

Project 5: ATmega32-Based Calculator with Keypad and LCD

Introduction

This project implements a scientific calculator using the ATmega32 microcontroller. The calculator processes user input via a keypad and displays real-time calculation results on an LCD. It supports arithmetic operations respecting operator precedence, decimal numbers, and negative values. This calculator also includes a green and red LED that lights up in correspondence with the validation of a calculator's computation.

Objective

The main objective is to design a **scientific calculator** with the following features:

- Input: Accepts string input in typical calculator style, supporting *, /, +, -, integer and decimal numbers.
- Output: Computes results with at least three decimal places and displays both positive and negative numbers. Green or red LED light up, depending on whether the expression inputted resulted in a valid computation or an error.

Components used:

1. Breadboard
2. Atmel ICE programmer
3. ATmega32 microcontroller
4. 9V Battery
5. Voltage regulator
6. 0.1uf Capacitor
7. Keypad
8. LCD
9. Jumper wires
10. USB wire
11. Laptop

Methodology

Micro controller: ATmega32

Input:

- A keypad is connected to designated I/O ports of the ATmega32
- The UI allows users to enter the date and time values automatically.

Output:

- A LCD is connected to the ATmega32 using appropriate data and control lines.
- The LCD is initialized and controlled using provided lcd.h and lcd.h.

Power Supply: 9V supplied; 5V regulated

- The system is powered by a regulated 5V supply derived from a 9V battery and voltage regulator.

Process

Hardware setup:

- With the initial set up from our project 1, the power supply set up remained the same. - We moved our microcontroller to the center of the breadboard to accommodate the bigger components used for this project.
- We connected the keypad to the microcontroller ports for input.
- Then, connected the LCD display to the microcontroller.
- Connected the push button, LED lights.

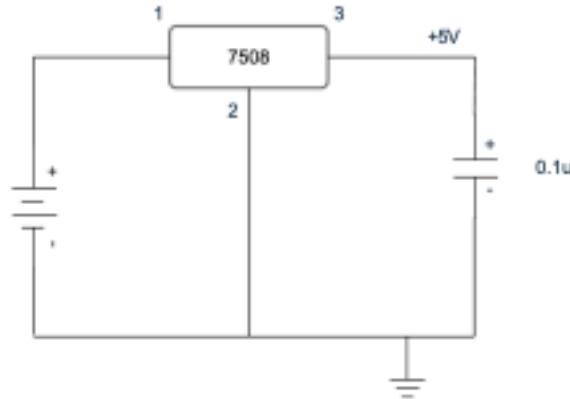
Code [all the files have been attached]

- Keypad input (keypad.c): Reads user button presses and debounces the signal.
- LCD display (lcd.c): Shows the user's input and the computed result.
Expression parsing & evaluation (calc_minimal.c): Handles arithmetic calculations, including operator precedence and decimals.
- Main control logic (main.c): Coordinates input, calculation, display, and LEDs.
- Push button (PA0): Computes the mathematical expression when pressed (acts as the “=” button).
- LED indicators (in main.c):
 - Green LED (PB3): Turns on when the answer is valid.
 - Red LED (PB4): Turns on to show errors (like dividing by zero).
- Hardware utilities (avr.h): Provides low-level functions for easy hardware control.

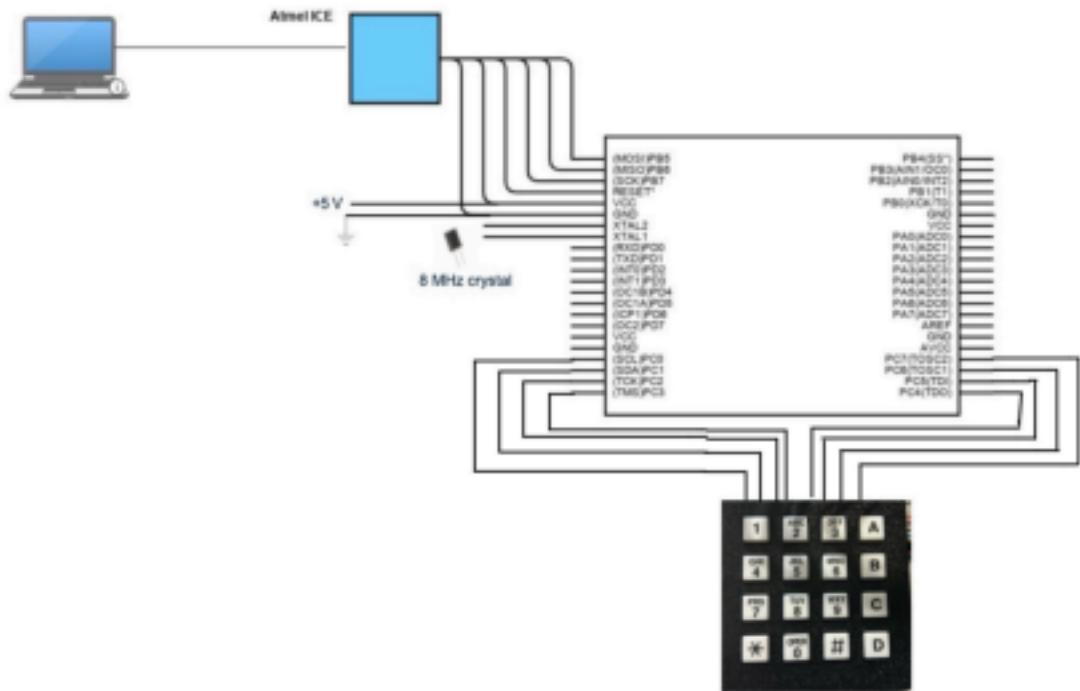
Diagram & Picture

For the schematic diagram having both keypad and LCD on the same diagram was difficult. Therefore, we made individual diagrams

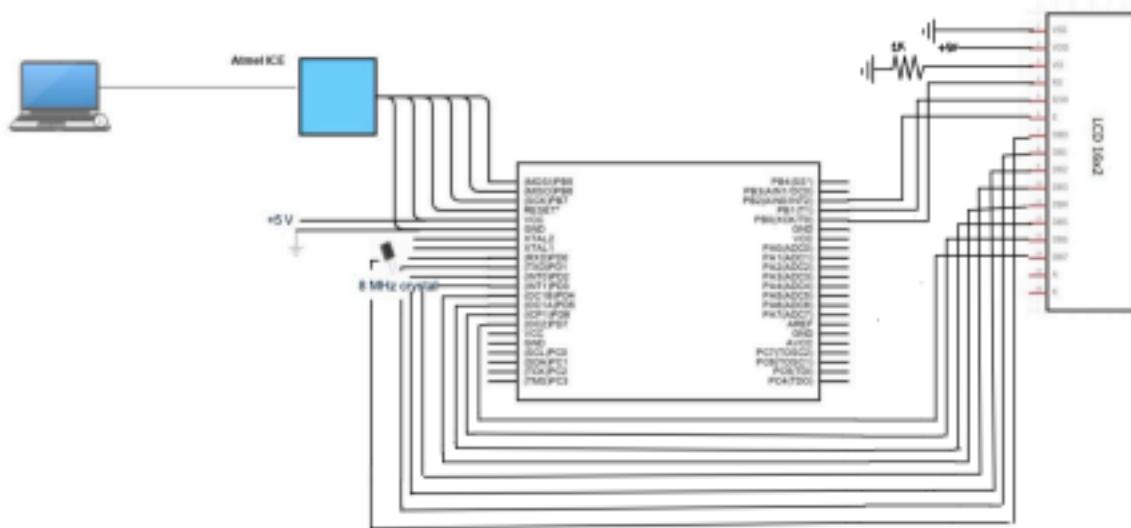
Power supply:



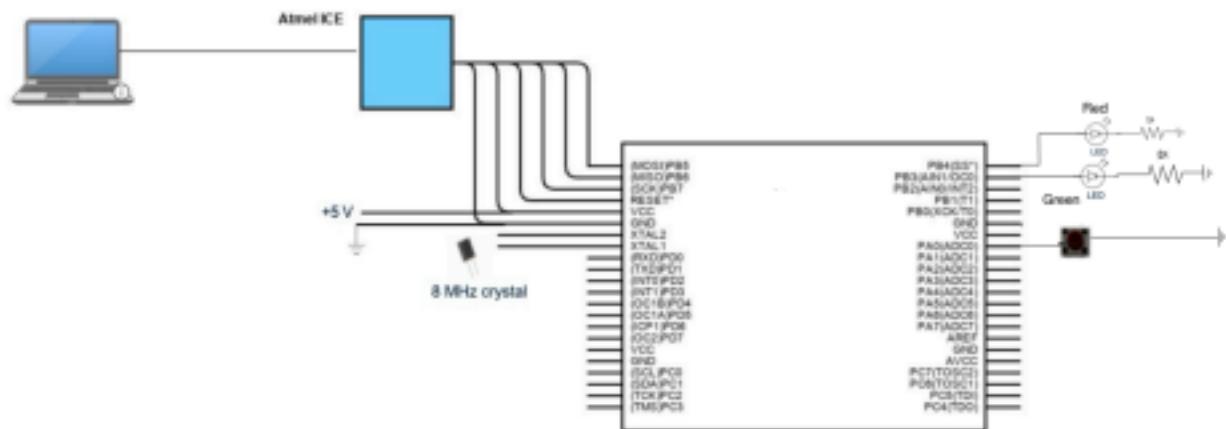
Keypad:



LCD:



Additional Schematic for push button & LED lights:



Picture(s):

