



# ECE 425

## Microprocessor Systems

### Final Project Arithmetic Calculator

Shailesh Alagan  
Nandu Krishna  
Instructor: Aaron Nanas  
Fall 2024

# Introduction

This project focuses on developing an arithmetic calculator using the TM4C123GH6PM microcontroller, an external 4x4 keypad, and UART communication to display the inputted numbers and the output. The calculator is capable of performing basic arithmetic operations, including addition, subtraction, multiplication, and division, while providing the user with seamless interaction using the keypad and the TeraTerm terminal.

The 4x4 membrane keypad serves as the primary input device, including the keypad on the EduBase Board, enabling the user to enter numbers, arithmetic operators, and control commands such as clear or equals. The keypad is interfaced with the microcontroller using GPIO pins configured in a row-column scanning format to detect key presses efficiently. This code was provided by the professor. The microcontroller processes the inputs, interprets them as either operands or operators, stores them in an array, and triggers the appropriate arithmetic operation.

The UART (Universal Asynchronous Receiver/Transmitter) module is utilized to display intermediate and final results on a serial terminal, allowing real-time feedback. The UART communication ensures seamless data transmission between the microcontroller and the computer.

The project integrates keypad scanning, arithmetic logic, and UART communication to deliver a smooth and functional user experience. Input validation, and error handling (e.g., division by zero) are implemented to enhance the calculator's reliability.

This implementation highlights the use of embedded system peripherals, such as GPIO, Keypad, and UART in designing a practical application while emphasizing real-time data processing and hardware-software interaction.

## Background and Methodology

Several key embedded systems concepts were implemented in the development of this project. They are listed below.

1. GPIO (General Purpose Input/Output):
  - The external 4x4 keypad is initialized using GPIO pins configured for row-column scanning.
  - Key detection is made by reading rows and scanning columns.
2. UART Communication (Universal Asynchronous Receiver/Transmitter):

- The UART module is configured to enable serial communication between the microcontroller and the TeraTerm terminal.
  - This allows real-time display of inputs, intermediate results, and the final output.
3. Data Processing and Control Logic:
- The calculator implements arithmetic operations such as addition, subtraction, multiplication, and division.
  - Input is validated, and special cases are handled to ensure accurate and error-free calculations.
  - The system processes sequential inputs in arrays and maintains states for operands and operators.
4. Row-Column Scanning Technique:
- To detect keypresses on the 4x4 keypad, a row-column scanning algorithm is applied, provided by the professor, which efficiently identifies the pressed key.

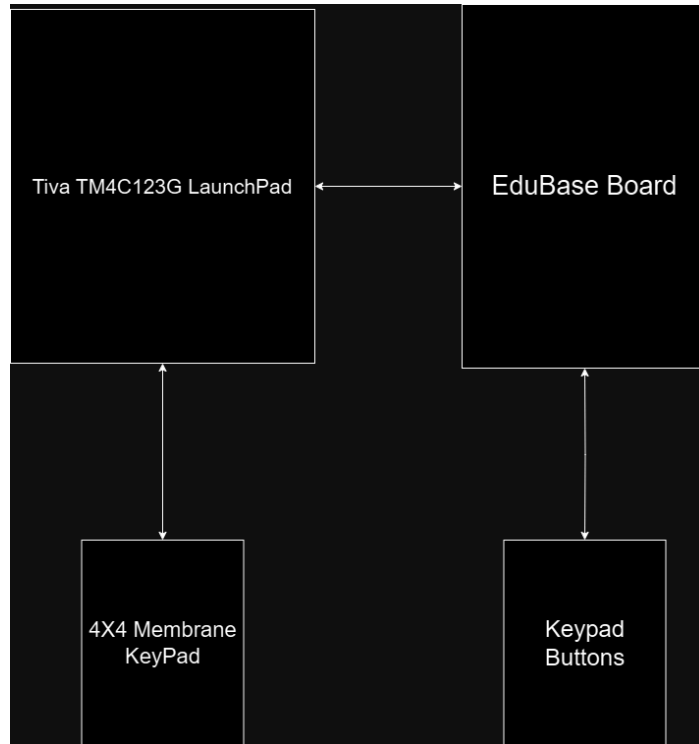
The hardware, software, and peripherals used are listed in the table below:

Hardware	Software	Peripherals
TM4C123GH6PM Microcontroller	Keil $\mu$ Vision IDE	GPIO
4x4 Keypad	Embedded C Programming	UART
USB-to-Serial Converter	TeraTerm Terminal Software	SysTick Timer
TeraTerm Terminal Software	TM4C123GH6PM Startup Files	
Power Supply (e.g., 5V via USB)		
EduBase Keypad		

The hardware and software components were combined to achieve a functional and reliable arithmetic calculator. The TM4C123GH6PM microcontroller effectively utilizes GPIO for keypad interfacing, UART for communication, and interrupts for efficient input handling to meet the project objectives.

# Block Diagram

The block diagram of this project is as follows:



# Components Used

The components are as follows:

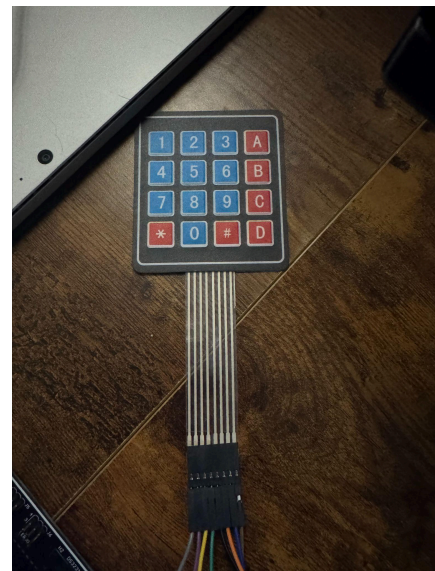
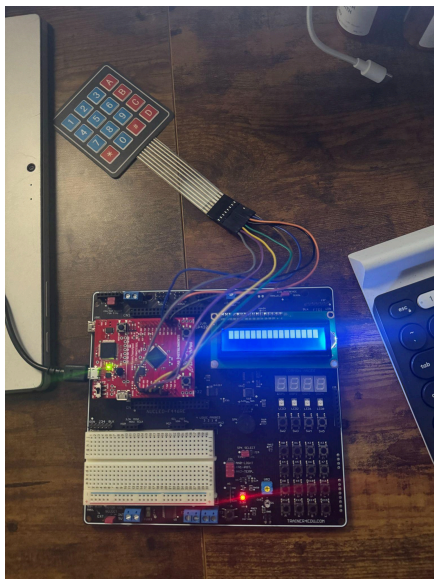
Component	Description
TM4C123GH6PM Microcontroller	The central processing unit for the project.
4x4 Membrane Keypad	External keypad for entering numbers and operators.
Serial Terminal Software (TeraTerm)	Displays inputs, operations, and results.
Keil $\mu$ Vision IDE	Software used for coding, compiling, and debugging.
Embedded C	To program the microcontroller.
GPIO	Used to detect input from the keypad and initialize other peripherals.
UART	Enables serial communication to display results to the terminal.
SysTick Timer	Used for debouncing or time delays.
5V USB Power	Provides power to microcontroller and peripherals.

## Pinout Used

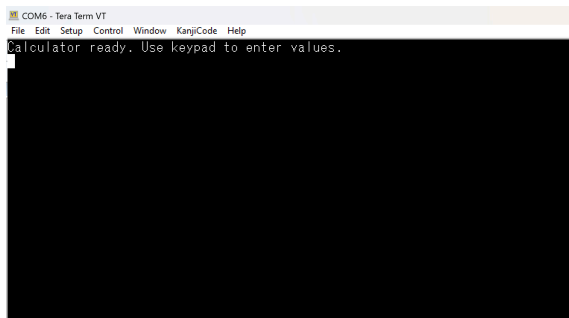
Pins	Connection
PD0	Row 1 (Keypad)
PD1	Row 2 (Keypad)
PD2	Row 3 (Keypad)
PD3	Row 4 (Keypad)
PA2	Column 1 (Keypad)
PA3	Column 2 (Keypad)
PA4	Column 3 (Keypad)
PA5	Column 4 (Keypad)
PA1	Transmit (UART)
PA0	Receive (UART)

## Analysis and Results

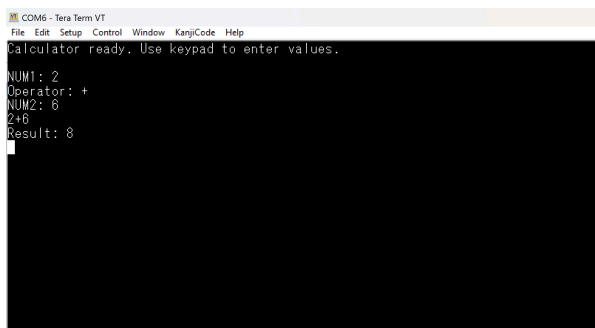
Pictures of the setup are shown below:



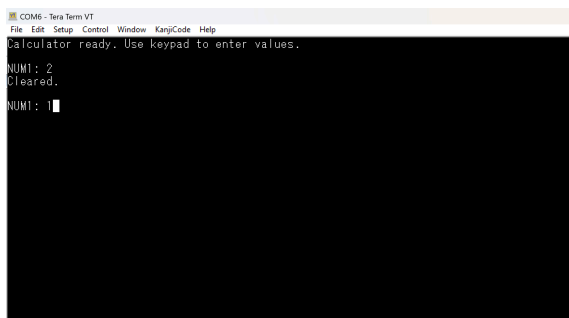
Pictures of the results from the terminal are shown below:



Starting Screen.



Final result after inputting from the keypad.



Screen Cleared.

We had a few issues while making this project. Some include the following:

- Not being able to make the external keypad fully functional. This resulted in having to use the EduBase Board keypad.
- Interrupts weren't working fully, which resulted in having to use states while detecting buttons.
- Since the keypad wasn't working fully, more operations (such as square root, exp) were not able to be made functional.
- Not being able to fully implement floating point numbers.

- Since states were used, to reset the calculator, the RESET button must be pressed on the microcontroller.

Video Links:

ECE 425 Final Project Full Implementation Video:

<https://youtu.be/W2aIZn-7rf0>

ECE 425 Final Project Clear Screen:

<https://youtu.be/OH64Lva-fgg>

## Conclusion

Through the development of this project, we gained hands-on experience and a deeper understanding of several key concepts in embedded systems. The following are the key concepts and skills learned from the project:

1. GPIO Configuration and Keypad Interfacing:
  - Learned how to configure GPIO pins for both input and output operations to interface with a 4x4 keypad. By implementing the row-column scanning technique, we efficiently detected key presses and converted them into usable data for the calculator logic.
2. UART Communication:
  - Working with the UART module from the lab provided valuable experience in configuring serial communication settings such as baud rate, stop bits, and data frame size. We also learned how to use UART to send and receive data between the microcontroller and a terminal, enabling real-time user interaction and debugging. Although we learned this in lab, doing this project further enhanced our understanding of UART.
3. Embedded C Programming and Low-Level Hardware Interaction:
  - We further enhanced our proficiency in Embedded C programming, including writing low-level code to directly configure microcontroller registers.
4. Arithmetic Logic Implementation:
  - I learned how to implement the core arithmetic operations (addition, subtraction, multiplication, and division) and manage sequential inputs to maintain operands, operators, and results.
5. Debugging and Testing:



- By using UART for output, we were able to debug the system in real-time, testing individual components and ensuring reliability. This improved our problem-solving skills and ability to troubleshoot embedded systems.

## Bibliography

- 1) Keypad Driver, Professor Nanas

[https://github.com/csun-fall24-ece-425/ECE425\\_Keypad\\_Example](https://github.com/csun-fall24-ece-425/ECE425_Keypad_Example)