ELECTRICAL ENGINEERING DEPARTMENT

EE143 California Polytechnic State University Lab #6

Arduino Ohmmeter

VIDEOS:

A link to Jeffrey Blum's playlist of Arduino Tutorials
https://www.youtube.com/watch?v=fCxzA9_kg6s&list=PLA567CE235D39FA84

Links to videos on how to use Tinkercad.com circuits; blinking LEDs with an Arduino
https://youtu.be/yyG0koj9nNY **https://youtu.be/MojSo7OtF9w**

PRELAB:

Use the Tinkercad circuit simulator to design an Arduino based circuit that blinks a RED LED once then blinks a GREEN LED twice and then blinks a BLUE LED three times and repeats indefinitely.

Only required to have a successful simulation.

PURPOSE:

- To use an Arduino in a practical application such as an ohmmeter.
- To gain further practice bread boarding circuits, this time circuity peripheral to an Arduino
- To gain further practice in modifying Arduino code.

This experiment relates to the following **course learning objectives**:

1. Ability to build a complete hardware / software Arduino system.
2. Acquire practice in recording data and results.
3. Ability to analyze and evaluate data.

STUDENT PROVIDED EQUIPMENT:

1 Arduino
1 Breadboard
1 10kΩ potentiometer
1 10kΩ ¼ Watt ±5% Resistor
1 4.3kΩ ¼ Watt ±1% Resistor
1 Push button switch

EXPERIMENTAL SECTIONS:

1) Arduino Ohmmeter

BACKGROUND:

Consult the Arduino web site links below for topics of interest to you.

What is Arduino? https://www.arduino.cc/en/Guide/Introduction

Arduino software (IDE) https://www.arduino.cc/en/Guide/Environment

Arduino libraries https://www.arduino.cc/en/Guide/Libraries

Arduino Ohmmeter

Arduino Troubleshooting https://www.arduino.cc/en/Guide/Troubleshooting

Switch Bounce

When a switch changes position, as when a push button switch is pressed, the contact is not made instantly. Instead, the contact is made over a small interval of time; the switch makes contact and then breaks contact repeatedly until it makes full contact. This make contact and break contact phenomenon called switch bouncing can cause unreliable data.

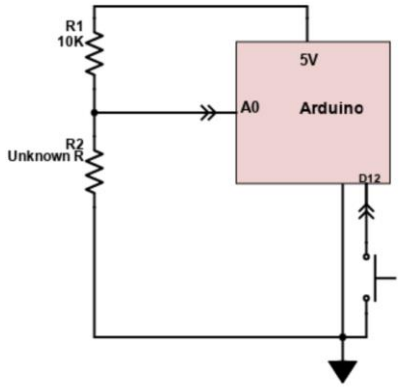
Fortunately, there is a "de-bouncing" library available for an Arduino to mitigate switch bouncing; this library is included in the Arduino ohmmeter code you will use.

PROCEDURE:

Section1) Arduino Ohmmeter

a) Connect an Arduino to a computer and use the Tools menu to select the appropriate board and COMport.

b) Build the circuit shown. Use a 10kΩ potentiometer for R2 (the Unknown R). Connect the potentiometer as a rheostat by connecting the middle lead (wiper contact) and one outer lead to ground (downward arrow symbol) and the other outer lead to A0 and R1.



c) Upload code below to Arduino.

```
#include <Bounce2.h> // Debounce switch to mitigate spurious readings due to mechanical vibrations


const float Vin            = 5.0;       //should start with Vin = 5.0
const float R1             = 10000;     //should start with R1 = 10K
const int   CIRCUIT        = 0;         //using analog input A0
const int   BUTTON         = 12;        //Button to ground on pin 12
const int   READ_DELAY_mS  = 1;         //delay between successive reads
const float VOLTS_PER_COUNT = Vin/1023.0; //4.89 mV / ADC count, assuming 5V reference
```

Arduino Ohmmeter

```
Bounce Button = Bounce();   //Button object from Bounce library

float Vout;                 //voltage out of voltage divier, read by ADC
char  VoutString[100];      //string representation of Vout

float R2;                   //"unknown" resistance
int   ADCvalue;             // value read from ADC

void setup()
   {
   // serial monitor used at 115200 bps
   Serial.begin(115200);

   // trigger button
   pinMode(BUTTON, INPUT_PULLUP);

   // using the Bounce2 library to debounce the button
   Button.attach(BUTTON,INPUT_PULLUP);
   Button.interval(25);

   // starting salutation
   Serial.println("Arduino Ohm Meter Started.\nPush button to initiate reading.\n");
   }

void loop()
   {
   // Is the button pushed?
```

Arduino Ohmmeter

```
Button.update();
bool ButtonPushed = !Button.read(); // invert logic to true = pushed

// if the button is pushed, generate a reading
if (ButtonPushed)
   {
   //read the ADC four times to get a stable reading
   for(int i=0;i<4;i++)
      {
      ADCvalue = analogRead(CIRCUIT);
      delay(READ_DELAY_mS);
      }

   //calculate the voltage that was read
   Vout = (float)ADCvalue * VOLTS_PER_COUNT;

   // calculate the resistance
   R2 = (Vout*R1)/(Vin-Vout);

   // print the results to the serial monitor
   Serial.print("ADC Value: ");
   Serial.print(ADCvalue);

   Serial.print("   Vout: ");
   dtostrf(Vout, 1, 4, VoutString);
   Serial.print(VoutString);
   Serial.print("V");
```

Arduino Ohmmeter

```
    Serial.print("   Measured Resistance: ");
    Serial.print(R2);
    Serial.println(" ohms");


    // wait for the button to be released before proceeding
    while (ButtonPushed)
       {
       Button.update();
       ButtonPushed = !Button.read();
       }
    }
  }
```

d) Adjust potentiometer and press button to view various resistances.

   Try potentiometer at the two extreme positions (all the way clockwise and all the way counter-clockwise) and try some intermediate positions; half way, quarter turn clockwise, etc.

e) Explain how the Arduino arrives at the R2 value displayed on the serial monitor.

f) Measure 4.3kΩ ±1% resistor with DMM.

Record measured value here _____

g) Replace potentiometer with 4.3kΩ ±1% resistor.

h) Measure 4.3kΩ ±1% resistor with Arduino Ohmmeter.

i) Record measured value here _____

j) Compare Arduino Ohmmeter resistance to DMM resistance (express as % error).

   % error =

k) How could system accuracy be improved?

   Hint: Is the Arduino 5V supply really 5.0V? Is R1 really 10.0kΩ?

l) Modify source code to improve system accuracy and upload to Arduino.

m) Measure 4.3kΩ ±1% resistor with improved (?) Arduino Ohmmeter.

   Record measured value here _____

Arduino Ohmmeter

n) Compare improved (?) Arduino Ohmmeter resistance to Arduino Ohmmeter resistance in step i) (express as % error).

   % error =

o) Add the following feature to the Arduino ohmmeter, when a short is the unknown R an LED is lit. Use a red LED, same LED used in the continuity-tester earlier this qtr.

p) Optional, if you have a second LED, use this LED to indicate when an open (no connection) is the unknown R.

DISCUSSION:

Section 1

1) Comment on performance of Arduino Ohmmeter improved in step n) compared to step i).

2) Insert modified code for step 0)