

# Документация прошивки ESP32-CONSOLE-CSI

## 1. Общее описание

Данная прошивка предназначена для микроконтроллеров серии ESP32 (рекомендуется ESP32-S3) и разработана на базе ESP-IDF v5.x. Основное назначение — проведение исследований в области физической безопасности беспроводных сетей (Physical Layer Security).

### Ключевые возможности:

- Сбор Channel State Information (CSI) в различных режимах (HT20/HT40).
- Генерация трафика (Ping) через ESP-NOW для возбуждения канала.
- Обмен текстовыми сообщениями для эмуляции коммуникации Алиса-Боб.
- Управление радиомодулем "на лету" (смена канала, мощности, MAC-адреса).
- Сканирование эфира для выбора оптимальных частот.

## 2. Архитектура модулей

Проект построен по модульному принципу. Каждый компонент регистрирует свои команды в консоли (REPL).

Компонент	Описание	Зависимости
<code>cmd_radio</code>	Центральный модуль управления Wi-Fi стеком. Инициализирует драйвер, управляет PHY (канал, мощность, полоса). Предотвращает конфликты инициализации.	<code>nvs</code> , <code>esp_wifi</code>
<code>cmd_csi_recv</code>	Модуль приёма CSI. Настраивает Promiscuous mode и CSI callback. Выводит сырье данные CSI в CSV-формате в консоль.	<code>cmd_radio</code>
<code>cmd_csi_ping</code>	Модуль активной генерации трафика. Отправляет широковещательные пакеты ESP-NOW с высокой частотой для получения CSI на приёмнике.	<code>cmd_radio</code> , <code>esp_timer</code>
<code>cmd_msg</code>	Модуль обмена сообщениями. Позволяет отправлять произвольный текст (Chat) между узлами через ESP-NOW. Используется для эмуляции легитимного трафика.	<code>cmd_radio</code>
<code>cmd_system</code>	Стандартные системные команды (рестарт, info, heap).	—

## 3. Справочник команд (Console API)

Все команды вводятся через последовательный порт (UART/USB) со скоростью **115200** или **921600** бод (зависит от конфигурации).

### 3.1. Управление радио (`cmd_radio`)

Базовая настройка физического уровня. Рекомендуется выполнять в первую очередь.

## radio\_init

Инициализирует Wi-Fi стек и применяет настройки. Если стек уже запущен, меняет параметры "на лету" (где возможно) или перезапускает интерфейс.

```
*    **Синтаксис:** `radio_init [-c <channel>] [-s <secondary>] [-b <bandwidth>] [-m <mac>] [-r]`
```

- **Аргументы:**

- **-c, --channel <1-14>**: Номер основного канала (по умолчанию 11).
- **-s, --secondary <none|above|below>**: Положение вторичного канала для HT40 (по умолчанию **below**).
- **-b, --bw <20|40>**: Ширина полосы в МГц (по умолчанию 40).
- **-m, --mac <aa:bb:cc:dd:ee:ff>**: Смена MAC-адреса (спуфинг).
- **-r, --restart**: Принудительно остановить и запустить Wi-Fi заново.

**Пример:** `radio_init -c 6 -b 20` (Переключиться на 6 канал, 20 МГц).

## radio\_info

Выводит текущее состояние радиомодуля (инициализирован ли стек, текущий канал, MAC-адрес).

## tx\_power

Управление мощностью передатчика. Полезно для тестов на дальность или для скрытности.

- **Синтаксис:** `tx_power [-d <dbm>]`

- **Аргументы:**

- **-d, --dbm <8-20>**: Мощность в dBm. (Максимум ~20 dBm, минимум ~2 dBm).

- **Без аргументов:** Выводит текущую мощность.

## scan

Сканирование эфира для поиска точек доступа и оценки занятости каналов.

- **Синтаксис:** `scan [-p]`

- **Аргументы:**

- **-p, --passive**: Пассивное сканирование (без отправки Probe Request). Более скрытое, но медленное.

---

## 3.2. Сбор данных (`cmd_csi_recv`)

### recv

Включает режим прослушивания (Promiscuous Mode) и вывода CSI.

- **Синтаксис:** `recv [-m <MAC>] [-t <seconds>]` (в текущей реализации аргумент времени может игнорироваться, работает до перезагрузки или остановки).
- **MAC** - адрес станции от которой планируется получать пакеты.
- **Формат вывода:**

```
CSI_DATA, count, mac, rss, rate, sig_mode, mcs, bw, len, "[byte1, byte2, ...]"
```

### 3.3. Генерация трафика (`cmd_csi_ping`)

#### `ping`

Запускает отправку пакетов через ESP-NOW для сбора CSI. Теперь поддерживает Unicast (подтверждаемая доставка) и настройку частоты.

```
* **Синтаксис:** `ping -t <sec> [-r <hz>] [-m <mac>]`
```

- **Аргументы:**

- `-t, --timeout <sec>`: Время работы в секундах.
- `-r, --rate <hz>`: Частота отправки пакетов (Герц). По умолчанию **100**.
- `-m, --mac <addr>`: MAC-адрес получателя.
  - Если указан конкретный MAC: используется **Unicast** (стабильнее, есть повторы).
  - Если не указан: используется **Broadcast** (FF:FF:FF:FF:FF:FF).

**Особенности модуляции:** Команда автоматически выбирает режим **HT20** или **HT40** в зависимости от настроек `radio_init`, но фиксирует индекс модуляции на **MCS0** (самая низкая скорость). Это сделано намеренно для получения максимально "чистых" и полных данных CSI без потерь на физическом уровне.

**Пример:** `ping -t 10 -r 200 -m 1a:00:00:00:00:02` — Отправлять 200 пакетов в секунду конкретно Бобу в течение 10 секунд.

### 3.4. Обмен сообщениями (`cmd_msg`)

#### `msg_listen`

Включает отображение входящих текстовых сообщений ESP-NOW в консоли.

- Если радио не инициализировано, запускает его с дефолтными настройками.

#### `msg_send`

Отправляет текстовое сообщение.

```
* **Синтаксис:** `msg_send [-m <mac>] <text>`
```

- **Аргументы:**

- `-m, --mac <addr>`: MAC-адрес получателя. Если не указан — **Broadcast** (всем).
- `<text>`: Текст сообщения в кавычках.

**Пример:** `msg_send -m 30:ae:a4:cc:dd:ee "Hello Bob"`

## 4. Сценарии использования

Сценарий А: Сбор датасета для генерации ключей (Алиса и Боб)

1. **Настройка Алисы (Передатчик):**

```
# Инициализация на 11 канале, 40 МГц
radio_init -c 11 -b 40 -s below
# Запуск генерации трафика на 60 секунд с частотой 100 Гц
ping -t 60 -r 100
```

2. **Настройка Боба (Приёмник):**

```
# Те же настройки радио!
radio_init -c 11 -b 40 -s below
# Включение записи лога в файл (на ПК)
# Запуск приема на 100 секунд
recv -t 100
```

Сценарий Б: Атака "Ева" (Пассивный перехват)

1. **Разведка:**

```
scan --passive
# Находим, на каком канале общаются Алиса и Боб (например, канал 6)
```

2. **Настройка Евы:**

```
# Встаем на тот же канал
radio_init -c 6 -b 40
# Скрываем свое присутствие (минимум мощности)
tx_power -d 8
# Слушаем переписку
msg_listen
# Или собираем CSI для криптоанализа
recv
```

## 5. Формат данных CSI

Данные выводятся в формате CSV. Каждая строка начинается с тега `CSI_DATA`.

Поле	Описание
<code>count</code>	Норядковый номер пакета (uint32).
<code>mac</code>	MAC-адрес отправителя пакета.
<code>rssi</code>	Уровень сигнала (dBm).
<code>rate</code>	Индекс скорости (MCS/Rate).
<code>sig_mode</code>	Режим: 0=Legacy, 1=HT, 2=VHT, 3=HE.
<code>mcs</code>	Modulation Coding Scheme (0-7).
<code>bw</code>	Полоса пропускания (0=20MHz, 1=40MHz).
<code>len</code>	Количество байт данных CSI.
[...]	Массив байт CSI (амплитуды и фазы поднесущих).

Для **HT40** (ESP32) массив содержит 128 комплексных чисел (или меньше, в зависимости от упаковки). ESP32 выдает данные в формате двух байт на поднесущую (Signed 8-bit Real, Signed 8-bit Imaginary).

## 6. Устранение неполадок

### Ошибка: `ESP_ERR_INVALID_STATE` при вводе `recv`

- Причина:** Попытка повторной инициализации Wi-Fi.
- Решение:** В версии v2.0 исправлено модулем `cmd_radio`. Если возникает, выполните `radio_init --restart`.

### Данные CSI не приходят (`recv` молчит)

- Проверьте, что Алиса и Боб на одном канале (`radio_info`).
- Проверьте, что Алиса действительно шлет пакеты (`ping`).
- Проверьте MAC-адрес фильтрации в коде `cmd_csi_recv.c` (`TARGET_MAC`).

### Мусор в консоли при старте

- Проверьте скорость порта. Для тяжелого трафика CSI рекомендуется **921600** или **2000000** бод.

### Ошибка `ESP_ERR_WIFI_IF` при смене MAC

- Нельзя менять MAC, если Wi-Fi активен и передает данные. Используйте `radio_init -m <mac> --restart`.