

# Análisis Detallado del Firmware: Línea de Comandos IEEE 802.15.4

Laura Daniela Alarcón Castaño (Autor del Proyecto)

**Plataforma:** ESP32-C6 (ESP-IDF v5.5)

**Archivo Principal:** esp\_ieee802154\_cli.c

# Índice

<b>1. Introducción y Contexto del Proyecto</b>	<b>3</b>
<b>2. Arquitectura de Software y Componentes</b>	<b>3</b>
2.1. Plataforma y SDK . . . . .	3
2.2. Componentes Clave y Dependencias . . . . .	3
<b>3. Análisis del Protocolo IEEE 802.15.4</b>	<b>3</b>
3.1. Capa Física (PHY) . . . . .	3
3.2. Mecanismos de Control . . . . .	3
<b>4. Análisis del Código Fuente: esp_ieee802154_cli.c</b>	<b>4</b>
4.1. Flujo de Ejecución Principal (app_main) . . . . .	4
4.2. Inicialización de la Consola . . . . .	4
<b>5. Comandos Críticos de la CLI y Uso</b>	<b>4</b>
5.1. Comandos de Configuración . . . . .	4
5.2. Comandos de Transmisión y Diagnóstico . . . . .	5
<b>6. Conclusiones</b>	<b>5</b>

## 1. Introducción y Contexto del Proyecto

Este proyecto demuestra el uso del estándar de comunicación inalámbrica **IEEE 802.15.4** en el microcontrolador **ESP32-C6** a través de una **Interfaz de Línea de Comandos (CLI)**.

El objetivo es exponer las funciones de configuración y diagnóstico de la capa física y MAC a través de una consola interactiva, permitiendo al usuario:

1. Configurar dinámicamente parámetros de red (PAN ID, Canal, Dirección).
2. Enviar y recibir paquetes de prueba (TX/RX).
3. Realizar evaluaciones de canal (CCA) y mediciones de energía (ED).

## 2. Arquitectura de Software y Componentes

### 2.1. Plataforma y SDK

- **Target del Chip:** esp32c6.
- **Framework:** ESP-IDF versión 5.5.0.

### 2.2. Componentes Clave y Dependencias

El proyecto se basa en el sistema de componentes modular de ESP-IDF, tal como se define en `CMakeLists.txt` e `idf_component.yml`.

Cuadro 1: Dependencias Clave del Firmware (Extraído de `CMakeLists.txt`)

Componente	Propósito en el Proyecto
<code>ieee802154</code>	Implementación del protocolo 802.15.4 de Espressif.
<code>console</code>	Provee el *Read-Eval-Print Loop* (REPL) interactivo.
<code>nvs_flash</code>	Inicialización del almacenamiento no volátil.
<code>cmd_ieee802154</code>	Registra los comandos específicos de 802.15.4 en la CLI.
<code>cmd_system</code>	Registra comandos generales del sistema (ej. <code>restart</code> , <code>heap</code> ).

## 3. Análisis del Protocolo IEEE 802.15.4

El estándar **IEEE 802.15.4** define la capa física (PHY) y de acceso al medio (MAC) para redes inalámbricas de área personal de baja tasa de datos (LR-WPANs). Es el cimiento de protocolos de alto nivel como Zigbee, Thread y algunos modos de Bluetooth.

### 3.1. Capa Física (PHY)

La PHY define la modulación, los canales y las características de radio. El ESP32-C6 incluye un transceptor dedicado de 2.4 GHz compatible con 802.15.4.

### 3.2. Mecanismos de Control

Dos funciones críticas probadas por la CLI son:

- **Energy Detect (ED):** Medición de la energía presente en un canal durante un período de tiempo, útil para evaluar el ruido o la ocupación.

- **Clear Channel Assessment (CCA):** Mecanismo utilizado antes de transmitir para asegurar que el canal está libre, minimizando colisiones y crucial para la eficiencia de la red.

## 4. Análisis del Código Fuente: esp\_ieee802154\_cli.c

### 4.1. Flujo de Ejecución Principal (app\_main)

La función de entrada del sistema orquesta la secuencia de inicialización en el siguiente orden:

```

1 void app_main(void)
2 {
3     // 1. Inicializa NVS para guardar configuración persistente.
4     initialize_nvs();
5
6     // 2. Habilita el driver 802.15.4 en la radio del chip.
7     esp_ieee802154_enable();
8
9     // 3. Inicializa el entorno de la consola (UART)
10    // para procesar la entrada de comandos del usuario.
11    initialize_console();
12
13    // 4. Inicia el Read-Eval-Print Loop (REPL).
14    esp_console_start_repl(repl);
15 }
```

Listing 1: Flujo Simplificado de app\_main

### 4.2. Inicialización de la Consola

El código configura el entorno de la consola ('esp<sub>c</sub>onsole') para usarla UART, lo que permite la interacción nativa.

**Prompt:** El indicador de comando se define como "ieee802154", facilitando la identificación del entorno: ieee802154>.

**Registro de Comandos:** Los módulos ieee802154\_cmd.h y cmd\_system.h son importados para registrar todas las funciones de control disponibles en la consola.

## 5. Comandos Críticos de la CLI y Uso

Los comandos habilitados a través de la CLI permiten la interacción directa con el stack 802.15.4.

### 5.1. Comandos de Configuración

Cuadro 2: Comandos de Configuración de Red

Comando (Ejemplo)	Módulo	Función y Uso
ieee802154 set_pan_id 0xABCD	ieee802154_cmd	Define el Identificador de Red de Área Personal (PAN ID).
ieee802154 set_channel 11	ieee802154_cmd	Selecciona el canal de radio (11-26 en 2.4 GHz).

Cuadro 2: Comandos de Configuración de Red

Comando (Ejemplo)	Módulo	Función y Uso
ieee802154 set_addr 0x0001	ieee802154_cmd	Establece la dirección corta del dispositivo.

## 5.2. Comandos de Transmisión y Diagnóstico

Cuadro 3: Comandos de Operación y Diagnóstico

Comando (Ejemplo)	Propósito	Resultado y Aplicación
ieee802154 tx 0x0002 "Hola"	Transmisión de un paquete corto.	Envía el mensaje "Hola." a la dirección 0x0002. Fundamental para la verificación de la capa física.
ieee802154 ed 5 1	Detección de energía (ED).	Mide el nivel de ruido en el canal (5) durante un intervalo (1 segundo), validando la función de diagnóstico de capa física.
ieee802154 start_rx	Habilita el modo receptor.	Pone el chip en escucha continua para recibir paquetes de prueba.

## 6. Conclusiones

Este firmware de CLI convierte el ESP32-C6 en una herramienta de diagnóstico y desarrollo para redes 802.15.4. La capacidad de configurar y probar la capa física y MAC mediante comandos interactivos valida la versatilidad del chip para aplicaciones de bajo consumo, como redes malladas (Mesh), Zigbee o Thread, donde el control directo de la radio es esencial.