Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Spring, Year: 2024), B.Sc. in CSE (Day)

Combinational Package for Data Communication

Course Title: Data Communication Lab
Course Code: CSE-308
Section: 221-D13

Students Details

Name	ID
Mahmudul Hasan Soad	221902250
Md. Abu Rayhan Imran	221002457

Submission Date: 20-06-2024
Course Teacher's Name: Farhan Mahmud

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

The "Combinational Package for Data Communication in C++" project aims to design and develop a comprehensive data communication library that can be integrated into various C++ applications. This library will provide standardized methods for data transmission, reception, and error handling across different communication protocols.

1.2 Motivation

In the modern era of technology, data communication is critical for various applications, from networking to embedded systems. The motivation behind this project is to create a versatile and efficient package that simplifies the development process for engineers and developers, enabling them to implement robust data communication features in their applications with minimal effort.

[**?**].

1.3 Problem Definition

1.3.1 Problem Statement

Developers often face challenges in implementing reliable and efficient data communication protocols in their applications. Existing solutions are either too complex or lack the flexibility to be adapted to different use cases. This project seeks to address these issues by providing a combinational package that streamlines the process of data communication in C++.

1.3.2 Complex Engineering Problem

Designing a data communication package involves addressing complex engineering problems such as ensuring data integrity, handling communication errors, and providing support for multiple protocols. This project will tackle these challenges through careful design and implementation of robust algorithms and data structures.

1.4 Design Goals/Objectives

The primary objectives of this project are:

To design a flexible and efficient data communication package. To ensure compatibility with various communication protocols. To provide comprehensive documentation and examples for easy integration.

1.5 Application

1.5.1 Education

The package can be used as a teaching tool for students learning about data communication and networking in C++.

1.5.2 Research

Researchers can utilize the package to prototype and test new communication protocols and algorithms.

1.5.3 Technological Advancement

The package will contribute to the advancement of technology by providing a reliable tool for developing communication systems in various industries, including IoT, telecommunications, and embedded systems.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

This section covers the detailed design, development, and implementation process of the "Combinational Package for Data Communication in C++". It includes the tools and technologies used, the source code structure, and the algorithms implemented.

2.2 Project Details

The project is structured to address common challenges in data communication, including efficient data transmission, error detection and correction, and support for multiple protocols. Key features include:

Streamlined data transmission and reception.
Robust error handling mechanisms.
Customize communication workflows.

bitStuffing
characterStuffing
crc
hammingCode
ipAddressConversion
manchesterEncoding
diffManchesterEncoding
nrziEncoding

2.3 Implementation

This section details the tools and technologies used to develop our project. We will also provide an overview of the project's source code structure.

2.3.1 Tools and Technologies

- Software : Block-codes
- Operating System : Windows
- Programming Language : C++

2.3.2 Implementation details (with screenshots and programming codes)

The project source code is structured into several modules, each responsible for a specific aspect of data communication. The following is a sample of the implementation details:

2.4 Algorithms

The project involves designing algorithms for data transmission, reception, and error handling. The following pseudocode outlines the core algorithm for data communication:

Algorithm DataCommunication Input: dataToSend Output: receivedData

1: Function sendData(dataToSend) 2: Establish connection 3: Transmit data 4: Handle transmission errors 5: End Function

6: Function receiveData() 7: Establish connection 8: Receive data 9: Handle reception errors 10: Return receivedData 11: End Function

12: Main 13: dataToSend ← "Hello, World!" 14: sendData(dataToSend) 15: receivedData ← receiveData() 16: Print "Received Data: " + receivedData 17: End Main

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

In this section, we discuss the experimental setup and environment installation required for simulating the outcomes of our Linux package manager project.

3.1.1 System Setup and Installation

The experimental setup for evaluating the "Combinations Package for Data Communication in C++" includes the following components:

Hardware: A computer system with sufficient processing power, memory, and storage capacity to run the Windows operating system and perform data communication tasks. Operating System: Windows 10 or higher. C++ Compiler: Block-codes

3.1.2 Simulation Procedure

The simulation procedure involves several steps to ensure comprehensive evaluation:

Test Scenarios: Define a variety of test scenarios encompassing data transmission, reception, and error handling.
Execution: Execute each test scenario utilizing the developed combinational package.
Analysis: Analyze the results to assess performance, reliability, and usability. Iterative Testing: Refine and repeat test scenarios to ensure thorough coverage and validation.

3.2 Results Analysis/Testing

3.3 Results Overall Discussion

The Universal Linux Package Manager (UPM) provides a flexible solution for managing software across multiple Linux distributions. By centralizing commands in a JSON configuration file and detecting the distribution at runtime, UPM can adapt to various

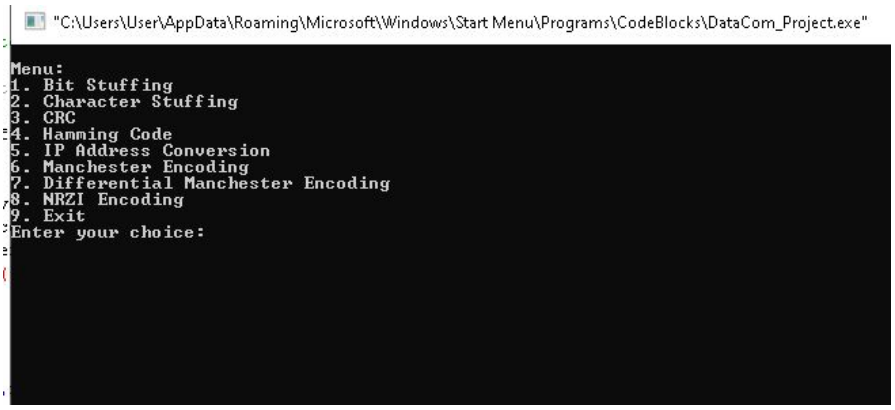

Figure 3.1: Main menu

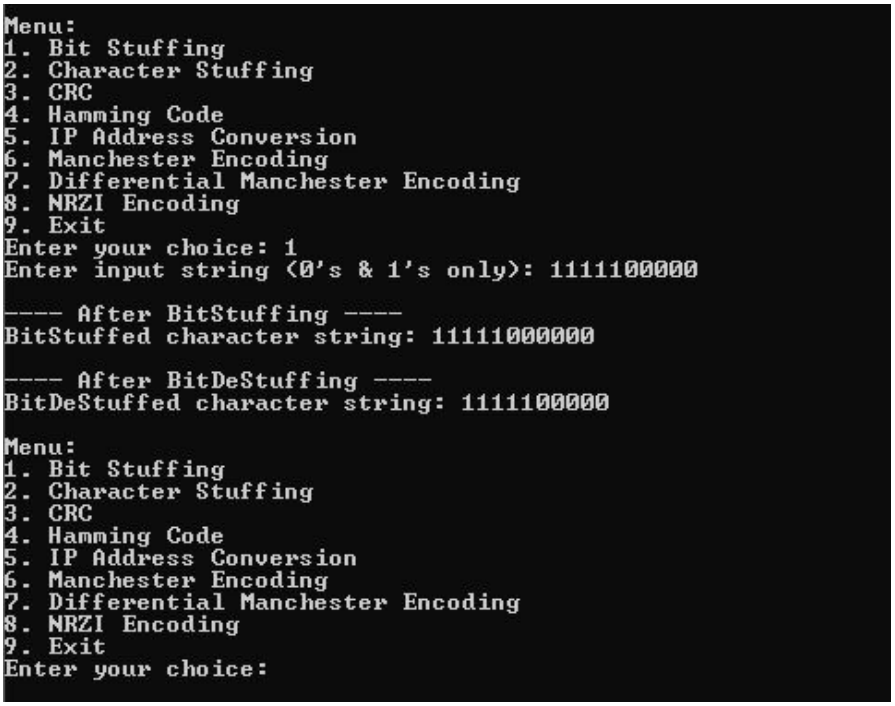

Figure 3.2: Stuffing and De-stuffing

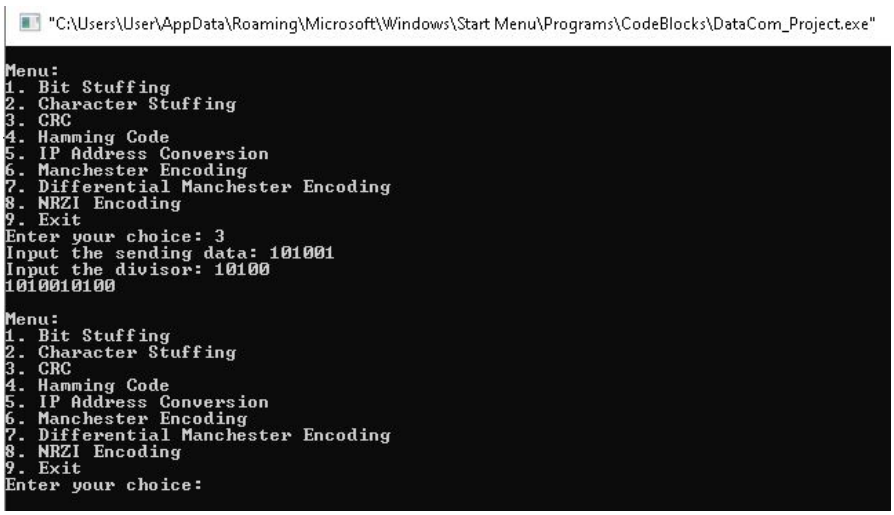

Figure 3.3: CRC

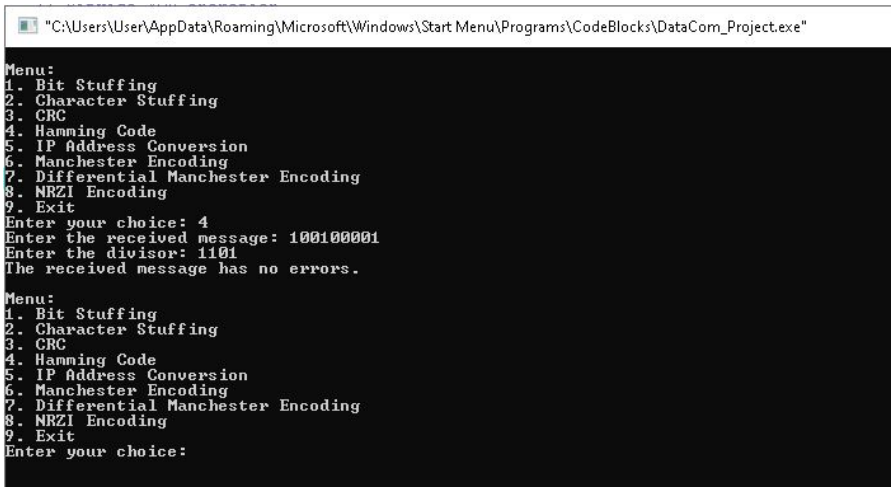

Figure 3.4: Hamming

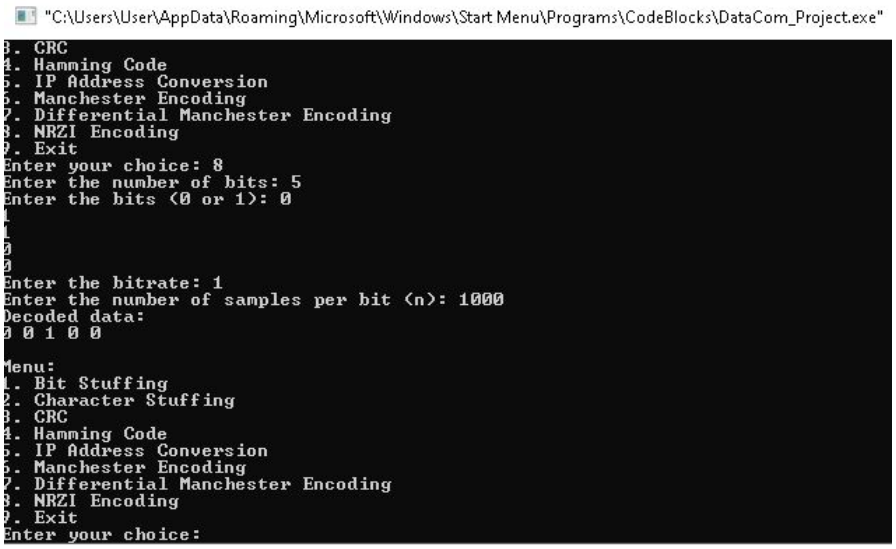

Figure 3.5: NRZ-I

environments, simplifying the process for users. The algorithm and implementation presented ensure a robust and extensible tool that can be easily maintained and expanded with additional applications and distributions

Chapter 4

Conclusion

4.1 Discussion

The "Combinational Package for Data Communication in C++" project stands out as a robust solution for managing data communication across different applications. By consolidating essential communication functions and supporting multiple protocols, the package facilitates efficient and reliable data exchange processes. This capability is crucial for enhancing application performance and ensuring seamless integration in various technological environments.

4.2 Limitations

Despite its strengths, the current version of the package has some limitations:

Limited support for certain niche communication protocols. Basic error handling mechanisms that may require further refinement. Absence of a graphical user interface (GUI), which could simplify usability for non-technical users.

4.3 Scope of Future Work

To enhance the package's capabilities and address its limitations, future development efforts could focus on:

Expanding support for additional communication protocols to cater to a broader range of application needs. Implementing advanced error handling mechanisms to improve reliability and robustness. Developing a user-friendly graphical user interface (GUI) for intuitive configuration and management of communication tasks. This would enhance accessibility and usability, especially in environments where direct command-line interaction may be challenging. By addressing these areas, the "Combinational Package for Data Communication in C++" project can further solidify its position as a versatile tool for modern data communication solutions.

4.4 References

- https://www.geeksforgeeks.org/implementation-of-bit-stuffing-and-bit-destuffing/
- https://github.com/Robetron/Hamming-Code/blob/master/Hamming
- https://www.upgrad.com/tutorials/software-engineering/software-key-tutorial/cyclic-redundancy-check/