

Оглавление

Предисловие от издательства	13
Введение	14
Предисловие.....	15
ЧАСТЬ I. ПОДГОТОВКА	21
Глава 1. Введение в IoT.....	23
1.1. Архитектура интернета вещей	23
Уровень восприятия и управления	23
Сетевой уровень	24
Уровень платформы.....	25
Прикладной уровень.....	26
1.2. Применение IoT в проекте «Умного дома»	26
Глава 2. Введение в практику IoT-проектов	29
2.1. Введение в типовые проекты IoT	29
2.1.1. Базовые модули для обычных устройств IoT.....	29
2.1.2. Базовые модули клиентских приложений.....	30
2.1.3. Введение в общие облачные платформы IoT	31
2.2. Практика: проект Smart Light.....	32
2.2.1. Структура проекта	33
2.2.2. Функции проекта	33
2.2.3. Подготовка оборудования.....	34
2.2.4. Процесс разработки	36
2.3. Резюме	37
Глава 3. Введение в ESP RainMaker.....	38
3.1. Что такое ESP RainMaker?	39
3.2. Реализация ESP RainMaker	40
3.2.1. Служба обработки заявок	41
3.2.2. RainMaker Agent	42
3.2.3. Облачный сервер	43
3.2.4. Клиент RainMaker.....	44
3.3. Практика: ключевые моменты разработки с ESP RainMaker.....	44
3.4. Особенности ESP RainMaker	46

3.4.1. Управление пользователями.....	46
3.4.2. Функции конечного пользователя.....	47
3.4.3. Функции администратора.....	48
3.5. Резюме	48

Глава 4. Настройка среды разработки..... 50

4.1. Обзор ESP-IDF	50
4.1.1. Версии ESP-IDF	51
4.1.2. Рабочий процесс ESP-DIFF Git	52
4.1.3. Выбор подходящей версии.....	53
4.1.4. Обзор каталога ESP-IDF SDK	53
4.2. Настройка среды разработки ESP-IDF	58
4.2.1. Настройка среды разработки ESP-IDF в Linux	58
4.2.2. Настройка среды разработки ESP-IDF в Windows.....	60
4.2.3. Настройка среды разработки ESP-IDF на Mac	66
4.2.4. Установка VS Code	67
4.2.5. Знакомство со сторонними средами разработки	67
4.3. Система компиляции ESP-IDF	68
4.3.1. Основные концепции системы компиляции.....	68
4.3.2. Структура файла проекта	69
4.3.3. Правила построения системы компиляции по умолчанию	71
4.3.4. Введение в сценарий компиляции	72
4.3.5. Введение в общие команды	73
4.4. Практика: компиляция примера программы Blink.....	74
4.4.1. Анализ примера	74
4.4.2. Компиляция программы Blink.....	77
4.4.3. Прошивка программы Blink.....	81
4.4.4. Анализ логов последовательного порта программы Blink	82
4.5. Резюме	85

ЧАСТЬ II. РАЗРАБОТКА ОБОРУДОВАНИЯ И ДРАЙВЕРОВ 87

Глава 5. Аппаратный дизайн продуктов Smart Light на базе ESP32-C3..... 89

5.1. Характеристики и состав продуктов Smart Light.....	89
5.2. Аппаратный дизайн базовой системы ESP32-C3	93
5.2.1. Источник питания	97
5.2.2. Порядок включения питания и сброс системы	97
5.2.3. SPI флеш-память.....	98
5.2.4. Источник тактовых импульсов	98
5.2.5. Радиочастотный сигнал (RF) и антенна	99
5.2.6. Выводы управления загрузкой ПО (Strapping Pins).....	102
5.2.7. GPIO и ШИМ-контроллер.....	102

5.3. Практика: создание системы умного освещения с помощью ESP32-C3.....	103
5.3.1. Выбор модулей.....	103
5.3.2. Настройка ШИМ-сигналов на выводах GPIO.....	105
5.3.3. Загрузка встроенного ПО и интерфейс отладки.....	105
5.3.4. Рекомендации по проектированию радиочастотой части	108
5.3.5. Рекомендации по проектированию источника питания.....	109
5.4. Резюме	109

Глава 6. Разработка драйверов 111

6.1. Процесс разработки драйверов	111
6.2. Периферийные приложения ESP32-C3	112
6.3. Основы построения драйверов светодиодов.....	113
6.3.1. Цветовые пространства.....	114
6.3.2. Светодиодный драйвер	118
6.3.3. Диммирование светодиодов.....	119
6.3.4. Введение в ШИМ	120
6.4. Разработка драйвера для регулирования светодиодов.....	122
6.4.1. Энергонезависимая память (NVS).....	122
6.4.2. Светодиодный ШИМ-контроллер (LEDC).....	123
6.4.3. Программирование ШИМ для светодиодов.....	125
6.5. Практика: добавление драйверов в проект Smart Light.....	128
6.5.1 Драйвер кнопки	128
6.5.2. Драйвер регулировки яркости светодиода	130
6.6. Резюме	134

ЧАСТЬ III. БЕСПРОВОДНАЯ СВЯЗЬ И УПРАВЛЕНИЕ 135

Глава 7. Настройка Wi-Fi-соединения 137

7.1. Основы Wi-Fi	137
7.1.1. Введение в Wi-Fi.....	137
7.1.2. Эволюция IEEE 802.11	138
7.1.3. Концепции Wi-Fi	139
7.1.4. Wi-Fi-соединение	141
7.2. Основы Bluetooth	149
7.2.1. Введение в Bluetooth.....	149
7.2.2. Концепции Bluetooth	150
7.2.3. Bluetooth-соединение	154
7.3. Конфигурация сети Wi-Fi	158
7.3.1. Руководство по настройке сети Wi-Fi.....	158
7.3.2. Программная точка доступа (Soft access point, SoftAP)	158
7.3.3. SmartConfig	159
7.3.4. Bluetooth	162
7.3.5. Другие методы.....	164

7.4. Программирование Wi-Fi	166
7.4.1. Компоненты Wi-Fi в ESP-IDF	166
7.4.2. Упражнение: соединение Wi-Fi	168
7.4.3. Упражнение: интеллектуальное подключение к Wi-Fi	172
7.5. Практика: конфигурация Wi-Fi в проекте Smart Light	184
7.5.1 Соединение Wi-Fi в проекте Smart Light	184
7.5.2. Умная настройка Wi-Fi	185
7.6. Резюме	186

Глава 8. Локальное управление..... 187

8.1. Введение в локальное управление	187
8.1.1. Применение локального управления	189
8.1.2. Преимущества локального управления	189
8.1.3. Обнаружение управляемых устройств с помощью смартфонов	189
8.1.4. Передача данных между смартфонами и устройствами	190
8.2. Общие методы локального обнаружения	190
8.2.1. Широковещательная передача	191
8.2.2. Групповая передача (Multicast)	197
8.2.3. Сравнение широковещательной и групповой рассылки	202
8.2.4. Протокол групповых приложений mDNS для локального обнаружения	203
8.3. Общие протоколы связи для локальных данных	206
8.3.1. Протокол управления передачей (TCP)	206
8.3.2. Протокол передачи гипертекста (HyperText Transfer Protocol, HTTP)	211
8.3.3. Протокол пользовательских датаграмм (User Datagram Protocol, UDP)	214
8.3.4. Протокол ограниченных приложений (Constrained Application Protocol, CoAP)	218
8.3.5. Протокол Bluetooth	222
8.3.6. Обзор протоколов передачи данных	228
8.4. Гарантии безопасности данных	230
8.4.1. Введение в безопасность транспортного уровня (TLS)	232
8.4.2. Введение в датаграмм-протокол безопасности транспортного уровня (DTLS)	238
8.5. Практика: локальное управление в проекте Smart Light	241
8.5.1. Создание локального управляющего сервера на базе Wi-Fi	241
8.5.2. Проверка функциональности локального управления с помощью скриптов	245
8.5.3. Создание локального сервера управления на базе Bluetooth	246
8.6. Резюме	248

Глава 9. Управление через облако 250

9.1. Введение в удаленное управление	250
9.2. Облачные протоколы передачи данных	251
9.2.1. Введение в MQTT	251

9.2.2. Принципы MQTT.....	252
9.2.3. Формат сообщения MQTT	254
9.2.4. Сравнение протоколов	258
9.2.5. Настройка MQTT Broker в Linux и Windows	259
9.2.6. Настройка клиента MQTT на основе ESP-IDF	260
9.3. Обеспечение безопасности данных MQTT.....	262
9.3.1. Значение и функция сертификатов.....	262
9.3.2. Локальная генерация сертификатов	263
9.3.3. Настройка MQTT Broker.....	266
9.3.4. Настройка клиента MQTT.....	266
9.4. Практика: дистанционное управление через ESP RainMaker.....	268
9.4.1. Основы ESP RainMaker.....	268
9.4.2. Протокол связи между узлом и серверной частью облака.....	269
9.4.3. Взаимодействие между клиентом и облачным бэкендом	274
9.4.4. Типы пользователей	277
9.4.5. Основные сервисы	278
9.4.6. Пример Smart Light	280
9.4.7. Приложение RainMaker и интеграция сторонних платформ	286
9.5. Резюме	293

Глава 10. Разработка приложений для смартфонов..... 295

10.1. Введение в разработку приложений для смартфонов	295
10.1.1. Обзор разработки приложений для смартфонов	296
10.1.2. Структура проекта Android.....	296
10.1.3. Структура проекта iOS.....	297
10.1.4. Жизненный цикл Android Activity	298
10.1.5. Жизненный цикл iOS ViewController.....	299
10.2. Создание нового проекта приложения для смартфона	301
10.2.1. Подготовка к разработке под Android	301
10.2.2. Создание нового проекта Android	301
10.2.3. Добавление зависимостей для MyRainmaker	302
10.2.4. Запрос разрешений в Android	303
10.2.5. Подготовка к разработке iOS.....	304
10.2.6. Создание нового проекта iOS.....	304
10.2.7. Добавление зависимостей для MyRainmaker	305
10.2.8. Запрос разрешений в iOS	307
10.3. Анализ функциональных требований приложения	307
10.3.1. Анализ функциональных требований проекта.....	307
10.3.2. Анализ требований к управлению пользователями	308
10.3.3. Анализ требований к подготовке и привязке устройства....	309
10.3.4. Анализ требований к удаленному управлению.....	310
10.3.5. Анализ требований к планированию	311
10.3.6. Анализ требований к пользовательскому центру	312
10.4. Разработка системы управления пользователями.....	312
10.4.1. Введение в API RainMaker.....	312
10.4.2. Инициализация связи через смартфон.....	313

10.4.3. Регистрация учетной записи.....	313
10.4.4. Вход в учетную запись.....	316
10.5. Разработка системы подготовки устройств	319
10.5.1. Сканирование устройств	320
10.5.2. Подключение устройств	321
10.5.3. Генерация секретных ключей	324
10.5.4. Получение идентификатора (ИД) узла	324
10.5.5. Подготовка устройств	326
10.6. Разработка управления устройствами	328
10.6.1. Привязка устройств к облачным учетным записям.....	328
10.6.2. Получение списка устройств	330
10.6.3. Получение статуса устройства	333
10.6.4. Изменение статуса устройства.....	336
10.7. Разработка расписания и пользовательского центра.....	338
10.7.1. Реализация функции планирования.....	338
10.7.2. Реализация центра пользователей	340
10.7.3. Дополнительные облачные API.....	343
10.8. Резюме	344

Глава 11. Обновление встроенного ПО и управление

версиями.....	345
11.1. Обновление прошивки	345
11.1.1. Обзор таблицы разделов	346
11.1.2. Процесс загрузки прошивки	348
11.1.3. Обзор механизма OTA	350
11.2. Управление версиями прошивки.....	353
11.2.1. Маркировка прошивки	353
11.2.2. Откат и защита от отката	354
11.3. Практика: пример OTA-обновления.....	355
11.3.1. Обновление прошивки через локальный хост.....	355
11.3.2. Обновление прошивки через ESP RainMaker.....	358
11.4. Резюме	364

ЧАСТЬ IV. ОПТИМИЗАЦИЯ И СЕРИЙНОЕ ПРОИЗВОДСТВО365

Глава 12. Управление питанием и оптимизация энергопотребления..... 367

12.1. Управление питанием ESP32-C3.....	367
12.1.1. Динамическое масштабирование частоты	368
12.1.2. Настройка управления питанием.....	370
12.2. Режимы пониженного энергопотребления ESP32-C3	370
12.2.1. Режим Modem-sleep	371
12.2.2. Режим Light-sleep	373
12.2.3. Режим глубокого сна Deep-sleep	378
12.2.4. Потребление тока в различных режимах питания.....	380

12.3. Управление питанием и отладка режима низкого энергопотребления	381
12.3.1. Отладка через логи	381
12.3.2. Отладка по состояниям GPIO	383
12.4. Практика: управление питанием в проекте Smart Light.....	385
12.4.1. Настройка функции управления питанием.....	386
12.4.2. Использование блокировки управления питанием	387
12.4.3. Проверка энергопотребления	388
12.5. Резюме	388

Глава 13. Расширенные функции безопасности устройства.... 389

13.1. Обзор безопасности данных IoT-устройств	389
13.1.1. Зачем защищать данные устройств интернета вещей?	390
13.1.2. Основные требования к безопасности данных IoT-устройств	391
13.2. Защита целостности данных.....	392
13.2.1. Основы метода проверки целостности	392
13.2.2. Проверка целостности данных прошивки	393
13.2.3. Пример.....	394
13.3. Защита конфиденциальности данных	394
13.3.1. Введение в шифрование данных	394
13.3.2. Введение в систему флеш-шифрования.....	396
13.3.3. Хранение ключей флеш-шифрования.....	399
13.3.4. Рабочие режимы флеш-шифрования	400
13.3.5. Процесс флеш-шифрования.....	402
13.3.6. Введение в шифрование NVS.....	403
13.3.7. Примеры флеш-шифрования и шифрования NVS	404
13.4. Защита легитимности данных	406
13.4.1. Введение в цифровую подпись	406
13.4.2. Обзор системы безопасной загрузки	408
13.4.3. Введение в программную безопасную загрузку	408
13.4.4. Введение в аппаратную безопасную загрузку	410
13.4.5. Примеры	415
13.5. Практика: функции безопасности в серийном производстве	416
13.5.1. Флеш-шифрование и безопасная загрузка	416
13.5.2. Включение флеш-шифрования и безопасной загрузки с помощью инструментов пакетной прошивки.....	417
13.5.3. Включение флеш-шифрования и безопасной загрузки в проекте Smart Light	418
13.6. Резюме	418

Глава 14. Запись и тестирование прошивки для серийного производства 420

14.1. Загрузка прошивки при серийном производстве	420
14.1.1. Определение разделов данных.....	421
14.1.2. Запись прошивки	423

14.2. Тестирование серийной продукции	424
14.3. Практика: производственные данные в проекте Smart Light.....	426
14.4. Резюме	426

Глава 15. ESP Insights: платформа удаленного мониторинга427

15.1. Введение в ESP Insights	427
15.2. Начало работы с ESP Insights	431
15.2.1. Начало работы с ESP Insights в проекте esp-insights.....	431
15.2.2. Пример выполнения в проекте esp-insights	433
15.2.3. Отчетность об информации дампа памяти	433
15.2.4. Настройка интересующих логов	434
15.2.5. Сообщение о причине перезагрузки	435
15.2.6. Отчетность по заданным показателям	435
15.3. Практика: использование ESP Insights в проекте Smart Light	438
15.4. Резюме	440

Предисловие от издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в основном тексте или программном коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Введение

ESP32-C3 – это одноядерный микроконтроллер, представляющий собой «систему на кристалле» (SoC) с интегрированными Wi-Fi и Bluetooth 5 (LE), основанный на архитектуре RISC-V с открытым исходным кодом. Он обеспечивает необходимый баланс мощности, возможностей ввода-вывода и безопасности, предлагая таким образом оптимальное экономичное решение для подключаемых устройств. Эта книга от компании Espressif продемонстрирует различные приложения семейства ESP32-C3, проведя вас по увлекательному пути, начинающемуся с основ разработки проектов интернета вещей (IoT) и настройки среды и заканчивающемуся практическими примерами. Первые четыре главы рассказывают об IoT, ESP RainMaker и ESP-IDF. В главах 5 и 6 кратко говорится о проектировании оборудования и разработке драйверов. По мере продвижения вы узнаете, как настроить свой проект через сети Wi-Fi и мобильные приложения. Наконец, вы научитесь оптимизировать свой проект и запускать его в серийное производство.

Если вы инженер в смежных областях, разработчик программного обеспечения, преподаватель, студент или просто любитель, интересующийся IoT, – эта книга для вас.

Примеры кода, использованные в данной книге, вы можете скачать с сайта Espressif на GitHub. Для получения последней информации о разработке интернета вещей, пожалуйста, следите за нашим официальным аккаунтом.

Предисловие

Информатизирующийся мир

Со времени появления Всемирной сети интернет вещей (IoT) совершил грандиозный скачок, став новым типом инфраструктуры в цифровой экономике. Чтобы приблизить технологию к широкой публике, компания Espressif Systems работает над тем, чтобы разработчики из всех слоев общества могли использовать IoT для решения некоторых насущных проблем современности. Мир «интеллектуальной сети вещей» – это то, чего мы ожидаем от будущего.

Разработка все новых чипов является важнейшим компонентом этого видения. Это марафон, требующий постоянных прорывов через технологические границы. От «изменившего правила игры» ESP8266 до серии ESP32, интегрирующей Wi-Fi и Bluetooth (LE), а затем ESP32-S3, оснащенной ускорителем искусственного интеллекта, компания Espressif никогда не прекращает исследования и разработку продуктов для решений IoT. С помощью нашего программного обеспечения с открытым исходным кодом, такого как фреймворк ESP-IDF для разработки интернета вещей, фреймворк разработки распределенных сетей ESP-MDF и платформа ESP RainMaker для подключения устройств, мы создали независимую платформу для создания приложений AIoT¹.

По состоянию на июль 2022 года совокупные поставки IoT-чипсетов Espressif превысили 800 млн, что обеспечивает огромное количество подключенных устройств во всем мире и вывело компанию в лидеры рынка Wi-Fi-микроконтроллеров. Стремление к совершенству делает каждый продукт Espressif популярным благодаря высокому уровню интеграции и экономической эффективности. Выпуск ESP32-C3 знаменует собой важную веху в развитии технологий собственных разработок Espressif. Это одноядерный 32-рядный микроконтроллер на базе RISC-V с 400 Кбайт SRAM, который может работать на частоте 160 МГц. В ESP32-C3 встроены Wi-Fi с частотой 2,4 ГГц и Bluetooth 5 (LE) с поддержкой большой дальности связи. Он обеспечивает прекрасный баланс мощности, возможностей ввода-вывода и безопасности, предлагая таким образом оптимальное экономичное решение для подключенных устройств. На примере столь мощного ESP32-C3 эта книга с подробными иллюстрациями и практическими примерами призвана помочь читателям разобраться в знаниях, связанных с IoT.

¹ AIoT – в отличие от всем известного сокращения IoT, означающего «интернет вещей» (Internet of Things), сокращение AIoT означает «искусственный интеллект вещей» (Artificial intelligence of things) – появившееся в последние годы направление, сочетающее технологии интернета вещей с технологиями искусственного интеллекта (ИИ). – *Здесь и далее прим. перев.*

Почему мы написали эту книгу?

Espressif Systems – это нечто большее, чем полупроводниковая компания. Это также компания, занимающаяся платформой интернета вещей, которая всегда стремится к прорывам и инновациям в области технологий. В то же время Espressif открыла исходный код и поделилась с сообществом своей самостоятельно разработанной операционной системой и программным обеспечением, сформировав уникальную экосистему. Инженеры, производители и энтузиасты технологий активно разрабатывают новые программные приложения на основе продуктов Espressif, свободно общаются и делятся своим опытом. Вы можете увидеть увлекательные идеи разработчиков, постоянно появляющиеся на различных платформах, таких как YouTube и GitHub. Популярность продуктов Espressif стимулировала рост числа авторов, которые выпустили более 100 книг на основе наборов микросхем Espressif, более чем на десяти языках, включая английский, китайский, немецкий, французский и японский.

Именно поддержка и доверие партнеров сообщества поощряют непрерывные инновации Espressif. «Мы стремимся сделать наши чипы, операционные системы, фреймворки, решения, облако, бизнес-практики, инструменты, документацию, тексты, идеи и т. д. еще более актуальными для ответов, необходимых людям для решения самых насущных проблем современной жизни. Это высочайшие амбиции и моральный ориентир Espressif», – сказал г-н Тео Суи Энн, основатель и генеральный директор компании Espressif.

Espressif ценит изложение идей. Поскольку непрерывное совершенствование технологий интернета вещей предъявляет все более высокие требования к инженерам, как мы можем помочь большему числу людей быстро освоить IoT-чипы, операционные системы, программные платформы, типовые схемы приложений и продукты облачных сервисов? Как говорится, лучше научить человека ловить рыбу, чем давать ему эту самую рыбу. Во время мозгового штурма нам пришло в голову, что мы могли бы написать книгу, в которой систематизировались бы ключевые знания об IoT-разработке. Мы договорились, быстро собрали группу старших инженеров, обобщивших опыт технической команды в области встроенного программирования, разработки аппаратного и программного обеспечения интернета вещей, что внесло свой вклад в публикацию этой книги. В процессе написания мы изо всех сил старались быть объективными и справедливыми, избавляться от шор и использовать максимально краткие выражения, чтобы рассказать о сложности и очаровании интернета вещей. Мы тщательно обобщили общие проблемы, отзывы и предложения сообщества, чтобы четко ответить на вопросы, возникающие в процессе разработки, и предоставить практические рекомендации по разработке интернета вещей для ответствующих технических специалистов и лиц, принимающих решения.

Структура книги

Эта книга ориентирована на инженеров и излагает необходимые знания для разработки проекта IoT шаг за шагом. Он состоит из четырех частей, а именно:

- **«Подготовка»** (главы 1–4): эта часть знакомит с архитектурой IoT, типичными структурами проектов, облачной платформой ESP RainMaker

и средой разработки ESP-IDF, закладывая прочную основу для разработки проекта IoT;

- **«Разработка оборудования и драйверов»** (главы 5–6): на основе чипсета ESP32-C3 в этой части подробно рассматриваются минимальная аппаратная система и разработка драйверов, а также реализуются диммирование, цветокоррекция и управление беспроводной связью;
- **«Беспроводная связь и управление»** (главы 7–11): в этой части объясняется интеллектуальная схема конфигурации Wi-Fi, основанная на чипе ESP32-C3, протоколах локального и облачного управления, а также локальном и удаленном управлении устройствами. Здесь также предоставляются способы разработки приложений для смартфонов, обновления встроенного ПО и управления версиями;
- **«Оптимизация и серийное производство»** (главы 12–15): эта часть предназначена для продвинутых приложений IoT, ориентированных на оптимизацию продуктов в области управления питанием, низкого энергопотребления и повышенной безопасности. Представлены примеры записи прошивки и тестирования в серийном производстве, а также диагностики рабочего состояния и ведения логов устройств через платформу удаленного мониторинга ESP Insights.

Об исходном коде

Читатели могут запускать примеры программ из этой книги либо путем ввода кода вручную, либо с использованием исходного кода, прилагаемого к книге. Мы подчеркиваем сочетание теории и практики и почти в каждую главу включаем раздел «Практика» на основе проекта Smart Light («Умного освещения»). Все коды открыты. Читатели могут скачать исходный код и обсудить его в разделах, связанных с этой книгой, на GitHub и нашем официальном форуме *esp32.com*. Открытый исходный код этой книги подчиняется условиям Apache License 2.0.

Авторское примечание

Эта книга официально выпущена компанией Espressif Systems и написана старшими инженерами компании. Она подходит для менеджеров и специалистов по исследованиям и разработкам в отраслях, связанных с IoT, преподавателей и студентов смежных специальностей, а также энтузиастам в области интернета вещей. Мы надеемся, что эта книга может служить и рабочим пособием, и справочником, и прикроватной книжкой, как хороший наставник и друг.

При составлении книги мы ссылались на некоторые актуальные результаты исследований экспертов, ученых и технических специалистов в стране² и за рубежом и сделали все возможное, чтобы привести их в соответствие с академическими нормами. Однако неизбежны некоторые упущения, поэтому здесь мы хотели бы выразить наше глубокое уважение и благодарность всем соответствующим авторам. Кроме того, мы цитируем информацию из интернета, поэтому хотели бы поблагодарить авторов и издателей оригиналов и прино-

² Так как Espressif Systems является китайской компанией со штаб-квартирой в Шанхае, то, очевидно, речь идет о Китае.

сим свои извинения за то, что не можем указать источник каждого фрагмента информации.

Для выпуска качественной книги мы организовали регулярные внутренние обсуждения, изучили предложения и отзывы читателей предварительных версий и редакторов издательства. Здесь мы хотели бы еще раз поблагодарить вас за вашу помощь, которая способствовала этой успешной работе.

Последнее, но самое главное: спасибо всем в Espressif, кто усердно работал для создания и популяризации нашей продукции.

Разработка IoT-проектов требует широкого круга знаний. В связи с ограниченным объемом книги, а также уровнем знаний и опыта авторов упущения неизбежны. Поэтому просим экспертов и читателей критиковать и исправлять наши ошибки. Если у вас есть какие-либо предложения по этой книге, пожалуйста, свяжитесь с нами по адресу book@espressif.com. Мы с нетерпением ждем ваших отзывов.

Как пользоваться этой книгой

Код изложенных в этой книге проектов находится в открытом доступе. Вы можете скачать его с нашего репозитория GitHub, поделиться своими мыслями и задать вопросы на нашем официальном форуме.

GitHub: <https://github.com/espressif/book-esp32c3-iot-projects>

Форум: <https://www.esp32.com/bookc3>

В книге выделяются следующие части текста:

Исходный код

В этой книге мы подчеркиваем сочетание теории и практики, и, таким образом, раздел о проекте Smart Light найдется почти в каждой главе. Соответствующие шаги и ссылка на исходный код будут размещаться между двумя линиями и помечаться заголовком **Исходный код**.

Примечание/Совет

Здесь вы можете найти важную информацию и напоминания об успешной отладке вашей программы. Она будет размещена между двумя толстыми линиями и помечена заголовком **Примечание** или **Совет**.

Большинство команд в этой книге выполняются под Linux и помечаются символом системного приглашения «\$». Если для выполнения команды требуются привилегии суперпользователя, приглашение будет заменено на "#". Командная строка в системах Mac помечается «%», как указано в разделе 4.2.3 «Установка ESP-IDF на Mac».

Основной текст этой книги будет напечатан обычным (пропорциональным) шрифтом, а примеры кода, компоненты, функции, переменные будут обозначены моноширинным шрифтом. Имена файлов, каталоги, пути и гиперссылки обозначаются курсивом.

Команды или тексты, которые должны быть введены пользователем, и команды, которые могут быть введены нажатием клавиши **Enter**, будут напечатаны **жирным моноширинным шрифтом**. Логи и блоки кода будут помечены голубым фоном. Гиперссылки обозначены синим курсивом: *[гиперссылка](#)*.

Пример:

Во-вторых, используйте `esp-idf/components/nvs_flash/nvs_partition_generator/nvs_partition_gen.py` для создания раздела бинарных файлов NVS на хосте разработки с помощью следующей команды:

```
$ python $IDF_PATH/components/nvs_flash/nvs_partition_generator/ nvs_partition_gen.py  
--input mass prod.csv --output mass prod.bin --size NVS_PARTITION_SIZE.
```


Часть I

Подготовка

Здесь начинается наше путешествие к полноценному проекту интернета вещей. В этой части мы сначала познакомимся с архитектурой и базовыми знаниями IoT, затем с сервисами ESP RainMaker на основе проекта Smart Light. Наконец, установим платформу разработки интернета вещей Espressif (фреймворк ESP-IDF) для дальнейшего изучения. К концу данной части вы будете знакомы с концепциями, описанными в этой книге далее, и сможете разрабатывать и компилировать примеры проектов.

ГЛАВА 1

Введение в IoT

ГЛАВА 2

Введение в практику IoT-проектов

ГЛАВА 3

Введение в ESP RainMaker

ГЛАВА 4

Настройка среды разработки



В конце XX века, с появлением компьютерных сетей и коммуникационных технологий, интернет быстро интегрировался в жизнь людей. В процессе развития интернет-технологий родилась идея интернета вещей (IoT). Буквально IoT означает связь вещей между собой с помощью интернета. В то время как обычный интернет ломает границы пространства и времени и сужает дистанцию между «человеком и человеком», IoT делает «вещи» важным участником, сближает «людей» и «вещи». В обозримом будущем интернет вещей станет движущей силой информационной индустрии.

Итак, что такое интернет вещей?

Точное определение интернету вещей дать трудно, поскольку его значение и масштабы постоянно развиваются. В 1995 году Билл Гейтс впервые поднял идею IoT в своей книге «*The Road Ahead*» («Дорога в будущее»). В его изложении IoT позволяет объектам обмениваться информацией друг с другом через интернет. Конечной целью является создание «интернета всего». Это ранняя интерпретация IoT, а также фантазия о технологиях будущего. Тридцать лет спустя, с бурным развитием экономики и технологии, фантазия становится реальностью. От «умных устройств», «умных домов», «умных городов», интернета транспортных средств и носимых устройств – к виртуальной «метавселенной», поддерживаемой технологиями IoT. Постоянно появляются новые концепции. В этой главе мы начнем с объяснения архитектуры интернета вещей, а затем познакомимся с одним из наиболее общих приложений IoT – «умным домом», чтобы помочь вам получить четкое представление.

1.1. Архитектура интернета вещей

Интернет вещей включает в себя несколько технологий, которые в различных отраслях имеют разные прикладные свойства и формы. Для того чтобы разобраться в структуре, ключевых технологиях и прикладных характеристиках IoT, необходимо установить единую архитектуру и стандартную техническую систему. В этой книге архитектура IoT просто разделена на четыре уровня: уровень восприятия и управления, сетевой уровень, уровень платформы и уровень приложений.

Уровень восприятия и управления

Как самый основной элемент архитектуры IoT, уровень восприятия и управления является ядром, реализующим всестороннее представление IoT. Его основная функция состоит в том, чтобы собирать, идентифицировать и управлять информацией. Он состоит из множества устройств, обладающих способностью

сбора информации, идентификации, управления и исполнения, а также отвечает за получение и анализ данных, таких как свойства материалов, поведенческие тенденции и состояние устройства. Таким образом, IoT получает способность познавать реальный физический мир. Кроме того, этот слой также может контролировать статус устройства.

Наиболее распространенными устройствами этого слоя являются различные датчики, играющие важную роль в сборе и идентификации информации. Датчики похожи на органы чувств человека: например, фоточувствительные датчики эквивалентны зрению, акустические датчики – слуху, газовые датчики – обонянию, а датчики, чувствительные к давлению и температуре, – осязанию. Со всеми этими «органами чувств» предметы становятся «живыми» и способными к осмысленному восприятию, узнаванию и манипулированию физическим миром.

Сетевой уровень

Основная функция сетевого уровня заключается в передаче информации, включая данные, полученные с уровня восприятия и управления, заданному целевому объекту, а также команд, выданных с прикладного уровня, обратно на уровень восприятия и управления. Сетевой уровень служит основным коммуникационным мостом, соединяющим различные уровни системы интернета вещей. Для создания базовой модели интернета вещей необходимо выполнить две операции по интеграции объектов в сеть: доступ к интернету и передачу данных через интернет.

Доступ к интернету

Интернет обеспечивает взаимосвязь между людьми, но пасует при включении вещей в сообщество. До появления интернета вещей большинство вещей не были «подключены к сети». Только непрерывное развитие технологий позволило подключать объекты к интернету, тем самым реализуя взаимосвязь между «людьми и вещами» и «вещами и вещами». Существует два распространенных способа подключения к интернету: доступ к проводной сети и доступ к беспроводной сети.

К проводным методам доступа относятся Ethernet, последовательная связь (например, RS-232, RS-485) и USB. Доступ через беспроводную сеть зависит от беспроводной связи, которую можно разделить на беспроводную связь ближнего действия и дальнюю беспроводную связь.

Беспроводная связь ближнего действия включает ZigBee, Bluetooth®, Wi-Fi, NFC³ и радиочастотную идентификацию (RFID). Беспроводная связь большого радиуса действия включает стандарты мобильной связи (2G, 3G, 4G, 5G), а так-

³ NFC (Near field communication, «связь в близком поле») – стандарт связи устройств на расстояниях 10 см и менее, представляющий собой расширение стандарта бесконтактных карт. В отличие от Bluetooth, может осуществлять связь только «точка–точка»; в отличие от радиочастотной идентификации (RFID), связь полноценно двухсторонняя. Всем известный пример применения NFC в быту – эмуляция бесконтактных банковских карт с помощью мобильного устройства при оплате покупок или проезда в общественном транспорте.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru