**Measuring the I-V characteristic of a device using an Arduino, MATLAB and an externally triggered function generator**

There are many ways to measure the current-voltage characteristics of electrical devices in the laboratory. We have measured one point at a time using multimeters and a dc power supply; and have also used a function generator and an oscilloscope in XY mode. Using the Arduino to facilitate the I-V measurement is a good way to reduce the tedium, increase the accuracy, and show our students another way to employ the Arduino.

The Arduino was programmed through MATLAB (<http://www.mathworks.com/hardware-support/arduino-matlab.html>) since we have it available and the data is easily plotted and saved. There are several other ways to store the data and later plot it. One way is to use the Adafruit data logging shield for Arduino (<http://www.adafruit.com/products/1141> ) and an open-source plotting utility to plot the data stored on the SD card. Another way is to capture the data using a serial port data capture utility like RealTerm (<http://realterm.sourceforge.net/>), save the data and then plot it.

The figure below is a schematic diagram of a setup used to measure the I-V characteristic of a diode.



The figure is a measured raw data plot much like that created as the points are measured.

To plot the diode’s current-voltage characteristic from the raw data, the student must calculate the voltage across the diode and the current through it. To determine the voltage across the diode, the student must subtract the voltage across the sampling resistor (Arduino analog 1) from the voltage across the function generator (Arduino analog 0). To calculate the current through the diode, the student divides the voltage across the resistor (Arduino analog 1) by its resistance. This is easily accomplished using either MATLAB or a spreadsheet.

The MATLAB script we used is provided below. (IV\_meas.m)

%% This is IV\_meas

% The goal of this script is to retrieve two analog voltage inputs

% from the Arduino, convert the digital word to Volts then

% plot and save the data in a file. A button will determine start time.

%

% This is for measurement of current voltage characteristics

%

% Settings on the function generator: 0 to 4.5V ramp in 10 seconds

% triggered externally

%

% Arduino analog 0 is tied to the voltage across both the device and the

% current sampling resistor (function gereator output)

%

% Arduino analog 1 is tied to the node between the device and the current

% sampling resistor (measuring voltage across sampling resistor)

%

% Arduino pin 13 is connected through a 300 ohm resistor to +5V and

% through a pushbutton switch (PBNO) to ground. When the switch is depressed

% pin 13 reads 0 and the Arduino triggers the function generator.

%

% Arduino pin 11 is tied to the external trigger of the function generator.

% Steve Weis, TCU Dept of Engineering

clear all;

%% create arduino object and connect to board

a=arduino('COM5');

%% setup

% specify pin mode for pin 13 that will be used for start button

a.pinMode(13,'input');

% specify pin mode for pin 11 that will be used to trigger fn generator

a.pinMode(11,'output');

%insure that the output is not floating

a.digitalWrite(11,0);

numpts=300; %number of points measured

xydata=zeros([numpts 2]); % preallocate

% ask for filename

xydataFileName=input('Filename for the x and y data? ','s');

%% basic analog and digital IO

disp('Push the button when you wish to start taking data');

dv=1;

% read digital input from pin 13

while dv

dv=a.digitalRead(13);

end

% send trigger to fn generator - trailing edge

a.digitalWrite(11,1);

pause(0.01);

a.digitalWrite(11,0);

for i=1:numpts

xydata(i,1)= a.analogRead(0)/1023\*5; %scaling factor, 5V full scale

xydata(i,2)= a.analogRead(1)/1023\*5;

plot(xydata(:,1),xydata(:,2),'bo'); %plots blue points with x =analog 0

% y = analog 1 input

grid

drawnow;

end

dlmwrite(xydataFileName, xydata)

%% close session

delete(a)