

ошибок, поступление текстовых сообщений в чате и изменение общего статуса звонка. Использование слушателей событий поможет более чётко контролировать пользовательский опыт и реагировать на различные ситуации в режиме реального времени.

Тестирование приложения. Особое внимание при разработке приложений с использованием ZEGOCLOUD следует уделить тщательному и всестороннему тестированию. Процесс тестирования должен охватывать не только базовый сценарий (успешный звонок между двумя устройствами), но и более сложные ситуации, такие как стресс-тестирование – групповые звонки с большим количеством участников, корректное переключение между фронтальной и тыловой камерами во время вызова, обработку перехода приложения в фоновый режим и его последующее восстановление, а также корректный приём входящих вызовов во время активной трансляции или другого сеанса связи. Крайне важно проверить, как SDK ZEGOCLOUD реагирует на нестабильный сетевой сигнал, кратковременные или длительные обрывы соединения и последующее автоматическое восстановление связи. Тестирование всех ключевых сценариев использования необходимо проводить на различных физических устройствах с разной производительностью, различными размерами экранов и разными версиями операционной системы Android, чтобы обеспечить стабильную и предсказуемую работу приложения на всём спектре поддерживаемых устройств.

Именно такой комплексный подход, включающий интеграцию всех возможностей SDK, тщательную кастомизацию пользовательского интерфейса в соответствии с фирменным стилем приложения, активное использование инструментов аналитики для мониторинга и оптимизации, а также всестороннее и поэтапное тестирование на различных устройствах и в различных условиях, позволит создать надёжные, удобные и востребованные пользователями продукты на базе ZEGOCLOUD, которые полностью соответствуют современным стандартам цифровых коммуникаций. Благодаря высокой стабильности работы, надёжным механизмам безопасности, качеству передаваемого звука и видео, а также широким возможностям гибкой настройки и масштабирования, платформа ZEGOCLOUD с лёгкостью может стать мощным и надёжным фундаментом для любого проекта.

Общение – неотъемлемая часть жизни современного человека. Однако не всегда и не у всех есть возможность общаться в режиме онлайн. Именно с этой целью и был создан видеочат – удобный и доступный инструмент для живого общения. Почему стоит выбрать именно видеочат? Такой формат давно зарекомендовал себя и получил множество положительных отзывов. Он не требует специальной подготовки, не ограничен правилами или строго заданными темами. Разговор строится свободно, по желанию самих участников. Чтобы начать беседу, достаточно самых простых и доступных средств – смартфона с выходом в интернет и установленного приложения. Всё это делает видеочат по-настоящему универсальным способом общения, подходящим для каждого.

Литература

1 Zegocloud Documentation. – URL: <https://www.zegocloud.com/docs> (дата обращения: 20.03.2025).

УДК 004.9

Р. Ю. Громыко

ДИНАМИЧЕСКАЯ ВИЗУАЛИЗАЦИЯ ДАННЫХ ИЗ НЕРЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ FIREBASE

В статье излагаются результаты динамического отображения метеоданных, поступающих в режиме реального времени с датчиков проекта «Метеостанция»,

реализованного с использованием микропроцессора ESP3. Описывается универсальное решение для сбора и визуализации данных (освещённости, давления, температуры. Приводятся примеры визуализации данных.

В ходе реализации проекта «Метеостанция» на базе микропроцессора ESP32 [1] была обнаружена проблема, связанная с визуализацией и хранением полученных данных с микроконтроллера ESP-32. Более простой способ её решения был в создании базы данных (БД) с конкретными полями и реализации приложения, которое будет работать с этой базой данных. Однако такой подход был бы максимально не гибким и подходил бы только для единственного случая, ведь если изменились бы входные данные, то всё перестало бы работать. В результате было найдено решение, более сложное, но крайне универсальное. Оно обеспечивает работу с любыми входными данными.

Универсальное решение для сбора данных. Во-первых, это способ хранения данных, для этого была выбрана облачная нереляционная БД Firebase, которая хранит данные в формате JSON. Благодаря формату JSON можно легко менять структуру сохраняемых данных, например, добавляя или убирая поля. В статье описывается работа с данными такого формата. На рисунке 1 представлены данные, поступающие в режиме реального времени с метеостанции.

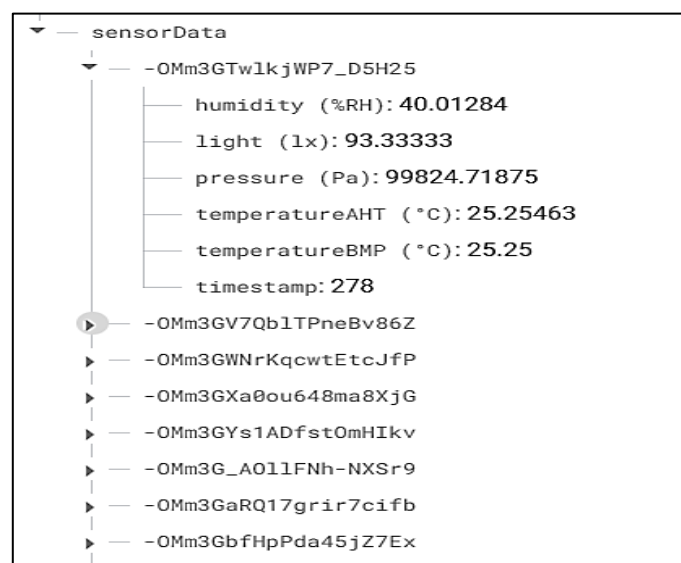


Рисунок 1 – Формат метеоданных

Как можно увидеть по рисунку 1, в БД хранится список “sensorData”, в который добавляются сами объекты. У объектов есть поля, для данной работы не имеет значение количество полей и их название, так как приложение динамически их обрабатывает. Важно знать только то, что название полей лучше делать понятными, так как из-за динамического подхода именно названия полей и будут использоваться в приложении для обозначения соответствующих данных.

Теперь же перейдем к самому приложению. Если приложение было запущено в первый раз и у него нет сохранённых данных, то первое, что выведется на экран, будет окно настроек (рисунок 2). Оно состоит из 2 вкладок – «Подключение» и «Поля для отображения». В окне подключения необходимо ввести название подключения (нужно только для более удобного выбора активной БД в дальнейшем) и поля для соединения с БД. Необходимым является только поле “Firebase URL”, а остальные нужны если у БД дополнительные параметры защиты и для подключения необходимо иметь ключ или токен.

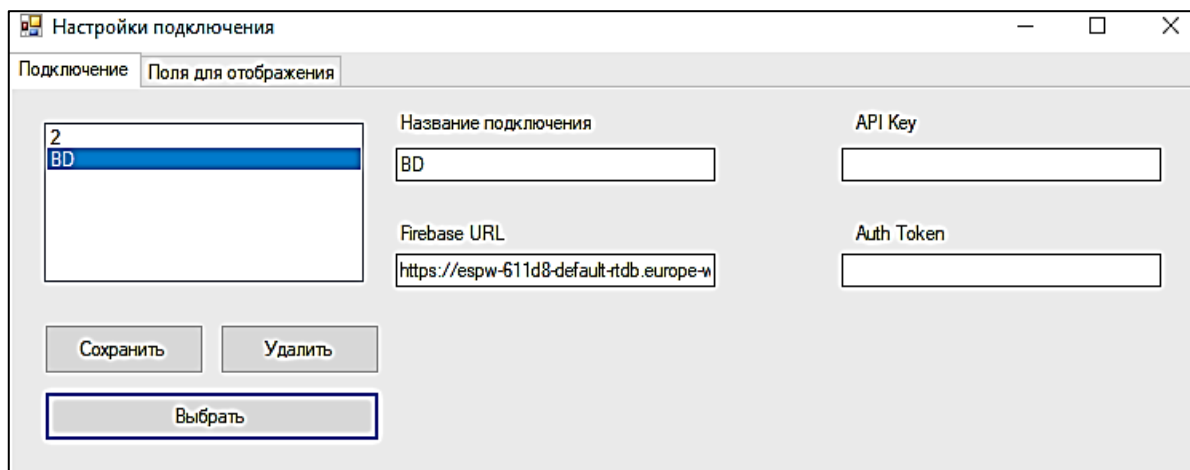


Рисунок 2 – Окно настроек

Информацию о БД можно сохранить в списке, она будет доступна и при дальнейших запусках приложения. После необходимо выбрать БД кнопкой «Выбрать».

Если данные подключения к БД были введены корректно, то приложение подключится к соответствующей БД и выведется информационное окно (рисунок 3), на котором можно увидеть сколько полей обнаружило приложение в БД.

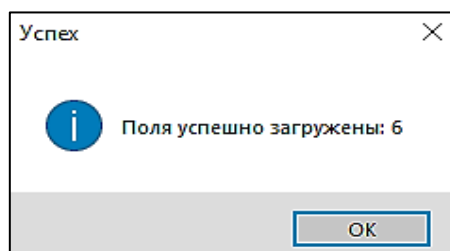


Рисунок 3 – Успешное подключение к БД

После этого можно переходить в вкладку «Поля для отображения». По рисунку 4 можно заметить, что здесь находятся два списка с полями из ранее показанной на рисунке 1 БД. Каждый элемент можно выбрать, кликнув по стоящему рядом полю выбора. Для оси Y можно выбрать от 1 до всех элементов, а для оси X только 1. После выбора нужных для отображения полей необходимо нажать кнопку «Применить и закрыть».

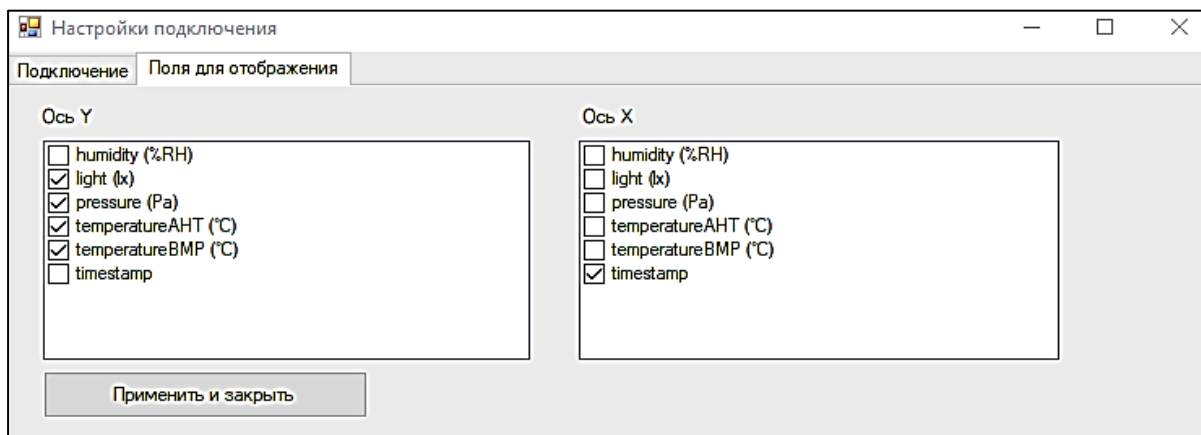


Рисунок 4 – Вкладка «Поля для отображения»

Как можно видеть на рисунке 5, для каждого элемента из списка с осью Y создавалась отдельная вкладка, в которой выбранный элемент будет в оси Y. На данных, полученных с датчика света, будет легче всего понять корректность отображения, так как, пару раз посветив фонариком на датчик и скрыв его от света, можно сразу понять, когда происходили эти действия по графику.

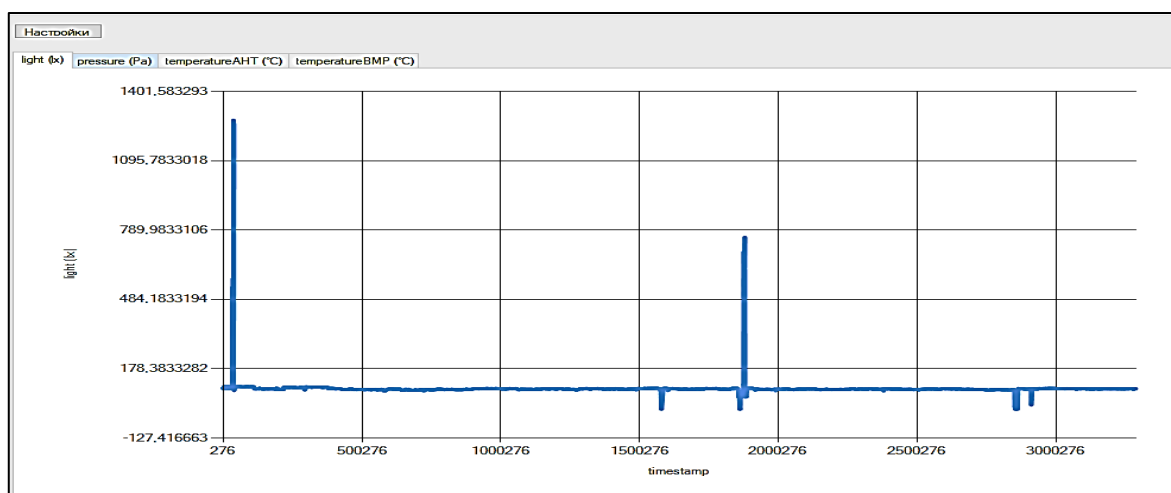


Рисунок 5 – Окно отображения (освещённость)

Также для более красивого отображение данных для каждого поля подбирается свой диапазон по Y на графике. На данных, полученных с датчика давления, можно увидеть колебание значений (рисунок 6). Между максимумом и минимумом по оси Y очень маленькая разница, если бы ось Y начиналась с 0, то тогда для пользователя была бы видна только линия, что помешало бы изучению данных.

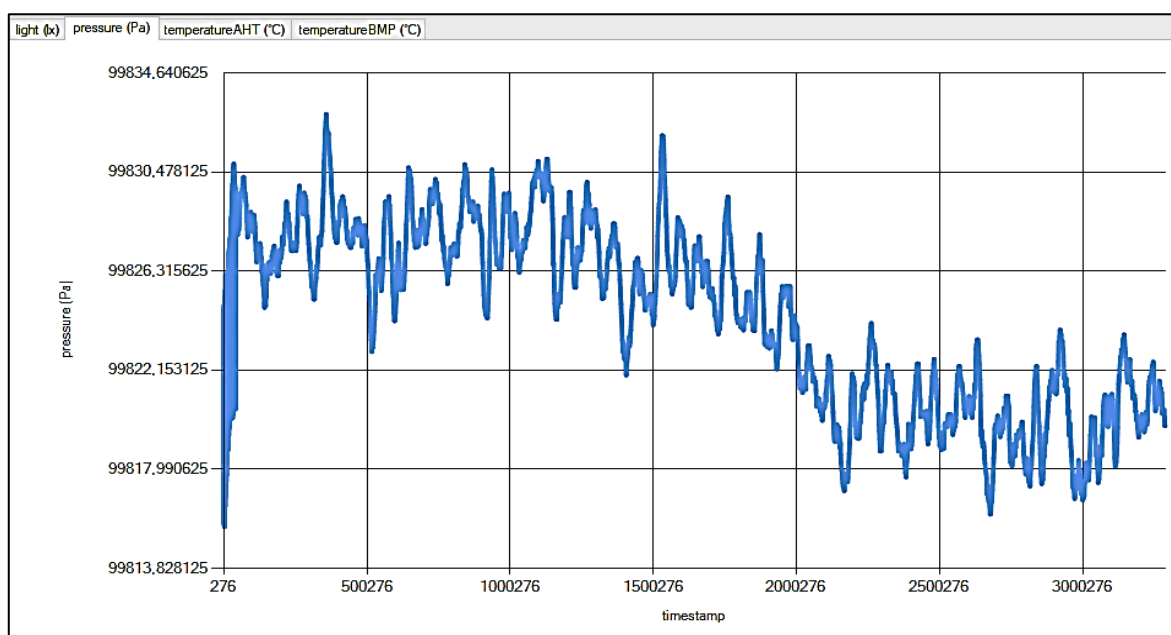


Рисунок 6 – Окно отображения(давление)

Также затронем ось X, с помощью неё выстраивается вторая ось, которая может быть как временной, для стандартного просмотра данных, так и любой другой, для просмотра зависимостей между данными. Например, к плате подключены 2 разных датчика

с температурой, на рисунке 7 ось X и ось Y одинаковы, и на графике получается ровная линия. На рисунке 8 в оси Y данные по температуре с другого датчика и по колебаниям графика мы можем заметить, что данные с датчиков немного отличаются друг от друга, хотя и максимально близки.

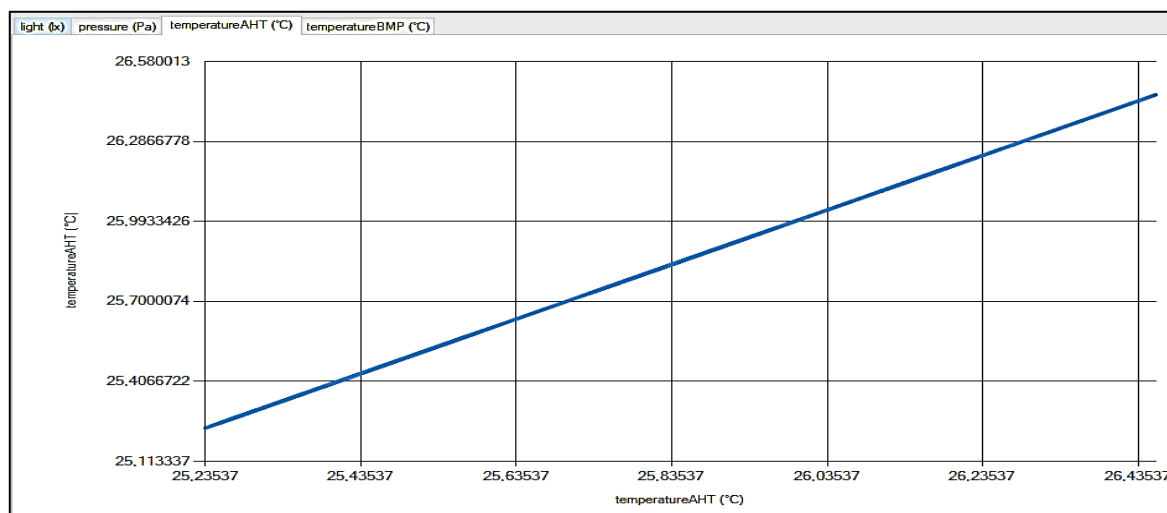


Рисунок 7 – Окно отображения (температура), оси X и Y одинаковые

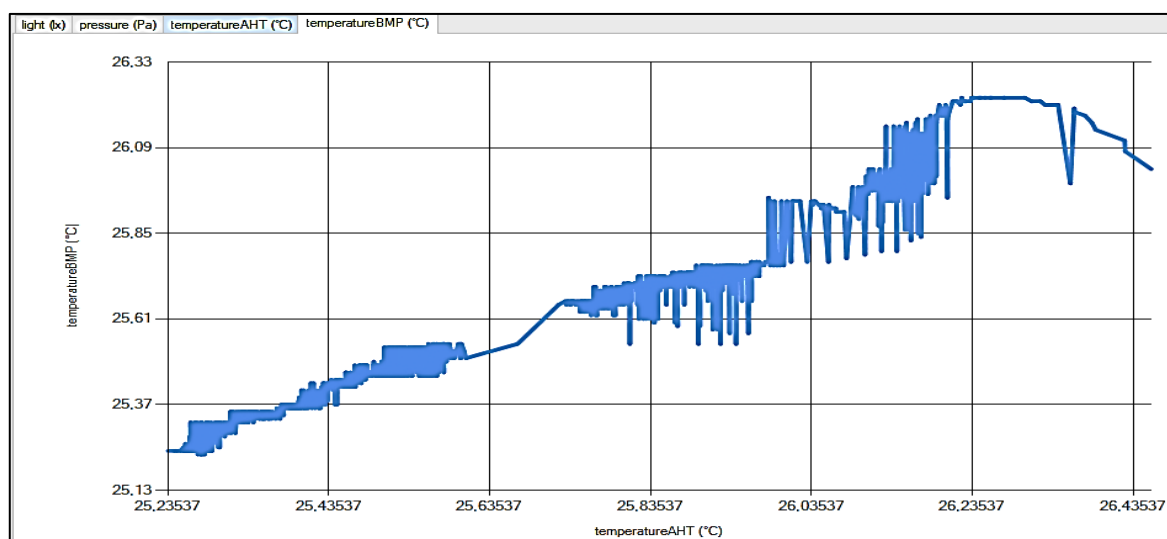


Рисунок 8 – Окно отображения (температура), оси X и Y различны

Подводя итоги, можно сказать, что благодаря динамическому подходу приложение можно использовать для визуализации различного типа данных, а у пользователя не будет нужды редактировать код приложение под свои нужды. Это делает приложение универсальным и удобным для пользования в различных сферах жизни.

Литература

1 Громыко, Р. Ю. Об одном способе сбора, хранения, передачи и систематизации больших данных / Р. Ю. Громыко // Творчество молодых ' 2024 : сб. науч. работ студентов, магистрантов и аспирантов : в 3 ч. / М-во образования Республики Беларусь, Гомельский гос. ун-т им. Ф. Скорины ; редкол.: Р. В. Бородич [и др.]. – Гомель : ГГУ им. Ф. Скорины, 2024. – Ч. 1. – С. 170–173.