa)
- Flexibility: Scanner can operate in various modes without affecting processor

Full-Stack Ownership & Data Sovereignty

Built from the Ground Up

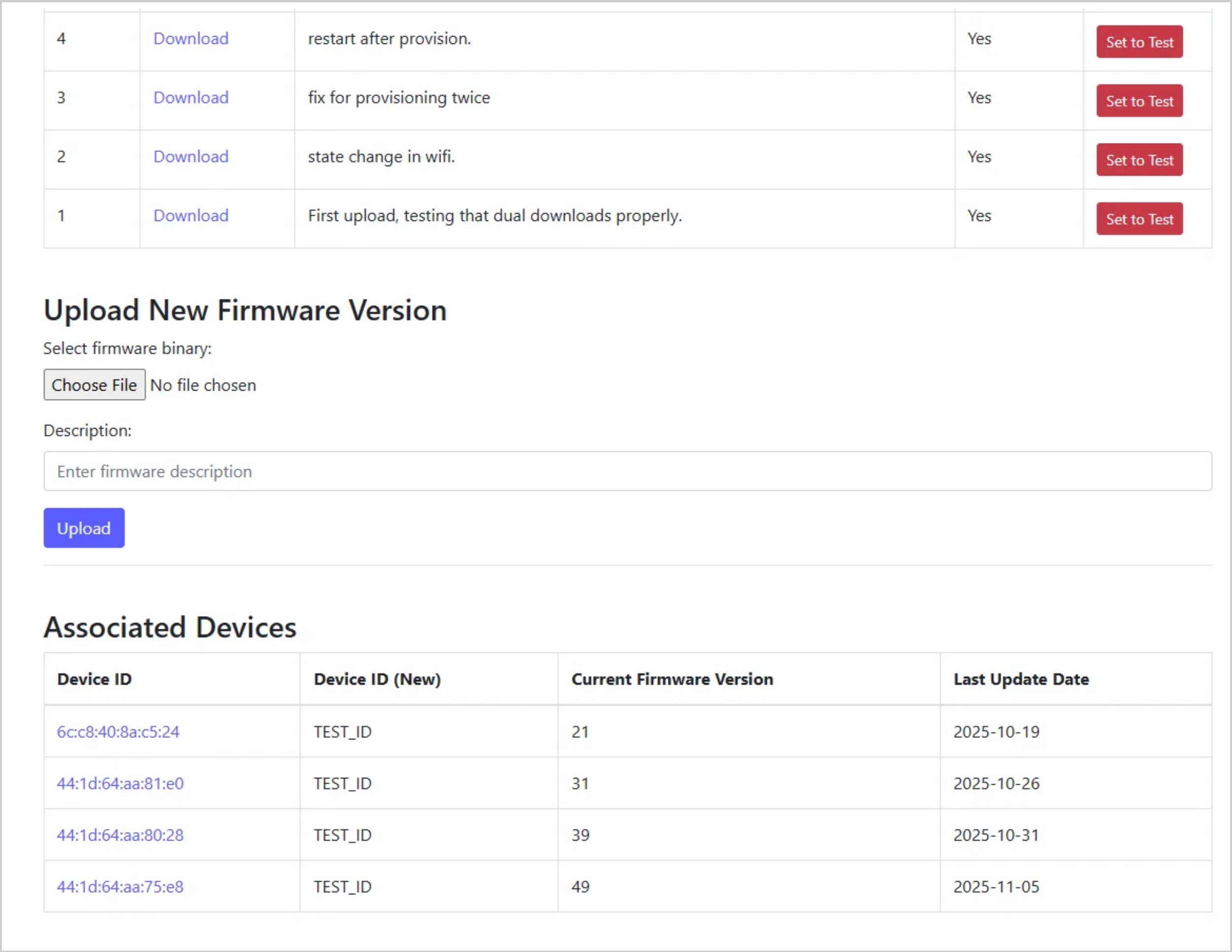
The DuoCore platform represents complete vertical integration - every layer of the stack has been custom-built in-house:

Firmware Layer (C)

- Boot firmware and provisioning system
- Inter-ESP communication protocol
- MQTT client implementation and OTA update system

Backend Infrastructure (PHP/MySQL)

- Device registration and management APIs
- Project assignment and demand tracking
- OTA firmware distribution servers



4	Download	restart after provision.	Yes	Set to Test
3	Download	fix for provisioning twice	Yes	Set to Test
2	Download	state change in wifi.	Yes	Set to Test
1	Download	First upload, testing that dual downloads properly.	Yes	Set to Test

Upload New Firmware Version

Select firmware binary:

Choose File No file chosen

Description:

Enter firmware description

Upload

Associated Devices

Device ID	Device ID (New)	Current Firmware Version	Last Update Date
6c:c8:40:8a:c5:24	TEST_ID	21	2025-10-19
44:1d:64:aa:81:e0	TEST_ID	31	2025-10-26
44:1d:64:aa:80:28	TEST_ID	39	2025-10-31
44:1d:64:aa:75:e8	TEST_ID	49	2025-11-05

Figure 3: OTA Firmware Management - Version control, firmware deployment, and device tracking

Communication Infrastructure

- Self-hosted MQTT broker with custom protocol design

Zero Third-Party Dependencies

■ No AWS IoT Core

■ No Azure IoT Hub

■ No Google Cloud IoT

■ No commercial MQTT services

■ No vendor-supplied SDKs or libraries

■ Own all source code

■ Host all infrastructure

■ Control all data flows

■ Set our own roadmap

Business & Technical Advantages

Cost Control	Data Privacy
No per-device fees	Complete data sovereignty
No usage-based charges	Customer data isolation
Predictable infrastructure costs	Privacy compliance (GDPR, HIPAA)
No licensing fees	No third-party data mining

Why This Matters: Most IoT platforms depend on commercial cloud services that charge per device, per message, or per API call. These costs can become prohibitive at scale, and companies have no control over pricing changes, feature deprecation, or how their data is used.

By building the entire stack in-house, DuoCore provides predictable economics for customers deploying at scale, complete privacy for sensitive applications, unlimited flexibility to adapt to changing requirements, and long-term stability independent of vendor business models.

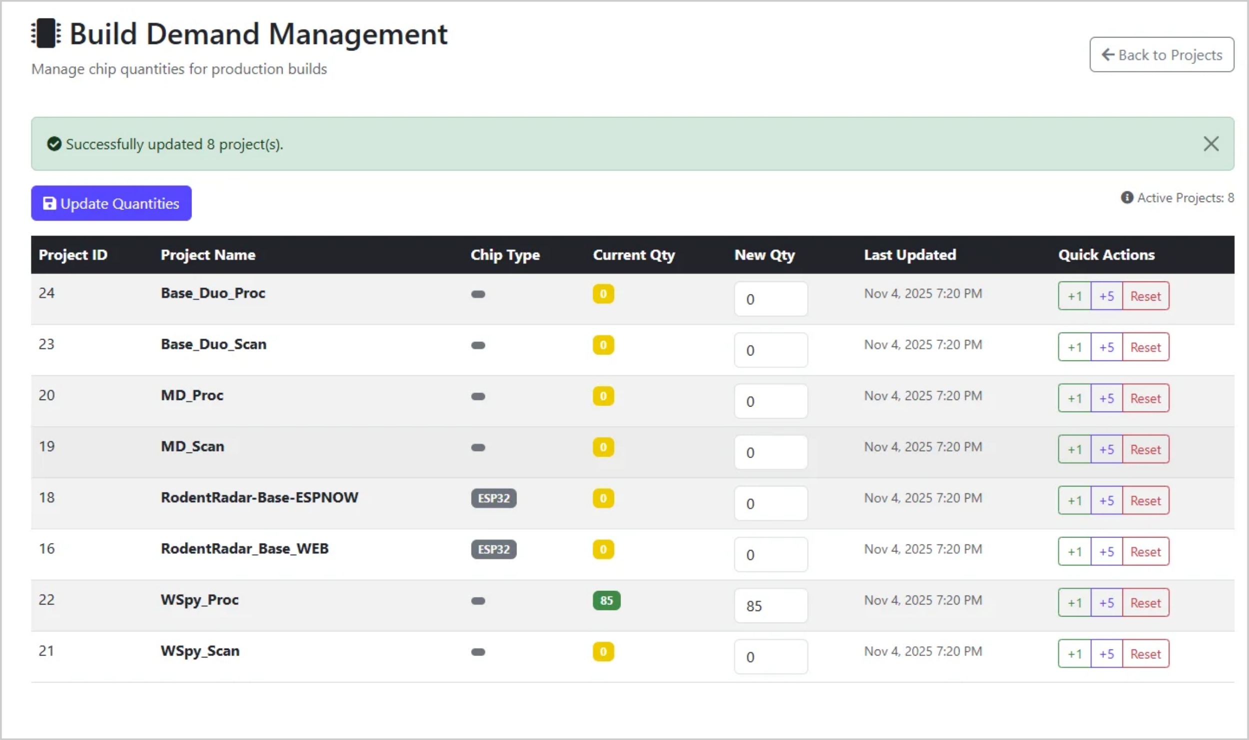
Manufacturing and Deployment Flow

Step 1: Factory Assembly

PCB assembled with two ESP32s and components. Both programmed with generic boot firmware containing factory WiFi credentials. Device marked as 'first boot' in flash memory.

Step 2: Factory Configuration

Device powered on in factory. Processor detects first boot, connects to factory WiFi, and calls server API for project assignment. Server responds with project ID based on current demand and decrements demand counter.



Build Demand Management

Manage chip quantities for production builds

✓ Successfully updated 8 project(s).

Active Projects: 8

Project ID	Project Name	Chip Type	Current Qty	New Qty	Last Updated	Quick Actions
24	Base_Duo_Proc	▬	0	0	Nov 4, 2025 7:20 PM	+1 +5 Reset
23	Base_Duo_Scan	▬	0	0	Nov 4, 2025 7:20 PM	+1 +5 Reset
20	MD_Proc	▬	0	0	Nov 4, 2025 7:20 PM	+1 +5 Reset
19	MD_Scan	▬	0	0	Nov 4, 2025 7:20 PM	+1 +5 Reset
18	RodentRadar-Base-ESPNOW	ESP32	0	0	Nov 4, 2025 7:20 PM	+1 +5 Reset
16	RodentRadar_Base_WEB	ESP32	0	0	Nov 4, 2025 7:20 PM	+1 +5 Reset
22	WSpy_Proc	▬	85	85	Nov 4, 2025 7:20 PM	+1 +5 Reset
21	WSpy_Scan	▬	0	0	Nov 4, 2025 7:20 PM	+1 +5 Reset

Figure 2: Build Demand Management - Server interface showing current demand for each project. The device queries this system to determine which project to become (e.g., WSpy_Proc has demand of 85).

Step 3: Factory OTA Update

Processor registers device with MQTT system, downloads project-specific firmware. Scanner receives project assignment via UART and downloads its firmware. Both ESPs reboot into production firmware.

Step 4: Factory Finalization

Device completes boot, performs self-test, confirms MQTT registration. Factory WiFi credentials are ERASED from memory. Device enters idle mode, ready to ship.

Step 5: Customer Site - First Power On

Customer unboxes and powers on device. Device detects no WiFi credentials and enters PROVISIONING MODE. Creates temporary WiFi Access Point 'Device-XXXXXX-Setup' with captive portal web interface.

Step 6: Customer Provisioning

Customer connects phone/laptop to device AP. Web browser opens to setup page. Customer selects WiFi network and enters password. Device validates credentials and saves to flash, then reboots.

Step 7: Normal Operation

Device connects to customer WiFi. Processor establishes MQTT connection. Scanner begins scanning. Device operates as assigned project with bidirectional cloud communication and ESP-NOW sensor support.

Manufacturing Advantages

Traditional Approach Problems:

- ■ Must know end customer at manufacturing time
- ■ Inventory management by product type
- ■ Cannot reassign units if demand forecast changes
- ■ Complex SKU management and tracking
- ■ Customer must configure WiFi (support burden)
- ■ Pre-configured WiFi credentials create security risks

DuoCore Platform Solution:

- ■ **Build and Configure in Factory:** Complete setup before shipping
- ■ **Demand-Based Assignment:** Server automatically assigns projects based on current orders
- ■ **Single SKU Inventory:** All units identical until configuration
- ■ **Zero Customer Setup:** Power on, connect WiFi via phone, done
- ■ **No Security Risks:** Factory credentials erased before shipping
- ■ **Flexible Production:** Respond instantly to demand changes

Business Benefits:

Manufacturing:

- Reduced Inventory Costs: One SKU instead of multiple variants
- Flexible Production: Build to forecast, assign during configuration
- Quality Control: Single test procedure for all units
- Rapid Response: Change production mix without retooling

Logistics:

- Simplified Warehousing: Generic configured units ready to ship
- No Waste: Unsold units can be reconfigured for different projects
- Fast Fulfillment: Configure and ship in single factory operation

Customer Experience:

- Plug-and-Play: Customer only connects power and WiFi
- Professional Setup: No exposed technical configuration

- Reliable Operation: Fully tested and configured before shipping

Infrastructure Economics:

- No Per-Device Cloud Fees: Self-hosted infrastructure eliminates AWS/Azure costs
- Predictable Scaling: Infrastructure costs don't multiply with device count
- Data Privacy Control: Complete sovereignty over customer data

MQTT Bidirectional Communication

One of the platform's most powerful features is the integrated MQTT system that provides real-time bidirectional communication between devices and the cloud.

MQTT Features

Cloud-to-Device Communication:

- Remote configuration changes
- State control commands
- OTA update triggers
- Device queries and diagnostics

Device-to-Cloud Communication:

- Real-time sensor alerts (via MQTT)
- Status updates and heartbeats (via MQTT)
- Detection events (via MQTT for urgent, HTTP batch for historical)
- Health monitoring data

Data Upload Strategy:

- **MQTT for Real-Time:** Urgent alerts, state changes, and immediate notifications use MQTT for instant delivery
- **HTTP Batch Upload:** Less time-sensitive data (historical logs, aggregate statistics, bulk sensor readings) is batched on the Processor and uploaded via HTTP to PHP/MySQL backend
- **Optimal Protocol Selection:** Reduces MQTT traffic, minimizes bandwidth, and allows efficient bulk data storage

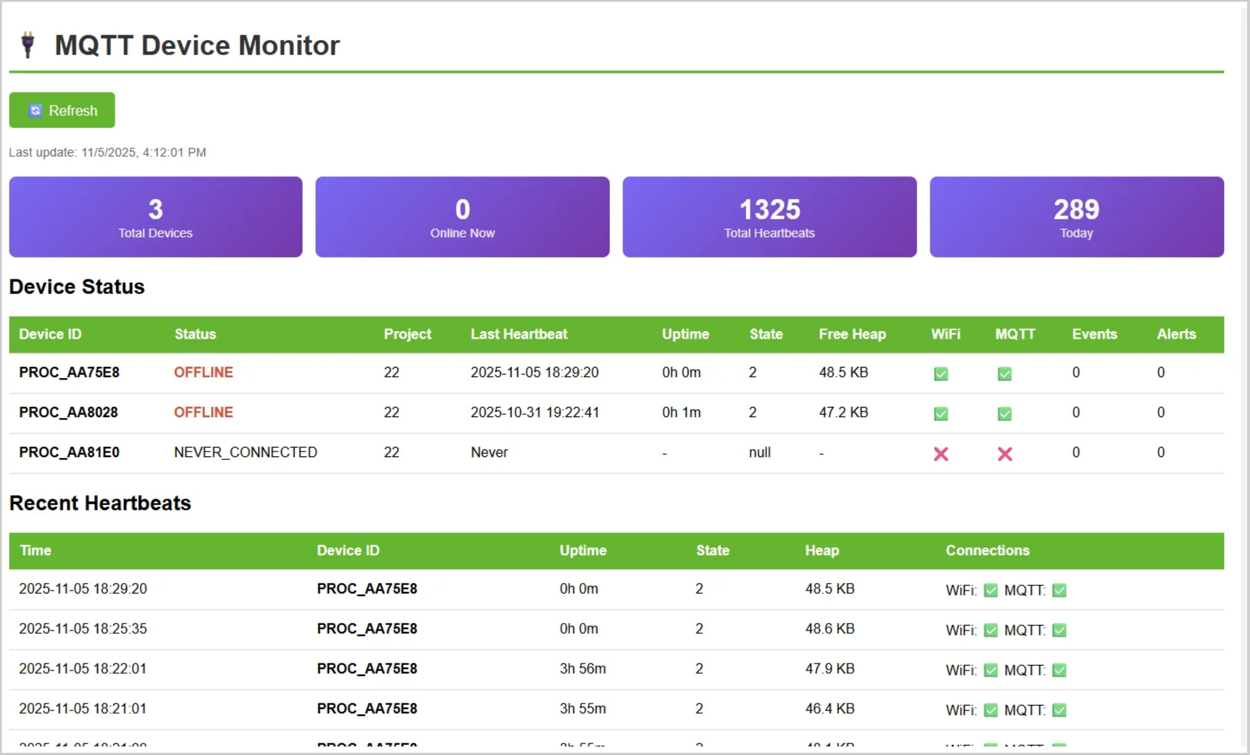
Cloud Processing Capabilities:

- Backend can handle data processing and analysis
- Alerts generated centrally from aggregated device data
- Reduces processing load on resource-constrained devices
- Enables sophisticated multi-device correlation and logic

ESP-NOW External Sensor Network

The Scanner ESP32 can communicate with external sensors via ESP-NOW protocol, enabling low-power wireless sensor networks including motion sensors, environmental sensors, door/window sensors, and any ESP-based peripheral device.

Data Flow: External sensor detects event via ESP-NOW → Scanner ESP32 receives and processes data → Scanner forwards to Processor via UART → Processor publishes to cloud via MQTT → Cloud can respond with commands via MQTT back through the chain.



MQTT Device Monitor

Last update: 11/5/2025, 4:12:01 PM

3 Total Devices

0 Online Now

1325 Total Heartbeats

289 Today

Device Status

Device ID	Status	Project	Last Heartbeat	Uptime	State	Free Heap	WiFi	MQTT	Events	Alerts
PROC_AA75E8	OFFLINE	22	2025-11-05 18:29:20	0h 0m	2	48.5 KB	✓	✓	0	0
PROC_AA8028	OFFLINE	22	2025-10-31 19:22:41	0h 1m	2	47.2 KB	✓	✓	0	0
PROC_AA81E0	NEVER_CONNECTED	22	Never	-	null	-	✗	✗	0	0

Recent Heartbeats

Time	Device ID	Uptime	State	Heap	Connections
2025-11-05 18:29:20	**PROC_AA75E8**	0h 0m	2	48.5 KB	WiFi: ✓ MQTT: ✓
2025-11-05 18:25:35	**PROC_AA75E8**	0h 0m	2	48.6 KB	WiFi: ✓ MQTT: ✓
2025-11-05 18:22:01	**PROC_AA75E8**	3h 56m	2	47.9 KB	WiFi: ✓ MQTT: ✓
2025-11-05 18:21:01	**PROC_AA75E8**	3h 55m	2	46.4 KB	WiFi: ✓ MQTT: ✓

Figure 1: MQTT Device Monitor Dashboard - Real-time monitoring of device status, heartbeats, and connectivity

Security Considerations

Boot Security: Factory WiFi credentials erased after project download. Secure boot options available. Certificate-based OTA verification.

MQTT Security: TLS/SSL encryption for all MQTT traffic. Device-specific credentials. Topic-based access control. Credential rotation capability.

OTA Security: Firmware signature verification. Encrypted firmware downloads. Rollback on verification failure. Version downgrade protection.

Monitoring and Diagnostics

Device Health Monitoring:

- Periodic heartbeat via MQTT
- Memory usage tracking
- WiFi signal strength
- Communication link quality
- Uptime and reboot tracking

Remote Diagnostics:

- MQTT command for state queries
- Log level adjustment via MQTT
- Remote configuration updates
- Scanner calibration commands

Conclusion

This dual ESP32 platform represents a sophisticated IoT infrastructure that solves real manufacturing and deployment challenges. By separating concerns between connectivity and sensing, implementing flexible factory-configured OTA project assignment with customer-site WiFi provisioning, building the entire stack in-house without third-party dependencies, and providing bidirectional MQTT communication with ESP-NOW sensor networks, the platform enables:

- Streamlined Manufacturing: Build and configure generic units in factory, ship ready-to-deploy
- Zero-Touch Deployment: Customer simply powers on, connects WiFi via phone, device operational
- Complete Cost Control: No per-device fees or usage charges from third-party IoT platforms
- Total Data Privacy: Self-hosted infrastructure means complete data sovereignty
- Unlimited Optimization: Full-stack ownership enables tuning at every layer
- Flexible Multi-Product Platform: Multiple products from single hardware design
- Robust Operation: Isolated processors prevent cascade failures
- No Vendor Lock-in: Complete independence from commercial IoT platforms

The result is a powerful, flexible IoT platform that dramatically reduces inventory complexity, eliminates customer setup complexity, provides complete cost and privacy control, and enables rapid deployment of new products and features.

What makes this platform unique:

- Factory configuration eliminates shipping logistics complexity
- Built-in WiFi provisioning means no customer technical knowledge required
- Dual-processor architecture provides unmatched reliability
- Full-stack ownership delivers cost predictability and optimization freedom
- Self-hosted infrastructure ensures complete data privacy and sovereignty
- MQTT bidirectional communication enables true IoT device management
- Single hardware platform serves unlimited product variations