

# Основы микропроцессорной техники

## Введение в Espressif Rainmaker

Хохлов С.А.

МГТУ им. Н.Э. Баумана

13 ноября 2025 г.

# Espressif RainMaker

# 1. Что это такое I

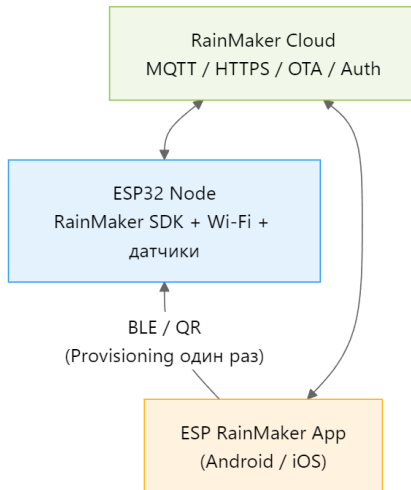
Espressif RainMaker — это облачная IoT-платформа, созданная компанией Espressif Systems для устройств на ESP32 и ESP32-S/C/H.

Она объединяет в себе:

- прошивку на ESP32 (SDK или Arduino-ядро),
- облачный сервер (RainMaker Cloud),
- мобильное приложение для Android / iOS,
- API и веб-панель для разработчиков.

Главная идея — максимально быстрое создание умных устройств с визуальным управлением без необходимости писать собственный облачный бэкенд или приложение.


## 2. Архитектура RainMaker I



## 2. Архитектура RainMaker II



- 1 Provisioning только один раз: приложение подключается к ESP32 по BLE или QR, передаёт Wi-Fi и токен устройства.
- 2 После этого всё через облако: ESP32 и приложение больше не общаются напрямую — оба работают с RainMaker Cloud.
- 3 MQTT — основа обмена: ESP32 публикует измерения и принимает команды, приложение получает обновления от облака.
- 4 Полная синхронизация: изменения в приложении сразу отражаются на устройстве и наоборот — облако хранит текущее состояние.

### 3. Почему это удобно I

- Всё готово “из коробки” — можно сосредоточиться на периферии и логике, а не на серверной части.
- Мобильное приложение генерируется автоматически по коду.
- Работает и с ESP-IDF, и с Arduino-ядром.
- Есть готовые примеры: Switch, Lightbulb, Temperature Sensor, Fan Controller.
- Видимый результат за одно занятие: “подключил датчик — и данные видны на телефоне”.
- EvaluationHub 

## 4. Установка и подготовка I

### 4.1. Требования

- Плата ESP32 DevKitC / NodeMCU-32S / WROOM-32
- ESP-IDF 4.4+ (или PlatformIO / Arduino IDE)
- Учётная запись RainMaker
- Смартфон с приложением
  - Android: ESP RainMaker в Google Play 
  - iOS: App Store – ESP RainMaker 

## 4. Установка и подготовка II

### 4.2. Установка SDK

```
git clone --recursive https://github.com/espressif/esp-rainmaker.git  
cd esp-rainmaker/examples
```

Далее можно открыть пример, например `temperature_sensor`.



## Примеры проектов

## 5. Пример проекта: “Выключатель” I

Цель: использовать кнопка BOOT и RGB-светодиод на плате ESP32-S2-Saola-1 или ESP32-C3-DevKitC для демонстрации работы переключателя.

- Состояние светодиода (зелёный цвет) показывает текущее состояние переключателя.
- Нажатие кнопки BOOT изменяет (переключает) состояние переключателя, а значит — и состояние светодиода. Это изменение также будет отображаться в мобильном приложении.

## 5. Пример проекта: “Выключатель” II

- Если переключить кнопку в приложении, светодиод на вашей плате тоже изменит состояние, а в мониторе ESP32-S2 появятся сообщения вида:
  - I (16073) app\_main: Received value = true for Switch - power

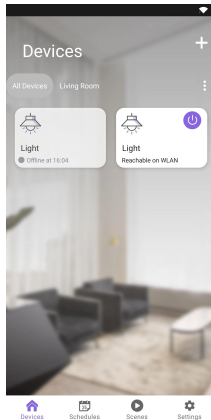
## 5.1 Код примера

- [Code Basics | ESP RainMaker Documentation Hub](#) 

## 5.2 Что произойдёт после прошивки I

- 1 Устройство включается и открывает BLE-канал для “provisioning”.
- 2 Пользователь в мобильном приложении добавляет новое устройство (QR-код или BLE).
- 3 RainMaker Cloud регистрирует узел и отображает устройство с выключателем.
- 4 На телефоне появляется устройство-выключатель.

## 5.2 Что произойдёт после прошивки II




## 6. Мобильное приложение RainMaker I

Приложение автоматически отображает все устройства и их параметры, полученные из прошивки.

- Для каждого параметра создаётся виджет (переключатель, слайдер, график, индикатор).
- Можно управлять устройствами (например, включить реле) или наблюдать показания.
- Есть уведомления, сценарии, таймеры, групповое управление.
- OTA-обновление можно запустить прямо из приложения.

## 7. Расширение проекта I

- 1 Добавить термодатчик: `esp-rainmaker/examples/temperature_sensor/main/app_main.c`  
at master · espressif/esp-rainmaker · GitHub 

В приложении появится кнопка индикатор температуры.



## 7. Расширение проекта II

- 2. Добавить OTA-обновление:

```
esp_rmaker_ota_enable_default();
```

Эта функция:

- регистрирует задачу OTA в RainMaker SDK,
- подписывает устройство на обновления от облака,
- использует безопасное HTTPS-соединение с проверкой сертификата.

## 7. Расширение проекта III

- 3 Добавить Google Home / Alexa через интеграции RainMaker Cloud.

# OTA I

OTA (Over-The-Air update) — важная часть экосистемы Espressif RainMaker, и она реализована на уровне SDK и облака, так что обновление прошивки можно делать прямо из RainMaker App или Developer Console, без FTP, TFTP или веб-интерфейсов.

- 1 Разработчик загружает новую прошивку — через RainMaker Developer Console или API. — указывает имя проекта и версию (например, v2.1.0).
- 2 RainMaker Cloud хранит бинарный файл — и метаданные (версия, CRC, дата, устройства, к которым применимо).

## ОТА II

- 3 ESP32 периодически проверяет наличие обновлений — SDK автоматически запрашивает у облака версию (`esp_rmaker_ota_check`). — если версия новее — начинается загрузка по HTTPS.
- 4 Загрузка и установка
  - прошивка скачивается во второй раздел flash (OTA partition B);
  - по завершении проверяется контрольная сумма и подпись;
  - происходит переключение активного раздела и перезагрузка;
  - после успешного старта раздел помечается как «valid».
- 5 Приложение показывает прогресс ОТА — пользователь видит статус «Updating...» и процент загрузки.

# ОТА III


## Безопасность ОТА

- TLS-шифрование трафика.
- Проверка цифровой подписи прошивки.
- Ключи и сертификаты хранятся в eFuse.
- Возможность ограничить обновление конкретными пользователями или группами устройств.


## 8. Преимущества RainMaker

Возможность	Описание
Provisioning	Настройка Wi-Fi и привязка устройства через BLE или QR-код
Автоматический UI	Приложение само создаёт элементы управления
Без серверов	Не нужно поднимать свой MQTT/HTTP-бэкенд
OTA-обновления	Встроенный механизм обновления прошивок
Безопасность	TLS, eFuse-ключи, управление доступом
Кросс-платформенность	Android, iOS, Web API
Совместимость с ESP-IDF	Использует FreeRTOS и стандартный стек Espressif

## 9. Альтернативные проекты I


- ESPHome
  - Home Assistant и локальные системы
- Платформа для прошивки ESP32/ESP8266 с минимальным кодом (YAML-конфигурации).
- Позволяет подключаться к Home Assistant «из коробки».
- [esphome.io](https://esphome.io) 

## 9. Альтернативные проекты II


- Tasmota
  - Home Assistant, OpenHAB, MQTT-экосистемы
- Прошивка для ESP-устройств, ориентированная на управление реле/датчиками, с интеграцией через MQTT, REST и др.
- Tasmota Documentation - Tasmota 



## 9. Альтернативные проекты III

- НАА – Home Accessory Architect (или “esp-homekit-devices”)
  - Apple HomeKit
- Специализированная прошивка для ESP32/ESP8266, добавляющая нативную поддержку Apple HomeKit.
- GitHub - RavenSystem/esp-homekit-devices 

## 9. Альтернативные проекты IV

- Mongoose OS
  - Облачные сервисы, IoT-интеграции
- Более широкая IoT-платформа, поддерживающая ESP32 и другие MCU, позволяет интеграции с AWS, Google IoT и др.
- Википедия 

## 10. Заключение

ESP RainMaker — это идеальный инструмент для:

- демонстрации IoT-концепций (датчик → сеть → облако → пользователь),
- быстрых проектов и лабораторных,
- расширения опыта после AVR-микроконтроллеров,
- обучения архитектуре систем с Wi-Fi, RTOS, облаком и безопасностью.

