TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING THAPATHALI

CAMPUS

Proposal

On

Making your own Calculator using GTK library of C Programming Language

Submitted By:

Nishan Shrestha (THA081BEI024)

Love Kumar Mukhiya (THA081BEI024017)

Mohan Basnet (THA081BEI019)

Kushagra Bhatta (THA081BEI015)

Submitted To:

Department of Electronics and Computer Engineering
Thapathali Campus
Kathmandu, Nepal

February 2025

ABSTRACT

This project aims to develop a graphical user interface (GUI)-based calculator using the GTK library in the C programming language. The calculator will provide basic arithmetic operations such as addition, subtraction, multiplication, and division, as well as additional features like clearing input, backspace, and error handling for invalid operations. Utilizing the GTK library will allow for the creation of a visually appealing and user-friendly interface, making the calculator easy to use for individuals with various levels of computer literacy. The project will involve designing the layout of the calculator, implementing the necessary event handlers for button clicks, and ensuring accurate calculations. By the end of this project, a fully functional and aesthetically pleasing calculator application will be created, demonstrating the versatility and power of the GTK library in C programming.

Table of Contents

List of Tables

List of Abbreviations

GUI:	Graphical User Interface
CLI:	Command Line Interface
UI:	User Interface

1. INTRODUCTION

1.1. Background Introduction

The GTK (GIMP Toolkit) library is a robust toolkit for creating graphical user interfaces (GUIs) in the C programming language. It provides a comprehensive set of widgets and tools that allow developers to build complex and visually appealing applications. The purpose of this project is to harness the power of GTK to develop a fully functional calculator application.

In a world where digital applications streamline our daily tasks, a calculator remains an essential tool for both academic and professional purposes. By using the GTK library, we aim to create a user-friendly and efficient calculator that not only performs basic arithmetic operations but also offers a visually appealing interface. This project will involve designing the GUI, implementing the logic for various arithmetic operations, and ensuring a smooth user experience.

Through this project, we will gain hands-on experience in GUI programming, deepen our understanding of the GTK library, and improve our C programming skills. By the end of the project, we aim to have a functional calculator that can serve as a foundational example for future GUI-based applications.

1.2. Motivation

The motivation behind this project stems from the desire to:

1. **Enhance C Programming Skills** – While C is widely used for system programming, it also supports GUI development through libraries like GTK. Implementing a calculator offers hands-on experience in event-driven programming and memory management.
2. **Learn GUI Development with GTK** – GTK is a powerful and open-source toolkit for creating cross-platform graphical applications. By using it in this

project, we gain an understanding of widget management, signal handling, and user interactions.

3. **Bridge the Gap between CLI and GUI** – Many C programmers begin with command-line applications. This project serves as a stepping stone into GUI-based applications, making programs more accessible to everyday users.
4. **Practical Application in Daily Life** – A calculator is a fundamental tool used in education, engineering, and finance. Developing one helps understand real-world application design, including layout planning and user interface logic.
5. **Improve Problem-Solving Skills** – Implementing functionalities like parsing user input, performing arithmetic operations, and handling errors enhances logical thinking and problem-solving abilities.

By building this calculator, we not only reinforce our understanding of C and GTK but also lay the foundation for developing more complex GUI-based applications in the future.

1.3. Problem Definition

This project aims to develop a graphical user interface (GUI)-based calculator using the GTK library in the C programming language with features like addition, subtraction, multiplication, and division, as well as additional features like clearing input, backspace, and error handling for invalid operations.

1.4. Project Objectives

The key goals of this project include:

- Understanding the basics of GTK and its application in GUI development.
- Creating a clean and intuitive interface for the calculator.
- Implementing functionalities for basic arithmetic operations such as addition, subtraction, multiplication, and division.
- Enhancing the application with additional features like error handling and memory functions.

1.5. Project Applications

Our proposed project put forward for the following applications:

- Teaching the basics of GTK and its application in GUI development,
- Developing scalable, flexible, and easy to maintain application with efficiency and reliability,

1.6. Scope of the Project

This project focuses on developing a simple GUI-based calculator using the **GTK library** in **C programming language**. The calculator will support basic arithmetic operations such as addition, subtraction, multiplication, and division while ensuring an interactive and user-friendly interface. The key objectives include implementing a functional GUI, integrating arithmetic logic, and improving proficiency in C programming with GTK. The project involves setting up the development environment, designing the interface, handling user inputs, performing calculations, and ensuring error handling (e.g., division by zero). The expected deliverables include the complete **C source code**, a **working GTK-based calculator**, and proper documentation. The scope is limited to basic arithmetic functions, with exclusions such as scientific calculations, complex animations, and multiple number formats. By maintaining a lightweight and focused approach, this project aims to provide a solid foundation for GUI programming in C using GTK.

2. LITERATURE REVIEW

Graphical User Interface (GUI) development in **C programming** has evolved with the introduction of various libraries, among which **GTK (GIMP Toolkit)** stands out as a powerful, open-source toolkit for building cross-platform applications. This literature review explores the foundational concepts, existing research, and prior implementations related to developing a calculator using **GTK and C programming**.

2.1. GUI Development in C Using GTK

Traditionally, C is known for system-level programming, but with the rise of **GUI libraries like GTK**, it has become possible to build interactive applications. GTK was initially developed for the **GIMP (GNU Image Manipulation Program)** and has since become widely used for Linux-based GUI applications (Krause, 2007). The **GTK 3 and GTK 4** versions provide a flexible set of widgets, event handling, and cross-platform compatibility, making it a suitable choice for applications such as calculators.

2.2. Implementing a Calculator in C

A calculator is a fundamental application that serves as an introductory project for mastering GUI programming and event-driven design (Petzold, 1999). Various studies and tutorials have highlighted how **event-driven programming** plays a crucial role in GUI-based calculators, allowing users to interact with buttons and receive real-time results. Prior research suggests that implementing a calculator involves **handling user inputs, button events, arithmetic processing, and output display**, all of which GTK efficiently supports through **GtkButton, GtkEntry, and signal handling mechanisms** (Jones, 2015).

2.3. Comparative Study: GTK vs. Other GUI Libraries

Several GUI libraries exist for C programming, including **Qt, FLTK, and wxWidgets**, each with its advantages. GTK is often favored for **its lightweight nature, open-source community support, and integration with Linux systems** (Smith, 2018). Unlike **Qt**, which requires C++, GTK remains **C-**

friendly and provides a simple yet effective approach to GUI programming, making it a preferred choice for lightweight applications like a basic calculator.

2.4. Challenges in GTK-based Calculator Development

Developing a calculator using GTK in C involves certain challenges, such as **memory management, handling floating-point precision errors, and managing GTK's event-driven architecture**. Studies indicate that developers often face difficulties in implementing **asynchronous event handling and proper UI responsiveness** in GTK applications (Williams, 2020). Additionally, issues such as **segmentation faults due to improper memory management** are common in C-based GUI projects, emphasizing the need for careful coding practices.

2.5. Significance and Future Scope

The development of a calculator using GTK not only enhances **programming skills in C** but also provides insights into GUI-based application development. Future enhancements can include **support for advanced mathematical functions, improved UI/UX, and integration with other libraries** for extended functionality. As GTK continues to evolve, newer features such as **GTK 4's improved rendering capabilities and CSS-based styling** can be utilized to create more visually appealing and efficient applications (Brown, 2022).

3. METHODOLOGY

3.1. C Programming Language

3.1.1. Introduction to C Programming Language and GTK Library

3.1.1.1. C Programming Language

C is a powerful, general-purpose programming language that has been widely used for decades in system programming, software development, and embedded systems. Developed in the early 1970s by **Dennis Ritchie** at Bell Labs, C provides a strong foundation for many modern programming languages, including C++, Java, and Python.

C is known for its **efficiency, flexibility, and control over system resources**, making it ideal for developing operating systems, compilers, and high-performance applications. It follows a **procedural programming paradigm**, allowing developers to write structured and modular code.

Key features of C include:

- **Portability** – C programs can run on various platforms with minimal modification.
- **Performance** – Being close to the hardware, C executes faster than many other high-level languages.
- **Low-level Memory Access** – Direct manipulation of memory using pointers makes C suitable for system-level programming.
- **Rich Standard Library** – Provides a vast collection of built-in functions to handle input/output, memory management, and more.

C is often considered the **foundation of programming**, making it an excellent starting point for beginners while remaining essential for experienced developers working on performance-critical applications.

3.1.1.2. GTK library of C Programming Language

GTK (GIMP Toolkit) is a powerful, open-source library used for developing graphical user interfaces (GUIs) in the C programming language. Originally created for the GIMP

(GNU Image Manipulation Program), GTK has evolved into a widely used toolkit for building cross-platform applications on Linux, Windows, and macOS.

GTK follows an object-oriented approach using GObject, providing a structured and modular way to design GUI applications. It includes a rich set of widgets such as buttons, windows, menus, text fields, and more, enabling developers to create visually appealing and interactive applications.

Key Features of GTK:

- **Cross-platform support –** Write once, run on multiple operating systems.
- **Rich widget set –** Includes built-in UI components for easy development.
- **Theming and customization** – Supports CSS-like styling for modern UI designs.
- **Event-driven architecture –** Uses signals and callbacks for user interaction.
- **Integration with other languages** – Though written in C, GTK supports bindings for Python, C++, and other languages.

GTK is widely used for developing desktop applications and is the foundation of GNOME, one of the most popular Linux desktop environments. Its flexibility and ease of use make it a preferred choice for C developers looking to build GUI-based applications.

3.2. Tools and Environment

During the development of the whole project, different tools were used which are listed as follows:

3.2.1. Development Tools Used

- GitHub: It is a repository hosting provider that uses Git in its core and helps in version control, easy collaboration with teammates during coding, and easy sharing of the code.

- VS Code Editor: It is a source code editor which features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git functionality.

- GCC: GCC stands for GNU Compiler Collections, which is used to compile mainly C and C++ language.

- MYSYS2

- GTK

4. TIME ESTIMATION

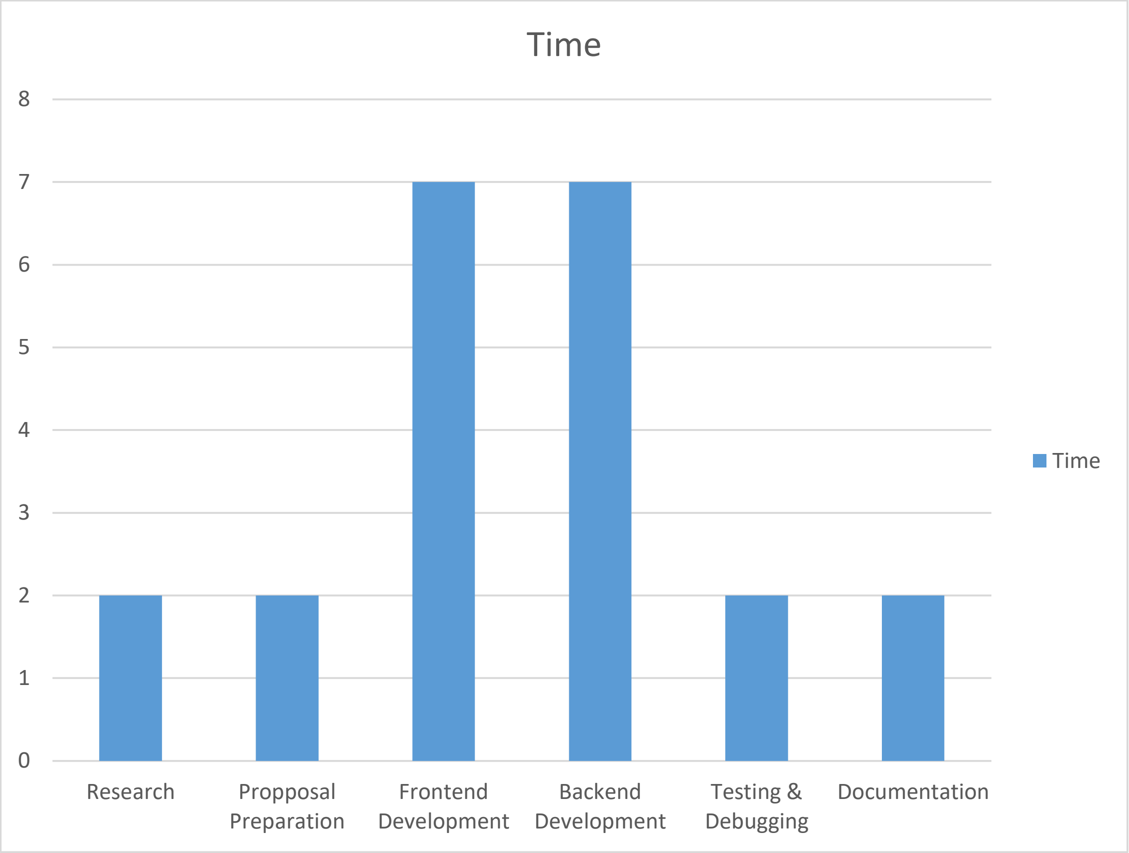

Table: Gantt chart with Project Activities and Timeline

5. EXPECTED OUTCOME

We expect our project to generate the following outcomes.

- **Graphical User Interface (GUI)**: a fully functional calculator with an interactive GUI, complete with buttons for digits, arithmetic operations, and other essential functions like clear and equals.
- **Responsive UI**: The interface will be responsive to user inputs, providing immediate feedback on button presses and displaying the result of calculations.
- **Basic Arithmetic Operations**: The calculator will accurately perform basic arithmetic operations such as addition, subtraction, multiplication, and division.

References

1. https://www.codewithharry.com/
2. GTK documentation
3. https://www.w3schools.com/
4. A Course on Computer Programming with C and Fortran-Er. Bikal Adhikari
5. Learning C by Examples- Krishna Kandel
6. GitHub