

Capacitance Meter Using Arduino

This project focuses on **measuring capacitance using an Arduino**. A capacitance meter is useful for checking unknown capacitors or verifying their values. The Arduino measures capacitance by analyzing the time it takes to charge or discharge a capacitor.

1. How the Circuit Works

Basic Principle (RC Time Constant)

The capacitance (C) is determined using the **RC charging equation**:

$$V = V_{\max} \times (1 - e^{-t/RC})$$

where:

- **V** = voltage across the capacitor
- **V_{max}** = supply voltage
- **t** = time taken to reach a certain voltage
- **R** = known resistor value
- **C** = capacitance

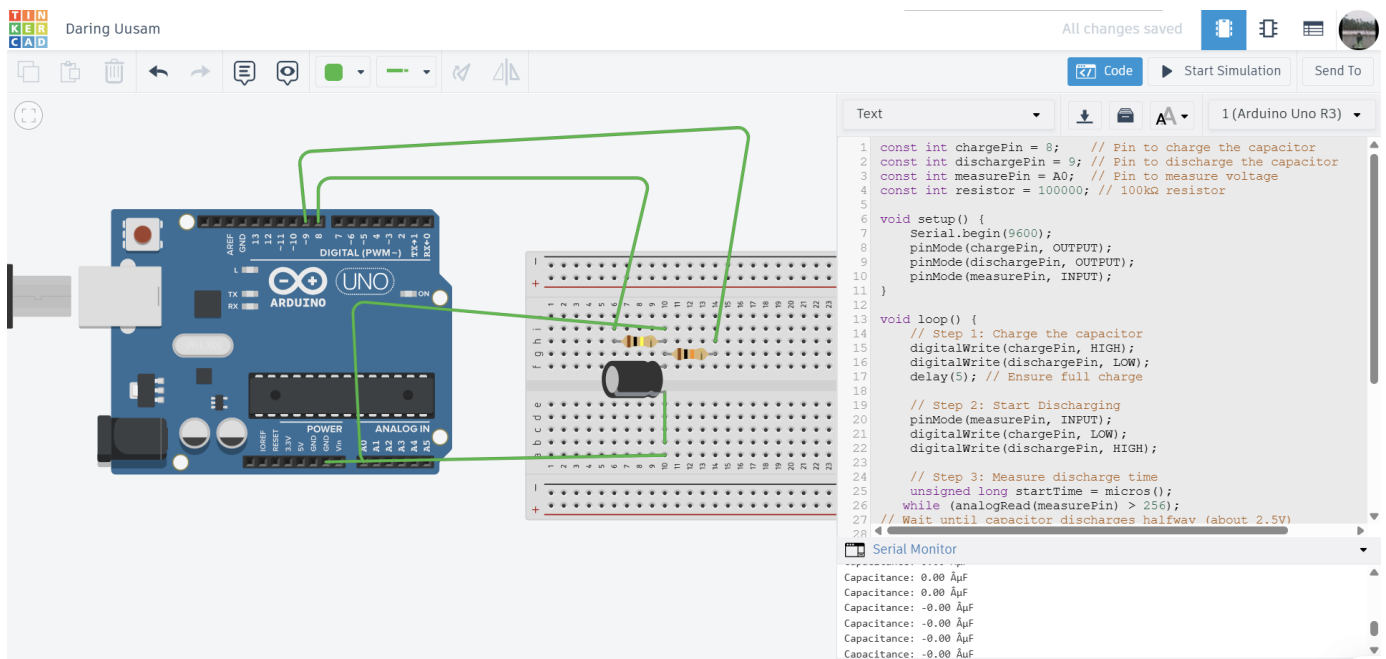
Rearranging the equation to solve for C:

$$C = t/R$$

where **t** is measured using the Arduino's **millis()** or **micros()** function.

2. Required Components

- **Arduino (Uno, Nano, or Mega)**
- **Resistor (known value, e.g., 10kΩ)**
- **Capacitor (to be measured)**
- **Breadboard & Jumper Wires**



3. Circuit Connections

1. Connect **one end of the capacitor** to **GND**.
2. Connect **the other end** to a **digital pin (e.g., D7)** and a **resistor (R)**.
3. The **other end of the resistor** is connected to **5V**.

Circuit Diagram

- **Arduino first sets the pin HIGH**, charging the capacitor.
- Then, it **switches to input mode** and measures how long it takes to discharge.

4. Arduino Code for Capacitance Measurement

```
const int chargePin = 8; // Pin to charge the capacitor
const int dischargePin = 9; // Pin to discharge the capacitor
const int measurePin = A0; // Pin to measure voltage
const int resistor = 10000; // 10kΩ resistor

void setup() {
    Serial.begin(9600);
    pinMode(chargePin, OUTPUT);
    pinMode(dischargePin, OUTPUT);
    pinMode(measurePin, INPUT);}

void loop() {
    // Step 1: Charge the capacitor
    digitalWrite(chargePin, HIGH);
    digitalWrite(dischargePin, LOW);
    delay(5); // Ensure full charge

    // Step 2: Start Discharging
    pinMode(measurePin, INPUT);
    digitalWrite(chargePin, LOW);
    digitalWrite(dischargePin, HIGH);

    // Step 3: Measure discharge time
    unsigned long startTime = micros();
    while (analogRead(measurePin) > 512); // Wait until capacitor discharges halfway (about 2.5V)
    unsigned long elapsedTime = micros() - startTime;

    // Step 4: Calculate Capacitance
    float capacitance = (float)elapsedTime / resistor; // Formula C = t / R
```

```
Serial.print("Capacitance: ");  
Serial.print(capacitance);  
Serial.println(" μF");  
// Step 5: Reset and repeat  
digitalWrite(dischargePin, LOW);  
delay(1000);}
```

5. Alternative Methods

Using an External Circuit (555 Timer)

- A **555 Timer** can be used in **astable mode** to generate a frequency.
- The capacitance can be derived from the frequency.

Using Arduino's Analog Pin (Direct ADC Measurement)

- Charge the capacitor using a known voltage.
- Use **analogRead()** to measure the voltage drop over time.

6. Accuracy and Limitations

- Works best for **capacitors > 100pF**.
- For **small capacitances**, parasitic capacitance affects accuracy.
- Use a **precision resistor** for better results.