Оглавление

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 заканчивая 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 AIoT[1].

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-разрядный 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 соответствующих 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 RainMakerr

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 стране[2] 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

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 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, которые можно разделить на беспроводную связь ближнего действия и дальнюю беспроводную связь.

Беспроводная связь ближнего действия включает ZigBee, Bluetooth®, Wi-Fi, NFC[3] 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.

же такие технологии, как LoRa, eMTC, узкополосный интернет вещей (NB-IoT)[4] и т. д.

Передача данных через интернет

Различные методы доступа в интернет определяют соответствующий физический канал передачи данных. Следующее, что нужно сделать, – это решить, какой протокол связи использовать для передачи. По сравнению с интернет-терминалами, большинство IoT-терминалов в настоящее время имеют меньше доступных ресурсов, таких как производительность обработки, емкость памяти, скорость передачи и т. д., поэтому в IoT-приложениях необходимо выбрать протокол связи, который занимает меньше ресурсов. В настоящее время широко используются два протокола связи: MQTT (Message Queuing Telemetry Transport, передача телеметрии с очередью сообщений) и CoAP (Constrained Application Protocol, протокол ограниченного приложения).

Уровень платформы

Уровень платформы в основном относится к облачным платформам IoT. Когда все терминалы IoT объединены в сеть, их данные должны быть агрегированы на облачной платформе IoT для расчетов и хранения. Уровень платформы в основном поддерживает приложения IoT для облегчения доступа и управления крупными устройствами. На уровне платформы IoT-терминалы подключаются к облачной платформе, собираются данные терминалов и выдаются команды дистанционного управления. Как промежуточный сервис для назначения оборудования отраслевым приложениям, уровень платформы играет связующую роль во всей архитектуре IoT, неся абстрактную бизнес-логику и стандартизированную базовую модель данных, которая может не только реализовать быстрый доступ к устройствам, но также обеспечивает мощные модульные возможности для удовлетворения различных потребностей в промышленных прикладных сценариях. Уровень платформы в основном включает в себя функциональные модули, такие как доступ к устройствам, управление устройствами, управление безопасностью, обмен сообщениями, мониторинг операций и эксплуатации, техническое обслуживание, а также приложения для работы с данными:

- доступ к устройствам, реализующий соединение и связь между терминалами и облачными платформами IoT;
- управление устройствами, включая такие функции, как подключение и обслуживание устройств, преобразование и синхронизация данных и распределение устройств;
- управление безопасностью, обеспечение передачи данных IoT с точки зрения безопасной аутентификации и связи;
- обмен сообщениями, включая три направления передачи, т. е. терминал отправляет данные на облачную платформу IoT, облачная платформа IoT

[4] LoRa (Long Range) – технология маломощной сети передачи данных со скоростью 0,3–50 кбит/с и дальностью от 1 до 15 км в нелицензируемых диапазонах частот (ISM). eMTC (Enhanced Machine Type Communication) и NB-IoT (Narrow Band Internet of Things) – специально разработанные для IoT разновидности сотовой связи, поверх LTE (eMTC) или LTE/GSM (NB-IoT).

отправляет данные на сторонний сервер или другие облачные платформы IoT, а серверная сторона удаленно управляет устройствами IoT;
- мониторинг O&M[5], включая мониторинг и диагностику, обновление прошивки, онлайн-отладку, сервисы ведения лог-файлов и т. д.;
- приложения данных, включая хранение, анализ и применение данных.

Прикладной уровень

Прикладной уровень использует данные уровня платформы для управления приложением, производит их фильтрацию и обработку с помощью таких инструментов, как базы данных и аналитическое программное обеспечение. Полученные данные можно использовать для реальных приложений IoT, таких как интеллектуальное здравоохранение, интеллектуальное сельское хозяйство, умные дома и умные города.

Конечно, архитектуру IoT можно разделить на большее количество уровней, но сколько бы ни было слоев, из которых он состоит, основной принцип остается по существу одним и тем же. Изучение архитектуры IoT помогает углубить наше понимание технологий и построить полнофункциональные IoT-проекты.

1.2. Применение IoT в проекте «Умного дома»

IoT проник во все сферы жизни. Наиболее близкое к нам приложение IoT – «Умный дом». Многие традиционные бытовые приборы теперь оснащены одним или несколькими IoT-устройствами, и многие недавно построенные дома с самого начала проектируются с использованием технологий IoT.

На рис. 1.1 показаны некоторые распространенные устройства умного дома.

Развитие умного дома можно просто разделить на стадию подбора «умного» продукта, этап взаимодействия сценариев и интеллектуальный этап, как показано на рис. 1.2.

Первый этап касается **умных продуктов**. В отличие от традиционных домов, в умных домах устройства IoT получают сигналы с датчиков и объединяются в сеть посредством беспроводной связи с помощью таких технологий, как Wi-Fi, Bluetooth LE или ZigBee. Пользователи могут управлять интеллектуальными продуктами различными способами – через приложения на смартфонах, с помощью голосовых помощников, интеллектуального управления колонками и т. д.

Второй этап фокусируется на **взаимодействии сценариев**. На этом этапе разработчики больше не рассматривают возможность управления одним умным продуктом, а соединяют два или более таких продукта, автоматизируя взаимодействие до определенной степени и, наконец, формируя общий пользовательский режим всего окружения. Например, когда пользователь нажимает любую кнопку установки режима, свет, шторы и кондиционеры выключаются, автоматически возвращаясь к предустановкам. Конечно, есть условие, что

[5] О функции Operation and Maintenance (O&M) см. стр. 25. Не путать с международным стандартом O&M (Observations and Measurements) на форматы наблюдений и измерений, относящимся к географическим информационным системам.

логика взаимодействия легко настраивается, включая условия срабатывания и действия для выполнения команды. Представьте, что режим обогрева кондиционера срабатывает при снижении температуры в помещении ниже 10 °C; что в 7 часов утра играет музыка, чтобы разбудить пользователя, умные шторы открываются и рисоварка или тостер запускается через смарт-розетку. Пользователь встает и заканчивает умываться, а завтрак уже подан, так что можно не опаздывать на работу. Как удобна стала наша жизнь!

Рисунок 1.1. Обычные устройства умного дома

Рисунок 1.2. Этапы разработки умного дома

Третий этап – **интеллектуальный**. По мере доступа к большему количеству умных домашних устройств, растет количество типов генерируемых данных. При участии облачных вычислений, технологий больших данных и искусственного интеллекта это похоже на то, как в умные дома подсадили «умный мозг», который уже не требует частых команд от пользователя. Он собирает данные о предыдущих действиях, изучает модели поведения и предпочтения пользователя, чтобы автоматизировать действия, в том числе предоставляя рекомендации для принятия решения.

В настоящее время большинство умных домов находятся на стадии межсетевого взаимодействия. По мере развития увеличиваются скорость и интеллект умных продуктов, устраняются барьеры между коммуникационными протоколами. В будущем умные дома обязательно станут действительно «умными», как ИИ Джарвис в «Железном человеке»[6], который может не только помочь пользователю управлять различными устройствами и справляться с повседневными делами, но также обладает супервычислительной мощностью и выдающимися мыслительными способностями. На интеллектуальной стадии люди будут получать более качественные услуги как по количеству, так и по качеству.

[6] Джарвис (Jarvis) – верный дворецкий на основе искусственного интеллекта, вымышленный персонаж американских комиксов, издаваемых Marvel Comics. Джарвис чаще всего изображается в виде супергероя, как, например, в фильме «Железный человек» (2008).

Глава 2 Введение в практику IoT-проектов

В главе 1 мы представили архитектуру интернета вещей, роли и взаимосвязи уровней восприятия и управления, сетевого уровня, уровня платформы и приложений, обозначили развитие проекта умного дома. Однако так же, как при обучении рисованию, теоретических знаний далеко недостаточно, чтобы по-настоящему овладеть технологией, мы должны «запачкать руки», чтобы внедрить IoT-проекты на практике. Кроме того, когда проект переходит на стадию серийного производства, необходимо учитывать большое количество факторов, таких как подключение сети, ее настройка, взаимодействие с облачной платформой IoT, управление прошивкой и обновлениями, управление массовым производством и настройки безопасности.

Итак, на что нам нужно обратить внимание при разработке полного проекта IoT?

В главе 1 мы упомянули, что умный дом является одним из наиболее распространенных приложений интернета вещей, а умное освещение – одно из самых простых и практичных устройств, которое можно использовать в домах, отелях, спортзалах, больницах и т. д. Поэтому в этой книге мы возьмем создание проекта умного освещения в качестве отправной точки, объясним его компоненты и функции и дадим рекомендации по развитию проекта. Мы надеемся, что из этого случая вы сможете сделать выводы для создания большего количества IoT-приложений.

2.1. Введение в типовые проекты IoT

С точки зрения развития основные функциональные модули IoT-проектов можно разделить на разработку программного и аппаратного обеспечения IoT-устройств, разработку клиентских приложений и разработку облачной платформы IoT. Важно уточнить содержание этих основных модулей проекта, которые будут дополнительно описаны в этом разделе.

2.1.1. Базовые модули для обычных устройств IoT

Программно-аппаратная разработка IoT-устройств включает в себя следующие основные модули:

Сбор данных

В качестве нижнего уровня архитектуры IoT устройства для наблюдения и управления соединяют датчики и устройства через их чипы и периферийные устройства для сбора данных и контроля работы.

Привязка аккаунта и первоначальная настройка

Для большинства IoT-устройств привязка учетной записи и первоначальная настройка объединяются в один рабочий процесс, например подключение устройств и пользователей путем настройки сети Wi-Fi.

Управление устройством

При подключении к облачным платформам IoT устройства могут взаимодействовать с облаком – регистрироваться, связываться или управляться. Пользователи могут запрашивать статус продукта и выполнять другие операции в приложении для смартфона через облачные платформы IoT или протоколы локальной связи.

Обновление прошивки

Устройства IoT также могут обновлять прошивку в зависимости от потребностей производителей. Обновление прошивки и управление версиями осуществляются через команды, отправленные из облака. С помощью этой функции обновления прошивки вы можете постоянно совершенствовать функции устройств IoT, исправлять дефекты и улучшать взаимодействие с пользователем.

2.1.2. Базовые модули клиентских приложений

Клиентские приложения (например, приложения для смартфонов) в основном включают следующие базовые модули:

Система аккаунта и авторизации

Эта система поддерживает авторизацию учетной записи и регистрацию устройства.

Управление устройством

Приложения для смартфонов обычно снабжены управляющими функциями. Пользователи могут легко подключаться к IoT-устройствам и управлять ими в любое время и в любом месте с помощью приложений для смартфонов. В реальном умном доме устройства в основном управляются через приложения для смартфонов, которые не только обеспечивают интеллектуальное управление устройствами, но и экономят затраты на рабочую силу. Таким образом, управление устройством является обязательным для клиентских приложений, таких как управление функциями устройства, управление сценариями, планирование, дистанционное управление, привязка устройств и т. д. Пользователи умного дома также могут настраивать сценарии в соответствии с личными потребностями, контролируя освещение, бытовую технику, входную дверь и т. д., чтобы сделать домашнюю жизнь более комфортной и удобной. Они могут задать время кондиционирования, выключить кондиционер удаленно, включить свет в коридоре автоматически, как только дверь разблокирована, или переключиться в режим «театр» одной-единственной кнопкой.

Уведомления

Клиентские приложения контролируют состояние устройств IoT в режиме реального времени и отправляют оповещения, когда устройства ведут себя ненормально.

Послепродажное обслуживание клиентов

Приложения для смартфонов могут предоставлять послепродажное обслуживание продуктов для своевременного решения проблем, связанных с отказами IoT-устройств и техническими операциями.

Рекомендуемые функции

Для удовлетворения потребностей разных пользователей могут быть добавлены другие функции, такие как управление жестами, NFC, GPS и т. д. GPS может помочь установить точность операций в зависимости от местоположения и расстояния, а функция управления жестами позволяет пользователям устанавливать команды, которые будут выполняться для определенного устройства или сценария с помощью перемещения или встряхивания смартфона.

2.1.3. Введение в общие облачные платформы IoT

Облачная платформа IoT – это универсальная платформа, объединяющая такие функции, как управление устройствами, связь для обеспечения безопасности данных и управление уведомлениями. Согласно их целевому назначению и доступности, облачные платформы IoT можно разделить на общедоступные (далее именуемые «**публичное облако**») и частные (далее называются «**частным облаком**»).

Публичное облако обычно означает общие облачные платформы IoT для предприятий или частных лиц, которые эксплуатируются и обслуживаются поставщиками платформ и управляются через интернет. Могут быть бесплатными или по низкой цене, а также предоставлять услуги в открытой общедоступной сети, такой как Alibaba Cloud, Tencent Cloud, Baidu Cloud, AWS IoT, Google IoT и т. д. В качестве вспомогательной платформы общедоступное облако может интегрировать вышестоящих поставщиков услуг и конечных пользователей для создания новой экосистемы.

Частное облако создано только для корпоративного использования, что гарантирует лучший контроль над данными, лучшую безопасность и качество обслуживания. Его услуги и инфраструктура обслуживаются отдельными предприятиями, а вспомогательное аппаратное и программное обеспечение также предназначено для конкретных пользователей. Предприятия могут настраивать облачные сервисы в соответствии с потребностями своего бизнеса. В настоящий момент некоторые производители умных домов уже получили частные облачные платформы IoT и разработали приложения для умного дома на их основе.

Публичное облако и частное облако имеют свои преимущества, которые будут объяснены позже. Для достижения коммуникационной связности необходимо завершить как минимум встроенную разработку на стороне устройства наряду с бизнес-серверами, облачными платформами IoT и приложениями для смартфонов. Столкнувшись с таким огромным проектом, общедоступное облако обычно предоставляет комплекты для разработки ПО приложений на стороне устройства и смартфона, чтобы ускорить процесс. Как общедоступное, так и частное облако предоставляет услуги, включая доступ к устройству, управление устройством, теневые копии устройства, а также эксплуатацию и техническое обслуживание.

Доступ к устройству

Облачные платформы IoT должны предоставлять не только интерфейсы для доступа к устройствам с использованием таких протоколов, как MQTT, CoAP, HTTPS и WebSocket, но также и функции безопасности – аутентификацию для блокировки поддельных и незаконных устройств, эффективно снижающую риск быть скомпрометированым. Такая аутентификация обычно поддерживает различные механизмы, поэтому, когда устройства выпускаются серийно, необходимо предварительное присвоение сертификата устройства согласно выбранному механизму аутентификации и запись его на устройство.

Управление устройствами

Функция управления устройствами, предоставляемая облачными платформами IoT, может не только помочь производителям следить за статусом активации и онлайн-статусом своих устройств в режиме реального времени, но также позволяет такие действия, как добавление/удаление устройств, извлечение, добавление/удаление группы, обновление прошивки и управление версиями.

Теневые копии устройства

Облачные платформы IoT могут создавать постоянную виртуальную версию (теневую копию) для каждого устройства, и статус теневой копии может быть синхронизирован и получен приложением или другими устройствами через интернет-протоколы. Теневая копия устройства хранит последний отчетный статус и ожидаемый статус каждого устройства, и даже если устройство находится в автономном режиме, приложение по-прежнему может получать его статус, вызвав API. Теневая копия обеспечивает постоянно активные API, что упрощает создание приложений для смартфонов, взаимодействующих с устройствами.

Эксплуатация и обслуживание (O&M)

Функция O&M включает в себя три аспекта:

- демонстрация статистической информации об устройствах и уведомлениях интернета вещей;
- управление лог-файлами позволяет извлекать информацию о поведении устройства, потоке сообщений о включении/отключении и их содержании;
- отладка устройства поддерживает доставку команд, обновление конфигурации и проверку взаимодействия облачных платформ IoT с сообщениями от устройств.

2.2. Практика: проект Smart Light

После теоретического введения в каждой главе вы найдете практический раздел, связанный с проектом Smart Light, который поможет вам получить практический опыт. Проект основан на чипе Espressif ESP32-C3 и облачной платформе ESP RainMaker IoT Cloud и включает аппаратный беспроводной модуль в продуктах для умного освещения, встроенное программное обеспечение для интеллектуальных устройств на основе ESP32-C3, приложение для смартфонов и взаимодействие с ESP RainMaker.

Исходный код

Для лучшего обучения и развития опыта проект, описанный в этой книге, находится в открытом доступе. Вы можете загрузить исходный код из нашего репозитория GitHub по адресу *https://github.com/espressif/book-esp32c3-iot-projects*.

2.2.1. Структура проекта

Проект Smart Light состоит из трех частей:

I. **Устройства умного света на базе ESP32-C3**, отвечающего за взаимодействие с облаком IoT-платформы, а также осуществляющего управление выключением, яркостью и цветовой температурой светодиодов ламповой гирлянды.
II. **Приложения для смартфонов** (включая приложения для планшетов, работающие на Android и iOS), ответственные за сетевую конфигурацию продуктов Smart Light, а также служащие для запроса и управления их статусом.
III. **Облачной платформы IoT на основе ESP RainMaker**. Для упрощения в этой книге считаем облачную платформу IoT и бизнес-сервер единым целым. Подробности о ESP RainMaker будут представлены в главе 3.

Соответствие структуры проекта Smart Light общей архитектуре IoT показано на рис. 2.1.

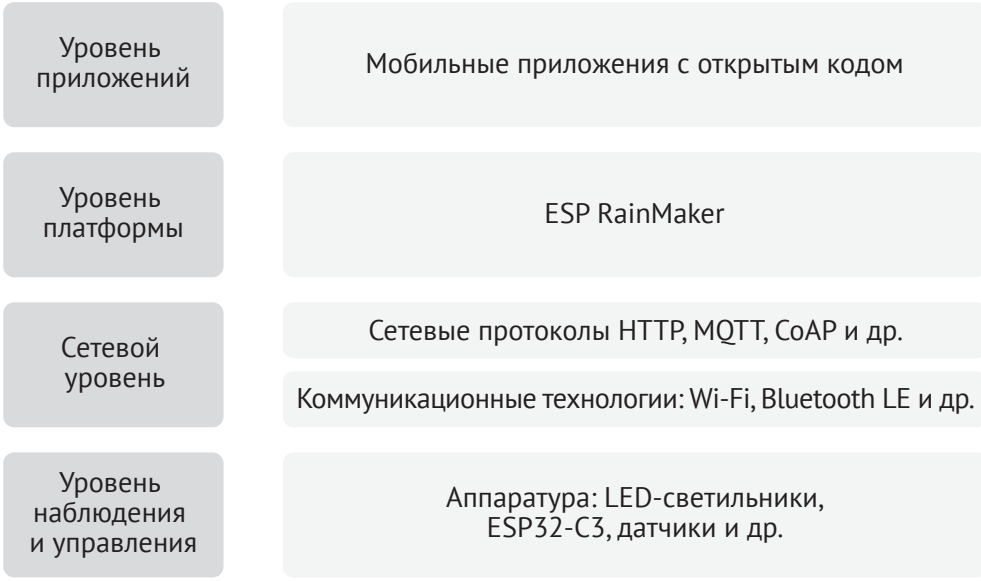

Рисунок 2.1. Структура проекта Smart Light

2.2.2. Функции проекта

В соответствии со структурой функции каждого уровня следующие.

Умные LED-светильники:

- настройка сети и подключение;
- ШИМ-управление светодиодами, например выключение, изменение яркости, цветовой температуры и т. д.;
- автоматизация или управление сценариями, например таймер;
- шифрование и безопасная загрузка с флеш-памяти;
- обновление прошивки и управление версиями.

Мобильные приложения:

- конфигурация сети и привязка устройств;
- интеллектуальное управление освещением, например выключение, изменение яркости, цветовой температуры и т. д.;
- настройки автоматизации или сценария, например таймер;
- местное/дистанционное управление;
- регистрация пользователя, вход в систему и т. д.

Облачная платформа интернета вещей ESP RainMaker IoT cloud:

- включение доступа к IoT-устройствам;
- предоставление API-интерфейсов управления устройством, доступных для мобильных приложений;
- обновление прошивки и управление версиями.

2.2.3. Подготовка оборудования

При реализации проекта вам потребуется следующее оборудование: умные светильники, смартфоны, Wi-Fi-роутеры и компьютер, соответствующий требованиям установки среды разработки.

Умные светильники

Умные светильники – это новый тип ламп, форма которых такая же, как у обычных ламп накаливания. Умный светильник состоит из понижающего регулируемого источника питания, беспроводного модуля (со встроенным ESP32-C3), светодиодного контроллера и светодиодной RGB-матрицы. При подключении к питанию выходное напряжение 15 В постоянного тока после понижения, диодного выпрямления и регулирования обеспечивает питанием светодиодный контроллер и светодиодную матрицу. Светодиодный контроллер может автоматически отправлять высокие и низкие уровни через определенные промежутки времени, переключая светодиодную RGB-матрицу между включенным (горит свет) и выключенным (свет выключен) состояниями, чтобы она могла излучать голубой, желтый, зеленый, фиолетовый, синий, красный и белый свет. Беспроводной модуль отвечает за подключение к Wi-Fi-роутеру, получающему и сообщающему сведения о состоянии умных светильников и отправляющему команды для управления светодиодом.

На ранней стадии разработки вы можете смоделировать умный свет, используя плату ESP32-C3 DevKitM-1, подключенную к светодиодной RGB-лампе (см. рис. 2.2). Но вы должны учитывать, что это не единственный способ собрать интеллектуальное осветительное устройство. Аппаратная часть проекта в этой книге содержит только беспроводной модуль (со встроенным ESP32-C3), но не полный аппаратный набор для реального проекта.

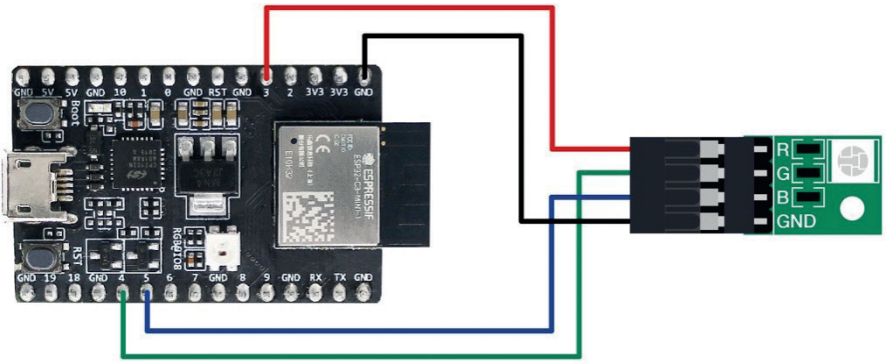

Рисунок 2.2. Имитатор умного света

Кроме того, Espressif также производит плату для разработки аудио на базе ESP32-C3 (ESP32-C3-Lyra) для управления светом со звуковым сопровождением. Плата имеет интерфейсы для микрофонов и динамиков и может управлять светодиодными лентами. Ее можно использовать для разработки сверхдешевых высокопроизводительных звуковых вещателей и с ритмическим световым сопровождением. На рис. 2.3 показана плата ESP32-C3-Lyra, соединенная со световой полосой из 40 светодиодов.

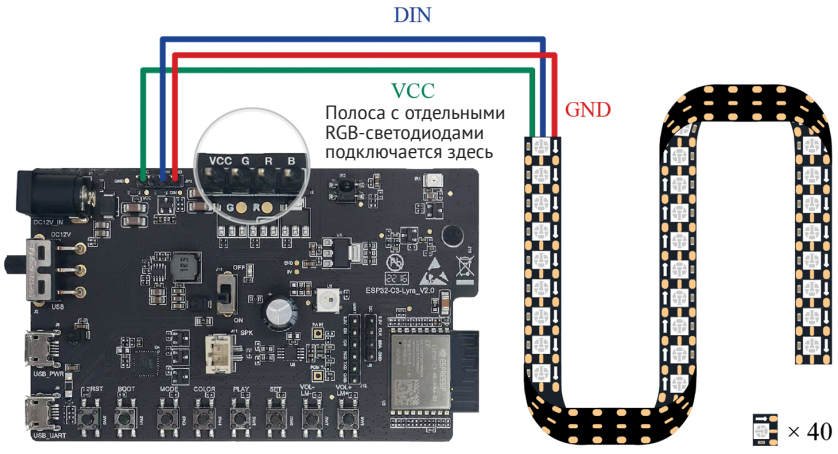

Рисунок 2.3. ESP32-C3-Lyra с подключенной полосой из 40 светодиодов

Смартфоны (Android/iOS)

Проект Smart Light предполагает разработку мобильного приложения для настройки и управления интеллектуальным осветительным устройством.

Wi-Fi-роутеры

Wi-Fi-роутеры преобразуют сигналы проводной сети и сигналы мобильной сети в сигналы беспроводной сети для компьютеров, смартфонов, планшетов и других беспроводных устройств, объединенных в сеть. Например, для широкополосного доступа в доме требуется только подключение к Wi-

Fi-роутеру для обеспечения беспроводной сети устройств Wi-Fi. Основной стандартный протокол, поддерживаемый маршрутизаторами Wi-Fi, – IEEE 802.11n со средней скоростью передачи 300 Мбит/с или максимум 600 Мбит/с. Он обратно совместим с IEEE 802.11b и IEEE 802.11g. Чип ESP32-C3 от Espressif поддерживает IEEE 802.11b/g/n, поэтому вы можете выбирать однодиапазонный (2,4 ГГц) или двухдиапазонный (2,4 ГГц и 5 ГГц) Wi-Fi-роутер.

Компьютер (Linux/macOS/Windows)

Среда разработки будет представлена в главе 4.

2.2.4. Процесс разработки

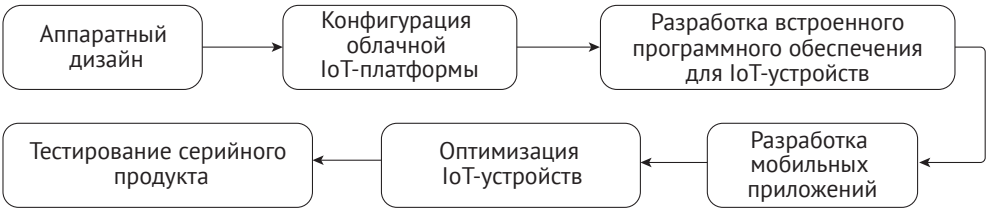

Рисунок 2.4. Этапы разработки проекта Smart Light

Аппаратный дизайн

Аппаратный дизайн устройств IoT имеет важное значение для проекта IoT. Полный проект Smart Light предназначен для изготовления лампы, управляемой по сети. Разные производители производят лампы разных стилей и типов драйверов, но их беспроводные модули обычно несут одни и те же функции. Для упрощения процесса разработки проекта Smart Light эта книга охватывает только проектирование оборудования и разработку программного обеспечения для беспроводных модулей.

Конфигурация облачной IoT-платформы

Чтобы использовать облачные платформы IoT, вам необходимо настроить проекты на серверной части, например задать продукты, создание устройств, настройку свойств устройств и т. д.

Разработка встроенного программного обеспечения для устройств IoT

Реализуйте необходимые функции с помощью ESP-IDF, Espressif SDK на стороне устройства, включая подключение к облачным платформам IoT, разработку драйверов светодиодов и обновление прошивки.

Разработка мобильного приложения

Разрабатывайте мобильные приложения для систем Android и iOS, чтобы реализовать регистрацию пользователей, вход в систему, управление устройством и другие функции.

Оптимизация IoT-устройств

После того как базовая разработка функций устройства IoT завершена, вы можете перейти к оптимизации задач, таких как оптимизация энергопотребления.

Тестирование серийного продукта

Проведение испытаний серийного продукта в соответствии с требованиями на функционирование оборудования, испытание старения, RF-тест[7] и т. д.

Несмотря на этапы, перечисленные выше, проект Smart Light не обязательно подлежит такой процедуре полностью, поскольку различные задачи могут выполняться одновременно. Например, встроенное программное обеспечение и мобильные приложения для смартфонов могут разрабатываться параллельно. Некоторые шаги также может потребоваться повторить, например оптимизацию устройств IoT и тестирование серийного производства.

2.3. Резюме

В этой главе мы изложили основные компоненты и функциональные модули проекта IoT, затем представили кейс практического проекта Smart Light, ориентируясь на его структуру, функции, подготовку оборудования и процесс разработки. Читатели могут сделать выводы для практики и уверенно выполнять проекты IoT с минимальными ошибками в будущем.

[7] RF-тест – измерение излучаемых сигналов радиочастотного спектра и их мощности на соответствие стандартам, гарантирующим отсутствие помех иным радиоустройствам, а также для исключения вредоносного воздействия на окружающую среду (см. главу 14).