

**Р. Ю. Громыко**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **Е. И. Сукач**, канд. техн. наук, доцент

## **АВТОМАТИЗАЦИЯ ПОЛУЧЕНИЯ И ИСПОЛЬЗОВАНИЯ МЕТЕОДАНЫХ В РЕАЛЬНОМ ВРЕМЕНИ С ИСПОЛЬЗОВАНИЕМ ESP-32**

**Введение.** Автоматизация сбора и анализа метеоданных играет ключевую роль в современном мире, позволяя оперативно реагировать на изменения погодных условий, оптимизировать сельскохозяйственные процессы и улучшать экологический мониторинг. Использование микроконтроллеров, таких как ESP-32, открывает новые горизонты для разработки умных систем сбора метеоданных в режиме реального времени. Благодаря высокой производительности, встроенной поддержке Wi-Fi и Bluetooth, а также способности интегрировать множество сенсоров, ESP-32 стал основой для создания компактных и энергоэффективных метеостанций.

В данной работе 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 облачные хранилища 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.

**Описание датчиков.** Для получения данных, а также их хранения, на плате были установлены специальные датчики. На рисунке 1 представлен микроконтроллер ESP32 с подключёнными к нему датчиками.

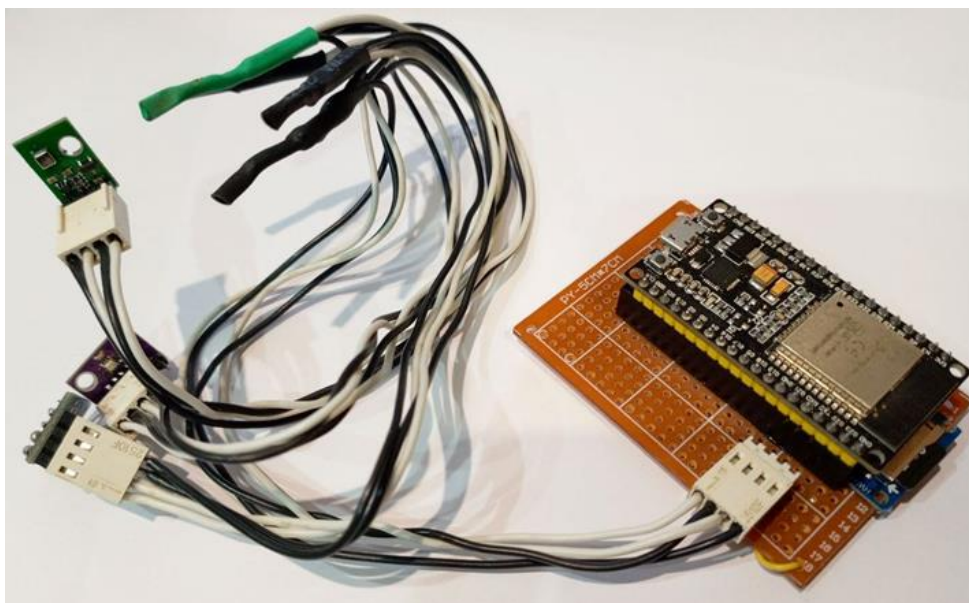


Рисунок 1 – Микроконтроллер ESP-32 с подключёнными датчиками

**BMP280.** Этот датчик от Bosch Sensortec измеряет атмосферное давление и температуру. 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.

**АНТ20.** Цифровой датчик температуры и влажности от AOSONG. Он обеспечивает высокую точность измерений ( $\pm 0.3^{\circ}\text{C}$  для температуры и  $\pm 2\%$  для влажности) и использует интерфейс I2C. Компактный размер и низкое энергопотребление делают его подходящим для мониторинга окружающей среды.

GY-302 (BH1750). Этот модуль оснащён датчиком освещённости BH1750, который измеряет уровень освещённости в люксах. Он использует интерфейс I2C и подходит для проектов, связанных с управлением освещением и солнечными трекерами.

HW-125. Этот модуль 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.

**Основные этапы работы средств автоматизации сбора данных.** 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. К основным этапам работы скрипта можно отнести следующие.

*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 Firebse. Соединение базы данных Firebase с микроконтроллером, таким как ESP-32, реализуется через библиотеку Firebase для ESP. Этот процесс включает несколько следующих этапов.

*Настройка Firebase.* В консоли Firebase создаётся проект, в котором настраиваются параметры базы данных, такие как URL и правила безопасности. Также необходимо получить API-ключ для доступа к базе данных.

*Инициализация соединения Wi-Fi.* Микроконтроллер подключается к интернету через Wi-Fi, используя SSID и пароль сети, чтобы иметь доступ к облачной базе данных.

*Настройка Firebase в коде.* В коде программы на ESP-32 импортируется библиотека Firebase и инициализируются основные переменные: ключ API, URL базы данных и параметры авторизации. Эти настройки позволяют связать устройство с проектом Firebase.

*Отправка данных:* Для взаимодействия с Firebase используются функции библиотеки. Например, данные можно отправлять в базу с помощью `Firebase.RTDB.setInt()` или `Firebase.RTDB.pushJSON()`. Также можно по-другому настроить отправку данных,

например, `Firebase.RTDB.update()` позволяет не добавлять новый элемент в БД, а перезаписывать существующий, что полезно, если нам не нужно хранить данные. В случае разработанного скрипта 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 для множества IoT приложений, от метеостанций до систем умного дома.

После правильной настройки датчика и базы данных, новые записи автоматически передаются с микроконтроллера в базу данных, если датчик успешно подключился к сети (рисунок 2).

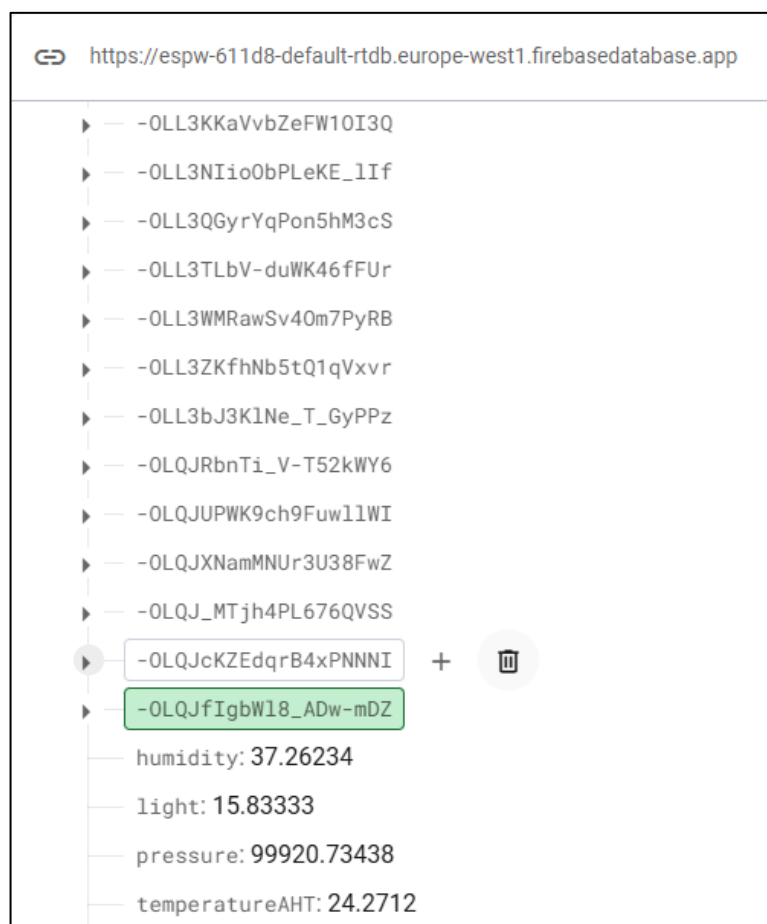


Рисунок 2 – Добавление новой записи в базу данных

Благодаря разработанному программному обеспечению пользователь может в реальном времени получать и сохранять данные с датчиков в облачной базе данных, что полезно для анализа и интерпретации данных и принятия управленческих решений.