

УДК 004.934

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ АЛГОРИТМІВ КЕРУВАННЯ ТА ЯКОСТІ ВІДТВОРЕННЯ АУДІО ДЛЯ MP3-ПЛЕЄРА НА БАЗІ ARDUINO

Дейно В.В.

e-mail: volodymyr.deino@nure.ua

Науковий керівник - д.т.н, проф. Хаханова Г.В.

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

This paper presents the development and experimental analysis of a portable MP3 player based on the Arduino platform. The study addresses the challenge of implementing high-quality audio playback while maintaining responsive control on an 8-bit microcontroller with limited resources. A comparative analysis of two control strategies is conducted: a simple loop-based approach versus an interrupt-driven algorithm for button handling. The hardware architecture utilizes an Arduino Nano, a DFPlayer Mini module for hardware MP3 decoding, and a set of control buttons. Experimental results demonstrate that the interrupt-driven approach significantly improves system responsiveness, reducing button latency from 150 ms to under 20 ms. The DFPlayer Mini module effectively offloads decoding tasks, achieving high audio quality with minimal CPU overhead. The proposed system provides a cost-effective solution for customized audio applications in educational and embedded systems contexts.

Вступ. Розробка портативних аудіоплеєрів на базі мікроконтролерів є класичним завданням, що дозволяє дослідити взаємодію апаратного та програмного забезпечення в системах реального часу. Сучасні мікроконтролери, такі як ATmega328P, мають обмежені обчислювальні ресурси для програмного декодування MP3, тому актуальним є використання спеціалізованих модулів з апаратним декодуванням. Метою роботи є проектування MP3-плеєра на базі Arduino та дослідження ефективності різних алгоритмів керування для забезпечення стабільного та чуйного інтерфейсу користувача.

Методика дослідження та апаратна реалізація. Для проведення експериментів було зібрано прототип портативного MP3-плеєра. Основні компоненти системи:

1. Мікроконтролер Arduino Nano (ATmega328P, 16 MHz).
2. Модуль відтворення: DFPlayer Mini (апаратне декодування MP3, WAV, WMA).
3. Носій даних: MicroSD карта (форматована у FAT16/FAT32) з аудіофайлами.
4. Органи керування: кнопки для керування відтворенням (Play/Pause, Next, Previous) та регулювання гучності.

5. Пристрій виведення: мініатюрний динамік (4 Ом, 3 Вт) або навушники через аудіороз'єм модуля.

Апаратне з'єднання модуля DFPlayer Mini з Arduino Nano виконано через програмний послідовний інтерфейс з обов'язковим використанням резистора для узгодження рівнів логічних сигналів. Кнопки керування підключені до цифрових входів з підтяжкою до живлення.

Програмна реалізація та алгоритми керування. Ключовим аспектом дослідження є порівняння двох підходів до обробки натискань кнопок.

Циклічний алгоритм. Стан кнопок опитується в кожній ітерації основного циклу програми. Для уникнення брязкоту контактів використовується функція затримки, що блокує виконання програми на час антибрязкоту.

Алгоритм на перериваннях. Кнопки підключені до пінів з апаратними перериваннями. Обробка натискань відбувається негайно у спеціальних функціях обробки переривань, що дозволяє уникнути блокувань основного циклу.

Для керування модулем DFPlayer Mini використовується стандартна бібліотека, яка надає простий інтерфейс для відтворення треків за індексом та регулювання гучності.

Результати експериментів. Для оцінки ефективності алгоритмів керування було проведено вимірювання часу реакції системи на натискання кнопок та завантаження центрального процесора. Результати порівняння наведено у таблиці 1.

Таблиця 1 – Порівняльний аналіз алгоритмів керування

Показник	Циклічний алгоритм	Алгоритм на перериваннях
Середній час реакції на натискання	150 мс	< 20 мс
Завантаження CPU (під час відтворення)	< 5%	< 5%
Чутливість до тривалих натискань	Задовільна	Висока
Складність реалізації	Низька	Середня

Аналіз якості відтворення показав, що використання модуля з апаратним декодуванням забезпечує стабільний вихідний сигнал без переривань та спотворень, незалежно від обраного алгоритму керування. Максимальна гучність забезпечує достатній рівень звукового тиску для невеликого приміщення. Важливою вимогою є правильне форматування

SD-карти (FAT16/FAT32) та дотримання правил іменування файлів, оскільки порядок копіювання файлів визначає їх індексацію модулем.

Висновки. У роботі реалізовано MP3-плеєр на базі Arduino Nano та модуля DFPlayer Mini. Експериментально доведено, що використання апаратного декодування дозволяє досягти високої якості звуку на 8-бітному мікроконтролері. Порівняльний аналіз показав, що алгоритм керування на основі переривань забезпечує значно кращу (в 7.5 разів) чутливість інтерфейсу користувача порівняно з простим циклічним опитуванням. Отримані результати можуть бути використані при розробці компактних аудіопристроїв з мінімальним енергоспоживанням для автономних систем відтворення звуку. Розроблена система може бути використана як основа для створення спеціалізованих аудіопристроїв, аудіогідів, інтерактивних експонатів та навчальних стендів. Подальші дослідження можуть бути спрямовані на додавання дисплея для відображення інформації про трек та реалізацію режимів відтворення (повтор, випадковий порядок).

Список використаних джерел:

1. Monk S. Programming Arduino: Getting Started with Sketches. New York : McGraw-Hill Education, 2016. 192 p.
2. McEwin L. New and old: Playing MP3s with an Arduino on an analog phone. *University of Waterloo Computer Museum Blog*. 2025.
3. DFPlayer Mini MP3 Player Module Documentation. *LCSC / EasyEDA Wiki*. 2024.
4. Hahanov V., Chumachenko S., Litvinova E., Hahanov I., Ponomarova V., Khakhanova H., Kulak G. Faults-as-address simulation. *IAES International Journal of Robotics and Automation*. 2024. Vol. 13, No. 4. P. 452–468. DOI: <http://doi.org/10.11591/ijra.v13i4.pp452-468>.
5. Seng J. Building a Music Player: Teaching Operating Systems Concepts via an Arduino-Powered Music Player. *2018 IEEE Frontiers in Education Conference (FIE)*. San Jose, CA, USA, 2018. P. 1–5.
6. DFPlayer - A Mini MP3 Player (DFR0299) [Електронний ресурс] // Farnell. 2025.