Capacitance Measurement Using Arduino

Author: Fuad Hasan, BME, KUET

---

1. Introduction

This project uses an Arduino board to measure the capacitance value of an unknown capacitor by calculating the charging time through a known resistor. It prints the result on the Serial Monitor either in microfarads (\u00b5F) or nanofarads (nF).

2. Key Components

- Arduino UNO (or similar)
- Capacitor (to be measured)
- Resistor (10k\u03a9)
- Breadboard and jumper wires
- USB Cable (for Arduino programming and serial monitoring)

3. Working Principle

1. **Initialization**:
   The Arduino initializes pins for charging, discharging, and analog reading.
2. **Charging Phase**:
   Arduino sets the charging pin HIGH, starting to charge the capacitor through the known resistor.
3. **Timing**:
   A timer starts, and the code continuously reads the analog voltage across the capacitor until it reaches about 63.2% of the supply voltage (approx. 610 in 10-bit ADC scale for 5V).
4. **Calculating Capacitance**:
   Based on the time taken to reach 63.2% charge and the known resistor value, capacitance is calculated using the formula:

C(μF)=(Time(ms)Resistance(Ω))×1000C (\mu F) = \left(\frac{Time (ms)}{Resistance (\Omega)}\right) \times 1000

5. **Displaying Results**:
   The calculated capacitance is displayed on the Serial Monitor in either microfarads or nanofarads.

6. **Discharging Phase**:
   Arduino stops charging, enables a discharge pin to quickly drain the capacitor, and waits until the capacitor voltage drops to zero before repeating.

4. Circuit Overview

- One end of the resistor is connected to 5V.
- The other end connects to one lead of the capacitor and to the analog input pin (A0).
- The other lead of the capacitor goes to Ground (GND).
- A discharge pin (Pin 9) is connected across the capacitor to speed up discharge.

5. Code

```
//Code written by -
//Fuad Hasan
//BME, KUET

//Initialize Pins
int analogPin = 0;
int chargePin = 8;
int dischargePin = 9; //speeds up discharging process, not necessary though

//Initialize resistor
int resistorValue = 10000;

//Initialize Timer
unsigned long startTime;
unsigned long elapsedTime;
```

```
//Initialize Capacitance Variables
float microFarads;
float nanoFarads;

void setup()
{
  pinMode(chargePin, OUTPUT);
  digitalWrite(chargePin, LOW);
  Serial.begin(9600);   //Necessary to print data to serial monitor over USB


}

void loop()
{
  digitalWrite(chargePin, HIGH);    //Begins charging the capacitor
  startTime = millis();     //Begins the timer

  while(analogRead(analogPin) < 610) // 648;
 {
   //Does nothing until capacitor reaches 63.25 of total voltage
 }

  unsigned long nowTime = millis();
  elapsedTime = nowTime - startTime;    //Determines how much time it took to charge capacitor
```

```
microFarads = ((float)elapsedTime / resistorValue) * 1000;
Serial.print(elapsedTime);
Serial.print(" mS   ");

if(microFarads >1)    //Determines if units should be micro or nano and prints accordingly
{
  Serial.print((long)microFarads);
  Serial.println(" microFarads");
}

else
{
  nanoFarads = microFarads * 1000.0;
  Serial.print((long)nanoFarads);
  Serial.println(" nanoFarads");
  delay(500);
}

digitalWrite(chargePin, LOW);   //Stops charging capacitor
pinMode(dischargePin,OUTPUT);
digitalWrite(dischargePin, LOW);    //Allows capacitor to discharge
while(analogRead(analogPin) > 0)
{
  //Do nothing until capacitor is discharged
}
```

```
 pinMode(dischargePin, INPUT);     //Prevents capacitor from discharging
}
```

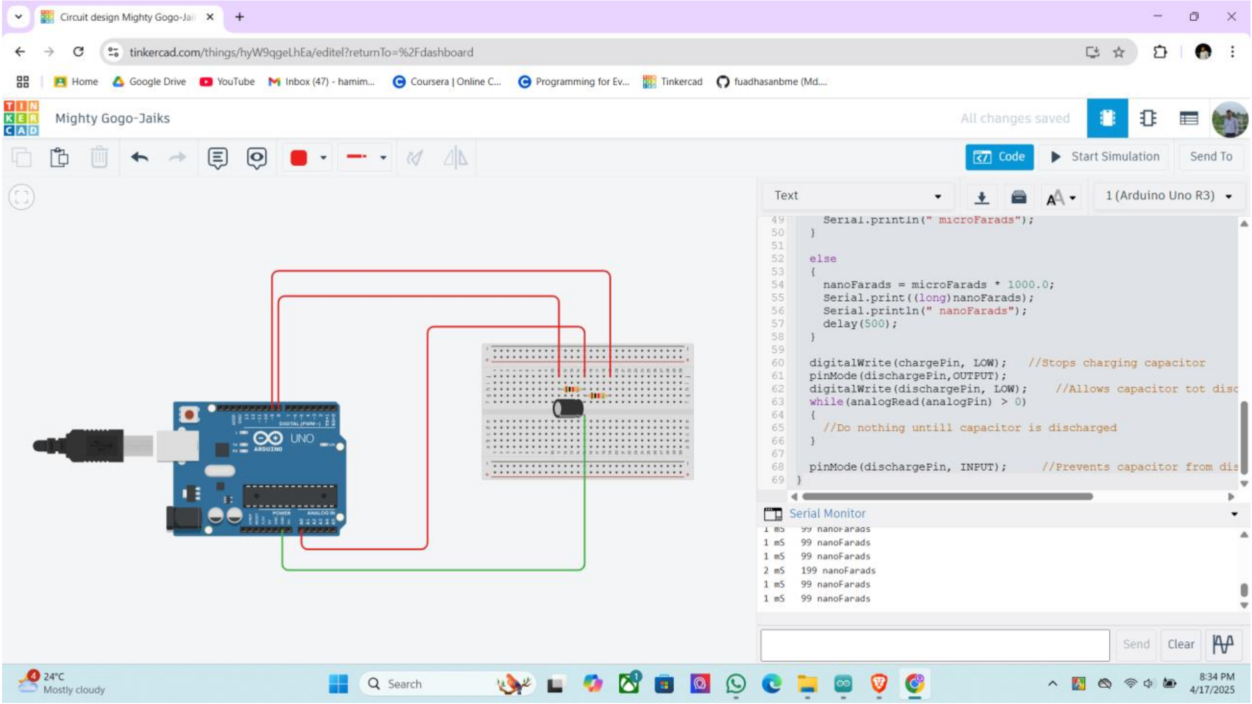
6. Code Explanation

- **Pin Initialization**:
  - analogPin = 0 -> Reads capacitor voltage.
  - chargePin = 8 -> Used to charge the capacitor.
  - dischargePin = 9 -> Used to quickly discharge the capacitor.
- **Resistor Value**:
  - 10k\u03a9 resistor used to control charging current.
- **Timer Variables**:
  - startTime and elapsedTime used to measure how long it takes to charge the capacitor.
- **Capacitance Calculation**:
  - Time is divided by resistance and scaled to get capacitance.
- **Charging the Capacitor**:
  - digitalWrite(chargePin, HIGH) starts charging.
  - while(analogRead(analogPin) < 610) waits until 63.2% charge.
- **Measuring and Printing**:
  - Depending on magnitude, prints either in microfarads or nanofarads.
- **Discharging the Capacitor**:
  - digitalWrite(dischargePin, LOW) discharges capacitor.
  - Waits until capacitor voltage is almost zero.

7. Conclusion

This simple Arduino-based capacitance meter provides an effective and low-cost method for measuring unknown capacitors using basic electrical principles. It offers a great introduction to both electronics and Arduino programming.

End of Document



```
    Serial.println(" microFarads");
  }

  else
  {
    nanoFarads = microFarads * 1000.0;
    Serial.print((long)nanoFarads);
    Serial.println(" nanoFarads");
    delay(500);
  }

  digitalWrite(chargePin, LOW);   //Stops charging capacitor
  pinMode(dischargePin,OUTPUT);
  digitalWrite(dischargePin, LOW);    //Allows capacitor tot disc
  while(analogRead(analogPin) > 0)
  {
    //Do nothing untill capacitor is discharged
  }

  pinMode(dischargePin, INPUT);     //Prevents capacitor from dis
}
```

```
1 mS   99 nanoFarads
1 mS   99 nanoFarads
1 mS   99 nanoFarads
2 mS   199 nanoFarads
1 mS   99 nanoFarads
1 mS   99 nanoFarads
```