AVR Calculator Application Report

Arnav Mahishi- EE24BTECH11006

March 25, 2025

Abstract

This report outlines the design and functionality of a sophisticated calculator application implemented using an AVR microcontroller and GCC. The application performs basic arithmetic operations such as addition, subtraction, multiplication, and division, as well as advanced functions such as inverse trigonometric operations, logarithms, and trigonometric calculations. The project demonstrates key concepts in embedded systems programming, including hardware interfacing, user input processing, and advanced mathematical computations on an AVR-based microcontroller.

Contents

1 Introduction

The AVR-based calculator application was developed using the AVR microcontroller platform and GCC toolchain to provide users with a versatile calculator capable of performing both basic and advanced mathematical operations. This project showcases the ability to integrate advanced mathematical functions such as inverse trigonometric functions (arcsin, arccos, arctan), trigonometric functions (sin, cos, tan), and logarithmic operations (ln, log) into an embedded system.

The purpose of this report is to provide an overview of the project, including the design choices, the implementation of various mathematical operations, the hardware setup, and the interaction between the user and the system.

2 System Requirements

To run this AVR-based calculator, the following hardware and software setup is required:

- **AVR Microcontroller:** Any AVR-based microcontroller (e.g., ATmega328P).
- **Development Environment:** GCC for AVR, AVRDUDE for programming, and a suitable text editor or IDE.
- **Input Devices:** Pushbuttons or switches for numeric and operational input.
- **Display:** 7-segment display, LCD, or similar output display for showing the result.
- **Mathematical Libraries:** Libraries for trigonometric and logarithmic functions, which are implemented using the AVR's floating-point math capabilities.

3 Features and Functionalities

The calculator offers several key features:

- **Basic Arithmetic Operations:** Addition, subtraction, multiplication, and division.
- **Advanced Trigonometric Functions:** Sin, Cos, Tan, and their inverses (arcsin, arccos, arctan).
- **Logarithmic Functions:** Natural logarithm (ln) and base-10 logarithm (log).
- **User Input:** Users enter numbers and operations via switches or buttons.
- **Output Display:** Results are displayed on a 7-segment display or an LCD screen.
- **Error Handling:** Error messages are displayed in case of invalid inputs or mathematical errors, such as division by zero or invalid operations in trigonometric calculations.

4 Code Structure Overview

The code structure for the AVR calculator is organized into several modules to handle initialization, input, calculations, and display. Key modules include:

- **Initialization:** This sets up the I/O pins for input and output, configures interrupts for button presses, and prepares the display.
- **Input Handling:** Button presses are captured and mapped to numeric values or operators, including advanced operations like trigonometric and logarithmic functions.
- **Arithmetic and Advanced Operations:** Functions to handle basic operations, as well as advanced operations like trigonometry and logarithms.
- **Display Logic:** The result is displayed on a 7-segment or LCD display. The display updates as calculations are performed.
- **Error Handling:** he system catches mathematical errors (e.g., division by zero, out-of-range values for inverse trig functions) and displays appropriate error messages.

5 Detailed Description of the Code

5.1 Initialization

The system is initialized by configuring the microcontroller's input/output ports for reading button presses and displaying the results. Timers are set up to handle input debouncing and to manage the refresh rates for the display. This initialization ensures that the system is prepared to receive inputs and perform calculations.

5.2 Input Handling

The user interacts with the calculator via physical buttons or switches, which are connected to the microcontroller's input pins. These buttons correspond to digits, operators (like +, -, *, /), and advanced mathematical functions (e.g., sin, cos, log). The program uses polling or interrupts to detect button presses and map the input to appropriate operations. The numeric values and operators are stored for further processing.

5.3 Arithmetic and Advanced Operations

The core functions of the calculator handle the following operations:

- **Basic Arithmetic Operations:** - Addition: Adds two numbers. - Subtraction: Subtracts one number from another. - Multiplication: Multiplies two numbers. - Division: Divides one number by another, with checks for division by zero.
- **Trigonometric Functions:** - Sine, cosine, and tangent functions are implemented based on the microcontroller's math libraries, providing the ability to compute these values for a given angle (in radians).
- **Inverse Trigonometric Functions:** - Arcsine (asin), arccosine (acos), and arctangent (atan) functions allow the calculation of angles based on the sine, cosine, and tangent values, respectively. Error handling ensures that input values are valid for inverse trigonometric calculations (e.g., -1 to 1 for arcsin and arccos).

- **Logarithmic Functions:** - The calculator supports natural logarithms (ln) and base-10 logarithms (log). These operations are performed using the AVR's math library, which supports floating-point operations for these advanced calculations.

5.4 Display Logic

The results of the arithmetic and advanced operations are displayed on a 7-segment display or an LCD screen. The program uses a lookup table or direct control of the display's segments to show numbers and letters (e.g., "Error" in case of invalid operations). The display is updated after every operation, and in cases of complex calculations (such as trigonometric or logarithmic results), it formats the output to fit the display constraints.

5.5 Error Handling

The system includes robust error handling to manage edge cases:

- Division by zero: The calculator checks for division by zero and displays an error message.
- Invalid values for inverse trigonometric functions: The arcsin and arccos functions handle input values outside the valid range of -1 to 1.
- Logarithm of non-positive values: The ln and log functions check for invalid input (log of 0 or negative numbers) and display an error message if needed.

6 GitHub Repository

For access to the full code and further documentation, visit the project's GitHub repository: `https://github.com/arnavmahishi/EE1003/tree/main/hardware/calculator`

7 Conclusion

This AVR calculator project demonstrates the integration of basic and advanced mathematical operations on an embedded system. By using an AVR

microcontroller, the project efficiently handles user input, performs arithmetic and trigonometric calculations, and displays results on a hardware interface. The inclusion of advanced functions like inverse trigonometry and logarithms enhances the functionality and versatility of the calculator.

8 Future Improvements

While the project is functional, several improvements could be made:

- **Graphical Output:** Using a graphical LCD to display more complex results and potentially allow for graphical representations of functions.
- **Memory Function:** Implementing memory storage to hold previous results for use in subsequent calculations.
- **Enhanced User Interface:** Adding more intuitive navigation