

DOCUMENTACIÓN TÉCNICA: GATEWAY MQTT/WEB SOCKETS ESP32

ÍNDICE








1. Descripción General
2. Arquitectura del Sistema
3. Configuración y Conexiones
4. Flujo de Operación
5. Mensajes MQTT
6. Estados del Sistema
7. Configuración Técnica

DESCRIPCIÓN GENERAL

¿Qué es este sistema?

Este código implementa un **Gateway IoT inteligente** basado en ESP32 que se conecta a servidores MQTT a través de **WebSockets**, permitiendo la comunicación bidireccional de dispositivos IoT con sistemas backend.

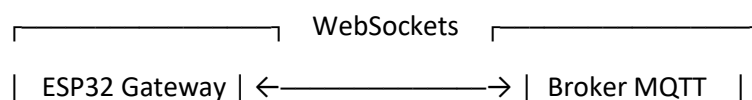
Características Principales

-  **Conexión MQTT sobre WebSockets** (puerto 8083)
-  **Modo AP para configuración** inicial WiFi
-  **Reconexión automática** inteligente
-  **Gestión de estados** robusta
-  **Publicación periódica** de datos y estado
-  **Interfaz LCD** para feedback visual
-  **Configuración persistente** en memoria flash

ARQUITECTURA DEL SISTEMA

Componentes Principales

text



	(Puerto 8083)	
- WiFi STA/AP	- EMQX Cloud	
- WebSocket	- HiveMQ	
- MQTT Manual	- Mosquitto	
- LCD Display		

Tecnologías Utilizadas

- **ESP32:** Microcontrolador principal
- **WebSockets:** Protocolo de comunicación
- **MQTT:** Protocolo de mensajería IoT
- **WiFi:** Conexión de red
- **Preferences:** Almacenamiento persistente
- **ArduinoJson:** Manipulación de JSON

CONFIGURACIÓN Y CONEXIONES

Configuración WiFi

El sistema soporta dos modos de operación:

1. Modo Estación (STA)

- Conecta a red WiFi existente
- Requiere SSID y password preconfigurados
- IP asignada por DHCP

2. Modo Punto de Acceso (AP)

- Crea red "ESP32-Gateway"
- Sin password por defecto
- IP: 192.168.4.1
- Duración: 3 minutos máximo

Configuración MQTT/WebSockets

cpp

// Configuración por defecto

```
String mqttServer = "broker.emqx.io";
```

```
int mqttPort = 8083;
```

```
String clientId = "ESP32-Gateway";
```

Brokers MQTT Compatibles

- **EMQX Cloud:** broker.emqx.io:8083
- **HiveMQ:** broker.hivemq.com:8083
- **Mosquitto:** tu.dominio.com:8083

FLUJO DE OPERACIÓN

Secuencia de Inicio

text

1. ⚡ POWER ON

↓

2. 📄 Cargar configuración WiFi desde memoria

↓

3. ❓ ¿Configuración existe?

└─ ☒ SI → Conectar WiFi → Paso 4

└─ ☐ NO → Modo AP → Esperar configuración

↓

4. 🌐 Conectar WebSocket al broker

↓

5. 🖱️ Establecer conexión MQTT

↓

6. 🟢 Sistema operativo

Diagrama de Estados

text

ESTADO_AP (Configuración)

↓ (WiFi configurado)

ESTADO_CONECTANDO_WIFI

↓ (WiFi conectado)

ESTADO_CONECTANDO_WS

↓ (WebSocket conectado)

ESTADO_CONECTANDO_MQTT

↓ (MQTT autenticado)

ESTADO_OPERATIVO 

↓ (Desconexión)

ESTADO_RECONECTANDO 

MENSAJES MQTT

Topics de Publicación

1. Estado del Gateway

json

```
{  
  "id": "ESP32-Gateway",  
  "ip": "192.168.1.100",  
  "rssi": -65,  
  "reconexiones": 2,  
  "heapLibre": 195232,  
  "estado": "OPERATIVO"  
}
```

- **Topic:** gateway/estado
- **Frecuencia:** 30 segundos

2. Datos de Sensores

json

```
{  
  "sensor": "DHT22",  
  "temperatura": 23.5,
```

```
"humedad": 65.2,  
"timestamp": 1234567890  
}  
  
• Topic: sensores/{tipo_sensor}/datos  
• Frecuencia: Personalizable
```

3. Heartbeat

```
json  
{  
  "clientId": "ESP32-Gateway",  
  "timestamp": 1234567890  
}  
  
• Topic: gateway/heartbeat (implícito via PING)  
• Frecuencia: 25 segundos
```

Topics de Suscripción

1. Comandos de Control

- **Topic:** gateway/control
- **Mensajes aceptados:**
 - "reset" → Reinicia el gateway

2. Datos de Sensores

- **Topic:** sensors/+ /data
- **Carácter comodín:** + para cualquier tipo de sensor

ESTADOS DEL SISTEMA

Descripción de Estados

Estado	Descripción	Duración	Acciones
ESTADO_AP	Modo configuración	3 min	Servir portal web

Estado	Descripción	Duración	Acciones
ESTADO_CONECTANDO_WIFI	Conectando a WiFi	10 seg	Intentar conexión
ESTADO_CONECTANDO_WS	Conectando WebSocket	Variable	Esperar conexión WS
ESTADO_CONECTANDO_MQTT	Autenticando MQTT	Variable	Esperar CONNACK
ESTADO_OPERATIVO	Sistema listo	Continuo	Publicar datos
ESTADO_RECONECTANDO	Recuperando conexión	10 seg	Reintentar conexión

Indicadores Visuales (LCD)

Cada estado muestra información específica en pantalla:

- **AP:** "MODO CONFIGURACIÓN" + IP del AP
- **Conectando:** Progreso de conexión
- **Operativo:** IP local y estado

CONFIGURACIÓN TÉCNICA

Parámetros de Tiempo

cpp

```
const unsigned long RECONNECT_INTERVAL = 10000; // 10 segundos
```

```
const unsigned long STATUS_INTERVAL = 30000; // 30 segundos
```

```
const unsigned long HEARTBEAT_INTERVAL = 25000; // 25 segundos
```

```
const unsigned long AP_TIMEOUT = 180000; // 3 minutos
```

Configuración de Memoria

- **Buffer MQTT:** 512 bytes
- **Client ID:** "ESP32-Gateway" + ID único
- **Preferences:** Namespace "gateway"

Manejo de Errores

- **Reconexión automática** después de 10 segundos

- **Contador de reconexiones** para diagnóstico
 - **Timeout de AP** después de 3 minutos
 - **Reinicio automático** en fallos críticos
-

PERSONALIZACIÓN

Adaptar para Sensores Reales

Reemplazar en publicarDatosSensor():

cpp

// Ejemplo con sensor DHT real

#include <DHT.h>

DHT dht(D4, DHT22);

float temperatura = dht.readTemperature();

float humedad = dht.readHumidity();

publicarDatosSensor("DHT22", temperatura, humedad);

Cambiar Broker MQTT

cpp

// En cargarConfigWiFi() o hardcodeado

String mqttServer = "tu.broker.com";

int mqttPort = 8083;

Agregar Nuevos Comandos

En processMessage(), agregar:

cpp

if (String(msg) == "update") {

 // Lógica de actualización

}

USO PRÁCTICO

Primera Configuración

1. **Conectar** al AP "ESP32-Gateway"
2. **Configurar** credenciales WiFi via portal web
3. **El sistema** se reinicia y conecta automáticamente

Monitoreo

- **Serial:** 115200 baudios para logs detallados
- **MQTT:** Suscribirse a gateway/estado para monitoreo
- **LCD:** Feedback visual en tiempo real

Mantenimiento

- **Reinicio remoto:** Publicar "reset" en gateway/control
- **Estadísticas:** Revisar contador de reconexiones
- **Diagnóstico:** Monitorear RSSI y heap libre






SOPORTE Y TROUBLESHOOTING

Problemas Comunes

- **✗ No conecta a WiFi:** Verificar SSID/password
- **✗ No conecta a MQTT:** Verificar broker y puerto
- **✗ Desconexiones frecuentes:** Verificar señal WiFi
- **✗ Memory leaks:** Monitorear heap libre

Logs de Diagnóstico

Los logs seriales incluyen emojis para fácil identificación:

-  Operaciones exitosas
-  Errores críticos
-  Reconexiones
-  Publicaciones
-  Recepciones