ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

Отчет о программном проекте на тему:

Программный комплекс управления контроллером отопления по протоколу OpenTherm

Выполнил студент:

группы #БКНАД, 3 курса Канищев Илья Андреевич

Принял руководитель проекта:

Воронин Игорь Вадимович
Научный сотрудник
ИПЛИТ РАН, г. Шатура

Москва 2024

Содержание

Аннотация

В рамках данной работы было написано кроссплатформенное мобильное приложение, нацеленное на работу с существующим аппаратным комплексом для управления отоплением. Оно позволяет дистанционно управлять температурой горячей воды, отопления и другими параметрами котла, а также отображать показания датчиков температуры, установленных в доме. Приложение написано для мобильных операционных систем iOS и Android с использованием фреймворка Flutter.

Ключевые слова

Мобильное приложение, Умный дом, Отопление, iOS, Android

1 Введение

1.1 Описание предметной области

Системы "умного дома"стремительно набирают популярность в современном мире. Возрастающее количество бытовых устройств получают возможность дистанционного управления через интернет, мобильные приложения или голосовые команды, благодаря интеграции с голосовыми ассистентами. Одним из таких инновационных устройств являются газовые и электрические водонагревательные котлы для отопления жилых домов и промышленных помещений, обеспечивающие регулировку температуры горячей воды и отопления. Эти устройства особенно актуальны для загородных домов, которые часто используются не постоянно, делая дистанционное управление системой отопления не только удобным, но и необходимым.

1.2 Цели и задачи проекта

Целью данного проекта является разработка мобильного приложения для операционных систем iOS и Android, позволяющего управлять водонагревательным котлом на расстоянии. Основой для управления котлом служит контроллер SmartTherm, работающий на базе микроконтроллеров ESP8266/ESP32. Особенностью SmartTherm является его открытый исходный код A и совместимость с различными прошивками, что позволяет настроить его под многие модели котлов с протоколом OpenTherm. На начальном этапе, мобильное приложение предоставит возможности для мониторинга и регулировки температуры воды и отопления, а также просмотра графиков температуры.

1.3 Структура и Методология Работы

Проект будет реализован в несколько этапов:

1 Анализ существующих решений.

2 Разработка структуры классов и архитектуры приложения.

3 Разработка прототипа пользовательского интерфейса.

4 Написание и тестирование кода.

Выбор пал на язык программирования Dart и фреймворк Flutter, поскольку они обеспечивают возможность создания кроссплатформенных приложений с единой кодовой базой

для iOS и Android.

Связь между мобильным устройством пользователя и котлом совершается в несколько этапов. Пользователь сможет сконфигурировать свое устройство SmartTherm, подключив его к домашней сети Wi-Fi, после чего на выбор предоставляется возможность получать с него информацию через TCP по IP-адресу или через MQTT при наличии собственного MQTT-брокера или развернутого на сервере Home Assistant.

Помимо непосредственно взаимодействия с котлом, приложение также должно предусматривать создание пары с новым контроллером. На первом этапе была реализована синхронизация при помощи ручной настройки по сети через TCP соединение. В перспективе возможно добавить в приложение сканирование QR-кодов и класть в комплект к контроллеру QR-код, содержащий информацию необходимую для подключения устройства.

2 Обзор литературы

2.1 Контекст разработки

С развитием технологий умного дома, рынок обогатился множеством решений для дистанционного управления различными домашними устройствами, включая системы отопления. Среди них выделяются продукты, использующие различные подходы к интеграции с существующими домашними системами. Основным и наиболее привычным методом взаимодействия с такими системами чаще всего являются именно мобильные приложения. При этом, при выборе решения для своего дома, пользователи рассматривают целый набор характеристик, таких как удобство настройки и эксплуатации, пользовательский опыт, возможность интеграции с существующей инфраструктурой, функционал и цена. Помимо перечисленного, существует запрос на локализацию сборки аппаратуры и ее импортозамещение, а также возможность тонкой настройки программного обеспечения и использования аппаратуры, принадлежащей заказчику. Такой подход часто позволяет сократить издержки и увеличить стабильность поставки и надежность программно-аппаратного комплекса.

2.2 Анализ существующих решений

2.2.1 Проекты умного дома на базе ESP8266/ESP32

В интернете можно найти описания систем умного дома собственного изготовления, в основном разрабатываемых читателями технических форумов для собственных нужд. Как

правило, такие проекты используют семейства контроллеров ESP8266/ESP32 и Arduino в качестве основы для умных устройств, однако ограничиваются технической стороной и не включают разработку дружелюбного пользовательского интерфейса. В качестве примера можно привести данный цикл статей [7] [5] [6]

2.2.2 Home Assistant

Home Assistant [2] - крупный open-source A проект, направленный на создание и интеграцию систем умного дома, а также разработку сценариев автоматизации. По сути Home Assistant представляет из себя целую платформу из готовых компонентов для управления различными домашними устройствами. Данный фреймворк включает в себя операционную систему для развертывания на домашнем сервере, веб-интерфейс с кастомизируемой панелью управления и возможностью использования расширений от сообщества. Авторы Home Assistant предлагают в том числе официальное приложение для iOS и Android, однако оно требует подключения к сервису Home Assistant Cloud, предоставляемого на платной основе на серверах иностранной компании.
Про возможности Home Assistant можно подробнее узнать из этого доклада с конференции разработчиков свободных программ [4]

2.2.3 Коммерческие проекты

Примером коммерческих проектов для управления котлами может послужить семейство термостатов и отопительных контроллеров Zont [3]. Они предоставляют оборудование специально для дистанционного считывания и регулировки температуры отопления. Их конкурентным преимуществом является возможность управления через GSM - в контроллер установлена сим карта, и он может, по сути, общаться с внешним миром через обычную мобильную связь. Таким образом, повышается надежность связи с котлом и упрощается настройка для конечного пользователя.

2.3 Позиционирование собственного проекта

Отличием предлагаемого мобильного приложения является его специализация на управлении водонагревательными котлами с использованием контроллера SmartTherm, работающего на основе протокола Open Therm . Это позволяет обеспечить более точное и гибкое управление, в отличие от общих решений для умного дома. Кроме того, наш проект включает в себя разработку интуитивно понятного пользовательского интерфейса, что часто от-

сутствует в DIY-проектах. Это делает наш продукт более доступным для широкого круга пользователей, в отличие от технически ориентированных решений.
В то же время, если у пользователя есть соответствующее желание и навыки, то наш продукт предоставляет гибкость для настройки практически под любые конфигурации в силу открытости исходного кода контроллера SmartTherm. Например, пользователь может выбрать управление устройством с собственного домашнего сервера, через локальную сеть или через поставщика виртуальных облачных серверов, таких как Yandex Cloud или AWS.

3 Обзор оборудования и инструментов

3.1 ESP32

Для тестирования и разработки продукта использовались две платы Espressif ESP Wroom 32 DevBoard соединенные следующим образом 3.1 3.2, распиновку esp32 можно увидеть в приложении A.1. На одной из них была установлена рабочая версия ПО SmartTherm v 0.7.4, которая поставляется с модулем SmartTherm для потребителей. Другая плата отвечала за эмуляцию работы котла по протоколу OpenTherm. Подробнее о ESP32 можно узнать на сайте производителя [1].

3.2 Home Assistant

Для обеспечения возможности построения графиков на основе исторических данных температуры и других показаний с датчиков необходима возможность сохранять эти данные в реальном времени. Поскольку мобильное приложение не может постоянно работать в фоновом режиме для сбора данных, оптимальным решением является использование домашнего сервера или облачной инфраструктуры.
Мы решили опереться на возможности Home Assistant для хранения и обработки данных. Home Assistant должен быть развернут на устройстве пользователя, предоставляя удобный способ визуализации данных через графики. Это позволяет пользователям видеть долгосрочные тренды и анализировать изменения в показаниях термостата и других датчиков. В приложении для этого была создана секция, отображающая веб-страницу с Home Assistant.

Рис. 3.1: Модуль SmartTherm в паре с эмулятором котла

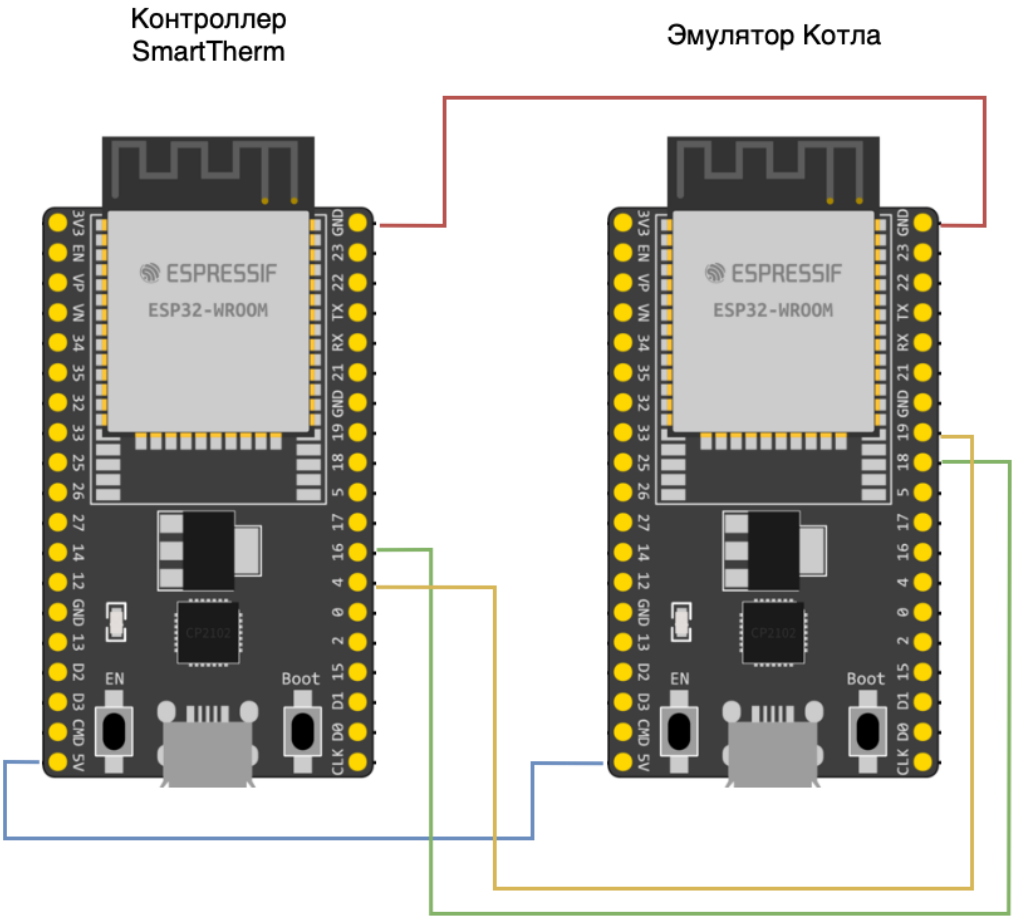

Рис. 3.2: Схема соединения

4 Ключевые детали реализции приложения

4.1 Протоколы

Как было указано ранее, в качестве основных протоколов для взаимодействия с модулем SmartTherm были выбраны TCP и MQTT. Такой выбор протоколов во многом продиктован аппаратными ограничениями платформы ESP32: из-за ее малой мощности и небольшого количества оперативной памяти необходимо использовать легковесные протоколы, поэтому TCP и MQTT часто используются для Интернета Вещей.

4.2 Бизнес-логика

Для управления состоянием приложения была выбрана библиотека BLoC для Flutter. Она позволяет удобно организовать управление состоянием через создание событий и обновление состояния в ответ на них.

4.2.1 Библиотека BLoC

Библиотека BLoC (Business Logic Component) для Flutter позволяет удобно разделить бизнес-логику и пользовательский интерфейс. Это достигается за счет использования потоков и реактивного программирования, что позволяет управлять состоянием приложения и обрабатывать события асинхронно.

Основные компоненты BLoC включают:

- BLoC: Отвечает за обработку событий и изменение состояний. BLoC получает события от пользовательского интерфейса, выполняет соответствующую бизнес-логику и обновляет состояние, которое затем передается обратно интерфейсу.
- Events: Представляют собой действия, которые требуют изменения состояния приложения. Например, при нажатии пользователем кнопки отправляется Event под названием UserButtonClick, которое обрабатывается коллбеком внутри BLoC.
- States: Представляют текущее состояние приложения, которое изменяется в ответ на события. UI-компоненты могут подписаться на изменение этого состояния и при необходимости обновлять интерфейс.

Использование библиотеки BLoC в приложениях Flutter имеет несколько ключевых преимуществ: она способствует четкому разделению бизнес-логики и UI, позволяет лучше масштабировать приложение и облегчает тестирование.

4.2.2 Компонент ThermostatControlBloc

Для управления большей частью бизнес-логики в приложении мной был разработан ThermostatControlBloc. Он отвечает за подключение и удаление устройств, изменение информации о них (например, имени или IP-адреса), а также за управление функциями котла и получение с него новой информации о текущем состоянии.

4.3 Сервис Network Service

Для взаимодействия по сети приложение использует сервис NetworkService, который выбирает одну из двух реализаций сетевого взаимодействия в зависимости от указанного пользователем протокола: TCPNetworkService или MQTTNetworkService. Оба сервиса реализуют интерфейс NetworkServiceImpl, описанный далее.

4.4 Интерфейс NetworkServiceImpl

Интерфейс NetworkServiceImpl определяет методы, которые должны реализовать все классы, которые могут быть использованы в качестве реализации NetworkService:

- ConnectToThermostat: подключение к котлу.
- GetThermostatStatus: получение статуса котла.
- SetParameters: установка параметров котла.

4.5 TCPNetworkService

TCPNetworkService реализует интерфейс NetworkServiceImpl для взаимодействия по протоколу TCP. Чтобы инкапсулировать низкоуровневое взаимодействие с TCP сокетами, мной был написан класс TCPConnection, который отвечает за подключение через TCP сокет, отправку запросов и получение ответов. Основные методы TCPConnection включают:

- Start: подключается к контроллеру SmartTherm через IP и порт.
- SubmitRequest: отправляет запрос к котлу, кодируя его в бинарном формате, и декодирует ответ, полученный от сокета.
- Close: закрывает соединение.

4.6 MQTTNetworkService

MQTTNetworkService реализует интерфейс NetworkServiceImpl для взаимодействия по протоколу MQTT. Он использует библиотеку mqtt_client, которая предоставляет классы для взаимодействия по MQTT. Она позволяет подписаться на MQTT-топики, созданные контроллером SmartTherm, чтобы получать обновленную информацию с котла.

5 Заключение

5.1 Итоги работы

В результате проделанной работы создано функциональное мобильное приложение для управления котлом с поддержкой интерфейса OpenTherm при помощи модуля SmartTherm. Оно предоставляет удобный пользовательский интерфейс для управления климатическими условиями в доме, используя протоколы TCP, либо MQTT в зависимости от потребностей и задач пользователя. Приложение позволяет в реальном времени наблюдать за различными показателями деятельности котла, от температуры отопления и горячей воды для мытья рук до более специализированных, таких как работа горелки и температура обратки, а при наличии настроенного Home Assistant, пользователь может также видеть графики температуры котла и других показателей. Приложение позволяет добавлять несколько устройств и управлять ими независимо друг от друга.

Также на главную страницу приложения был интегрирован виджет погоды, который показывает прогноз на день в текущей локации, что может быть актуально пользователю, чтобы настроить комфортный для себя уровень отопления или не забыть включить его в день наступления заморозков.

Код приложения был выложен на github по ссылке в приложении: A.

Итоговую схему основных классов в проекте можно увидеть на рисунке 5.1

5.2 Перспективы улучшения

Несмотря на то что большая часть поставленных задач была выполнена, нам удалось выявить несколько направлений, по которым можно улучшить приложение в будущем.

Во-первых, в приложении можно реализовать систему подсказок и туториалов, чтобы при нажатии на знак вопроса рядом с непонятным термином пользователь мог увидеть объяснение того, за что отвечает данный параметр. Также можно добавить туториалы, описывающие

некороые процессы внутри приложения в целом.

Во-вторых, в будущем хотелось бы реализовать механизм поиска контроллера SmartTherm в локальной сети. Для этого необходимо доработать прошивку контроллера и приложение, используя выбранный алгоритм, чтобы смартфон и контроллер могли узнать IP друг друга в автоматическом режиме.

В-третьих, в данный момент в прошивке контроллера отсутствует возможность управлять температурой горячей воды для мытья рук через MQTT. Когда эта возможность будет добавлена, приложение можно будет обновить для поддержки этой опции.

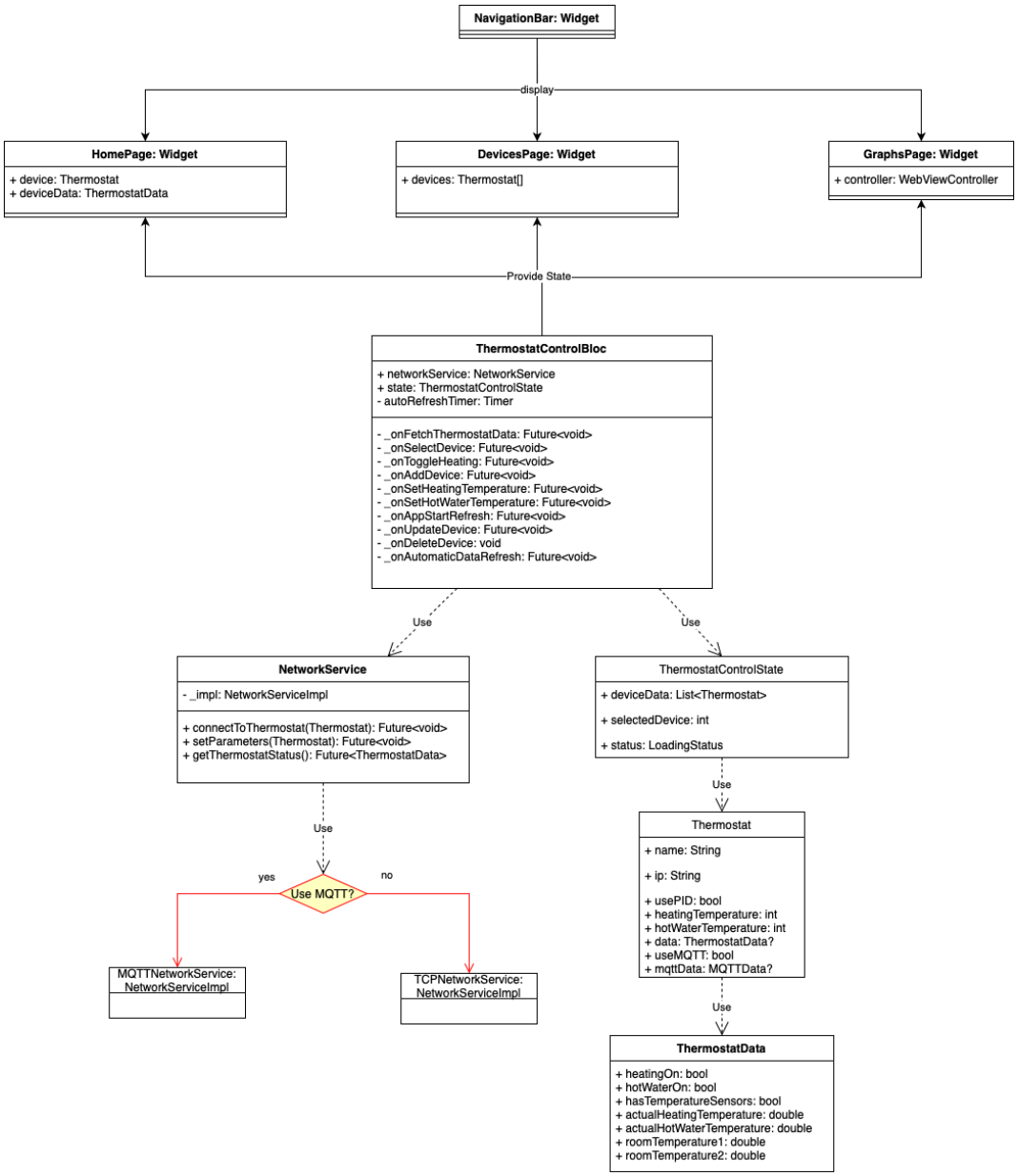

Рис. 5.1: Диаграма основных классов итогового приложения

Список литературы

[1] *ESP32 series datasheet*. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (дата обр. 25.05.2024).

[2] Home Assistant Core Team и Community. *Home Assistant*. URL: https://www.home-assistant.io (дата обр. 31.01.2024).

[3] *Автоматика ZONT - официальный сайт производителя*. URL: https://zont-online.ru (дата обр. 31.01.2024).

[4] Игорь Воронин, Евгений Коцюба, Ростислав Воронин. "Использование СПО HOME ASSISTANT в среде Alt Linux для автоматизации хозяйственной деятельности организации". В: *Девятнадцатая конференция разработчиков свободных программ* (2023), с. 43—47.

[5] Святослав Хусамов. *Мой умный дом на ESP8266, часть 2*. URL: https://habr.com/ru/articles/544328/ (дата обр. 31.01.2024).

[6] Святослав Хусамов. *Мой умный дом на ESP8266, часть 3*. URL: https://habr.com/ru/articles/545904/ (дата обр. 31.01.2024).

[7] Святослав Хусамов. *Схема моего умного дома на основе ESP8266, часть 1*. URL: https://habr.com/ru/articles/543536/ (дата обр. 31.01.2024).

A Приложение

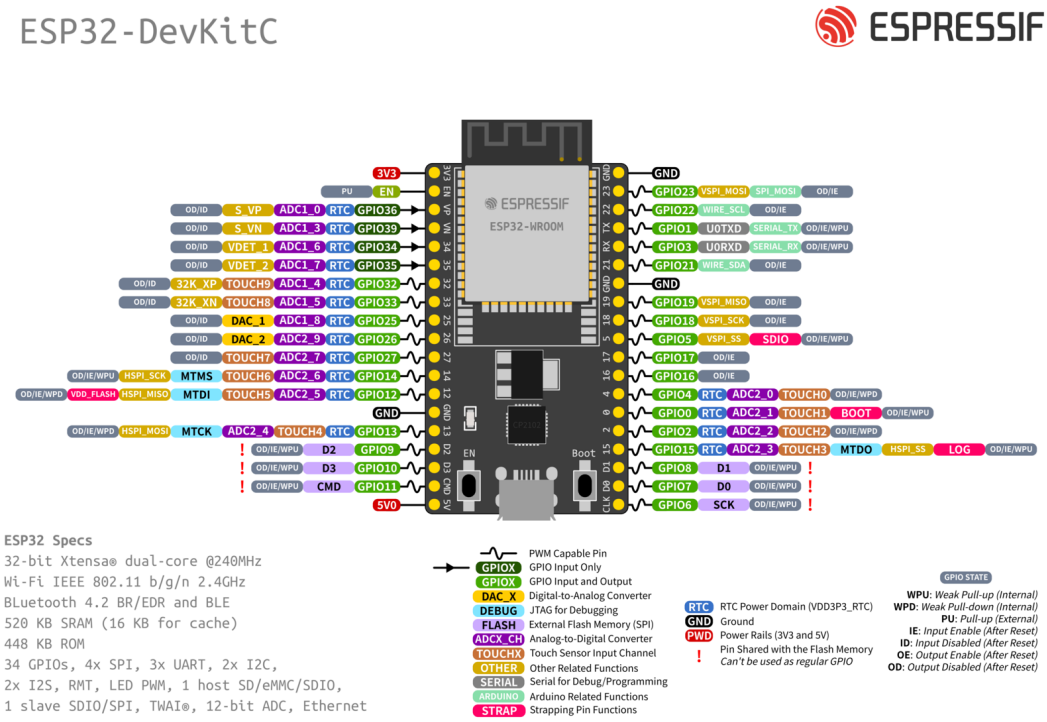

Рис. A.1: Распиновка платы ESP32

Ссылка на репозиторий проекта: https://github.com/kanishev2002/smart_therm

Ссылка на репозиторий прошивки SmartTherm: https://github.com/Evgen2/SmartTherm

Ссылка на репозиторий Home Assistant: https://github.com/home-assistant

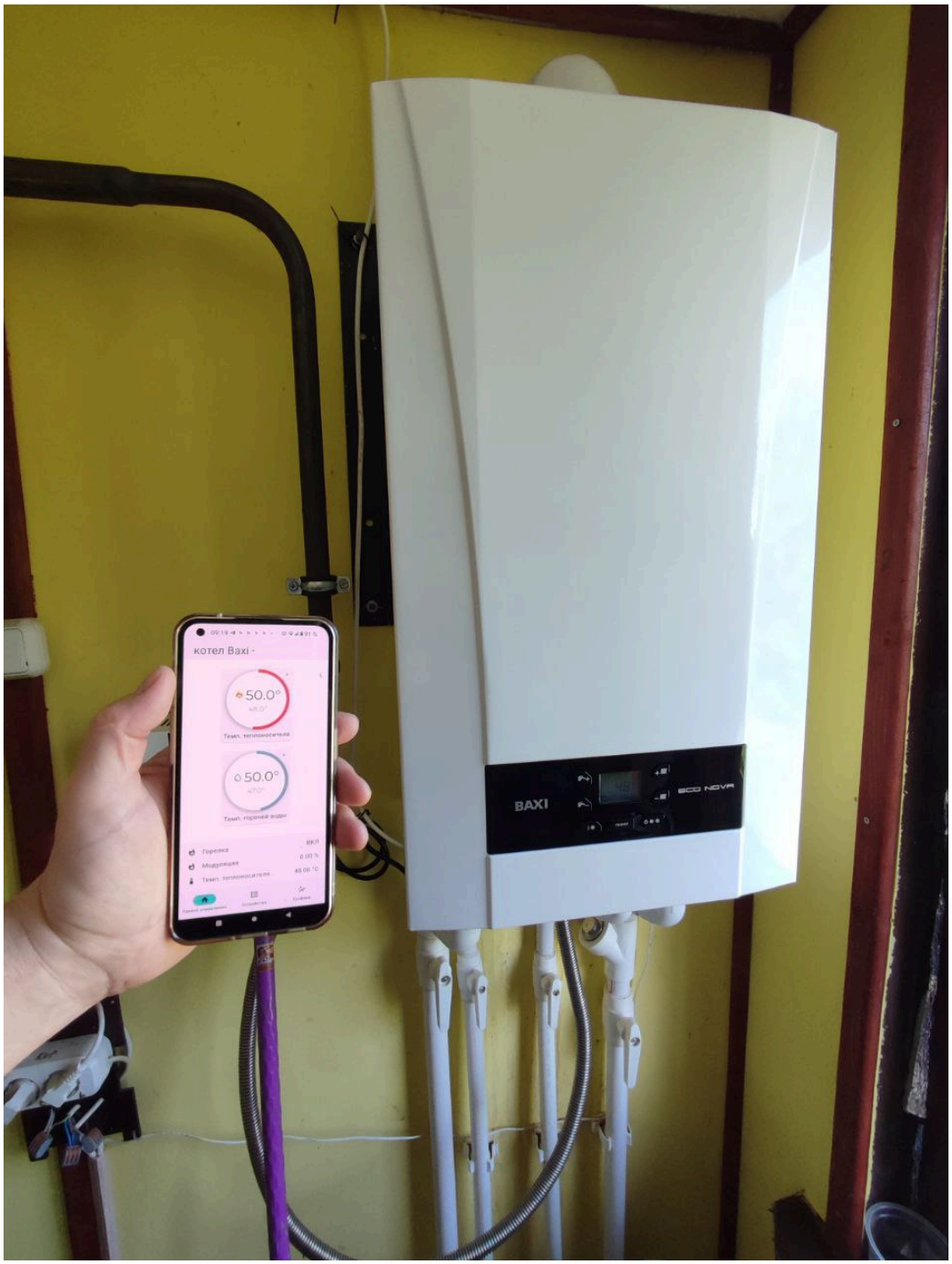

Рис. А.2: Использование тестовой сборки приложения с настоящим котлом