

U1 — UMIDIFICADOR 1 (com potência)

Tópicos MQTT

Tipo	Tópico	Payload	Observações
Publi	alissondev007/umidificador/ ca umidade	número (float, ex: "72.5")	Umidade lida pelo sensor SHT30
Publi	alissondev007/umidificador/ ca status	"Ligado", "Desligado", "Sistema desativado"	Estado da tomada e/ou do sistema
Publi	alissondev007/umidificador/ ca potencia	"1", "2", "3"	Nível atual de potência
Publi	alissondev007/umidificador/ ca config-atual	{"umidadeMinima":85,"umidadeMaxima":90}	Limits salvos
Rece	alissondev007/umidificador/ be comando	"potencia1", "potencia2", "potencia3", "ativar", "desligar"	Ajusta potência ou ativa/desativa sistema
Rece	alissondev007/umidificador/ be config	JSON igual ao /config-atual	Atualiza limites e salva em NVS

Funcionamento interno

- Quando o loop detecta umidade < umidadeMinima → aciona o relé por pulsos, simulando botão físico, de acordo com a potência.
- Quando umidade >= umidadeMaxima → desliga o relé com a quantidade de pulsos necessária.

- Quando recebe "desligar":
 - sistemaAtivo = false;
 - não executa mais a lógica de controle;
 - publica status = "Sistema desativado".
 - Quando recebe "ativar":
 - sistemaAtivo = true;
 - retoma a lógica automática;
 - publica status = "Ligado" ou "Desligado" conforme o relé.
-

O que o backend precisa fazer

→ Assinar (subscribe)

alissondev007/umidificador/umidade

alissondev007/umidificador/status

alissondev007/umidificador/potencia

alissondev007/umidificador/config-atual

→ Publicar (publish)

alissondev007/umidificador/comando

alissondev007/umidificador/config

→ Mensagens que o backend pode enviar

Ação	Tópico	Payload	Descrição
Mudar potência	/comando	"potencia1", "potencia2", "potencia3"	Envia pulso de troca de modo
Desligar sistema	/comando	"desligar"	Desativa o loop (sistemaAtivo = false)
Ligar sistema	/comando	"ativar"	Reativa o loop (sistemaAtivo = true)
Configurar limites	/config	{"umidadeMinima": 80, "umidadeMaxima": 90}	Salva limites na flash e publica /config-atual

U2 — UMIDIFICADOR 2 (SHT30 simples)

Tópicos MQTT

Tipo	Tópico	Payload	Observações
Publi	alissondev007/umidificador/31a7 ca dbcc/umidade	número (float, ex: "55.2")	Umidade atual
Publi	alissondev007/umidificador/31a7 ca dbcc/status	"1" ou "0"	1 = relé ligado, 0 = relé desligado
Publi	alissondev007/umidificador/31a7 ca dbcc/config-atual	{"umidadeMinima":50,"umidadeMaxima":60}	Limites salvos
Rece	alissondev007/umidificador/31a7 be dbcc/comando	"LIGAR" ou "DESLIGAR"	Ativa/desativa o loop automático
Rece	alissondev007/umidificador/31a7 be dbcc/config	JSON igual ao /config-atual	Atualiza limites

Funcionamento interno

- Lê o sensor SHT30 a cada 5 segundos.
- Quando sistemaAtivo == true, controla o relé conforme a umidade.
- Quando sistemaAtivo == false, força o relé **OFF** e pausa a lógica de controle.
- Não possui controle de potência.
- Publica status do relé, **não** do sistema automático.
→ portanto, o **estado “sistema ativo/inativo”** deve ser controlado e lembrado pelo backend.

O que o backend precisa fazer

→ Assinar (subscribe)

alissondev007/umidificador/31a7dbcc/umidade

alissondev007/umidificador/31a7dbcc/status

alissondev007/umidificador/31a7dbcc/config-atual

→ Publicar (publish)

alissondev007/umidificador/31a7dbcc/comando

alissondev007/umidificador/31a7dbcc/config

→ Mensagens que o backend pode enviar

Ação	Tópico	Payload	Descrição
Ligar sistema automático	/comando "LIGAR"		Ativa sistemaAtivo = true
Desligar sistema automático	/comando "DESLIGAR"		Desativa loop e relé
Atualizar limites	/config	{"umidadeMinima": 55, "umidadeMaxima": 65}	