**Constant Load Battery Tester**

Authors:

John Drumright III

Silas Renner

**Abstract:**

The purpose of this device is to test the performance of batteries in order to deduce information about characteristics of their capacity. This device will put a constant load on the battery in order to gather said information. An inherent behavior of batteries is their inability to output a constant current while they are being discharged. Using an Arduino platform with an accompanying circuit, the load on the battery is kept constant while data is gathered about the voltage and current of the battery with respect to time. To see the endurance of the battery under a given load one must plot its voltage versus time. The plot will display a characteristic “knee” in the curve that marks the point of the battery being dead. Using this tester, you can put a load of 50-550 mA on the battery, and see how the voltage changes with time. Since the current is being kept constant, the resultant curve will give you an idea of the true useful capacity of your battery. As an added bonus, the device is a relatively simple circuit that can be assembled using inexpensive components.

**Theory:**

There are two fundamental concepts in which this tester is based on. The first, Ohm’s law, is shown below:

*V = IR*

It is a simple yet powerful equation which relates the three basic electrical properties: voltage, current and resistance. With simple algebra, any of the three variables can be solved for using the other two. Specifically, in this case, we utilize Ohm’s law to match the current output of the battery to the user’s input current. The second underlying theory is not explicitly used however it serves as the model for the tester’s results. Based on an empirical relation derived by the German scientist William Peukert we can find evidence to support the data presented by this tester. Peukert’s law is as follows:

*CP = Ikt*

where CP = is the capacity at a one-ampere discharge rate

I  = discharge current

k = Peukert’s constant; a dimensionless constant characteristic of the battery being tested

t  = time

Although the equation 2 is not rigorously correct, it highlights the non-linearity of batteries. Based on a given load (i.e. current draw) the battery will exhibit a different charge capacity. As a result of the equation, higher discharge rates will yield shorter use times since the capacity is fixed. Peukert’s constant is a misnomer as it is not actually fixed, however we will refer to it as such. The constant k physically indicates the general “health” of the battery. This number decreases as the use of the battery increases. The purpose of the test is to support the theory of nonlinear discharge.

**Procedure:**

The following pages will outline the necessary steps on how to build the battery tester. The main components of this circuit are as follows:

1. Arduino Uno microcontroller
2. Digital analog converter (DAC) MCP4922
3. MOSFET, n-type transistor, CSD19506KCS

\*see accompanying data sheets for additional info\*

First, connect the Arduino microcontroller and the DAC to the breadboard (see circuit diagram for the appropriate pin connections). The Arduino 5V and GND pins will connect to the red and blue lines respectively on the breadboard. Those two lines will be the power for the DAC and the common ground the circuit. To maintain a clean setup, use a wire to connect the 5V red line on one side of the bread board to the red line on the opposing side. Repeat for the ground. This will allow for the use of shorter wires when making the other connections on the bread board, therefore reducing clutter and confusion.

Referring to the circuit diagram and the pictures below, connect the Arduino to the DAC. Note that there are two capacitors between the DAC and the 5V power supply, these serve to attenuate the 5V signal, removing any high-frequency noise on the line.

Picture picture picture

Pin 14 is the voltage output of the DAC and should be connected to the gate (g on the circuit diagram) of the MOSFET. It is important to place another bypass capacitor here as well to attenuate extra noise in the output signal of the DAC, please see the circuit diagram. The 1Ω sand bar resistor will be used to read the current coming from the battery and should be connected to the source pin of the MOSFET and in series with the positive terminal of the battery. The last pin on the MOSFET should of course be connected to ground.

Picture picture picture

Lastly, create two voltage dividers that connect in series on either side of the 1Ω resistor, using 33 kΩ and 47 kΩ resistors. The A0 and A1 pins of the Arduino should then be connected in-between the two large valued resistors. Once the battery is connected to ground the circuit will be complete and ready to run.

**Method:**

This battery tester uses the Arduino microcontroller, DAC, and MOSFET to keep a constant current load coming from the battery whilst taking voltage readings across the 1 Ω resistor in the circuit. To begin, if you were to simply connect the battery across the 1 Ω resistor, there would be a voltage drop across that resistor. To calculate the current coming from the battery you would simply measure the voltage on either side of the resistor, find the difference, and divide by the resistance. That is what pins A0 and A1 on the Arduino are being used for here.

Due to the relatively high current draw occurring in the circuit, the A0 and A1 Arduino connections are placed between two very large valued resistors. This essentially acts a voltage divider and prevents high current from going through the Arduino. Nonetheless, the pins are then connected on either side of the resistor, corresponding to the connections indicated on the circuit diagram. So when you hook the battery up across this resistor now, you will get a reading in both the A0 and A1 pins. However, the two pins do not read analog values of the voltages going into them, instead they return digital values between 0 and 1023. Those digital values scale with the incoming 0-5 Volts, so through a simple conversion in the software it is easy to convert that digital signal into an analog voltage reading. You also need a conversion for the voltage divider that separates the pins from the current going through the circuit, which is handled in the software as well. Using those two voltages and the known value of the resistor, it is possible to calculate the current going through the resistor. Conveniently enough, the battery and resistor are connected in series so the current going through the resistor will be the same as the current coming from the battery.

Now, in order to set the current being drawn to a desired level, it is necessary to change the resistance that the battery is experiencing. Using Ohms Law, one can see that the current is inversely proportional to resistance. So, the higher the resistance, the lower the current coming from the battery will be. This is where the MOSFET comes into play. It essentially acts as a digital potentiometer, that ranges from near 0 Ω to an infinite resistance, in other words a short in the circuit. However, this new source of resistance will not affect the validity of the current calculation from the 1 Ω resistor. It is important to note here that the max current that can be drawn from the battery will be due only to the 1 Ω resistor with the MOSFET being necessarily open, i.e. at near 0 Ω, because these circumstances will give us the lowest possible resistance.

So with that information in mind, when the MOSFET is completely open, the current is maxed out. Therefore, the tester starts out by having the MOSFET closed, not allowing any current to flow through the circuit. It will then read the user inputted current level and “open” up the MOSFET until the current flowing from the battery equals the user inputted level. In other words, the current flowing from the battery will start at zero, and slowly ramp up until it reaches the appropriate value. Once that value is reached, the device will begin printing the time, voltage, and current data. As the test is running, the current will slowly begin to drop, as is the nature of batteries. However, the Arduino will see this change and cause the MOSFET to open up and allow more current through, essentially keeping the current at a constant level.

The MOSFET here can be thought of as a water valve. The more you open the valve, the more water will come through. In this case, the openness or closeness of the MOSFET is determined by an input voltage of 0-5V to the gate. If there is 5V being sent to the gate, then the MOSFET will be completely open and if there is 0V being sent to it then it will be completely closed. The Arduino doesn’t have the capability of sending out a customizable analog voltage to the gate, thus is it necessary to include the intermediate DAC, which does output an analog voltage.

In order to have the MOSFET let the correct amount of current through, the Arduino must set the MOSFET to the correct level via the DAC. The DAC will accept two 8 bit words that encode the voltage you want it to output. So, using the Arduino SPI library, the software sends the appropriate message to the DAC that causes it to then send out the necessary voltage to adjust the MOSFET.

Using this set up allows the user to input a desired current level to be drawn from the battery and have that load be set and constant for the duration of the test. The resulting data is printed to the serial monitor and can then be plotted using excel to be analyzed.

**Uncertainty:**

The main source of uncertainty stems from floating resistance. The resistor used has a 5% uncertainty within itself. The calibration factors in the Arduino code itself are put in place to account for the floating resistance throughout the circuit. However, they are still not exact and depend upon each given situation (the battery, breadboard, number of connectors, etc.). With that being said, the voltages displayed in the program should be read as given with 3% uncertainty. The current readings should have a … uncertainty.

Improvements:

We understand the limitations of this device however we have already begun creating ways to further develop the device. For starters, we wish to use a smaller resistor so that we can put a larger load on the battery. The current resistor albeit small being only 1 Ohm, can only put a max load of 600 mA on the battery before the DAC reaches its maximum output, in turn completely opening the MOSFET gate. More experiments need to be taken so that we can minimize the discrepancies between the actual the values and the data taken by the tester. Also in the future would like to expand the tester to generate data on not only discharge curves but charging curves as well.