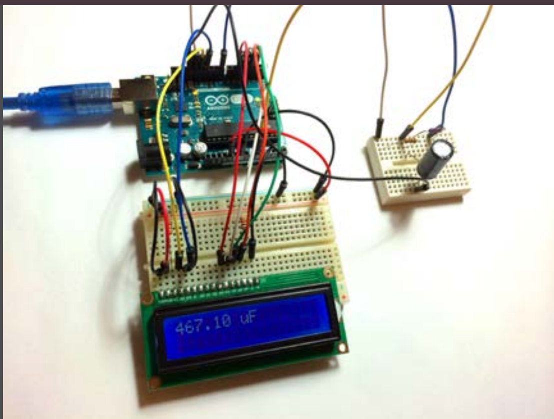
Circuit Basics

Raspberry Pi, Arduino, and DIY Electronics Tutorials

QUICK START GUIDE

How to Make an Arduino Capacitance Meter



SUMMARY

Unfortunately, I wasn't able to find a single Arduino capacitance meter circuit that could accurately measure the full range of commonly used capacitors. However, I found three different circuits that together accurately measure capacitance in the 10 pF to 3,900 μF range. The circuit diagrams and Arduino code for each circuit are provided below.

Each of these capacitance meters rely on a basic property of RC circuits- the time constant. The time constant is defined as the time it takes the voltage across the capacitor in an RC circuit to reach 63.2% of its voltage when fully charged:

TC = R x C

TC: time constant of the capacitor (in seconds)

R: resistance of the circuit (in Ohms)

C: capacitance of the capacitor (in Farads)

This formula can be rearranged to solve for capacitance:

C = TC / R

Each capacitance meter has an RC circuit with known resistor values and an unknown capacitor value. The Arduino will measure the voltage at the capacitor and record the time it takes to reach 63.2% of it's voltage when fully charged (the time constant). Since the resistance value is already known, we can use the formula above in a program that will calculate the unknown capacitance.

MATERIALS

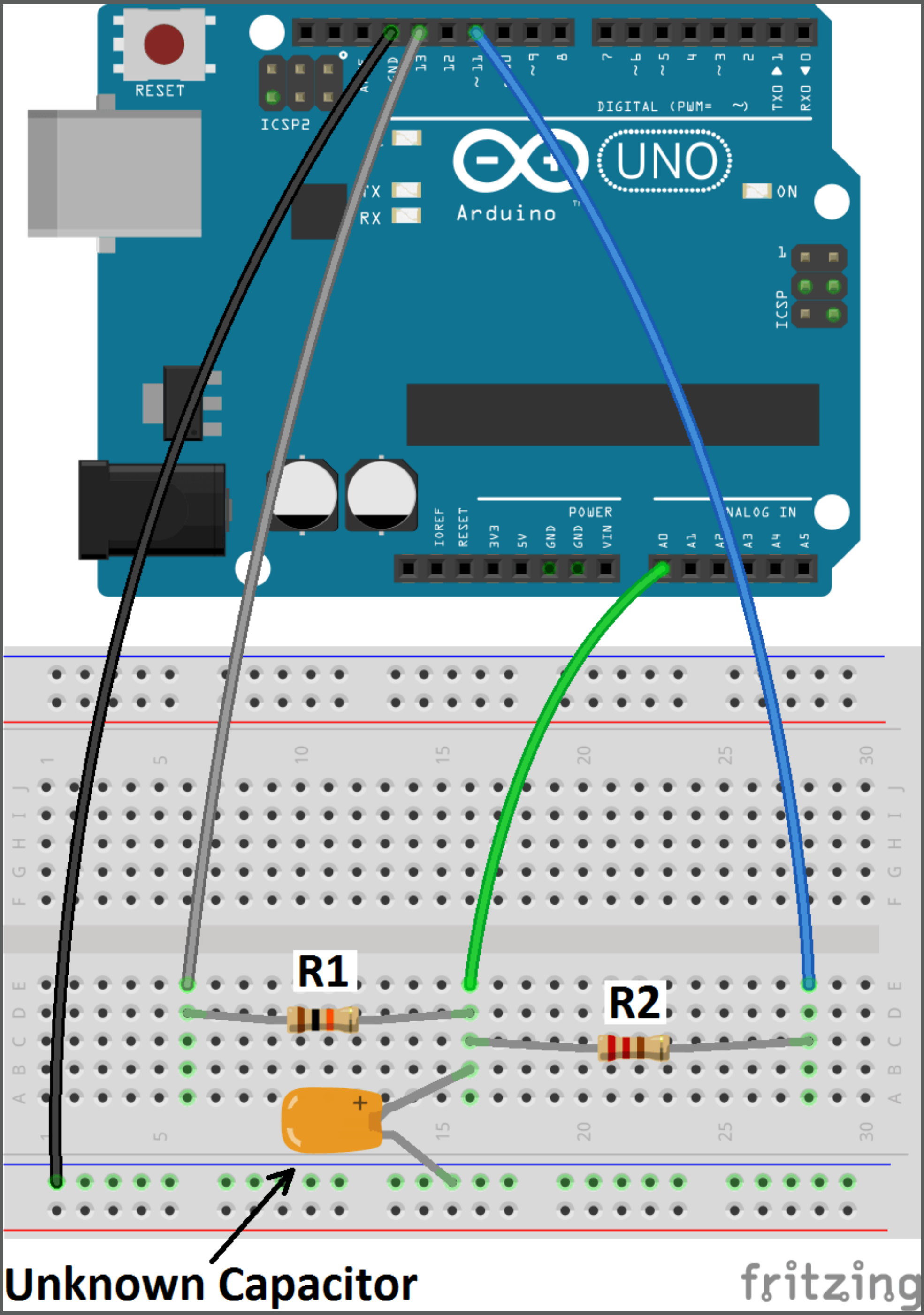
- Arduino Uno
- Breadboard
- 16X2 LCD (Optional)
- Jumper Wires
- 10K Ω, 220 Ω, 3.1K Ω, 1.8K Ω Resistors

RESOURCES

- Arduino Forums
- Circuit Basics

3900 µF to 1 µF Capacitance Meter

Follow this diagram to set up this capacitance meter:



R1 = 10K Ohms

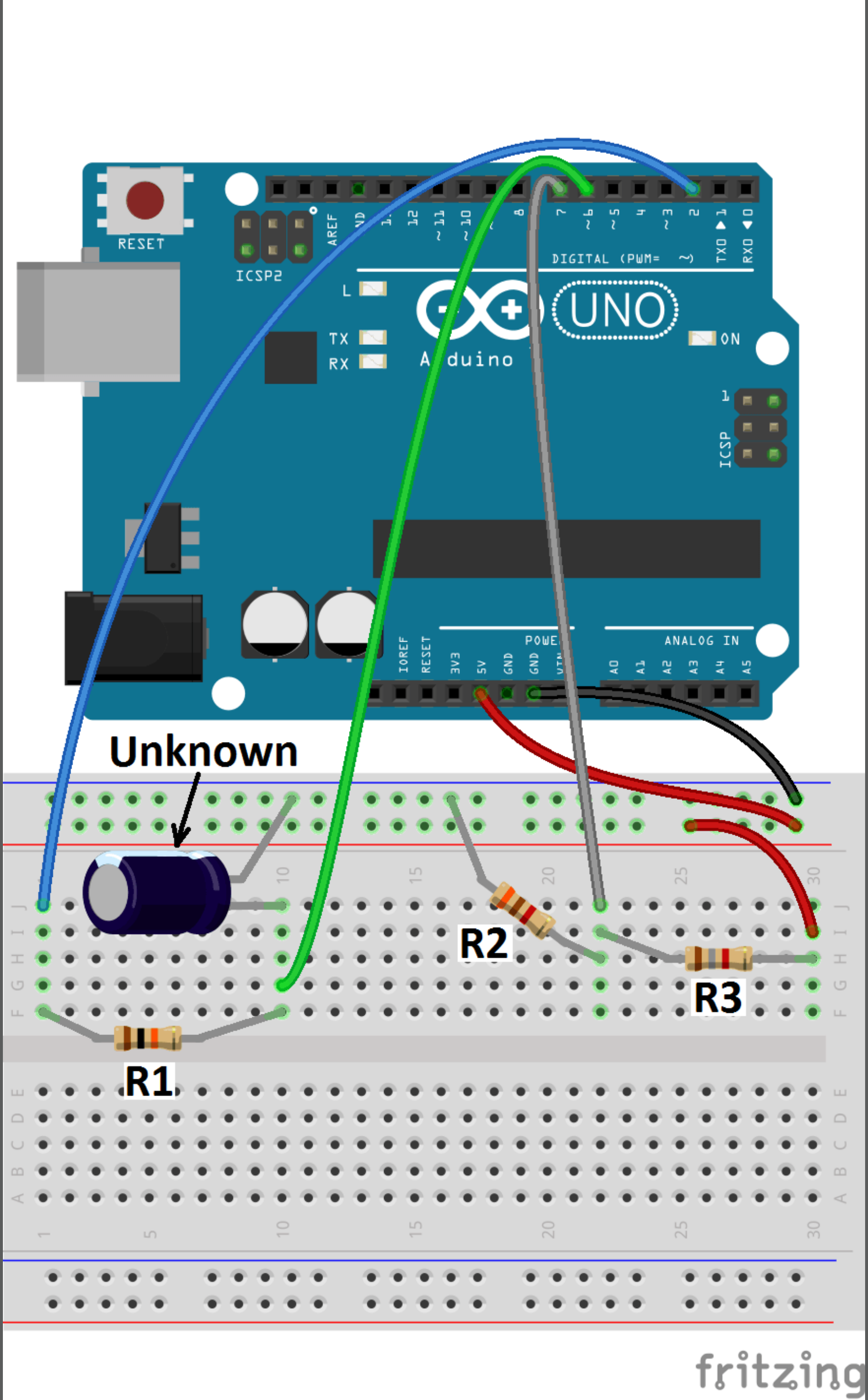
R2 = 220 Ohms

Upload this code to the Arduino, then open the serial monitor:

```
#define analogPin      0
#define chargePin      13
#define dischargePin   11
#define resistorValue  10000.0F

unsigned long startTime;
unsigned long elapsedTime;
float microFarads;
float nanoFarads;

void setup(){
  pinMode(chargePin, OUTPUT);
  digitalWrite(chargePin, LOW);
  Serial.begin(9600);
}

void loop(){
  digitalWrite(chargePin, HIGH);
  startTime = millis();
  while(analogRead(analogPin) < 648){
  }

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

  if (microFarads > 1){
    Serial.print((long)microFarads);
    Serial.println(" microFarads");
  }

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

  digitalWrite(chargePin, LOW);
  pinMode(dischargePin, OUTPUT);
  digitalWrite(dischargePin, LOW);
  while(analogRead(analogPin) > 0){
  }

  pinMode(dischargePin, INPUT);
}
```

180 µF to 0.0047 µF Capacitance Meter

Wire up the circuit like this:



R1= 10K Ohm

R2= 3.1K Ohm

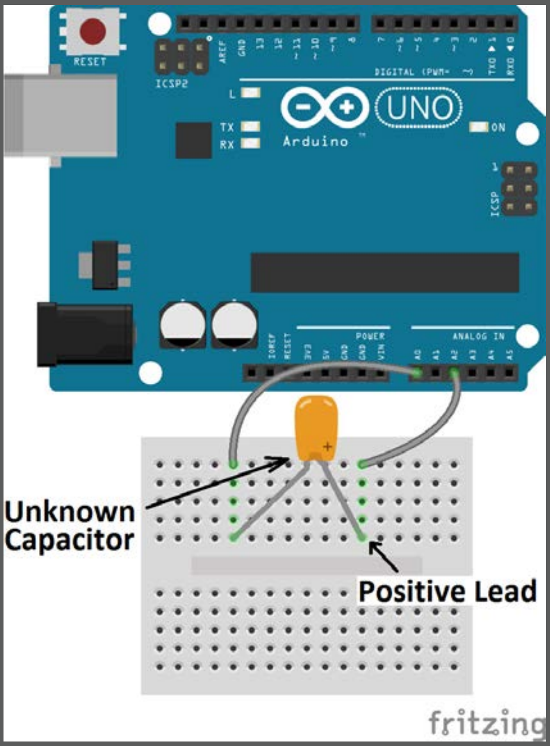
R3= 1.8K Ohm

Upload this code to the Arduino and open the serial monitor:

```
const byte pulsePin = 2;
const unsigned long resistance = 10000;

volatile boolean triggered;
volatile boolean active;
volatile unsigned long startTime;
volatile unsigned long duration;

ISR (ANALOG_COMP_vect)
  {
  unsigned long now = micros ();
  if (active)
    {
    duration = now - startTime;
    triggered = true;
    digitalWrite (pulsePin, LOW);
    }
  }

void setup ()
  {
  pinMode(pulsePin, OUTPUT);
  digitalWrite(pulsePin, LOW);
  Serial.begin(9600);
  Serial.println("Started.");
  ADCSRB = 0;
  ACSR =  _BV (ACI)
        | _BV (ACIE)
        | _BV (ACIS0) | _BV (ACIS1);
   }

void loop ()
  {
  if (!active)
    {
    active = true;
    triggered = false;
    digitalWrite (pulsePin, HIGH);
    startTime = micros ();
    }

  if (active && triggered)
    {
    active = false;
    Serial.print ("Capacitance = ");
    Serial.print (duration * 1000 / resistance);
    Serial.println (" nF");
    triggered = false;
    delay (3000);
    }
}
```

470 µF to 18 pF Capacitance Meter

Wire up the circuit like this:



Upload this code to the Arduino, then open the serial monitor:

```
const int OUT_PIN = A2;
const int IN_PIN = A0;
const float IN_STRAY_CAP_TO_GND = 24.48;
const float IN_CAP_TO_GND  = IN_STRAY_CAP_TO_GND;
const float R_PULLUP = 34.8;
const int MAX_ADC_VALUE = 1023;

void setup()
{
  pinMode(OUT_PIN, OUTPUT);
  pinMode(IN_PIN, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
    pinMode(IN_PIN, INPUT);
    digitalWrite(OUT_PIN, HIGH);
    int val = analogRead(IN_PIN);
    digitalWrite(OUT_PIN, LOW);
    if (val < 1000)
    {
      pinMode(IN_PIN, OUTPUT);

      float capacitance = (float)val * IN_CAP_TO_GND / (float)(MAX_ADC_VALUE -
val);

      Serial.print(F("Capacitance Value = "));
      Serial.print(capacitance, 3);
      Serial.print(F(" pF ("));
      Serial.print(val);
      Serial.println(F(") "));
    }
    else
    {
      pinMode(IN_PIN, OUTPUT);
      delay(1);
      pinMode(OUT_PIN, INPUT_PULLUP);
      unsigned long u1 = micros();
      unsigned long t;
      int digVal;

      do
      {
        digVal = digitalRead(OUT_PIN);
        unsigned long u2 = micros();
        t = u2 > u1 ? u2 - u1 : u1 - u2;
      } while ((digVal < 1) && (t < 400000L));

      pinMode(OUT_PIN, INPUT);
      val = analogRead(OUT_PIN);
      digitalWrite(IN_PIN, HIGH);
      int dischargeTime = (int)(t / 1000L) * 5;
      delay(dischargeTime);
      pinMode(OUT_PIN, OUTPUT);
      digitalWrite(OUT_PIN, LOW);
      digitalWrite(IN_PIN, LOW);

      float capacitance = -(float)t / R_PULLUP
                              / log(1.0 - (float)val / (float)MAX_ADC_VALUE);

      Serial.print(F("Capacitance Value = "));
      if (capacitance > 1000.0)
      {
        Serial.print(capacitance / 1000.0, 2);
        Serial.print(F(" uF"));
      }
      else
      {
        Serial.print(capacitance, 2);
        Serial.print(F(" nF"));
      }
      Serial.print(F(" ("));
      Serial.print(digVal == 1 ? F("Normal") : F("HighVal"));
      Serial.print(F(", t= "));
      Serial.print(t);
      Serial.print(F(" us, ADC= "));
      Serial.print(val);
      Serial.println(F(")"));
    }
    while (millis() % 1000 != 0)
      ;
}
```