

Firefly alongside Tasmota (shared MQTT broker)

Firefly does not emulate Tasmota devices; it integrates with them the standard way - both talk to the same MQTT broker, each with its own topic scheme. Anything subscribed to the broker (Tasmota switches, Node-RED, Home Assistant) can read Firefly sensors and drive Firefly outputs, and vice versa.

The topic map

- Firefly publishes panel states (default `stat/<panel>/STATE`, configurable template) and `tele/<device>/LWT` availability.
- Firefly listens on `cmd/<device>/<panel>` for `ON`, `OFF`, `TOGGLE` or numeric payloads - publish there from a Tasmota rule, Node-RED flow or `mosquitto_pub` to drive relays, servos and timers.
- Set the broker in dashboard Settings (`mqtt_server`, `mqtt_user`, `mqtt_pass`) and enable per-panel **MQTT** buttons for exactly the panels you want shared.

Design philosophy: local first

- **Network up:** full remote control, cross-device automation, history and dashboards.
- **Network down:** Firefly rules, timers and programs keep running on the board; nothing safety-relevant should depend on the broker.
- **Broker down but LAN up:** the dashboard HTTP API (`/api/dashboard`, `/api/event`) still controls everything directly.
- **Power loss:** configuration is persisted and resumes where it was (timers restart per their saved state).

Example: a pump + float-switch loop runs as Firefly rules locally (works through outages), while a Tasmota switch elsewhere publishes to `cmd/<device>/pump` for manual override when the network is healthy.