

# SunFusion Energy Systems

## MQTT WebSocket Setup Guide for Solar Assistant

This guide walks you through enabling the MQTT WebSocket broker on Solar Assistant so the SunFusion Live Energy Monitor can connect and display real-time data.

### What You Need

Item	Details
Solar Assistant device	Raspberry Pi or similar running Solar Assistant
SSH access	Terminal (Mac/Linux) or PuTTY (Windows)
SSH credentials	Username: solar-assistant   Password: solar123
Site ID (cloud proxy)	Your Solar Assistant cloud address e.g. ellard.us.solar-assistant.io
Local network IP	Your Solar Assistant local IP (find in SA web UI → Settings → Network)

■ **Where to find your Site ID:** Log in to Solar Assistant web UI → go to **Settings** → **Remote Access**. Your Site ID is the address shown under "Cloud proxy" — it looks like `yourname.us.solar-assistant.io`

■ **Replace placeholders below with YOUR values:**

[[SITE-ID](#)] → Your Solar Assistant Site ID (e.g. `ellard.us.solar-assistant.io`)

[[YOUR-SA-IP](#)] → Your Solar Assistant local IP (e.g. `192.168.10.63`)

### Step 1 — Connect via SSH

You can SSH into Solar Assistant two ways — **either one works**. Use the Site ID from anywhere, or the local IP when on the same network.

Method	Command	When to Use
<b>Site ID (cloud proxy)</b>	<code>ssh solar-assistant@[SITE-ID]</code>	From anywhere (requires internet)
<b>Local IP</b>	<code>ssh solar-assistant@[YOUR-SA-IP]</code>	On same WiFi/LAN (faster, no internet needed)

**Example using Site ID (from anywhere):**

```
ssh solar-assistant@ellard.us.solar-assistant.io
```

Example using local IP (on same network):

```
ssh solar-assistant@192.168.10.63
```

When prompted for password, type: **solar123**

(The password will NOT appear on screen as you type — this is normal.)

■ **First time connecting?** You will see "The authenticity of host... can't be established." Type **yes** and press Enter to continue.

■ **Windows users:** If you don't have SSH, open **Windows Terminal** or **PowerShell** (built into Windows 10/11) — SSH is included. Alternatively, download **PuTTY** and enter the host and credentials there.

## Step 2 — Check Current MQTT Configuration

Once logged in, check if MQTT is already running:

```
sudo systemctl status mosquitto
```

You should see **active (running)** in green. If not, Solar Assistant may need an update.

Now check the current config:

```
cat /etc/mosquitto/mosquitto.conf
```

Look for any **listener** lines. By default, Solar Assistant runs MQTT on port **1883** (native MQTT protocol). The SunFusion monitor needs a **WebSocket** listener on port **9001**.

## Step 3 — Add WebSocket Listener on Port 9001

**This is the key step.** We need to add a WebSocket listener so the browser-based monitor can connect. Run this command (copy/paste the entire block):

```
sudo bash -c 'cat >> /etc/mosquitto/mosquitto.conf << EOF

# SunFusion WebSocket listener
listener 9001
protocol websockets
allow_anonymous true
EOF'
```

What this does:

- **listener 9001** — Opens port 9001 for connections
- **protocol websockets** — Makes it a WebSocket (browser-compatible) port
- **allow\_anonymous true** — No username/password required (local network only)

■ **Security note:** This allows any device on your local network to connect. For setups exposed to the internet, add authentication — see the Advanced section at the end.

## Step 4 — Verify the Configuration

---

Check the config looks correct:

```
cat /etc/mosquitto/mosquitto.conf
```

You should see these lines at the bottom:

```
# SunFusion WebSocket listener
listener 9001
protocol websockets
allow_anonymous true
```

■ **Tip:** The original `listener 1883` line should still be there too. Both ports run simultaneously — 1883 for native MQTT clients, 9001 for the web monitor.

## Step 5 — Restart the MQTT Broker

---

Restart Mosquitto to apply the changes:

```
sudo systemctl restart mosquitto
```

Verify it restarted successfully:

```
sudo systemctl status mosquitto
```

You should see `active (running)`. If you see `failed`, there is a typo in the config — go back to Step 3 and check the file.

Confirm port 9001 is listening:

```
sudo ss -tlnp | grep 9001
```

You should see output like:

```
LISTEN 0 128 0.0.0.0:9001 0.0.0.0:* users:(("mosquitto",pid=xxxx,fd=x))
```

## Step 6 — Test the Connection

Open the SunFusion monitor in your browser. You can test with either:

Method	URL / Address	Port
Local network	<code>http://[YOUR-SA-IP]</code>	9001
Cloud proxy (Site ID)	<code>https://[SITE-ID]</code>	9001
File on PC	Open <code>monitor.html</code> directly from Downloads folder	9001

On the splash screen (or Fleet Manager → Add System), enter:

**Solar Assistant IP:** `[YOUR-SA-IP]`

**Port:** 9001

Click **START MONITORING**. If connected, the status pill turns green and says **LIVE**.

■ **The SunFusion monitor auto-tries both ports.** It will attempt 9001 first. If that fails within 8 seconds, it automatically tries 1883. Whichever works gets saved.

## Troubleshooting

Problem	Solution
<b>Cannot SSH via local IP</b>	Make sure you are on the same WiFi/network as Solar Assistant. Try pinging the IP first: <code>ping [YOUR-SA-IP]</code> If that fails, try SSH via Site ID instead.
<b>Cannot SSH via Site ID</b>	Check that Remote Access is enabled in Solar Assistant. Verify your Site ID at Settings → Remote Access. Make sure device has internet connection.
<b>Mosquitto failed to start</b>	Check config for typos: <code>sudo nano /etc/mosquitto/mosquitto.conf</code> Look for misspellings, extra spaces, or missing lines.
<b>Port 9001 not listening</b>	Check if another program is using it: <code>sudo ss -tlnp   grep 9001</code> If empty, restart: <code>sudo systemctl restart mosquitto</code>
<b>Monitor says "Connecting..." but never connects</b>	Verify IP is correct and both devices are on same network. Check browser console (F12) for WebSocket errors. Try port 1883 in the monitor settings.
<b>Monitor connects but no data appears</b>	Solar Assistant may not be publishing MQTT topics. Check SA web interface → Settings → MQTT is enabled. Test: <code>mosquitto_sub -h localhost -t "solar_assistant/#" -v</code>

**Works on local but not from cloud proxy**

Cloud proxy forwards HTTP only, not WebSocket.  
For remote MQTT, use VPN or port forwarding.  
See Advanced section below.

## Advanced — Adding MQTT Authentication

If the Solar Assistant device is exposed to the internet (port forwarded), add a username and password:

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd sunfusion
```

Enter a password when prompted. Then update the config:

```
sudo nano /etc/mosquitto/mosquitto.conf
```

Change `allow_anonymous true` to:

```
allow_anonymous false
password_file /etc/mosquitto/passwd
```

Restart Mosquitto:

```
sudo systemctl restart mosquitto
```

Then in the SunFusion monitor settings, enter the username `sunfusion` and the password you created.

## Quick Reference — Copy/Paste Commands

Action	Command
SSH via Site ID	<code>ssh solar-assistant@[SITE-ID]</code>
SSH via local IP	<code>ssh solar-assistant@[YOUR-SA-IP]</code>
Password	<code>solar123</code>
Add WebSocket port	<pre>sudo bash -c 'cat &gt;&gt; /etc/mosquitto/mosquitto.conf &lt;&lt; EOF listener 9001 protocol websockets allow_anonymous true EOF'</pre>
Restart MQTT	<code>sudo systemctl restart mosquitto</code>
Check status	<code>sudo systemctl status mosquitto</code>
Confirm port open	<code>sudo ss -tlnp   grep 9001</code>
View config	<code>cat /etc/mosquitto/mosquitto.conf</code>
Edit config	<code>sudo nano /etc/mosquitto/mosquitto.conf</code>

<b>Test MQTT topics</b>	<code>mosquitto_sub -h localhost -t "solar_assistant/#" -v</code>
<b>View MQTT logs</b>	<code>sudo journalctl -u mosquitto -f</code>