Department of Electronic and Telecommunication Engineering
University of Moratuwa
Sri Lanka

EN3251 - Internet of Things

Project Proposal
IoT Based Music Player

Group 9

Name	Index Number
M. P. Wickramarathne	210703V
A.A.W.L.R.Amarasinghe	210031H
A.D.T. Dabare	210089P

This report is submitted as a partial fulfillment of module EN3251
2024.10.24

Contents

1 Introduction and Problem Statement

With the growing popularity of streaming services and smart devices, there is an increasing demand for portable solutions that allow users to enjoy their music conveniently, with features such as remote control and playlist management. Traditional music players do not typically offer internet streaming services or provide flexibility for remote management, which limits their convenience in today's connected world.

This project aims to design and develop a **portable IoT-enabled music player** that allows users to stream music from online services, control playback functions remotely via a web-based dashboard, and manage their playlists easily. By combining hardware components (ESP32 microcontroller and speakers) and software (Node-RED dashboard for remote control), this project will deliver a flexible and modern music player solution that enhances the user experience.

1.1 Problem

Traditional music players are often restricted by local storage, lack streaming capabilities, and do not offer advanced features for remote control. This project addresses these limitations by integrating IoT technologies into a music player system that allows for both manual control (via physical buttons) and remote control (via a web dashboard). This gives users the flexibility to control their music experience more effectively, whether at home or on the go.

1.2 Relevance

As more users adopt smart devices and streaming services, IoT integration in devices like music players enables more efficient interaction and enhanced functionalities. These include the ability to manage playlists, control volume, skip tracks, and stream from multiple services. This project aligns with the growing trend of creating smart, connected devices that improve user interaction and provide better control over personal media consumption.

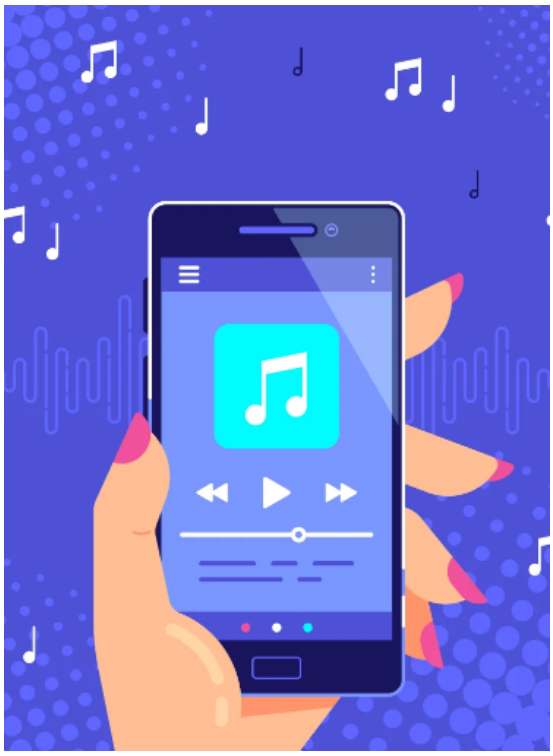

Figure 1: music

2 Objectives

- **Design and Development**: To design and develop a **portable IoT-enabled music player** that is capable of streaming music from online services.
- **Node-RED Dashboard**: To implement a dashboard using **Node-RED** for controlling various functions such as play/pause, volume control, playlist management, and track skipping.
- **Dual Control Modes**: To enable both manual control via physical buttons on the music player and remote control via the web-based dashboard, providing users with flexible control options.
- **Seamless Streaming**: To ensure that the system provides seamless streaming and control for an enhanced user experience, minimizing interruptions and providing responsive feedback.

3 System Overview

3.1 High-Level Architecture:

The IoT speaker system consists of hardware components, including an ESP32 microcontroller and attached speakers, as well as software elements like the Node-RED dashboard for user control. The system streams Internet radio and provides a dashboard for managing speaker functions.

3.2 End-Devices:

The primary end-device is the ESP32 microcontroller, which will handle both the speaker operations and the streaming of Internet radio. The ESP32 will be connected to speakers for audio output and equipped with physical buttons for basic control like play/pause, volume adjustment, and channel changes.

Figure 2: ESP 32

We chose the ESP32 microcontroller due to its versatility and robust features, which make it an ideal choice for IoT applications. The ESP32 comes with integrated Wi-Fi and Bluetooth connectivity, allowing it to seamlessly stream music from online platforms and communicate with external devices like smartphones or dashboards.

Its dual-core processor provides sufficient processing power for handling audio streaming while managing user inputs, making it efficient for real-time performance. Additionally, the ESP32 is energy-efficient, which is important for a portable music player, and it supports various communication protocols (e.g., MQTT, HTTP) needed for remote control via a web-based dashboard like Node-RED. This makes the ESP32 an excellent option for combining hardware control with IoT features, ensuring a smooth user experience.

3.3 Management Interface (Node-RED Dashboard):

A dashboard will be created using Node-RED, providing a graphical interface where users can remotely control the speaker's volume, manage playlists, skip/pause tracks, and switch between different radio stations. The dashboard will be accessible via a web browser, enabling control from any device with an internet connection.

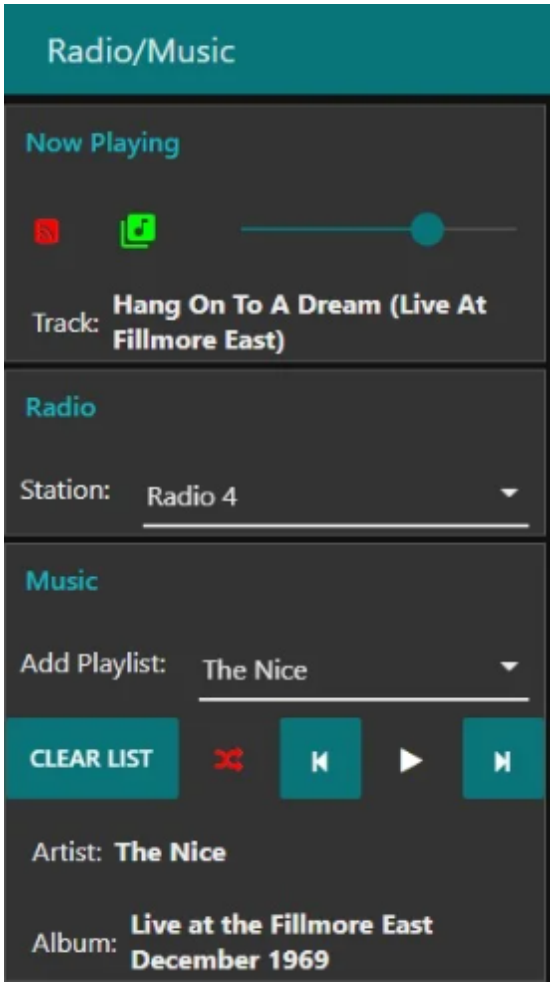

Figure 3: Sample Node RED Dashboard

We chose Node-RED for the dashboard because of its user-friendly, visual programming interface, which allows for rapid development and easy customization. Node-RED is particularly well-suited for IoT applications, as it enables seamless integration with a wide variety of protocols, including MQTT and HTTP, which are crucial for communication between the ESP32 microcontroller and the dashboard.

Additionally, Node-RED's flow-based development model simplifies the creation of complex interactions between hardware and software, such as controlling the music player functions (e.g., play/pause, volume control, track skipping) from a web-based interface. Its flexibility allows for real-time control and status updates, making it ideal for remotely managing IoT devices like a music player. The dashboard created with Node-RED is accessible from any web browser, allowing users to control

the music player from multiple devices (smartphones, tablets, desktops) with ease. Moreover, the rich library of pre-built nodes in Node-RED accelerates the development process, enabling quick integration of new features and APIs for future expansion.

4 User Interaction

The IoT-based music player provides users with two convenient methods for interacting with the system, offering both flexibility and ease of use depending on the situation or user preference:

- **Manual Controls**:
  Physical buttons on the music player will allow users to perform essential functions directly on the device. These controls will include buttons for:
  - **Adjusting volume**: Users can increase or decrease the volume using dedicated buttons, providing real-time feedback for precise control.
  - **Skipping tracks**: Buttons will allow users to move forward or backward through the playlist or music queue, giving them immediate access to their favorite songs.
  - **Switching channels or playlists**: Users can easily switch between different streaming services, playlists, or stations (if Internet radio is integrated) with a simple button press. These manual controls are especially useful in situations where accessing the web-based dashboard is not convenient, such as when outdoors or on the move.
- **Web-Based Dashboard**:
  The **Node-RED** dashboard provides a comprehensive and intuitive user interface for controlling all aspects of the music player remotely. Accessible from any device with an internet connection (smartphone, tablet, desktop), the dashboard will allow users to:
  - **Adjust the volume** remotely with a slider or button controls on the interface, providing fine-tuned control over sound levels.
  - **Manage tracks and playlists**: Users can view and control the current playlist, pause or resume tracks, skip songs, and rearrange tracks without needing to be physically near the device.
  - **Switch between different music services or channels**: The dashboard enables users to change between different music sources, whether it's a specific playlist, a streaming service, or Internet radio channels.
  - **Alerts and notifications**: The dashboard can be enhanced with real-time notifications or alerts, such as when the volume reaches a certain limit (e.g., maximum volume), the device is disconnected from the network, or the playlist ends. These notifications provide users with feedback on the system's status, ensuring they are always aware of important system events, even when they are controlling the device remotely.

Figure 4: listen music

5 Block Diagram

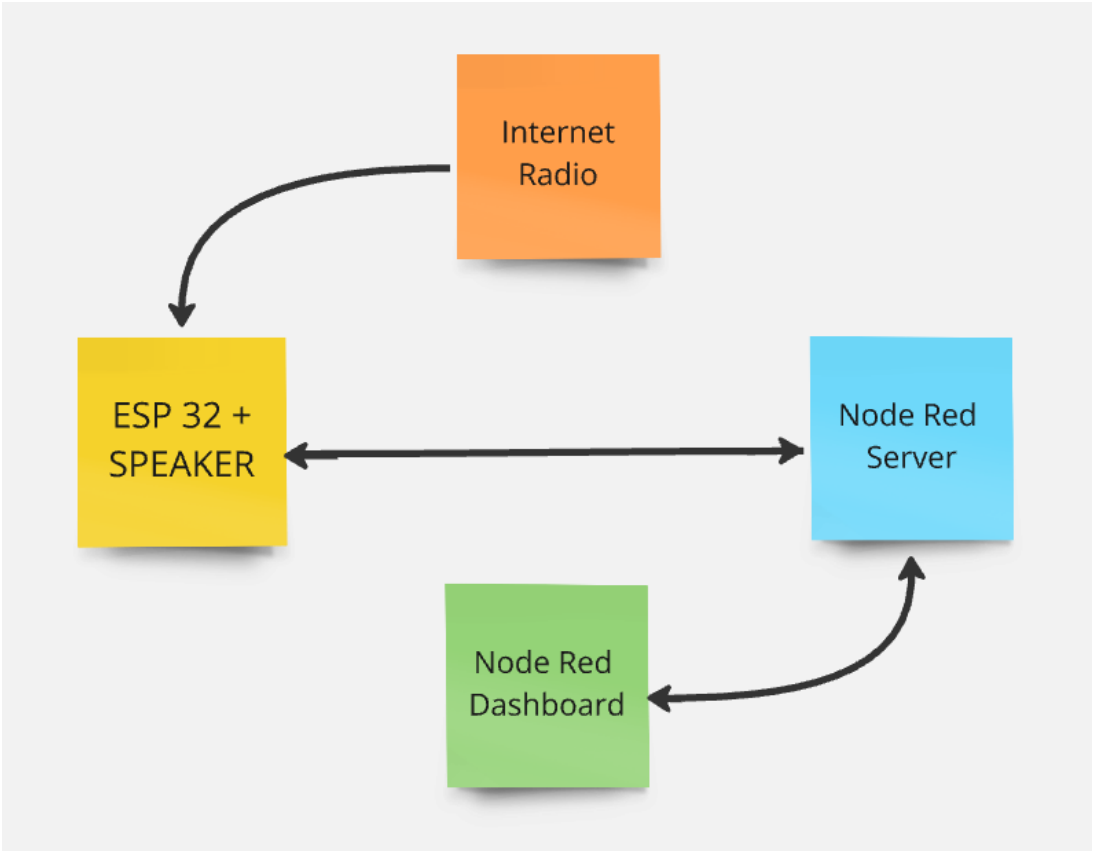

Figure 5: Block Diagram

The block diagram outlines the architecture of the **IoT-enabled music player** system. It shows the interaction between the various components involved in the system, which include:

- **ESP32 + Speaker**:
  This block represents the core hardware of the music player, including the **ESP32 microcontroller** and the speaker itself. The ESP32 handles music streaming from the Internet radio source and responds to control commands received either from physical buttons on the speaker or from the Node-RED dashboard.

- **Node-RED Server**:
  The **Node-RED Server** acts as the central hub for handling communication between the Node-RED dashboard and the ESP32. It manages the processing of commands received from the dashboard (such as play/pause, volume adjustment, and track switching) and forwards these commands to the ESP32, ensuring real-time interaction between the user interface and the hardware.

- **Node-RED Dashboard**:
  The **Node-RED Dashboard** provides the user interface through which users can control the music player remotely. It allows users to send commands (adjusting volume, switching tracks) to the ESP32 via the Node-RED server. The dashboard is accessible from any web-enabled device (such as a phone or computer) and enables the user to manage the music player remotely.

- **Internet Radio**:
  The **Internet Radio** block represents the external source from which the ESP32 streams music. The ESP32 connects to the Internet to fetch streaming audio data from the radio service and plays it through the connected speaker.

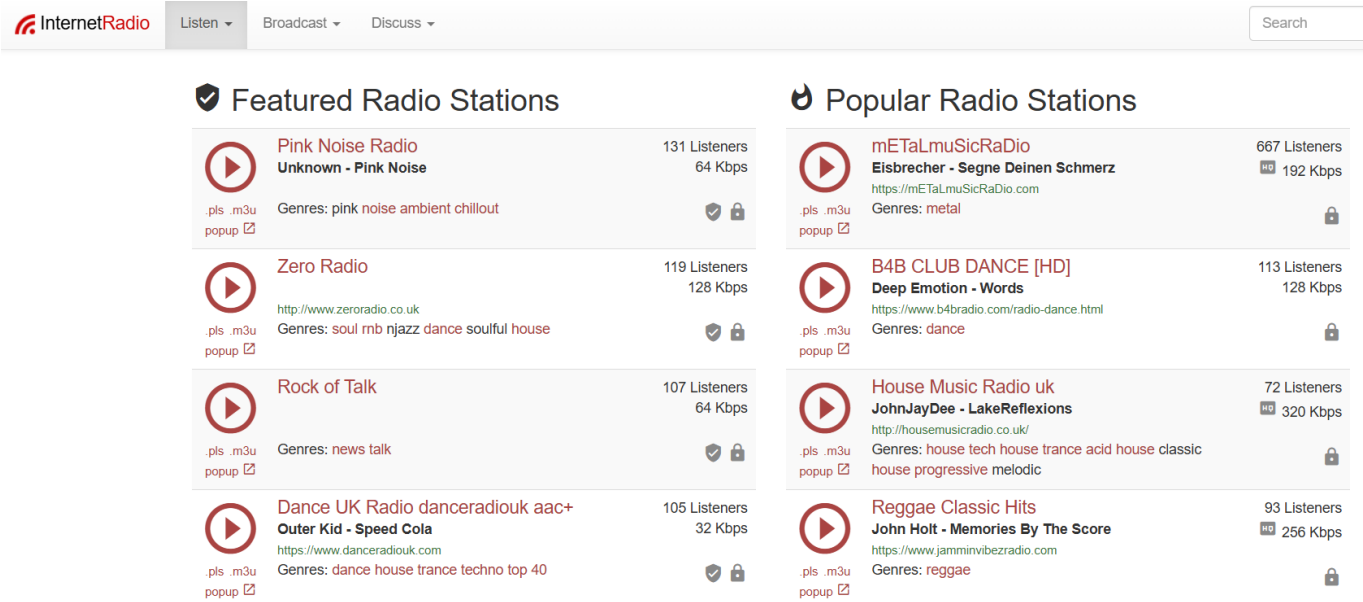

Figure 6: internet radio

Workflow:

- The **ESP32** fetches and streams music from the **Internet Radio** source and outputs it through the speaker.
- The **Node-RED Server** facilitates communication between the ESP32 and the **Node-RED Dashboard**.
- Users can interact with the system via the **Node-RED Dashboard**, sending commands (like volume adjustments, playlist management, etc.) to the Node-RED server.
- The **Node-RED Server** processes these commands and forwards them to the **ESP32**, which executes the necessary actions on the music player.

6 Design and Technical Details

6.1 Hardware

The **ESP32 microcontroller** is at the heart of the IoT-enabled music player. It is selected due to its robust features, including **Wi-Fi** and **Bluetooth** connectivity, making it ideal for IoT applications. The ESP32 is responsible for processing both **manual inputs** through physical buttons and **remote commands** via the Node-RED dashboard, allowing the system to respond in real time. Additionally, the ESP32 will be connected to the speakers to ensure **high-quality audio output**. **Physical buttons** will be included to allow for essential control functions such as play, pause, volume adjustment, and track skipping.

6.2 Software

The software aspect of the system revolves around the **Node-RED platform**, a powerful and intuitive flow-based development tool. The **Node-RED dashboard** will serve as the **remote user interface**, accessible from any internet-enabled device such as smartphones or computers. This dashboard provides users with the flexibility to manage and control their music experience remotely. The platform allows for various functionalities, including playlist management, volume control, and switching between stations, all through an easy-to-use graphical interface.

6.3 Communication and Connectivity

The system will utilize **Wi-Fi** to connect the ESP32 microcontroller to the Internet, enabling the music player to stream radio stations or music from online sources. The Wi-Fi connection will also facilitate remote control via the Node-RED dashboard. The communication between the dashboard and the ESP32 will primarily be handled using the **MQTT protocol**, which allows for low-latency, real-time interaction. However, **HTTP** can be used if broader compatibility is required.

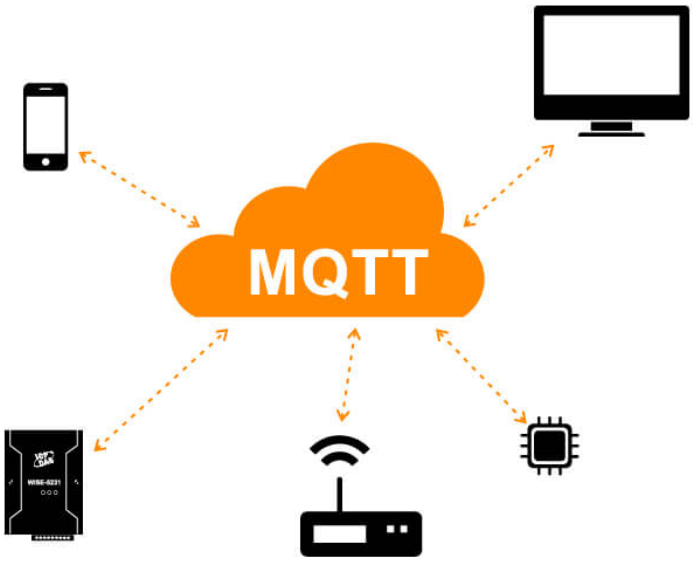

Figure 7: MQTT

7 Expected Outcomes and Benefits

7.1 Outcomes

The expected outcome of this project is a fully functional **IoT-enabled portable music player** capable of streaming music and Internet radio, controllable both manually via physical buttons and remotely through a Node-RED dashboard. The system will be able to:

- Stream music or Internet radio via the ESP32.
- Allow manual control of playback functions (play, pause, volume adjustment, track skipping) through physical buttons.
- Enable remote control via a Node-RED dashboard, allowing users to interact with the music player from any internet-enabled device.

7.2 Benefits

The IoT-enabled music player offers a range of benefits, including:

- **Enhanced User Experience**: Users will enjoy a flexible, modern music player with both manual and remote control options. The ability to remotely manage playlists, adjust volume, skip tracks, and switch radio stations via the Node-RED dashboard will provide greater convenience.
- **Seamless Music Streaming**: The ESP32's Wi-Fi capabilities will enable users to stream music or Internet radio without interruption, providing a smooth listening experience.
- **Portability**: The compact design of the music player, combined with the ESP32's low power consumption, ensures the system is portable and suitable for use in various environments, whether at home or on the go.
- **IoT Integration**: By integrating IoT technologies, the system represents a step toward the future of connected devices. Users can interact with the music player from anywhere, adding flexibility and improving their overall audio experience.
- **Cost-Effective Solution**: The use of ESP32 and open-source software (Node-RED) makes the system affordable to build while still offering advanced functionalities and control options typically found in more expensive, proprietary systems.

Figure 8: music streaming

8 Conclusion

This project will deliver an innovative and functional **IoT-enabled music player system** that integrates both hardware and software to offer modern, interactive functionalities. The use of the ESP32 microcontroller ensures that the system remains cost-effective while delivering advanced features such as Wi-Fi connectivity, Internet radio streaming, and IoT control. The integration of the **Node-RED dashboard** will enhance the user experience by enabling real-time, remote control over the device from any internet-connected device.

Overall, the music player will be a powerful yet portable solution that combines the traditional features of a music player with the flexibility and convenience of IoT technologies, making it a practical and user-friendly device.

9 Codes

10 References