

A Low Cost Automatic Curve Tracer

A PC, a sound card and software combine to make a powerful I-V curve tracer.

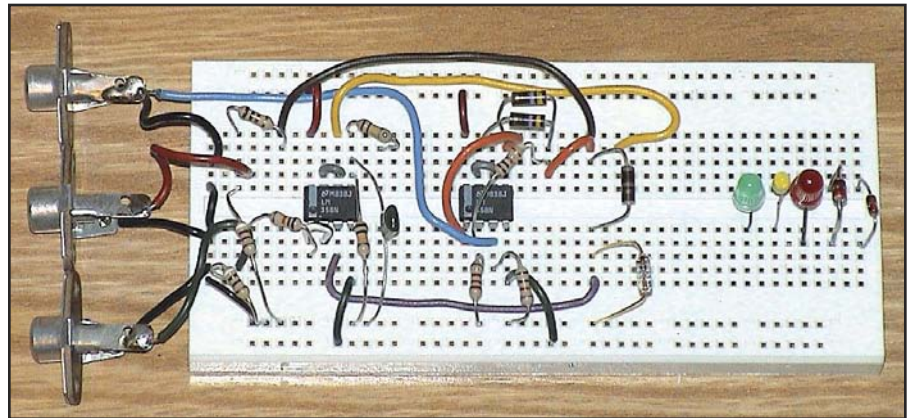
Dr George R. Steber, WB9LVI

Curve tracing of electronic components is a nifty way to gain more information about how they operate. By curve tracing, we mean making a graph of current versus voltage (an I-V curve) to display the basic characteristic of the electronic device. The I-V curves of diodes, Zener diodes, LEDs, transistors and other devices are helpful in understanding their nonlinear operation. Also, I-V curves are often used to obtain practical device data such as the forward voltage of a red LED operating at 15 mA or the reverse breakdown voltage of a 1N753A Zener diode at a current of 20 mA. But, alas, curve tracing seems to be a lost art and many experimenters rely only on curves supplied by the device manufacturer.

So What if We Can't Find the Curves?

Unfortunately, manufacturers' curves are not always handy. Indeed, the I-V curves of many devices may not be documented at all, like those diodes you picked up at the local hamfest. Of course, there are commercial curve tracers that will do the job but they are outside the realm of most experimenters. In this article I will describe an economical way of obtaining the I-V curves of two and three terminal devices using a PC, a simple circuit, a Windows compatible sound card and a bit of software.

To my knowledge, no one has previously used the sound card inputs of the PC for curve tracing without resorting to fancy modulation schemes. There is good reason for this, because most sound card inputs are ac coupled. That is, there is a capacitor in series with each input that effectively removes the dc value of the signal. When tracing a nonlinear device, like a diode, the actual dc value must be used or accuracy will suffer. A novel and completely new method that retains the dc value when using a sound card was developed by the author and disclosed in a January 2004 *Circuit Cellar* article.¹ One outstanding feature of the procedure is that it can be



The PC curve tracer breadboard setup.

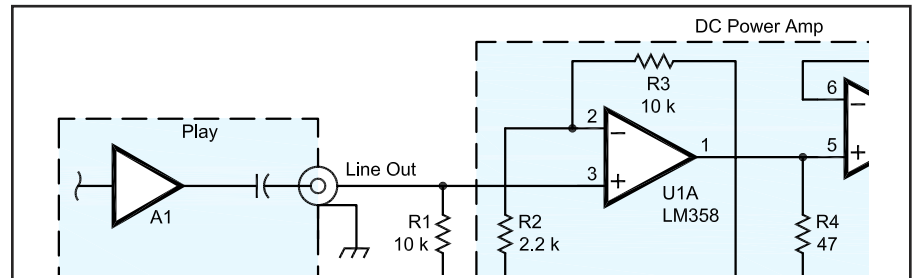


Figure 1 — Simple dc curve tracer setup (A) and I-V plot example (B).

implemented without modifying the sound card or even opening the PC case.

For this project I have designed a new circuit that is much simpler than the one described in that article, and I also wrote new software to make it easier to use. In this article I will cover how to build your own version of this new curve tracer, dubbed Curve Tracer II, and how to get it up and running. More in-depth technical details about this project are covered in a Jul/Aug *QEX* article.² It's a project that should appeal to almost anyone involved in electronics.

This project requires a PC running Windows. You can use one of the newer 3 GHz PCs or dust off your old 200 MHz PC. Although there are no guarantees, of course, that this project will work with your system, it has been tested with a 200 MHz Pentium

Pro, a 500 MHz Pentium III, and a 1.1 GHz AMD Athlon processor running Windows 98SE and Windows XP with a Sound Blaster (SB) Live!, Value Edition sound card.

If you have a Pentium or AMD PC with a Windows compatible full-duplex sound card, you probably have the basis for a very good curve tracer. All you need to do is build the simple circuit described, connect it to your computer sound card and run the program. You will be able to obtain accurate plots of the I-V characteristics of many devices, including resistors, diodes, Zener diodes, LEDs and transistors. Data is captured via the sound card stereo input, processed and plotted on the screen. The circuitry is very low cost (less than \$2) and uses readily available parts. You can build the circuit on a solderless breadboard as I did, or design

¹Notes appear on page 36.

a circuit board for it. In any case you will only need a good digital voltmeter (DVM) for calibration purposes. To get started it will be helpful to review the fundamentals of curve tracing.

Basic DC Curve Tracing

Obtaining an I-V curve is simple in principle. Refer to Figure 1 to see an example of a setup for obtaining I-V data for a device under test (DUT). An adjustable dc power supply, ammeter, voltmeter and current limiting resistor R are required. In this case only a positive voltage is shown but both polarities are often used for tracing. Adjust the supply voltage while monitoring the meters and write down the meter readings as a table of I and V values. From the table, plot data points to obtain a graph, like the one shown in Figure 1B. While this works well, it has several drawbacks. One has to manually adjust the voltage at reasonable intervals, read and write down the data, and finally plot it. And you have to switch power leads if you want to plot negative voltages. If more than a few devices need characterization, an automatic method is preferable. Clearly, if you have a bag of diodes, LEDs or Zeners that you just bought at a hamfest and want to test them, this is not the way to do it.

There are other ways to characterize your devices such as using an ac curve tracer. Basically you apply an ac voltage to the device under test (DUT) and plot its current versus voltage on a two-channel dc-coupled

oscilloscope with x-y plotting capability. You need an ac voltage source, an isolation transformer, a few resistors and, of course, the oscilloscope. Details of this method are covered in the *QEX* article.

While the ac curve tracing approach is straightforward, there are some things to consider. In addition to building the circuit, one needs to have access to an oscilloscope, know how to use it and calibrate it properly. Another consideration is saving and printing the curves for analysis or comparison. A PC can be of considerable help in these areas. So now let's consider how we can use a PC and a sound card for curve tracing.

Sound Card Interface Circuit

The lead photo shows my breadboard setup for interfacing to the PC sound card. Only a few components are required. In this instance, I am using a solderless breadboard, as it is an easy way to get started. Designing a circuit board may be a better option for those interested in a long term, less fragile system. Switch S1, the power supply and connecting cables are not shown. A Zener, 1N753A, is shown as the DUT located below the 100 Ω resistor used as R_m . Observe the devices on the right side of the breadboard; the LEDs and diodes. They're not part of the circuit and are just sitting there patiently waiting their turn to be traced. Note that RCA type connectors are used on my breadboard for the sound card inputs and output, but this was only

for convenience; any other kinds of jacks could be used. Sound cards typically have a 3.5 mm stereo jack for input and output so you will need a cable, preferably shielded, to convert to RCA. Actually, you will need two stereo cables (one for input signals and one for output signals) with the 3.5 mm stereo plug on one end and RCA plugs on the other end. Such cables are commonly found at your local electronics or audio-video store.

The interface circuit schematic is shown in Figure 2. A few general comments are in order. First make sure to use LM358 op-amps. Substitutes like the TL082 or TL272 will not work well here. Secondly, use a good, well-regulated dual power supply of ± 12 V with at least 100 mA capability. The DUT may require a pulsed current of 60 mA or more and the supply voltage should not droop during this time.

The interface uses one sound card output (either left or right channel) and two sound card inputs, the left and right channels. Although the schematic shows the inputs separately for clarity, they are in fact combined into a single stereo jack. The line output will provide a special sine wave as described later. Resistor R1 provides a ground reference for the sound card line output. U1A and U1B form a dc power amplifier to boost the line output and provide more current capability. This power amp does not provide equal current source and sink capability since the LM358 is unbalanced in this respect.

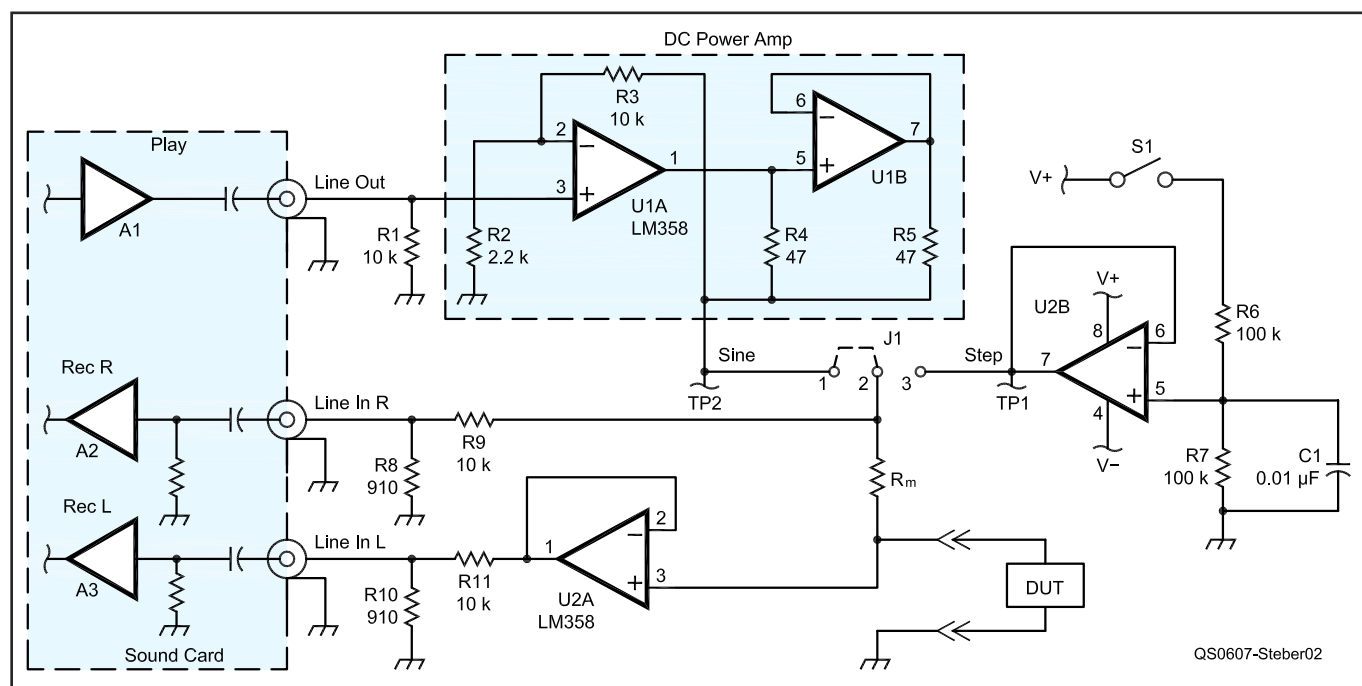


Figure 2 — Curve tracer circuit and the interface to the PC sound card. All resistors are $\frac{1}{4}$ W, 5%.

DUT — Device under test; see text.
C1 — 0.01 μ F, 50 V ceramic capacitor.
R1, R3, R9, R11 — 10 k Ω resistor.

R2 — 2.2 k Ω resistor.
R4, R5 — 47 Ω resistor.
R6, R7 — 100 k Ω resistor.
R8, R10 — 910 Ω resistor.

R_m — Current measurement resistor; see text.
U1, U2 — LM358 dual op-amp.

Typically an LM358 op-amp can source 40 mA and sink 20 mA. Since we are using two op-amps in our power amp, we should be able to double those figures. My breadboard circuit actually produces less, sourcing only up to 65 mA and sinking up to 32 mA.

Nonetheless, this is a stable, overload protected circuit with little dc offset, which is why I used it. Observe that in the case of diodes or LEDs, one can always orient the DUT so that current is sourced to the DUT, if desired. Note that resistor R2 controls the gain of the power amp. If you need to increase the gain, decrease the value of R2.

U2B is a simple step generator providing a slow step voltage, at about $\frac{1}{2}$ the positive supply voltage, at test point TP1 when S1 closes. Use a single action *momentary* switch for S1. Don't use a slide switch. Jumper block J1 is used to select the step or sine wave signal for the sound card. The step voltage is only used once, to calibrate the system. Note that there is also a test point, TP2, near pin 1 of J1. With no signal applied to the power amp, the voltage at TP2 should be less than 5 mV dc. This will avoid having a dc voltage applied to the sound card inputs and causing an error. In my circuit, the voltage at TP2 is around 3 mV.

A current measuring resistor, R_m , typically 100 to 1000 Ω , is connected in series with the DUT. The exact value can be chosen to suit your requirements. R8 and R9 form an attenuator to reduce the voltage to something the sound card line input *right* channel can accept without overloading, typically around 1 V. (My *SB Live* begins clipping at 0.82 V.) The voltage across the DUT is buffered by U2A, similarly attenuated and fed to the *left* channel. The *current* through the DUT is the difference between the right and left channel voltages, divided by R_m . So we need to know the value of R_m accurately, say within 1%. Given the value of R_m , the software calculates the current in the DUT.

The circuit is potentially harmful to some devices since, in normal operation, the sine wave sweeps through both positive and negative voltages up to 10 V. To restrict the voltage sweep to only positive or negative excursions, a diode can be placed appropriately across the DUT. The sine wave may flattop as more current is drawn from power amplifier by the DUT. This is normally not a problem. Increasing R_m may help reduce this effect. This circuit can easily be modified to handle higher voltages and currents. It is only a starting point. A large number of devices can be curve traced with this circuit as it stands.

PC Curve Tracer Operation

Basically the curve tracer works by applying a test signal, a few cycles of a 240 Hz sine wave, to the DUT and measuring

the current through the device and the voltage across it. Referring to Figure 2, the shortened sine wave is generated in the computer and is output to the line output of the sound card amplifier A1 when the START button on the computer screen is clicked. The sine wave is boosted by the dc power amplifier and presented to the top of R_m via pin 2 of J1. This sine wave produces a current in the DUT and a voltage across it, just as in the simple dc curve tracer discussed earlier. The voltage at the top of R_m is sent to the right channel of the sound card and the voltage at the bottom of R_m is sent to the left channel. Knowing the voltage across R_m enables calculation of the current via Ohm's Law in the computer. Notice that the voltage at the bottom of R_m is also the voltage across the DUT, which is sent to the left sound card channel.

If the sound card inputs were dc coupled, this would be the end of the story. But most sound card inputs are ac coupled by means of large capacitors, which allow the frequency response to go down to very low frequencies, typically 20 Hz. See the sound card block diagram, specifically line input amps A2 and A3, as shown in Figure 2, as an example of input coupling. Because of these capacitors the voltage tends to decay in a specific manner. Because they are capacitors, they tend to look like a short circuit to the initial applied voltages. Knowing these characteristics, we can model the input on the computer and build an inverse model that will reverse these effects. Details of how this is done are explained in more detail in the *QEX* article and involves circuit theory and complex variables. Fortunately we don't need to know these details to build a fine curve tracer.

We need to calibrate the software model so that it exactly reverses the effects of the ac coupling. For this purpose we use a specific test input, namely a step voltage and the circuitry associated with U2B. With the DUT removed and the jumper on J1 set between pins 2 and 3, we start a software capture. Shortly after the "start" we initiate a step voltage by pressing S1. Holding S1 closed, the step voltage amplitude is also measured with a digital voltmeter at TP1. The program captures the step voltage and is now ready for calibration of the software model, given some input from you. More specific details on how to do this are given later. The calibration is then saved and used for all further curve tracer operations. The calibration values have been found to hold up well over many months without drift, but can always be redone when needed.

Sound Card Considerations

A low distortion, low noise, full duplex sound card is desirable. The economical *Sound Blaster Live, Value Edition* (SB) fills

the bill nicely and probably many others will too. Since I cannot test them all, I will restrict my attention to this one. Referring to Figure 2, we see that A1 is the line output amplifier, and A2 and A3 are the right and left channel line inputs of the sound card. The mixer control in your *Windows* software controls the gains of these amps. The level of the line output signal is controlled in the *PLAY* section of the mixer via the *WAVE* and *SPKR* sliders. To control the amplitude of the sine wave output, just adjust the sliders.

The gain of the line input amps A2 and A3 are controlled in the *RECORD* section of the mixer by the *LINE* slider. They will saturate if the input voltage is too high, *regardless* of the *RECORD* setting in the mixer. On the SB this occurs at 820 mV. Fortunately this is easy to detect, as the captured signal will flattop. At saturation, the linear gain of the amplifier is lost, so we may also see a fictitious current bump in the DUT current trace at the extreme voltage limits. A real time oscilloscope function is provided in the software to help monitor for this condition. So, if you see a current bump in the I-V trace when there is no DUT in place, it is likely caused by an amplifier saturating. Just back off a little with the *PLAY* output signal or just ignore it.

A note is in order on earlier Sound Blaster sound cards such as the SB16 and AWE 32 since there are so many of these still around. Unfortunately they do not provide true full duplex operation and are noisy. The same may be said of SB *compatible* cards, so be wary. My advice is not to use any of these cards.

Curve Tracer Software Installation and Operation

The *Curve Tracer* software is available on the ARRL web site and is zipped for fast downloading.³ Unzip it to a new folder and you are ready to go. Just run the executable (exe) program. It was tested with *Windows* 98 and *XP*. When you run the software you may get a message like "Required DLL file MSVBVM60.DLL was not found." This is a Visual Basic run time file and is on many systems. If not found, you will need to obtain it and install it on your system. It is available free from Microsoft and other sites on the Internet. It is usually available as *Visual Basic 6.0 SP5: Run-Time Redistribution Pack* (VBRUN60sp5.exe) and is a self-extracting file. Download takes about 6 minutes at 28.8 kbps.

If you get the message "Component 'COMDLG32.OCX' or one of its dependencies is not correctly registered: a file is missing or invalid." when you try to run the program, you will need to register it on your system. This is also available free from Microsoft and other sites on the Internet. More details are included with the software.

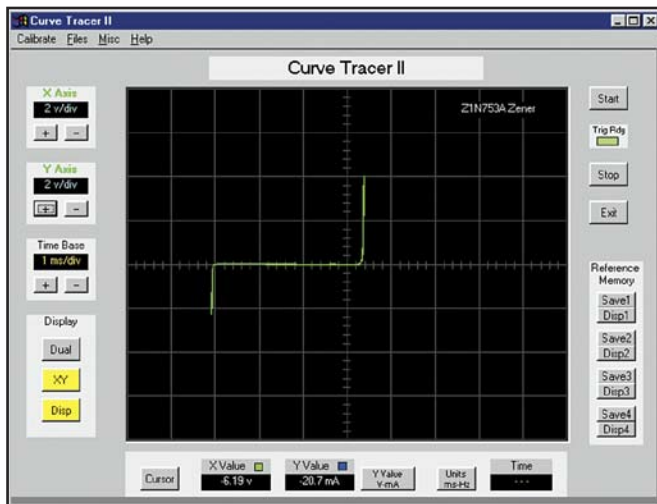


Figure 3 — The main window of the curve tracer program showing a Zener diode plot.

If you just want to experiment with the Curve Tracer program, go ahead as it does not modify the registry or install any other material on your computer. You can remove it by just deleting the entire folder where it is located.

Figure 3 shows a screen shot of the Curve Tracer II. In the middle of the screen is the oscilloscope display. It functions like a real oscilloscope and has controls for dual trace and X-Y display as well as sensitivity settings on the left side. The X-Y mode is essentially the I-V mode with current the vertical axis and voltage the horizontal. The *origin* is at the center of the screen. This screen shot shows the I-V display of a 1N753A Zener diode. It is rated by the manufacturer at 6.2 V at 20 mA. I measured 6.19 V at 20.7 mA with my Zener.

Below the scope screen is the cursor readout, which functions with the on-screen cursor and is controlled by your left mouse button. It works in both dual and X-Y mode. On the right side are the START button, STOP button and TRIGRDY LED. The START button initiates a capture, which automatically stops after 1 to 2 seconds while processing the data and displaying it on the screen. In case something is wrong, such as forgetting to turn on the circuit power supply, the STOP button may be used to stop the capture. Below the STOP button are four temporary memories for saving and comparing data. Additional features include a second cursor (right mouse button) for time measurement, provisions for printer output, screen capture and file saving for use by Microsoft *Excel* or other programs.

System Setup and Calibration

First make sure the circuit is powered and connected to the sound card line inputs as shown in Figure 2. The mixer that came with your sound card or the one that came with

Windows needs to be checked as you may have changed its settings. When you start the program, a little notice comes on the screen to remind you to do this. Basically you want to set the OUTPUT LEVEL, INPUT GAIN and STEREO BALANCE. The details on how to do this will obviously vary from system to system. Here is how it's done with the SB.

In the mixer PLAY section, enable WAVE and SPKR, set the sliders to max and mute all others (including LINE to avoid audio feedback). In the RECORD section, enable LINE, set it to maximum and mute all others. Set the STEREO BALANCE to center for all controls. These settings are *important*. For example, if you forget to mute LINE in the PLAY section there will be errors.

You only need to perform the calibration once, since parameters are saved in file *CTinit* upon exit. Follow these steps closely so good accuracy can be achieved.

First, remove the DUT. Next, set the jumper J1 on the circuit so that the step voltage is applied to R_m . Run the program. Click START and wait for the TRIGRDY LED to change from green to red. Now create the step voltage by closing switch S1 on the circuit. Upon capture of the step voltage data, the TRIGRDY LED will revert to green. With S1 still closed, measure the step voltage at TP1 with a good voltmeter, preferably a digital type, and write it down. View the data captured by the program using the left cursor in *dual* mode. You should see a large step voltage on L channel (green) and small voltage on the right minus left (R – L) channel (blue). See Figure 4 for an example. The data should start out clean without much noise. If there is a lot of noise at the start of capture (probably due to S1 bounce), repeat the capture until you get clean data.

The captured data is used to calibrate the program. Click on the CALIBRATE button

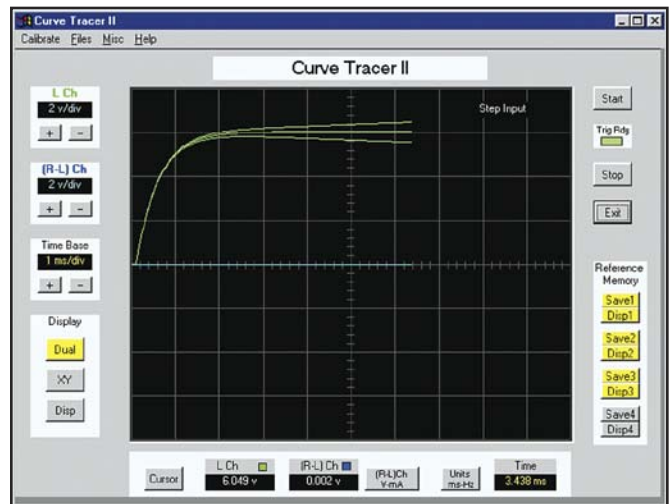


Figure 4 — The main window of the curve tracer program showing the response to a step voltage input.

to begin. Change the values shown in the CALIBRATE box as needed. All calibration is done on the originally captured step voltage. It does not need to be recaptured each time. Use the left cursor (left mouse button) to measure the data on the left (L) and (R – L) channels as needed.

Adjust the CHANNEL BALANCE value until the (R – L) channel is close to zero. This is essentially the voltage across R_m , which should be zero since the current in it is zero because the DUT is not present.

Adjust the COMPENSATION value until there is no droop or rise of the step voltage of the L channel. After its initial rise the step should flatten out to be a straight horizontal line across the window of the scope. If it is over corrected the line will rise, while if under corrected it will decline. See Figure 4 for an example of over, under and proper compensation. Make corrections in *very* small amounts. Use the cursor to verify it is straight by moving it across the waveform while looking at the readout.

The volts per bit calibration is done on the L channel of the sound card. Use the left cursor and measure the voltage of the L channel. Adjust this value until the cursor value is close to the value you measured with the voltmeter earlier. Each time you click the APPLY button in the CALIBRATION box, these values are applied to the captured step data. You will immediately see the results on the screen. Repeat these steps until the best calibration is achieved.

Other Calibration Box Values

The TRIGGER LEVEL controls the voltage level at which the beginning of the input signal is captured. If you get false captures, you can increase this value. If the value is too high, you will miss the leading edge of the captured waveform. If it is too low, you will

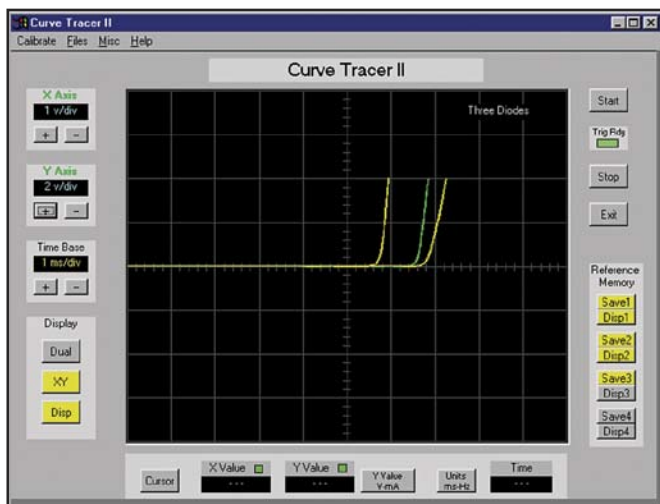


Figure 5 — The main window of the curve tracer program showing reference diode and LED traces.

get false triggers due to noise.

The reason for the INVERT INPUT box is that some sound cards invert the signal. The *SB Live* inverts the data, so this box needs to be checked to correct the input polarity. To determine if your sound card inverts the data, look at the step waveform of the L channel. It should start out positive for a positive input such as provided by the hardware circuit. If not, you need to check the box to invert the data.

Enter the value of the current measuring resistor in ohms in the R_m box. The cursor readout will display device current based on this value using Ohm's law.

Here are values for my *SB Live* card. CHANNEL BALANCE: 0.9551; COMPENSATION: +10; VOLTS PER BIT: 0.00033982; TRIGGER LEVEL: 200; R_m : 98.8 and INVERT: Box checked.

Obtaining I-V Traces

It is easy to curve trace a component. Make sure the hardware circuit is connected to the sound card, J1 connected to *sine*, the program calibrated, and the DUT connected. So you know what to expect, use a known resistor, diode or Zener as your first DUT. To curve trace it, follow these steps.

Click the START button. The green TRIGRDY LED will *not* turn red as the sine wave generated is only a few cycles long. After 1 to 2 seconds the data will be captured and processed, and will appear on screen. You have now curve traced your first component! If for some reason, the trigger did not occur, the TRIGRDY LED will turn red. A manual reset with the STOP button should restore operation. Normally this is not required unless there is something wrong with your setup. You cannot exit the program unless the TRIGRDY LED is green, a precaution to make

sure all processes are shut down.

Curve tracing various components is interesting and instructive. Resistors, of course, plot as straight lines. Their value can be checked by choosing a point on the line with the cursor (yielding the voltage and current at a given point) and using Ohm's Law. Diodes, Zeners, and LEDs show plots similar to Figure 3. Capacitors and inductors do not plot well as there are not enough cycles in the test sine wave to reach steady state.

As one last example, Figure 5 shows what you can expect when you trace an ordinary silicon diode and red and green LEDs. They clearly have different I-V curves; not what many would expect. This also illustrates the use of the reference memories to display and compare results. Files of some other

components that I have traced are included in the zipped file package mentioned earlier. You can use the program to read them and see what a trace looks like without having to build the circuit. The *QEX* article gives other examples, including how to curve trace a BJT transistor, an FET, a lambda diode and others. You can also observe negative resistance in some regions of the devices.

My PC curve tracer was compared to an ac curve tracer built using my Tektronix TDS360 Digital Real Time Oscilloscope and the results were in good agreement. With any luck, you may achieve similar results. In any case, I hope I haven't thrown too many curves your way!

Notes

¹G. Steber, "Tracing Voltage and Current," *Circuit Cellar*, Jan 2004.

²G. Steber, "I-V Curve Tracing with a PC," *QEX* Jul/Aug 2006.

³You can download the software associated with this article from the ARRLWeb at www.arrl.org/qexfiles/. Look for 7x06Steber.zip.

George R. Steber, PhD, is Emeritus Professor of Electrical Engineering and Computer Science at the University of Wisconsin-Milwaukee. George, WB9LVI, has an Advanced class license and is a Life Member of ARRL and the IEEE. Dr Steber has written several articles for QST, the most recent being "A Low Cost Automatic Impedance Bridge" in the October 2005 issue. He has industrial experience as a company officer, consultant and product designer, with 18 patents issued. In his spare time he enjoys racquetball, reading, editing video, astronomy and playing classical, Broadway and big band favorites with his trumpet in a local band. You may reach him at 9957 N River Rd, Mequon, WI 53092 or at steber@execpc.com with "Curve Tracer" in the subject line. Please send your message in text mode.

QST

NEW PRODUCTS

SOLDER-FREE PL-259 COAX CABLE SHIELD CONNECTION

◇The K4AVU Coax Crimper offers an alternative to soldering the shield connections on standard PL-259 coaxial cable connectors. The Coax Crimper fits around the part of the connector where the solder holes for the shield are located and crimps the body of the connector firmly in place. Installation is completed by soldering the center pin connection and twisting on the connector's coupling ring. The K4AVU

Coax Crimper is said to work with nearly all varieties of 1/2-inch diameter coax cables (RG-8 size), and also with standard UG-175 and UG-176 reducers for smaller diameter coax cable types such as RG-58, RG-59 and RG-8X. Price: \$29. For more information or to order, contact Paul Marsha, K4AVU, 200 Garden Trail Lane, Lexington, SC 29072; k4avu@yahoo.com; www.hamstation.com/coaxcrimper2.htm.

