

ОПТИМІЗАЦІЯ КОРИСТУВАЦЬКОЇ ВЗАЄМОДІЇ ЧЕРЕЗ ДИСТАНЦІЙНЕ КЕРУВАННЯ СМАРТФОНОМ ЗА ДОПОМОГОЮ АУДИОГАРНІТУРИ

Вінницький національний технічний університет

Анотація

Запропоновано підхід до використання аудіо гарнітури як пристрою дистанційного керування смартфоном для покращення взаємодії. Досліджено механізми перехоплення ширококомовних повідомлень ОС Android від медіа-кнопок. Наведено приклад програмної реалізації обробника подій, що дозволяє виконувати кастомні команди без візуального контакту з екраном.

Ключові слова: дистанційне керування, смартфон, Android, BroadcastReceiver, MediaSession, KeyEvent, Java.

Abstract

The paper proposes an approach to using an audio headset as a remote control device for a smartphone to enhance user interaction. The mechanisms for intercepting Android OS broadcast messages from media buttons are investigated. An example of the software implementation of an event handler is provided, allowing for the execution of custom commands without visual contact with the screen.

Keywords: remote control, smartphone, Android, BroadcastReceiver, MediaSession, KeyEvent, Java.

Вступ

Сучасні мобільні інтерфейси переважно орієнтовані на активне сенсорне введення, що створює значні експлуатаційні обмеження в умовах динамічного середовища, активного руху або за несприятливих погодних умов. Існуюча парадигма взаємодії вимагає від користувача постійного візуального контролю та задіяння дрібної моторики рук, що не завжди є можливим або безпечним.

Використання аудіо гарнітури як віддаленого пульта дозволяє розширити функціональність смартфона, перетворюючи її на допоміжний інтелектуальний контролер для виконання фонових завдань. Такий підхід відповідає концепції «Eyes-free interaction» та дозволяє знизити когнітивне навантаження на користувача, забезпечуючи безперервний доступ до функцій пристрою без необхідності витягнення його з кишені чи сумки. Актуальність дослідження зумовлена потребою у створенні більш адаптивних інтерфейсів, які інтегруються у повсякденну активність користувача, не перериваючи його основну діяльність.

Актуальність

Сьогодні ми все частіше взаємодіємо з гаджетами «на ходу», коли дивитися на екран просто незручно або навіть небезпечно — чи то за кермом, під час занять спортом, чи в складних погодних умовах. У 2026 році тренд на Ambient Computing (невидимі технології) робить голосове та кнопочке керування пріоритетом.

Проте сенсорні екрани стають перешкодою, коли руки зайняті або на вулиці дощ. Використання звичайної гарнітури як інтелектуального пульта дозволяє керувати смартфоном наосліп (Eyes-free interaction). На технічному рівні це реалізується через низькорівневий доступ до систем Android (MediaSession), що гарантує миттєву реакцію додатку, навіть якщо телефон лежить у кишені

заблокованим. Поєднання точного алгоритму розпізнавання натискань із вібровідгуком (Haptic Feedback) створює надійний інструмент, який працює стабільно там, де сенсорне керування пасує [4].

Метод взаємодії	Технологічний стек	Рівень автономності (1–10)	Ключові елементи автоматизації	Статус (2026)
MediaSession Control	Android API / Java	8.5–9.0	Високий пріоритет фоновому процесу, алгоритм Δt	Стандарт для аудіо-додатків
Voice AI(Assistant)	ML / NLP	7.0–8.0	Обробка природної мови, залежність від шуму	Потребує постійної мережі
Haptic Gestures	IMU Sensors / Bluetooth	8.0–8.5	Акселерометри гарнітури, розпізнавання жестів	Пілотні версії (Pro-сегмент)
Agentic AI Audio	On-device LLM / NPU	9.0–9.5	Автономне прийняття рішень, предиктивний вибір дій	Впровадження в Android 15+

Таблиця Порівняння методів дистанційної взаємодії з мобільними пристроями (2026 р.)

Основними задачі

Автоматизації дистанційного керування є насамперед створення стійкого фоновому сервісу на базі MediaSessionCompat, який зберігає пріоритет у пам'яті Android навіть при високому навантаженні. Ключовим етапом виступає розробка алгоритму аналізу часових інтервалів Δt , що дозволяє перетворити поодинокі імпульси фізичної кнопки на складні команди через розпізнавання одинарних, подвійних чи тривалих натискань. Для реалізації концепції «Eyes-free interaction» у систему інтегрується невізуальний зворотний зв'язок у вигляді Haptic Feedback та звукових індикаторів. Важливою складовою є енергооптимізація через подієво-орієнтовану архітектуру BroadcastReceiver, яка залучає ресурси процесора лише в момент активності. Окрім цього, стабільність системи забезпечується методами програмної фільтрації шумів та налаштуванням індивідуальних порогів чутливості (thresholds) під темп користувача. Завершує архітектуру принцип локальної обробки даних, що гарантує повну конфіденційність та захист від перехоплення команд через аудіо-порт[8-10].

Можливий шлях вирішення задачі

Технічна реалізація дистанційного керування базується на перехопленні подій KeyEvent.KEYCODE_MEDIA_*. Основною проблемою є забезпечення пріоритету обробки сигналів додатком який використовується для управління системою користувача. Для цього використовується реєстрація BroadcastReceiver [1-5].

Нижче наведено фрагмент коду на мові Java, який демонструє базовий принцип фільтрації подій натискання кнопки гарнітури:

```
public class RemoteControlReceiver extends BroadcastReceiver {
```

```
    @Override
```

```

public void onReceive(Context context, Intent intent) {
    if (Intent.ACTION_MEDIA_BUTTON.equals(intent.getAction())) {
        KeyEvent event = intent.getParcelableExtra(Intent.EXTRA_KEY_EVENT);
        if (event != null && event.getAction() == KeyEvent.ACTION_DOWN) {
            int keyCode = event.getKeyCode();
            if (keyCode == KeyEvent.KEYCODE_HEADSETHOOK) {
                performCustomAction(context);
            }
        }
    }
}

```

Для стабільної роботи у фоновому режимі додаток повинен ініціалізувати `MediaSession`. Це дозволяє системі Android ідентифікувати програму як активний медіа-центр, навіть якщо екран пристрою заблокований або процесор перейшов у режим енергозбереження [6].

Суть запропонованого методу полягає в алгоритмічній інтерпретації часових патернів натискань. Оскільки фізичний інтерфейс більшості гарнітур обмежений лише однією кнопкою (Hook), розширення функціоналу досягається шляхом аналізу інтервалів між імпульсами. Це дозволяє розділити стандартні операції відтворення від кастомних команд управління. Математично вибір команди C залежить від часового інтервалу Δt між двома послідовними подіями:

$$C = \begin{cases} Action_1, \text{ якщо } \Delta t > 500 \text{ ms} \\ Action_2, \text{ якщо } \Delta t \leq 500 \text{ ms} \end{cases}$$

Така модель дозволяє перетворити обмежений апаратний ресурс на розгалужену систему дистанційного керування, забезпечуючи взаємодію за принципом «*Eyes-free*» (без візуального контролю) [6-9].



Рисунок 1 - Результат інтерфейсу програми після віддаленого управління при активній сесії

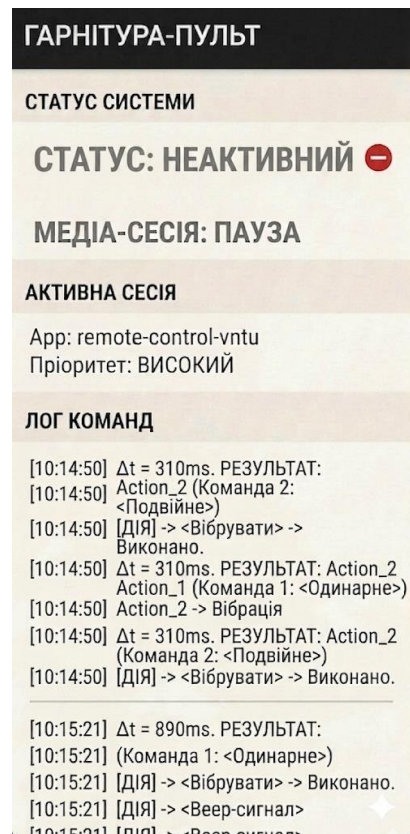


Рисунок 2 - Результат інтерфейсу програми після віддаленого управління при неактивній сесії

Висновки

Розроблена модель та наведена програмна реалізація дозволяють перенести частину рутинних операцій на аудіо інтерфейс. Це підвищує безпеку користувача та забезпечує безперервність експлуатації пристрою в складних зовнішніх умовах. Результати дослідження підтверджують ефективність використання часових інтервалів для кодування команд, що дозволяє обійти обмеження мінімальної кількості фізичних кнопок на гарнітурі. Запропонований підхід не лише покращує ергономіку взаємодії, а й створює підґрунтя для розробки спеціалізованих асистентів для людей з порушеннями зору.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Сьєрра К. Head First Java. Вивчення Java / К. Сьєрра, Б. Бейтс. — К.: Фабула, 2017. — 720 с.
2. Харді Б. Програмування під Android. 2-ге вид. / Б. Харді, Б. Філліпс, К. Стюарт, К. Марсікано. — СПб.: Питер, 2016. — 640 с.
3. ДСТУ ISO/IEC 25062:2014. Розроблення програмного забезпечення. Загальний формат галузевих звітів з придатності до використання (Usability). — [Чинний від 2015-07-01]. — К.: Держспоживстандарт України, 2015. — 32 с.
4. Шнейдерман Б. Проектування інтерфейсу користувача: Стратегії ефективної взаємодії людини та комп'ютера / Б. Шнейдерман, К. Плайзонт. — 6-те вид. — Pearson, 2016. — 624 с.

5. Блок Д. Kotlin. Програмування для професіоналів / Д. Блок, Е. Джетт. — СПб.: Питер, 2020. — 480 с.
6. Мартин Р. Чиста архітектура. Мистецтво розроблення програмного забезпечення / Р. Мартин. — К.: Фабула, 2019. — 368 с.
7. Official Android Documentation. MediaSession overview [Електронний ресурс]. — Режим доступу: <https://developer.android.com/guide/topics/media-apps/mediasession-overview> — Загол. з екрана.
8. ISO 9241-210:2019. Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems. — International Organization for Standardization, 2019. — 30 p.
9. Кумар А. Android Background Processing: Threads, Services, and WorkManager / А. Кумар. — Packt Publishing, 2021. — 350 p.
10. Нільсен Я. Дизайн інтерфейсів. 10 евристик юзабіліті / Я. Нільсен. — [Електронний ресурс]. — Режим доступу: <https://www.nngroup.com/articles/ten-usability-heuristics/>

Химич Олександр Володимирович — студент групи 2ПКТ-24б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: sasahimic39@gmail.com.

Науковий керівник: **Юхимчук Марія Сергіївна** — д-р техн. наук, професор кафедри комп'ютерних систем управління, Вінницький національний технічний університет, e-mail: umcmasha@gmail.com.

Лесько Владислав Олександрович — канд. техн. наук, доцент кафедри електричних станцій і систем, Вінницький національний технічний університет, e-mail: leskovlad@ukr.net.

Oleksandr Volodymyrovych Khymych – student of group 2PKT-24b, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: sasahimic39@gmail.com.

Supervisor: **Mariia Serhiivna Yukhymchuk** – Doctor of Technical Sciences, Professor of the Department of Computer Control Systems, Vinnytsia National Technical University, e-mail: umcmasha@gmail.com.

Vladyslav Oleksandrovykh Lesko – Ph.D. in Technical Sciences, Associate Professor of the Department of Electric Stations and Systems, Vinnytsia National Technical University, e-mail: leskovlad@ukr.net.