iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** VI **Month of publication:** June 2023

DOI: https://doi.org/10.22214/ijraset.2023.53442

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VI Jun 2023- Available at www.ijraset.com

Connection with the Multiple Bluetooth Speaker with the Single Device

Mrs. G. Jeyalakshmy[1], R. Geethapriya[2], G. Chithrakala[3], E. Swathi[4], A. Divya[5]
[1]*M. Tech, Assistant Professor/CSE*
[2, 3, 4, 5]*Final Year/CSE, Achariya College of Engineering Technology, Puducherry*

Abstract: ***This paper presents the design and development of an Audio Connector application that enables users to route audio from a single input source to multiple output devices simultaneously. It is able to communicate with a wide range of devices without an interface. Users can concurrently stream audio to two wireless speakers or headphones using the Dual Audio functionality. Additionally, users can independently adjust the media output volume of each audio device. The times when you and your buddies would constantly quarrel about the volume of the audio are long gone. Theoretically, all Bluetooth 5.0 and higher-capable devices can use the dual audio capability. However, other elements like the hardware capabilities of your device and its operating system, for example, the Android version, also affect your ability to use this feature. The Bluetooth signal's maximum range is still 10 metres, despite advancing technical requirements for wireless communication.***
Keywords: Audio routing, Audio Connector application, NAudio library, Multiple output devices, Real-time audio processing, Cross-platform support, Broadcasting, Video conferencing, Gaming, Performance evaluation, Latency and buffering, Usability, Hardware compatibility, Open-source collaboration, Accessibility, Privacy and security.

I. INTRODUCTION

In today's multimedia applications, audio routing plays a crucial role in facilitating a seamless audio experience across various platforms, including broadcasting, video conferencing, and gaming. The demand for routing audio from a single source to multiple output devices is frequently encountered to cater to diverse use cases. However, many operating systems offer limited native support for this functionality, posing challenges for users attempting to route audio to multiple devices simultaneously. This paper describes the development and execution of an Audio Connector application that facilitates the distribution of audio signals from a singular source to multiple output devices. The primary objective of this system is to offer a user-friendly graphical user interface (GUI) that simplifies the configuration and management of the audio routing process. By employing this application, users can easily control and customize the audio distribution according to their preferences. The Audio Connector application employs the Model-View-Controller (MVC) architectural pattern to ensure a clear separation between the user interface and the underlying audio processing logic. By adhering to MVC principles, the application achieves a modular and maintainable design. To handle various audio processing tasks, such as audio capture, buffering, and output, the application relies on the NAudio library. NAudio provides a robust set of tools and functionalities specifically tailored for audio-related operations. The application utilizes the NAudio library's WasapiLoopbackCapture class to capture audio from the default input device. The captured audio is buffered using a BufferedWaveProvider object. This buffering mechanism allows for the storage of audio data temporarily, enabling smooth playback across different devices. User initiates audio routing by clicking "Start" button on GUI. App iterates through selected output devices, initializes WaveOutEvent object for each, and starts playback. The application has error handling mechanisms to display helpful error messages when issues occur, like device initialization failures or audio processing errors. This paper introduces an Audio Connector application enabling users to route audio from one source to multiple output devices simultaneously.

II. EXISTING SYSTEM

Smartphones today have both BLE and BT in them. BLE can be used for peer to peer communication. In the current problem statement, there is no way by which the connecting device (Smartphone) can communicate to already connected device (Tablet/Speaker), regarding it’s need to establish a connection with BTA device. We used BLE advertisement and scanning features to enable this communication between connecting device and already connected device. During our literature survey, we haven’t found competition that uses BLE to solve BT connection issue. The sequence diagram in Fig. 3 depicts the proposed solution. Whenever user tries to connect smartphone to the speaker, the smartphone sends a Disconnection Request (DR) message using BLE advertisement to nearby devices, conveying its interest to connect to the speaker, simultaneously it starts BLE scan for receiving

©IJRASET: All Rights are Reserved

Disconnect Successful message (DS). The nearby devices such as Tablet in our case, which is already connected to the BTA device would be performing an energy efficient BLE scan in the background to receive (DR). The Tablet on receiving the DR message from the smartphone would compare the BT MAC address from the packets payload. If the MAC address matches with the Speaker's MAC address, it can result in a popup on the Tablet whether to allow disconnection with the Speaker. Once the user agrees (setting can be saved for further attempts) the connection with the Speaker is disconnected. Further the Tablet, sends response message called as Disconnect Successful (DS) message using BLE advertise conveying that it has successfully disconnected from speaker. Smartphone on receiving DS message via BLE scan, can then attempt a fresh connection to speaker. This will result in successful D2D connection shift from Tablet to Smartphone with the speaker.

III. ARCHITECTURE

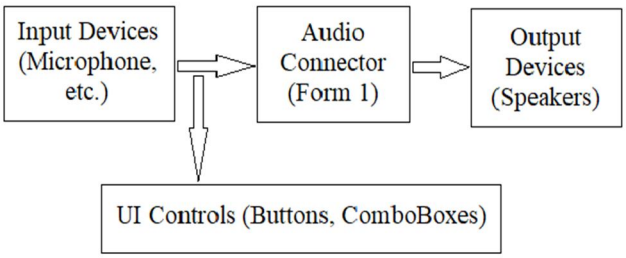


Fig.no. Architecture of the proposed system

1) *Input Devices (Microphone):* A microphone is an input device that captures audio or sound and converts it into an electrical signal. It is commonly used for recording voices, musical instruments, or any other sound source.
2) *Audio Connector (Form 1):* The audio connector refers to the physical interface used to connect audio devices together. It is used to connect headphones, speakers, or external microphones.
3) *Output Devices (Speaker):* A speaker is an output device that converts electrical signals into audible sound. It reproduces audio or sound from various sources such as computers, televisions, or music players.
4) *UI Controls (Buttons):* The Buttons are interactive graphical elements used in user interfaces. They can be clicked or pressed to trigger an action or change the state of a program or device. Buttons are commonly used for commands like "OK," "Cancel," or "Submit."
5) *UI Controls (Comboboxes):* Comboboxes, also known as dropdown menus, provide a list of options from which the user can select one. They typically consist of a text box and a down arrow. When the arrow is clicked or pressed, a menu appears with a list of selectable options. The user can choose one option from the list, which is then displayed in the text box.
6) UI controls like buttons and comboboxes are essential elements of graphical user interfaces (GUIs). They enable users to interact with software applications or devices by providing a visual representation of actions or choices. Buttons and comboboxes improve user experience by making it easier to navigate and control software or devices.

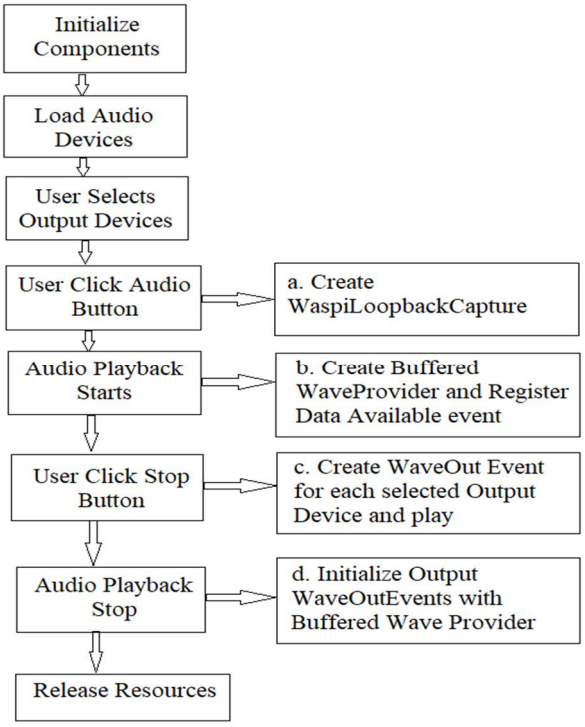
IV. PROPOSED SYSTEM

These tests utilized a variety of input sources, as well as multiple output devices with differing configurations. The results demonstrated that the application was able to maintain low latency and perform effective audio buffering. To address this challenge could be to optimize the audio processing algorithms and explore the use of hardware acceleration techniques. By doing so, the application may be able to achieve better performance and maintain real-time processing. During the development of the Audio Connector application, a significant challenge was ensuring compatibility with a diverse range of audio devices. To overcome this challenge, the application leverages the NAudio library, which provides support for a broad range of audio formats and devices. Despite this effort, it is still possible that there could be certain devices or configurations that are not fully compatible with the application. Currently, the Audio Connector application can support up to three output devices, but there may be a need to expand its capability to support a greater number of output devices in the future. Achieving this goal may require further optimizations to ensure that the application can maintain low latency and efficient resource usage while handling multiple devices simultaneously.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The aim of this paper was to introduce the Audio Connector application, which allows users to route audio from a single input source to multiple output devices. The system was built using the NAudio library and features a user-friendly GUI for ease of use and control. The application has been tested and shown to provide low latency and efficient buffering, making it ideal for a variety of multimedia scenarios, including broadcasting, video conferencing, and gaming. Further work will include the implementation of advanced features, support for additional platforms, and expanding the application's functionality to accommodate a broader range of use cases.

V. METHODOLOGY



1) *Initialize Components:* In the Form1() constructor, the user interface components are initialized, and the output device combo boxes are assigned to an array.
2) *Load Audio Devices:* The LoadDevices() method retrieves the available output devices and populates the output device combo boxes.
3) *User selects Output Devices:* The user selects the desired output devices using the combo boxes.
4) *User clicks Start Button:* When the user clicks the start button, the startButton_Click() method is executed.

a) A WasapiLoopbackCapture object is created to capture the system's audio.
b) A BufferedWaveProvider is created and the DataAvailable event is registered, which adds the captured audio samples to the buffer.
c) A WaveOutEvent is created for each selected output device.
d) The output WaveOutEvents are initialized with the BufferedWaveProvider as the source of audio samples.

5) Audio Playback starts: The audio playback starts on the selected output devices.
6) User clicks Stop Button: When the user clicks the stop button, the stopButton_Click() method is executed.
7) Audio Playback Stops: The audio playback stops, and the resources are released.
8) Release resources: All resources are released, such as the WasapiLoopbackCapture, the BufferedWaveProvider, and the WaveOutEvent objects.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VI Jun 2023- Available at www.ijraset.com

VI. CONCLUSION

This paper details the development and design of an Audio Connector application that facilitates routing audio from a single input source to multiple output devices simultaneously. The application utilizes the NAudio library and incorporates a user-friendly GUI for simplified configuration and control. Potential applications encompass diverse multimedia scenarios such as broadcasting, video conferencing, and gaming. Future endeavors may involve incorporating advanced features, extending platform compatibility, and enhancing functionality to cater to additional use cases.

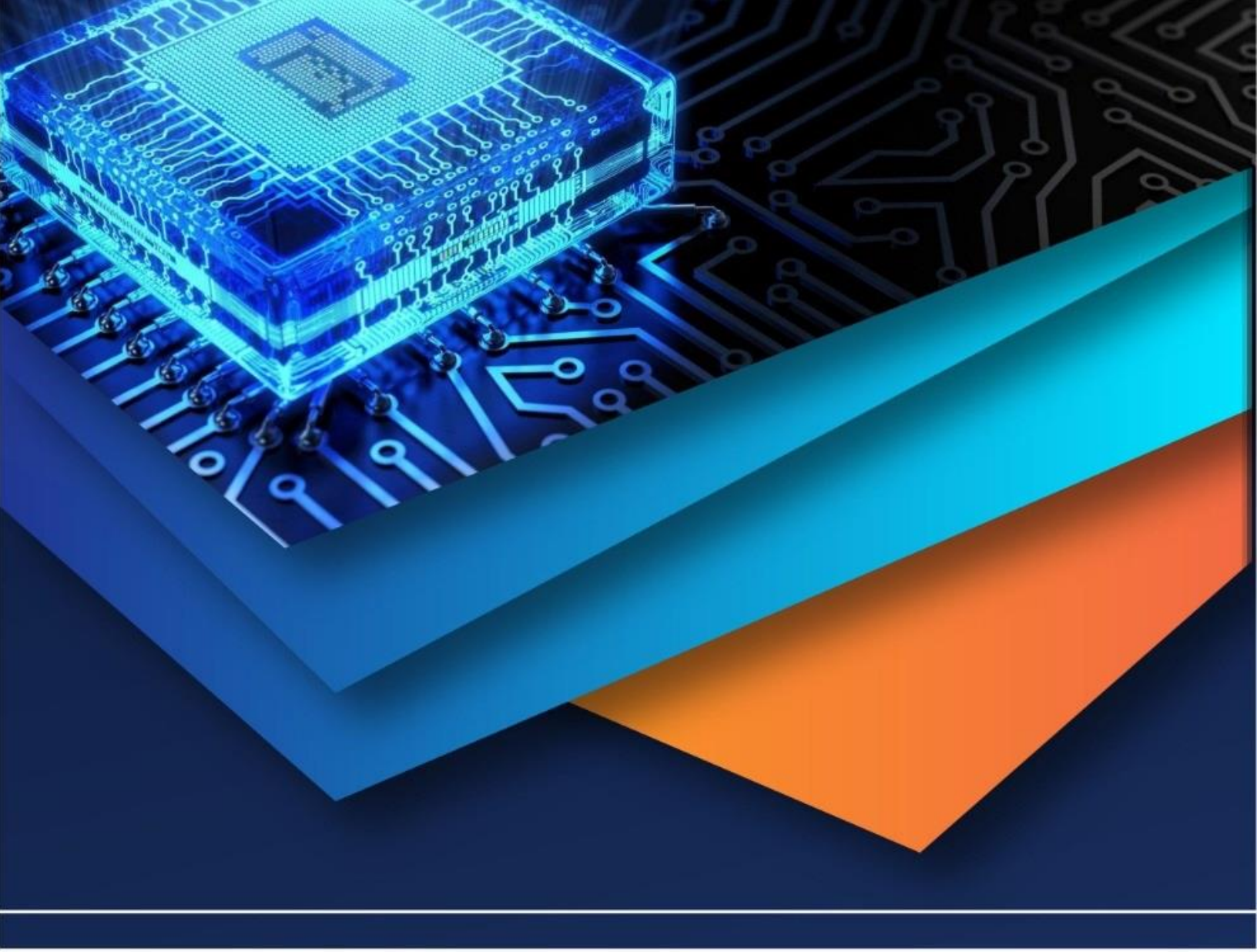
REFERENCES

[1] NAudio. (2021). NAudio: An open source .NET audio and MIDI library. [Online].
[2] M. Heath, "C# Programming Yellow Book: Learn to program in C# from first principles," 2021.
[3] P. McDermott-Wells, "What is Bluetooth?", IEEE Potentials, pp 33-35, 2005.
[4] Bluetooth SIG, “Bluetooth core specification version 5.0”, vol III, Part C, Generic Access Profile, pp 1967-2064, Dec. 2016.
[5] Bo-Ren Chen,Shin-Ming Cheng,Jia-Jhun Lin, "EnergyEfficient BLE Device Discovery for Internet of Things", Fifth International Symposium on Computing and Networking (CANDAR) - 2017, pp1-5, 2018.
[6] Bluetooth SIG, “Bluetooth core specification version 5.0”, vol III, Part C, Generic Access Profile, pp 2086-2087, Dec. 2016.
[7] Gaoyang Shan, Sun-young Im, Byeong-hee Roh, "Optimal AdvInterval for BLE scanning in different number of BLE devices environment", IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp 1-2, 2016.
[8] Wha Sook Jeon; Made Harta Dwijaksara; Dong Geun Jeong, “Performance Analysis of Neighbor Discovery Process in Bluetooth Low-Energy Networks”, IEEE Transactions on Vehicular Technology (Volume: 66 , Issue: 2 , Feb. 2017), pp 2-4, 2017.
[9] C. Bisdikian, “A overview of the Bluetooth wireless technology,” IEEE Communications Magazine, Vol.39, no. 12, pp. 86-94, Dec. 2001.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VI Jun 2023- Available at www.ijraset.com

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)