

РОЗРОБКА ГОЛОСОВОГО АСИСТЕНТА НА БАЗІ МІКРОКОНТРОЛЛЕРА ESP32

Михайлов Ю.О.

Науковий керівник – асистент Радченко С.С.

Харківський національний університет радіоелектроніки, каф. ЕОМ,
м. Харків, Україна

e-mail: yurii.mykhailov@nure.ua.

This development is devoted to the creation of a personal voice assistant based on the ESP32 microcontroller. Various technologies were used to solve the tasks, including processing and reproduction of analog signals, saving files on flash memory, interacting with external services via the HTTP protocol. The development has significant potential for improvement and expansion of functionality, including taking advantage of the dual-core microcontroller for multi-tasking, optimization of processing and data transmission to external services, smart home control, voice wake-up capability, etc.

Голосовий помічник – це сервіс на основі штучного інтелекту, який може розпізнавати мову людини та здатність виконувати певні дії у відповідь на голосові команди. З розвитком нейромереж, можливості та якості взаємодії людини з ШІ (штучний інтелект) стрімко збільшуються. До найпопулярніших голосових помічників у 2024 році можна віднести Siri (Apple), ChatGPT (OpenAI), GoogleAssistant (Google), тощо. Принципи роботи голосових асистентів засновані на технологіях розпізнавання та обробки мовлення, які виконуються за допомогою інтелектуальних алгоритмів та нейронних мереж. Вони включають у себе етапи аналізу звукового сигналу, перетворення його в текстовий формат, розпізнавання команди або запиту користувача, та виконання відповідної дії.

Для створення особистого голосового помічника було використано мікроконтролер ESP32 через наявність технології Wi-Fi та двоядерного процесору з частотою до 240 МГц, що робить його оптимальним вибором для реалізації DIY (Do It Yourself) проєктів. Наявність функцій енергозбереження також було однією з причин вибору мікроконтролера, з урахуванням можливих сценаріїв використання пристрою. Крім того, ESP32 має широкий спектр доступних бібліотек, змістовну документацію та підтримує різноманітні периферійні пристрої.

Процес обробки вхідного аналогового сигналу є ключовим етапом функціонування голосового асистента. Для цього використовується аналого-цифровий перетворювач (АЦП), який перетворює аналоговий сигнал у цифровий формат для подальшої його обробки. У мікроконтролері ESP32 вбудований АЦП, але його частота дискретизації обмежена 4095 Гц. У той

час як мікрофон INMP441 пропонує значно вищу частоту дискретизації – 16000 Гц. Це спрощує процес програмування, оскільки немає потреби в обробці аналогового сигналу самим мікроконтролером; замість цього, перетворений в цифровий формат вищої якості сигнал надходить через інтерфейс I2S [1]. Такий підхід дозволяє забезпечити більш точну та ефективну обробку аудіоданих для голосового асистента.

Протокол I2S (Inter-IC Sound) – стандарт інтерфейсу електричної послідовної шини, що використовується для з'єднання цифрових аудіопристроїв. Складається з трьох основних ліній передачі даних: тактового сигналу бітової синхронізації (CLK), тактового сигналу кадрової (за словами) синхронізації (WS) та лінії даних (SD), що передає фактичні аудіодані. Потребує програмної ініціалізації.

Для збереження записаних даних було використано файлову систему SPIFFS, яка дозволяє оперувати виділеною для неї Flash-пам'яттю мікроконтролера.

Для перетворення збережених аудіоданих в текстовий формат, необхідно передати ці дані на зовнішній сервіс, такий як Google Cloud Text-to-Speech API [2]. Цей API надає можливість конвертувати аудіофайли у мовлення, використовуючи алгоритми машинного навчання та штучного інтелекту для обробки мови. Передача даних реалізується через HTTP запит у форматі JSON. Якщо запит сформовано вірно і дані контенту записано у відповідному форматі, здатному для оброблення зовнішнім сервісом (наприклад, зашифровані у форматі base64), API повертає розшифрований текст у вигляді JSON з відсотковою точністю розпізнавання [3].

На даному етапі отриманий текст можна використовувати для керування периферійними пристроями мікроконтролера, або для подальшої взаємодії з іншими сервісами. Наприклад, розшифрований текст можна надіслати на API OpenAI, зокрема ChatGPT, де штучний інтелект згенерує відповідь відповідно до запиту. Отриманий текст можна вивести на екран або використати для подальшої обробки, наприклад, створення аудіофайлу з текстового формату та його відтворення на мікроконтролері, відповідно завершуючи основний цикл роботи голосового асистента. Перетворення аудіоданих в текстовий формат можна розглядати як аналогічний процес запису аудіо, але у зворотному напрямку.

З урахуванням можливості використання голосового асистента без прив'язки до стаціонарного джерела живлення, тобто від акумуляторної батареї, було впроваджено режим очікування з метою збільшення тривалості автономної роботи пристрою. Мікроконтролер ESP32 має різноманітні режими енергозбереження, що дозволяють значно зменшити споживану енергію. Режим Deep Sleep, який був використаний у розробці, припиняє роботу більшості його функцій, залишаючи активними лише необхідні для підтримки зовнішніх переривань або таймерів компоненти

[4]. В цьому режимі вимикаються ядра процесора, частини пам'яті та інші периферійні пристрої, що дозволяє зменшити енергоспоживання від кількох мікроампер до кількох міліампер, в залежності від конфігурації системи та підключених пристроїв. Вхід та вихід з цього режиму задається програмно: вхід – функцією, вихід – торканням піну (touch pin).

Використання голосових асистентів стає невід'ємною частиною сучасного життя, і ця розробка передбачає створення особистого голосового помічника на базі мікроконтролера ESP32. Для вирішення поставлених задач було використано різноманітні технології, включаючи обробку та відтворення аналогових сигналів, збереження файлів на флеш-пам'яті, взаємодію з зовнішніми сервісами через HTTP протокол, а також другорядні, типу режиму енергозбереження, для підвищення ефективності роботи пристрою. Основні переваги цього голосового помічника полягають у його доступності за ціною та простоті виготовлення. Крім того, він має значний потенціал для вдосконалення та розширення функціональності, включаючи використання переваг двоядерного мікроконтролера для багатозадачності, оптимізацію обробки та передачі даних на зовнішні сервіси, керування розумним будинком, можливість голосового пробудження тощо. Таким чином, така розробка надає широкі можливості для подальшого розвитку голосового асистента згідно з потребами та вимогами користувачів.

Список використаних джерел:

1. Sound with ESP32 - I2S Protocol // DroneBot Workshop. URL: <https://dronebotworkshop.com/esp32-i2s/> (дата звернення: 15.02.2024).
2. Transcribe short audio files // Cloud Speech-to-Text Documentation. Google Cloud. URL: https://cloud.google.com/speech-to-text/docs/sync-recognize#performing_synchronous_speech_recognition_on_a_local_file (дата звернення: 19.02.2024).
3. ESP32 HTTP GET and HTTP POST with Arduino IDE // Random Nerd Tutorials. URL: <https://randomnerdtutorials.com/esp32-http-get-post-arduino/> (дата звернення: 20.02.2024).
4. ESP32 deep sleep with arduino IDE and wake up sources // random nerd tutorials. URL: <https://randomnerdtutorials.com/esp32-deep-sleep-arduino-ide-wake-up-sources/> (дата звернення: 20.02.2024).