Project title : Capacitor Measurement System Using ATmega32

1. Introduction:

This project is designed to measure the capacitance of unknown capacitors using an ATmega32 microcontroller. The method used involves charging and discharging the capacitor through a known resistor and measuring the time it takes for the voltage across the capacitor to reach a certain threshold.

2. Objective:

To build a cost-effective and accurate Capacitance Meter capable of measuring a wide range of capacitor values using simple hardware and embedded software.

3. Components Used:

- ATmega32 Microcontroller
- Resistors (1ok ohm,1.8k ohm,470 ohm,220ohm)
- Capacitors (100nF,1microF,33pF,...etc)
- Analog-to-Digital Converter (ADC)
- LCD Display (for showing results)
- Push button

- Led as indicator of charging and discharging of capacitors

4. Methodology:

The principle is based on the exponential charging and discharging behavior of a capacitor in an RC (Resistor-Capacitor) circuit. The microcontroller applies a known voltage across the RC circuit and starts a timer.

5. Software Implementation:

The code is written in C using Atmel Studio for Atmega8 , here is our implementation :

- Test the code of Atmega8 using Atmel Studio
- Test the code of Atmega8 on Proteus Simulation
- Convert the code of Atmega8 into Atmega32 using Atmel Studio by changing i/o peripherals into Atmega32 schematic
- Convert the code of Atmega32 from Atmel Studio into CodeVision by removing the functions written in assembly in pure embedded C language , modified the syntax of Interrupt service routine to be compatible with CodeVision syntax also include the atmega32 library

- Test the Atmega32 code using CodeVision on Proteus Simulation

6. Results:

The system was tested with capacitors of various known values. The measured values were found to be reasonably accurate within an acceptable error margin, considering hardware limitations and environmental factors.

7. Conclusion:

The project successfully demonstrates the implementation of a digital capacitance meter using the ATmega32. It is a practical example of applying embedded systems in real-world electronics testing and measurement

---

8. Bonus:

To measure very large capacitance values, such as super capacitors (e.g., 100 Farads), the standard RC charging method needs some modifications in both hardware and software. Here's what's required:

1. Use Very Large Resistors (High R value):

Since the basic formula is: $C = t / R$

If the capacitance C is very large, the time (t) required for charging/discharging becomes very long unless you use a very large resistor (R). By increasing R, you can keep the charging time within a measurable range.

However:

- Very high resistance can make the voltage change very slow.
- It may also introduce noise and instability in measurement.
- You may need precision resistors to maintain accuracy.

2. Longer Timer Ranges / Slower Clock:

- The microcontroller's timer must be able to count long durations, possibly several seconds or minutes.
- You may need to use external timers or slow down the clock to increase time resolution over longer periods.

3. Protection Circuits:

   - Supercapacitors can draw a huge inrush current when charged suddenly.
   - You’ll need current-limiting resistors and possibly MOSFETs or relays to control the charging path safely.

4. Power Supply Considerations:

   - A 100F capacitor can store and release a significant amount of energy.
   - Make sure your power supply can handle it safely, especially during charging.

5. Software Adjustments:

   - You’ll need to allow for longer measurement timeouts.
   - The ADC should still sample the voltage periodically and wait for the threshold (e.g., 63.2% of Vcc).
   - Implement averaging or filtering to reduce noise.