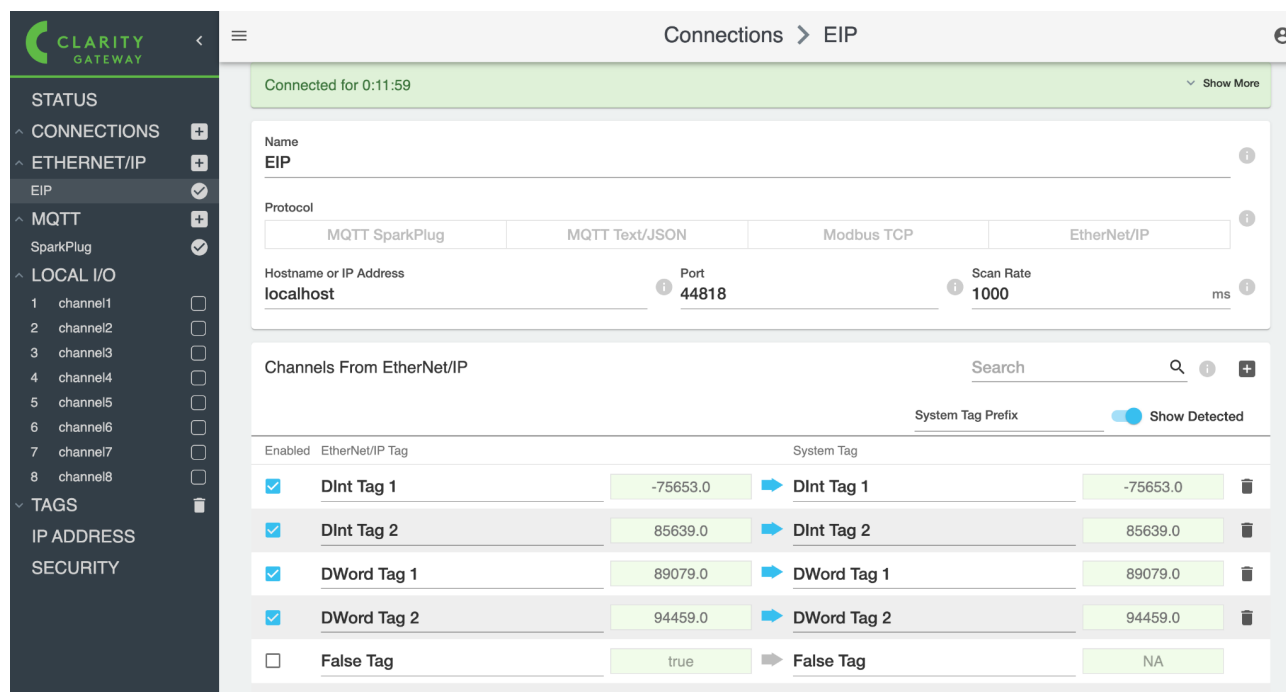


IoT Edge Gateway Software



Clarity Gateway makes it easy to move data from legacy devices to modern cloud applications. The fully responsive web application lets you explore the data available in your devices and control how it's sent to the cloud. Store and forward support prevents permanent data loss when your device's Internet connection temporarily goes down.

Supports Modern and Legacy Protocols

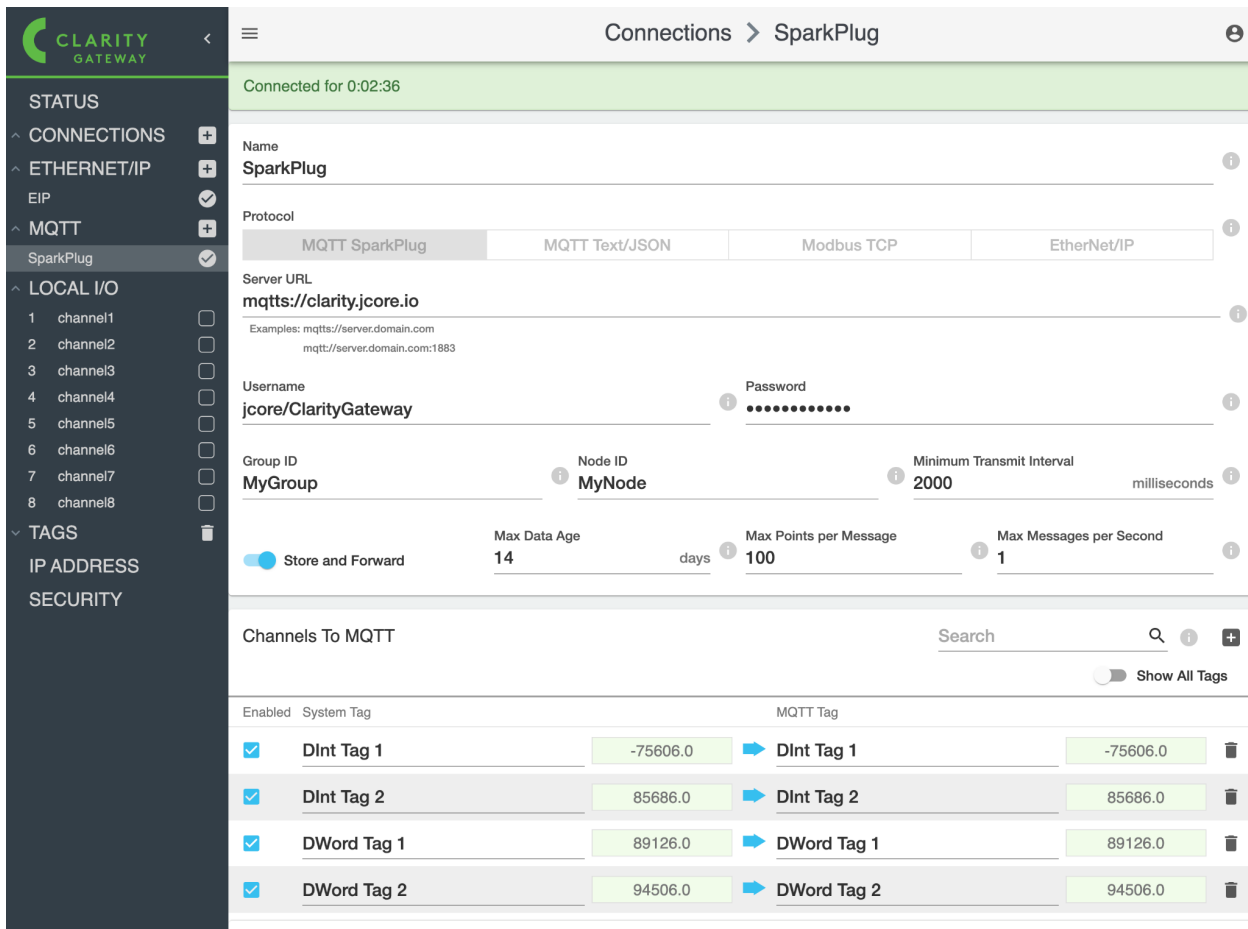
Clarity Gateway allows mapping between any of the supported protocols:

MQTT / SparkPlug	<ul style="list-style-type: none"> • Publishes standard SparkPlug birth and data messages • Supports store and forward
MQTT / JSON	<ul style="list-style-type: none"> • Publishes data using standard human-readable JSON • Supports store and forward
EtherNet / IP	<ul style="list-style-type: none"> • Automatically detects tags available in device

	<ul style="list-style-type: none"> • Browse tags and select which ones to import • Supports struct and array types
Modbus / TCP	<ul style="list-style-type: none"> • Supports Input Registers, Output Registers, Discrete Inputs, and Discrete Outputs

Store and Forward

Clarity Gateway prevents data loss by detecting communications outages, storing data during the outage, and forwarding data when communications are restored. You control the maximum amount of data to store, and how quickly the stored data is sent when the device goes back online.



The screenshot shows the Clarity Gateway web interface. The left sidebar contains navigation options: STATUS, CONNECTIONS, ETHERNET/IP, MQTT, LOCAL I/O, TAGS, IP ADDRESS, and SECURITY. The main content area is titled 'Connections > SparkPlug' and shows the connection status 'Connected for 0:02:36'. The configuration details for the SparkPlug connection are as follows:

- Name:** SparkPlug
- Protocol:** MQTT SparkPlug (selected), MQTT Text/JSON, Modbus TCP, EtherNet/IP
- Server URL:** mqtt://clarity.jcore.io (Examples: mqtt://server.domain.com, mqtt://server.domain.com:1883)
- Username:** jcore/ClarityGateway
- Password:** [Redacted]
- Group ID:** MyGroup
- Node ID:** MyNode
- Minimum Transmit Interval:** 2000 milliseconds
- Max Data Age:** 14 days
- Max Points per Message:** 100
- Max Messages per Second:** 1
- Store and Forward:** Enabled (toggle switch)

Below the configuration details, there is a section titled 'Channels To MQTT' with a search bar and a 'Show All Tags' toggle. The table below lists the channels and their corresponding MQTT tags:

Enabled	System Tag	MQTT Tag
<input checked="" type="checkbox"/>	DInt Tag 1	DInt Tag 1
<input checked="" type="checkbox"/>	DInt Tag 2	DInt Tag 2
<input checked="" type="checkbox"/>	DWord Tag 1	DWord Tag 1
<input checked="" type="checkbox"/>	DWord Tag 2	DWord Tag 2

Hardware and OS Support

Clarity Gateway supports a wide range of edge computing environments, including Docker on ARM/Linux and Windows 10:

JCore Iron Pi	<ul style="list-style-type: none">● Officially supported on the JCore Iron Pi● Access Iron Pi local analog and digital I/O via Clarity Gateway
Docker on ARM/Linux	<ul style="list-style-type: none">● Runs in a modern Docker runtime● Requires ≥ 1GB RAM, ≥ 2GB available storage
Windows 10	<ul style="list-style-type: none">● Installed via MSI installer● Runs as a Windows Service

Licensing

Clarity Gateway license levels are based on the total number of connections. For example, if you were connected to two Modbus devices and had two upstream MQTT connections, you would have 4 total connections and you could use the CGW-MOD-4 license.

CGW-MOD-4	<ul style="list-style-type: none">● MQTT● Modbus● 4 connection limit
CGW-EIP-4	<ul style="list-style-type: none">● EtherNet/IP● MQTT● Modbus● 4 connection limit
CGW-EIP-10	<ul style="list-style-type: none">● EtherNet/IP● MQTT● Modbus● 10 connection limit
CGW-EIP-50	<ul style="list-style-type: none">● EtherNet/IP● MQTT● Modbus● 50 connection limit

- Licenses are perpetual, so the licensing system will not disable the version of the app you've licensed
- Licenses are not locked to a specific CPU or hardware ID, so you can replace failed hardware without re-generating license keys
- JCore tracks which license keys are in use so that we can detect license theft