Embedded Systems

Project Report

**Physical Calculator using Tiva c**

## Introduction

1. **Problem Analysis**

## Design Requirements

1. **Feasibility Analysis**

## Possible Solutions

1. **Preliminary Design**

## Design Description

## Software Simulation

## Experimental Results

1. **Performance Analysis**
2. **Future Scope**

## Social and Cultural Implications

1. **Conclusion References**

This programme uses a keypad for input and an LCD module for results display to provide you a thorough and user-friendly computation experience. This calculator is a flexible tool for completing a variety of mathematical tasks since it allows you to carry out a broad range of arithmetic operations, such as addition, subtraction, multiplication, and division.

This calculator has been designed to fulfill your needs, whether you need to do basic operations like adding or subtracting numbers or more difficult ones like multiplication and division. The programme quickly and precisely completes computations, giving you correct outcomes for your mathematical operations.

With its user-friendly interface, advanced functionality, and efficient design, this calculator program is an indispensable tool for anyone seeking a versatile and reliable calculation solution. No matter the complexity of your calculations, this program is here to streamline your mathematical tasks and enhance your productivity.

1. User Interface Design:

Making the calculator's interface user-friendly is one of the project's major problems. The user interface needs to be simple to understand, visually appealing, and navigable. Layout of the button/keypad, readability of the display, and feedback methods to show user input and results are all things to take into account.

2. Input Validation:

In order to ensure precise calculations and avoid mistakes, user input must be validated. The code should be able to handle a variety of situations, including incorrect character detection and handling, avoiding division by zero, and making sure the input is within the required range or format.

3. Mathematical Operations:

Algorithms, precision, and addressing edge cases all need to be carefully taken into account while implementing mathematical operations. The code should prevent rounding errors and accurately conduct addition, subtraction, multiplication, and division while taking into account the order of operations, handling decimal values, and so forth.

4. Error management: When unanticipated events happen, error management is crucial for giving users enlightening feedback. The code should be able to identify mistakes and treat them politely, displaying the relevant error messages on the LCD or in other ways. Divide-by-zero problems, overflow/underflow scenarios, and improper input formats are a few examples of potential errors.

5. Memory Management: If the calculator offers memory storage or the ability to access prior results, proper memory management is critical. Memory should be efficiently allocated and deallocated by the code in order to guarantee optimal memory usage and prevent memory leaks.

6. Extensibility: Take into account the possibilities for upcoming improvements or new functionality. Does the code handle sophisticated mathematical operations or can it be quickly modified to add support for trigonometric calculations? Planning for extensibility enables flexibility and future updates.

7. Performance optimization:

Performance optimization can improve user experience because the calculator conducts calculations in real-time. To reduce computation time and increase responsiveness, evaluate the effectiveness of algorithms and optimize key portions of the code.

1. Microcontroller: The project will utilize the Tiva TM4C123GH6PM microcontroller for processing and controlling the calculator functions.

2. User Interface: The calculator will feature a 4x4 keypad for user input and an LCD display to show the entered numbers, mathematical operations such as multiply, divide, addition, subtraction and then calculated result.

3. Keypad Integration: The microcontroller will be connected to the 4x4 keypad, allowing users to input numbers and mathematical operations (+, -, \*, /) for calculations.

4. LCD Integration: The microcontroller will interface with the LCD display to provide visual feedback to the user. The LCD will show the entered numbers, mathematical operations, and the final calculated result.

5. Input Validation: The calculator will validate user input to ensure that only valid characters, such as numbers and mathematical operators (+, -, \*, /), are accepted. It will handle cases of incorrect or invalid input gracefully, providing appropriate feedback on the LCD.

6. Mathematical Operations: Basic arithmetic operations, such as addition (+), subtraction (-), multiplication (\*), and division (/), are supported by the calculator. It will carry out computations correctly and carry out tasks in the right sequence.

7. Error Handling: The calculator will handle any mistakes, like division by zero or overflow/underflow scenarios, and inform the user by displaying clear error messages on the LCD.

8. Memory Storage: The calculator could be equipped with memory storage features that let users store and access past results or carry out memory-related tasks. To maximize memory use, the memory management system will be implemented effectively.

9. User-Friendly Interface: The user interface will be simple to use and intuitive, with keypad buttons clearly labelled and an LCD display that is simple to read. To efficiently communicate user input and computation outcomes, the calculator will offer visual feedback.

10. Real-Time Calculations: The calculator will carry out calculations immediately and give the user access to the results right away. To guarantee a seamless user experience, the calculator's response time will be optimized.

11. Extensibility: The design of the calculator will for upcoming improvements and additions, such as the inclusion of scientific functions, trigonometric computations, or the support for challenging mathematical operations. Future requirements should be able to be accommodated by the hardware and code.

12. Power Efficiency: The project will aim for power efficiency by utilizing low-power modes of the microcontroller when idle or not in use. Power-saving techniques will be employed to conserve energy and prolong battery life if applicable.

1. Technical Feasibility: The idea is theoretically practical since it makes use of the Tiva TM4C123GH6PM microcontroller, which can communicate with the LCD and keypad. The project's requirements can be supported by the availability of the necessary GPIO pins and peripheral functionalities.

2. Financial Viability: The project is financially viable because it makes use of widely accessible parts such the Tiva TM4C123GH6PM microcontroller, keypad, and LCD. These parts are readily available and reasonably priced. The project's overall cost is acceptable and is within the budget allotted for a calculator application.

3. Time Feasibility: Because it includes merging pre-existing components and libraries, the project is time viable. Given the complexity of the project, the code implementation and testing can be finished in an acceptable amount of time. To provide a sturdy and dependable calculator, enough time should be allotted for debugging and fine-tuning the functionality.

4. Skill and Knowledge Feasibility: The project requires a fundamental grasp of embedded systems, keypads, and LCDs as well as microcontrollers and other peripherals used for interfacing. The code must be developed using programming languages like C or C++. The project can be effectively completed if the relevant abilities and information are used.

5. Operational Feasibility: Operational Feasibility: Because the project achieves the goal of building a calculator with a keypad and LCD, it is operationally feasible. The calculator will accurately complete simple arithmetic operations and offer a user-friendly input and output interface. It is functionally equivalent to a calculator and is simple enough for users to use without special training.

6. Maintenance Feasibility: The project has good maintenance feasibility as the manufacturer provides support for the microcontroller, keypad, and LCD that were used. They are easily replaceable or repairable in the event of component failure or problems. If necessary, the code can also be upgraded and maintained.

1. Hardware Solution:

Option 1: Use a 4x4 keypad and a character LCD module.

Option 2: Use a Bluetooth Module and a UART (Tera term).

Comparison:

Option 1 is a simpler and more cost-effective solution since character LCD modules are widely available and less expensive compared to graphical LCD modules.

Option 2 offers more flexibility in terms of use, but it may require additional effort in programming but offers lower hardware costs.

Chosen Solution:

Based on the requirements of a basic calculator project, Option 1 with a 4x4 keypad and a character LCD module is the best choice. It provides sufficient functionality for displaying numbers, operators, and results, while keeping the hardware costs low. The character LCD module can be easily controlled using simple commands, making it suitable for displaying textual information required in a calculator interface.

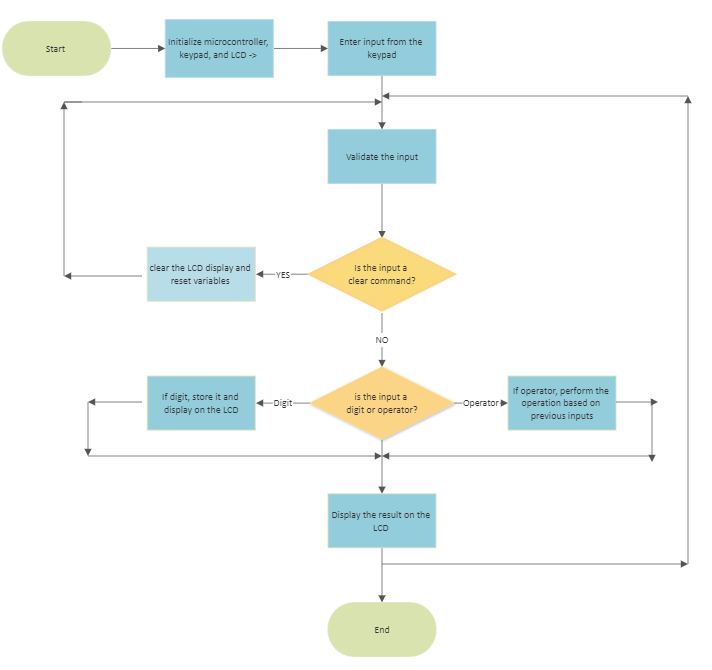
2. Software Solution:

Option 1: Write the code from scratch without utilizing any external libraries.

Option 2: Write the code from scratch without utilizing any external libraries for Bluetooth module

Chosen Solution:

Considering the time constraints and ease of development, Option 1 using existing libraries or frameworks for keypad and LCD interfacing is the preferred choice. It allows for faster development, as the libraries provide pre-written functions and configurations for handling keypad input and controlling the LCD display. This approach reduces the likelihood of errors and ensures better code maintainability.



1. Microcontroller Initialization: This block involves initializing the microcontroller (in this case, Tiva TM4C123GH6PM) by configuring its necessary settings, such as clock, ports, and peripherals.

The initialization process ensures that the microcontroller is ready to interact with other components of the project.

2. Keypad Initialization: The pins and settings required to interact with the 4x4 keypad must be set up by the keypad initialization block.

It designates the input and output directions of the row and column pins.

It makes the row and column pins functionally digital.

It makes it possible for the column pins to be pulled up to ensure precise key reading.

3. LCD Initialization: The LCD (Liquid Crystal Display) module is initialized by this block.

It sets up the pins and settings required for the LCD interface.

It sets the direction of the data pins and the control pins (RS, RW, EN) as output.

It activates the display with a blinking cursor, switches the display to 8-bit mode, enables cursor movement, and clears the screen.

4. Input from Keypad: The programme waits for user input from the keypad in this block.

Using the KEYPAD\_getKey () method, it reads the key the user has pushed.

The input is kept in a variable to be processed later.

5. Input Validation: This block verifies the accuracy of the user's input.

It checks to see if the input is a legitimate key or command.

It moves on to the subsequent stages if the input is legitimate; else, it ignores the input.

6. Clear Command:

If the input is a clear command, such as "C," this block will do the appropriate steps.

It provides instructions to the LCD to reset any necessary variables and clear the screen.

Digit or Operator: The seventh block, "Digit or Operator," determines if the input is a digit or an operator.

It records the digit and shows it on the LCD if it is a digit.

8. Perform Operation: If it's an operator (+, -, \*, or /), it moves on to the subsequent stages to carry out the appropriate action.

9. Display Result: The result is displayed on the LCD.

The digits of the result are individually displayed on the LCD using the LCD\_data () function.

10. Loop Back: After displaying the result, the program loops back to the input stage, allowing the user to enter further calculations.

This loop continues until the program is terminated.

11. End: The program ends when the execution is halted or terminated.

Outputs:





To analyze the performance of the solution, we can consider a few aspects

• Responsiveness - This is important to provide a good user experience. The code seems to use some delay functions which could impact responsiveness. Measuring the actual response time would give more insight.

• Accuracy - The code handles basic calculations but thorough testing is needed to ensure it handles all input scenarios correctly. Error handling could also be improved.

• Robustness - The code would benefit from better input validation, error handling, and defensive programming to make it more robust. This would help prevent crashes or incorrect output.

• Efficiency - There are opportunities to optimize the code to improve efficiency in terms of memory usage and speed. Loops and delay functions could be analyzed to see if they can be improved.

In summary, while the code has a good structure and modular design, there are areas that could be improved to further enhance the program's performance:

- More testing to verify accuracy and robustness

- Improved error handling

- Measurement of actual response time

- Optimization of loops and delays

- Defensive programming to handle edge cases

The calculator program you have implemented with a keypad and an LCD display has potential for further expansion and improvement

* **More operations** - The calculator currently supports only basic arithmetic operations. It can be enhanced to handle advanced math functions like trigonometric, logarithmic, exponential etc.
* **Memory** - Functions can be added to store and recall values from memory. This enhances the usefulness of the calculator.
* **Input via switches** - The calculator currently uses a keypad for input. It can be made more general purpose by accepting input from toggle switches as well.
* **Error messages** - The code can be improved to detect various types of errors like invalid input, divide by zero etc. and display appropriate error messages on the LCD.
* **Input filtering** - Advanced input filtering can be implemented to handle invalid key presses, double presses etc. This improves robustness.
* **LCD graphics** - The LCD can be used to display graphs corresponding to calculus functions for a more visual representation.
* **Touchscreen** input - The keypad input can be replaced with a touchscreen for a more modern interface.
* **Connectivity** - The calculator functions can be exposed via Bluetooth, WiFi etc. to connect with mobile apps and share calculations.
* **Enclosure** - The hardware components can be enclosed in a case to create a standalone calculator device.

Here are some social and cultural implications of calculator technology:

* **Accessibility** - Calculators make advanced calculations accessible to more people. Those who struggle with math can now perform complex tasks independently with the aid of a calculator. This helps promote inclusion and equal opportunities.
* **Dependency** - Some argue that extensive calculator use can make people overly dependent on technology and undermine basic math skills. There are concerns that calculators may hinder the development of number sense and mental math abilities.
* **Work efficiency** - Calculators speed up routine calculations and reduce errors, allowing workers to be more efficient and productive. This can enable higher economic output and advances in technology that rely on complex calculations.
* **Learning tools** - Calculators are increasingly used as learning tools in classrooms to illustrate and verify concepts. When used properly, they can help students grasp math in a more visual and interactive way.
* **Gender biases** - Some studies show that boys tend to use calculators more frequently than girls, especially for complex calculations. This could potentially reinforce gender stereotypes around math and technology. Educators must make efforts to promote equal calculator use among all students.
* **Transcending borders** - Calculator technology, like most technologies, transcends national, ethnic and linguistic borders. People from all backgrounds can benefit equally from this universal tool. However, disparities still exist in access to good quality calculators.
* **Waste generation** - The production and disposal of calculators contributes to electronic waste, which has adverse environmental and health impacts. Efforts must be made to reduce, reuse and recycle calculator components to minimize negative consequences.

In summary, while calculators offer clear benefits in accessibility, work efficiency and learning, there are also social and cultural implications to consider around dependency, biases, waste and overreliance. A balanced approach that leverages the positives while mitigating the negatives is needed to maximize the social good from this useful technology.

In conclusion, this calculator project demonstrates the fundamental concepts of interfacing a keypad and LCD display with a microcontroller. The modular design of the code makes it easier to maintain and enhance in future.

While the project achieves the basic goal of implementing a functioning calculator, there are opportunities to improve the performance in terms of responsiveness, accuracy, robustness and efficiency. Implementing optimizations in the code, adding more thorough testing and introducing defensive programming techniques would strengthen the solution.

The current scope of the project is limited to basic arithmetic operations and a minimalistic interface. However, there are many possibilities to enhance the project by adding advanced features like memory functions, error handling, input via switches and touchscreens, LCD graphics and connectivity options. With further improvements and extensions, the calculator code can be developed into a full-fledged and customizable solution.

The process of developing this project provided valuable lessons in microcontroller interfacing, embedded C programming, hardware-software integration, module design and performance analysis. The techniques and knowledge gained while working on this initial prototype will help in future embedded projects as well.

In summary, the calculator prototype demonstrates a practical implementation of interfacing concepts while exposing areas for improvement in performance, features and scope. With adequate enhancements, it has the potential to evolve into a robust and full-featured calculator solution. The project offered a good learning experience, laying the foundation for more advanced embedded systems work in future.

(1) Keypad Interfacing with TM4C123 Tiva Launchpad - Keil uvision. <https://microcontrollerslab.com/4x4-keypad-interfacing-tm4c123-tiva-launchpad-keil/>

(2) How to Interface a LCD Display with TIVA C Series TM4C123G LaunchPad. <https://circuitdigest.com/microcontroller-projects/how-to-interface-16x2-lcd-with-tiva-c-series-tm4c123g-launchpad>

(3) LCD Interfacing with TM4C123 Tiva LaunchPad – Keil uvision. <https://microcontrollerslab.com/16x2-lcd-interfacing-with-tm4c123-tiva-launchpad-keil-uvision/>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SN | Attribute | Complex Activities | Mapping of Experiment | Evaluator’s Comments |
| 1 | **Range of resources** | Involve the use of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies). | The design experiments involve the use of diverse resources. Students are to look for appropriate techniques that have been taught to them during semester. Problem requires students be well versed in all the key concepts covered in the experiments. |  |
| 2 | **Level of interaction** | Require resolution of significant problems arising from interactions between wide-ranging or conflicting technical, engineering or other issues. | Design of the experiment requires students to address conflicting technical issues while designing the solution/algorithm as the solution/algorithm needs to be accurate i.e to handle all the exceptions and it should be efficient i.e good in terms of time/space complexity. |  |
| 3 | **Innovation** | Involve creative use of engineering principles and research-based knowledge in novel ways. | N/A |  |
| 4 | **Consequences to society and the environment** | Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation. | N/A |  |
| 5 | **Familiarity** | Can extend beyond previous experiences by applying principles-based approaches | This design experiment requires the students to develop a solution which is an extension to all the key concepts delivered during the semester particularly the experiments dealing with the interprocess communication. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Attributes | Description | Evaluator’s Comments |
| - | Preamble | Assignment 3 is an individual assignment based on a complex control engineering design problem |  |
| 1 | Range of conflicting requirements | Design requirements given by the client include transient and steady state specifications which are inherently conflicting in nature. |  |
| 2 | Depth of analysis required | As the plant transfer function is based on each student roll number, every assignment is different and requires original thinking from every student. The design problem has no obvious solution and requires considerable analysis, iterations and simulations. |  |
| 3 | Depth of knowledge required | Depth of control engineering knowledge is required to design and simulate the two controllers and compare them. |  |
| 4 | Familiarity of issues | Because of the diverse nature of the problem and the conflicting design specifications students would encounter infrequent issues such as insolvability of the problem or implementation issues while designing the circuit for the controller and the Plant. |  |
| 5 | Extent of applicable codes | - NA - |  |
| 6 | Extend of stakeholder involved and level of conflicting requirements | The design problem involves a client (instructor) with conflicting requirements and the control engineer (Student). |  |
| 7 | Consequences | The assignment gives the students a firsthand experience regarding difficulties in meeting conflicting client requirements and designing and implementing a control system. |  |
| 8 | Interdependence | The problem involves designing a controller and suggesting a circuit for its implementation. The circuit implements integrators and differentiators which require different components and parts depending on their nature (active or passive). |  |