

Получение и мониторинг метеоданных в реальном времени с использованием ESP-32

Е.И. СУКАЧ¹, А.П. КОНЧИЦ², Р.Ю. ГРОМЫКО¹

Предложен способ получения и передачи данных с микроконтроллера через Wi-Fi связь в облачную базу данных. Рассматривается получение данных с использованием различных датчиков. Разработаны и описаны аппаратная и программная части программно-аппаратного комплекса для сбора и мониторинга метеоданных.

Ключевые слова: микропроцессор ESP-32, получение метеоданных, Wi-Fi соединение, большие данные.

A method for receiving and transmitting data from a microcontroller via Wi-Fi communication to a cloud database is proposed. The data acquisition using various sensors is considered. The hardware and software parts of the hardware and software complex for collecting and monitoring meteorological data have been developed and described.

Keywords: ESP-32 microprocessor, receiving meteorological data, Wi-Fi connection, big data.

Введение. Использование микроконтроллеров ESP-32 открывает большие возможности для разработки умных систем сбора, систематизации и анализа больших данных в режиме реального времени [1]. Благодаря высокой производительности, встроенной поддержке Wi-Fi и Bluetooth, а также способности интегрировать множество сенсоров, ESP-32 стал основой для создания компактных и энергоэффективных программно-аппаратных комплексов в различных предметных областях. Платы, подобные ESP-32, демонстрируют высокую гибкость и простоту в работе благодаря поддержке двух ядер, встроенной беспроводной связи (Wi-Fi и Bluetooth), множеству доступных библиотек и примеров кода, а также разнообразию интерфейсов для подключения датчиков и периферийных устройств. Они подходят как для начинающих, так и для опытных разработчиков и могут использоваться в самых разнообразных проектах [2].

В статье рассматривается подход к автоматизации получения метеоданных с использованием ESP-32, который позволяет не только собирать данные о температуре, влажности, давлении и освещённости, но и передавать их в облачное хранилище для дальнейшей обработки. Реализация таких решений делает данные доступными в режиме реального времени, что особенно важно для аграрного сектора, метеорологических служб, а также умных городов и домов. Реализованный способ получения информации позволяет избежать ошибок при опросе датчиков, присоединённых к плате ESP-31, и осуществить контроль метеоданных в любой географической точке при наличии интернета. Он обладает универсальностью и может заменить метод получения больших объемов данных с удаленных серверов [3].

Программно-аппаратный комплекс «Метеостанция» на базе ESP-32. В рамках проекта «Метеостанция» плата ESP-32 использовалась как оборудование для сбора данных. Прежде всего плата была настроена физически, а именно, на микроконтроллер были установлены необходимые датчики и настроено соединение по Wi-Fi для возможности получения данных с платы и их хранения.

Датчик BMP280 от Bosch Sensortec использовался для измерения атмосферного давления и температуры. Он поддерживает интерфейсы I2C и SPI, что делает его удобным для интеграции с микроконтроллерами. BMP280 отличается высокой точностью и низким энергопотреблением, что делает его идеальным для метеостанций и навигационных систем.

Цифровой датчик температуры и влажности АНТ20 от AOSONG обеспечил высокую точность измерений температуры ($\pm 0,3^{\circ}\text{C}$) и влажности ($\pm 2\%$). Он также использует интерфейс I2C, имеет компактный размер и низкое энергопотребление, что делает его подходящим для мониторинга показателей окружающей среды.

Модуль GY-302 оснащён датчиком освещённости BH1750, который оценивает уровень освещённости в люксах. Он использует интерфейс I2C и подходит для проектов, связанных с управлением освещением и солнечными трекерами.

Модуль HW-125 представляет собой SD-картридер, который используется для записи данных на SD-карту. Он поддерживает интерфейс SPI и совместим с библиотеками FS.h и SD.h, что позволяет легко интегрировать его в проекты с ESP-32.

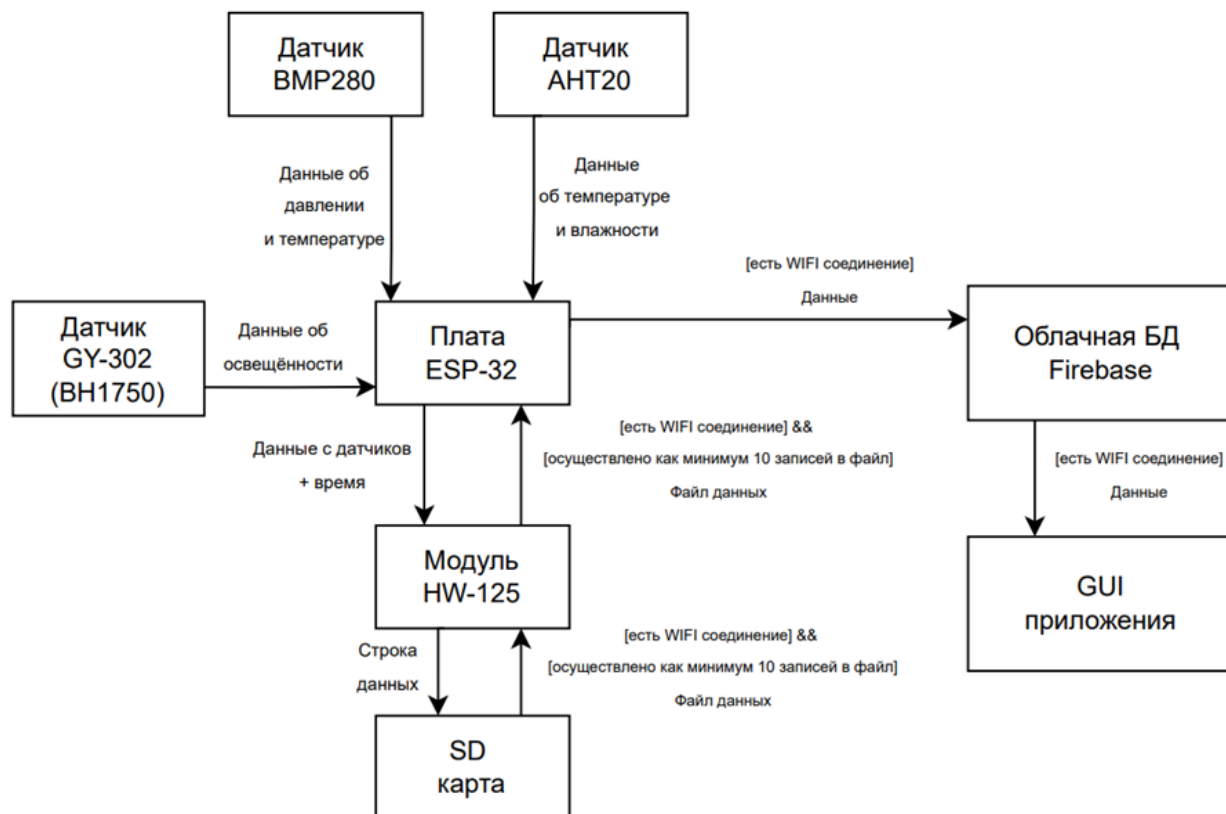


Рисунок 1—Структурная схема программно-аппаратного комплекса «Метеостанция»

Для хранения данных использовалась не реляционная база данных Firebase. Выбор облачной базы данных (БД) обеспечивает хранение данных в JSON формате, что гарантирует возможность масштабирования проекта при появлении новых датчиков различной конфигурации. Схема программно-аппаратного комплекса представлена на рисунке 1.

Для мониторинга метеоданных используется графический интерфейс рабочего приложения (GUI приложения), выбор которого определяется запросами пользователя. В зависимости от требований заказчика данные можно анализировать различными способами: от мобильных и десктопных приложений до Telegram-ботов и SMS сообщений на телефон. БД Firebase допускает использование различных языков программирования, что обеспечивает гибкость разработанной системы.

Схема работы скрипта для получения и передачи метеоданных в облачную БД. Для реализации функций автоматизации сбора метеоданных с помощью среды разработки Arduino был разработан скрипт. Он представляет собой интеграцию нескольких датчиков с микроконтроллером ESP-32, который обрабатывает данные, сохраняет их на SD-карту и отправляет в облачную базу данных Firebase. Схема работы скрипта реализуется следующим образом.

Инициализация компонентов и подключение к Wi-Fi. На этапе запуска скрипт считывает данные о Wi-Fi-сетях из файла на SD-карте (inet.txt). Если подключение к сети успешно, ESP-32 продолжает дальнейшую работу. В противном случае предпринимаются повторные попытки подключения.

Инициализация оборудования. Производится настройка подключенных датчиков, таких как BMP280 (температура и давление), АНТ20 (температура и влажность), BH1750 (освещённость), а также SD-карты для хранения данных. Дополнительно настраивается Firebase для отправки данных в облако.

Сбор и обработка данных. В главном цикле скрипт собирает показания с датчиков. Полученные значения температуры, давления, влажности и освещённости сохраняются в JSON-формате.

Сохранение данных на SD-карту. Собранные данные записываются в файл `sensor.json` на SD-карте. Это обеспечивает их сохранность даже при отсутствии подключения к интернету.

Отправка данных в Firebase. Если устройство подключено к интернету, данные отправляются в облако Firebase через API. После успешной отправки файл `sensor.json` удаляется с SD-карты, чтобы избежать дублирования.

Мониторинг состояния системы. Скрипт постоянно проверяет состояние Wi-Fi подключения и Firebase. В случае потери соединения предпринимаются попытки восстановления, что гарантирует стабильность работы системы.

Таким образом, этот скрипт реализует автоматизированное взаимодействие ESP-32 с облачной базой данных и локальным хранилищем, обеспечивая сбор и обработку данных для IoT-приложений.

Создание облачной базы данных. Для создания предметной базы данных была выбрана облачная база данных NoSQL Firebase. Соединение базы данных Firebase с микроконтроллером, таким как ESP-32, было реализовано через библиотеку Firebase для ESP 32 следующими последовательными этапами.

Настройка Firebase. В консоли Firebase был создан проект, в котором настроены параметры базы данных, такие как URL и правила безопасности. Получен API-ключ для доступа к базе данных.

Инициализация соединения Wi-Fi. Микроконтроллер был подключен к интернету через Wi-Fi с использованием SSID и пароля сети. В результате был получен доступ к облачной базе данных.

Настройка Firebase в коде. В код программы на ESP-32 была импортирована библиотека Firebase и инициализированы основные переменные: ключ API, URL базы данных и параметры авторизации. Эти настройки позволили связать устройство с проектом Firebase.

Отправка данных. Для взаимодействия с Firebase использовались функции библиотеки. Для этой цели могут быть использованы функции `Firebase.RTDB.setInt()` или `Firebase.RTDB.pushJSON()`. А использование функции `Firebase.RTDB.update()` обеспечивает перезапись существующего элемента в БД, что полезно в случае, когда не требуется хранение данных. В разработанном скрипте отправка всего файла происходит один раз в 10 итераций, поэтому используется `Firebase.RTDB.pushJSON()`.

Обработка ошибок. Код программы включает проверки подключения к Wi-Fi и доступности Firebase, чтобы обрабатывать возможные сбои соединения. Например, при потере доступа устройство может автоматически повторить попытку подключения.

Этот процесс позволяет микроконтроллеру ESP-32 интегрироваться с Firebase для хранения, обновления и извлечения данных в реальном времени, что является основой работы программно-аппаратного комплекса «Метеостанция» и множества других IoT приложений.

Визуализация и анализ данных. После правильной настройки датчиков и базы данных новые записи автоматически передаются с микроконтроллера в базу данных. Так как БД не реляционная, то данные хранятся в древовидной структуре формата JSON. При получении информации соответствующая запись добавляется в конец структуры. Firebase подсвечивает новые добавленные данные. Любую запись можно развернуть и посмотреть хранящуюся в ней информацию (рисунок 2).

Для организации мониторинга полученные данные визуализируются. Это можно сделать многими способами. В данном проекте это происходит через консольное приложение, реализованное на C#. Приложение подключается к облачному БД и для каждого типа данных строит свой график.

На рисунке 3 представлен график изменения данных освещенности во времени. С целью проведения тестирования работы датчика он несколько раз закрывался и один раз освещался. По данным на графике можно отследить, когда такие моменты происходили.



Рисунок 2 – Просмотр записи в базе данных

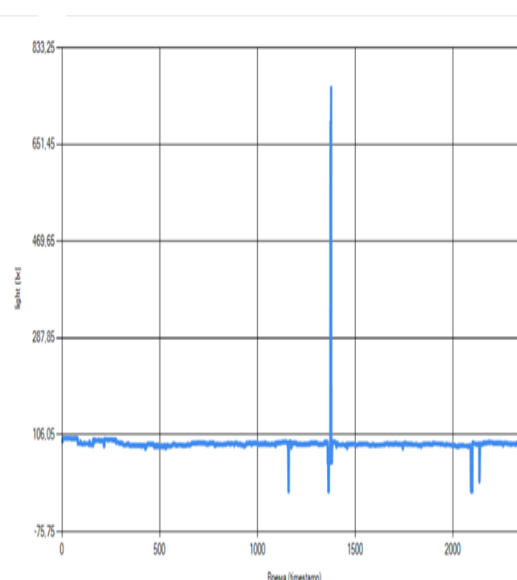


Рисунок 3 – Мониторинг данных освещенности

Каждый график хранится на отдельной вкладке на форме приложения (рисунок 4).



Рисунок 4 – Добавленные вкладки

Как нетрудно заметить, название для вкладки и оси Y приложение получает из данных БД. Ось X указывает время, но, так как плата ESP-32 не оснащена датчиком реального времени, берётся время с начала работы скрипта, а консольное приложение вместо миллисекунд использует порядковый номер. Также учтена возможность перезапуска платы и сброса времени, поэтому данные не будут перемешиваться.

Консольное приложение каждые пять секунд обновляет данные с сервера, но при этом фокус на вкладке приложение не меняется, что позволяет удобно изучить данные. При этом также стоит отметить, что консольное приложение динамическое и если ему предоставить ссылку на БД с другим типом данных, то он их успешно визуализирует.

Закключение. Благодаря разработанному программному обеспечению пользователь может в реальном времени получать и сохранять данные с датчиков в облачной базе данных, что полезно для мониторинга и принятия управленческих решений. Можно использовать как одну плату для просмотра состояния погоды в определённом месте, так и группу плат, которые будут покрывать значительную территорию.

Литература

1. Кэмерон, Н. Электронные проекты на основе ESP8266 и ESP32 / Н. Кэмерон. – М., 2022. – 456 с.
2. Громыко, Р. Ю. Об одном способе сбора, хранения, передачи и систематизации больших данных / Р. Ю. Громыко // Творчество Молодых – 2024 : сб. науч. работ студ. и асп. УО «ГГУ им. Ф. Скорины» / М-во обр-я РБ, Гомельский гос. ун-т ГГУ им. Ф. Скорины. – Гомель, , 2024. – С. 170–173.
3. Сукач, Е. И. Об опыте получения, хранения и анализа Big Data с удаленных сайтов / Е. И. Сукач, В. Ю. Бурикин, А. А. Кончиц // Известия Гомельского гос. ун-та им. Ф. Скорины. – 2022. – № 3 (132). – С. 93–96.

¹Гомельский государственный университет имени Франциска Скорины

²Институт леса НАН Беларуси