Bluetooth Console Messaging: Peer-to-Peer Text over RFCOMM

Group - B11
Aasim Mohammed M S (CB.AI.U4AID24101)
Aatish Ayyapath (CB.AI.U4AID24102)
Narendra R (CB.AI.U4AID24133)
Pravin Raj R P (CB.AI.U4AID24141)

Amrita Vishwa Vidhyapeetham

October 14, 2025

Table of Contents

Quick Recap

Problem Statement:

- Traditional Bluetooth applications require continuous pairing and stable connections.
- There is a lack of lightweight, device-to-device chat systems for short-range communication without internet dependency.

Project Objectives:

- Design and develop a peer-to-peer Bluetooth chat application using the RFCOMM protocol.
- Implement asynchronous message handling for reliable communication.
- Build an interactive user interface using Windows Presentation Foundation (WPF).
- Enable real-time device discovery, connection, and secure message exchange between devices.

Introduction

Technologies and Frameworks Used:

- **.NET 8.0:** Provides the runtime environment and tools for developing cross-platform applications with strong support for asynchronous programming.
- **WPF (Windows Presentation Foundation):** Used for building the interactive desktop user interface with XAML-based design.
- **32feet.NET Library:** Offers Bluetooth API support in .NET, enabling RFCOMM communication, device discovery, and data exchange.
- **C#:** The programming language used to implement the core logic of the software, Bluetooth handling, and event-driven UI interactions.

Literature Review / Background Study

Year	Publisher	Topic	Summary
2023	LNNS (Springer)	BLE Messaging	Showed browser-based BLE chat with low overhead.
2021	IEEE IoT J.	Reliability	Studied jammer detection and impact on Bluetooth links.
2019	P2P Netw. Appl.	Ad-hoc text	Proposed peer-to-peer text routing in ad-hoc networks.
2021	arXiv	Mesh perf.	Evaluated latency and reliability in Bluetooth mesh networks.
2020	Tech Tutorial	RFCOMM	Described serial port emulation for Bluetooth messaging.

System Design

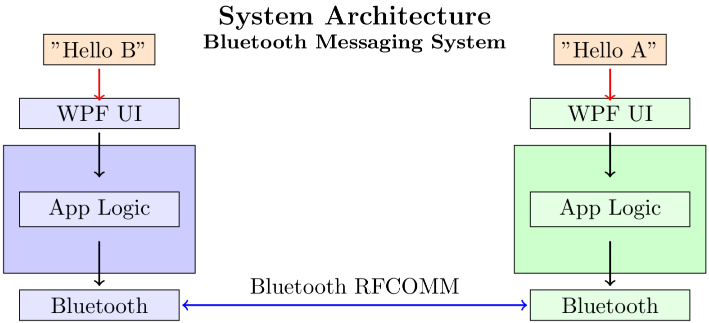
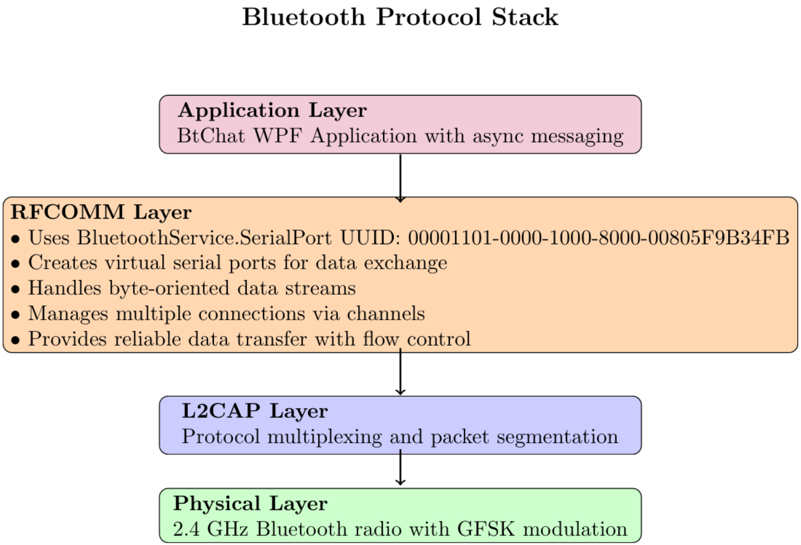
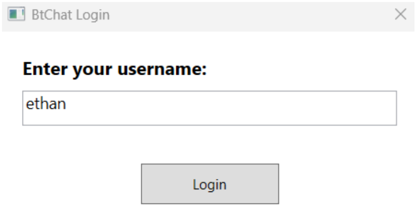


Figure: System Architecture Diagram

Methodology


Bluetooth Protocol Stack
Application Layer
BtChat WPF Application with async messaging
RFCOMM Layer
• Uses BluetoothService.SerialPort UUID: 00001101-0000-1000-8000-00805F9B34FB
• Creates virtual serial ports for data exchange
• Handles byte-oriented data streams
• Manages multiple connections via channels
• Provides reliable data transfer with flow control
L2CAP Layer
Protocol multiplexing and packet segmentation
Physical Layer
2.4 GHz Bluetooth radio with GFSK modulation

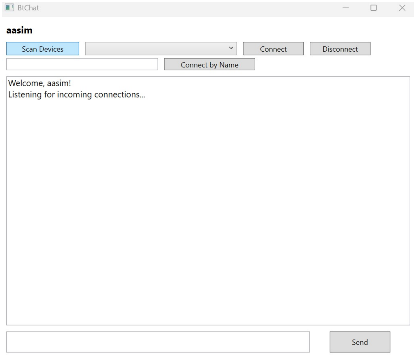
Implementation - Development Screenshots



Enter your username:

ethan

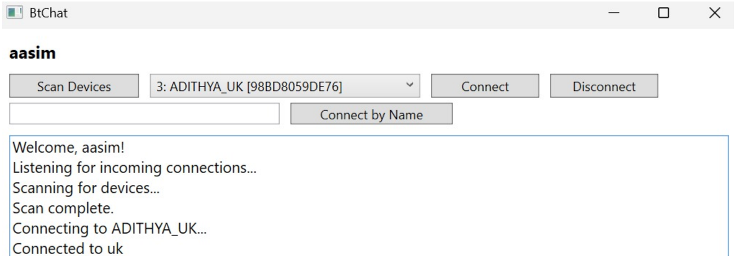
Login Window



Welcome, aasim!
Listening for incoming connections...

Main Chat Interface

Results - Connection Management



```
Welcome, aasim!
Listening for incoming connections...
Scanning for devices...
Scan complete.
Connecting to ADITHYA_UK...
Connected to uk
```

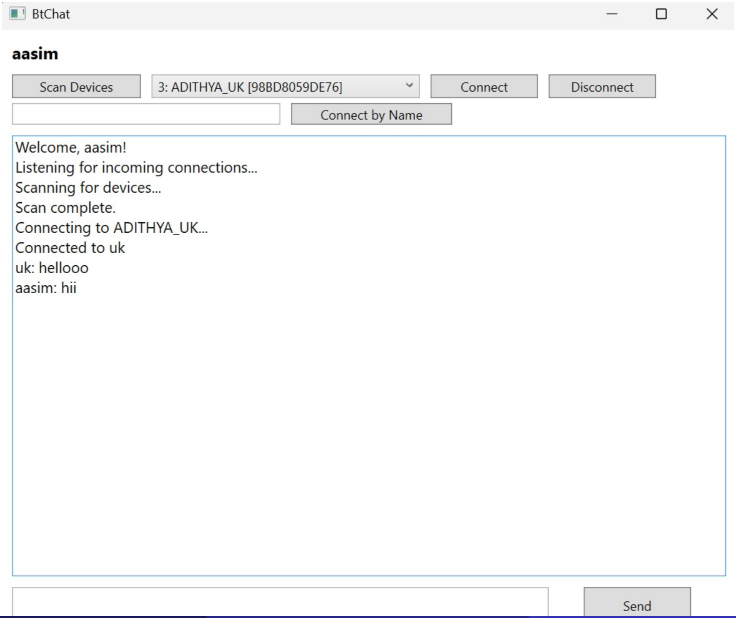
Figure: Bluetooth Device Discovery Results

Results - Performance Metrics

Metric	**Value**	**Unit**
Connection Time	2-3	seconds
Message Success Rate	100	%
Maximum Range	10-15	meters
Supported Devices	2	nodes

Table: System Performance Measurements

Results - Message Flow



```
Welcome, aasim!
Listening for incoming connections...
Scanning for devices...
Scan complete.
Connecting to ADITHYA_UK...
Connected to uk
uk: hellooo
aasim: hii
```

Discussion

Key Achievements:

- Implemented peer-to-peer Bluetooth chat using RFCOMM protocol
- Achieved stable asynchronous message transmission
- Designed an intuitive and responsive WPF user interface
- Enabled reliable device discovery and connection handling

Technical Insights:

- 32feet.NET library provides a strong API for Bluetooth communication in .NET
- Stream-based data exchange ensures properly reliable message delivery
- Asynchronous programming (async/await) prevents UI freezing during communication
- Application GUID uniquely identifies the RFCOMM service across various devices

Discussion

Limitations:

- Single active connection per device
- Windows-only due to WPF & 32feet.NET dependency
- Basic functionality (no advanced features, eg: group chat, file transfer)
- No message encryption implemented

Signaling and Communication Principles:

- Uses RFCOMM protocol for reliable serial data transfer
- Device discovery and addressing handled through Bluetooth SDP
- Connection setup involves mutual username exchange and stream creation
- Error handling through connection loss detection and cleanup routines

Conclusion & Future Work

Conclusion:

- Developed functional Bluetooth messaging system
- Implemented asynchronous communication
- Created user-friendly WPF interface
- Demonstrated signaling principles

Future Work:

- Add message encryption
- Support multiple simultaneous connections(mesh)
- Cross-platform development
- Add file transfer capabilities

References

Bluetooth SIG. (2023). *Bluetooth Core Specification*.

32feet.NET Library Documentation. (2023).

Microsoft. (2023). *.NET 8.0 WPF Documentation*.

Microsoft. (2023). *Asynchronous Programming with Async and Await*.

Thank You!