

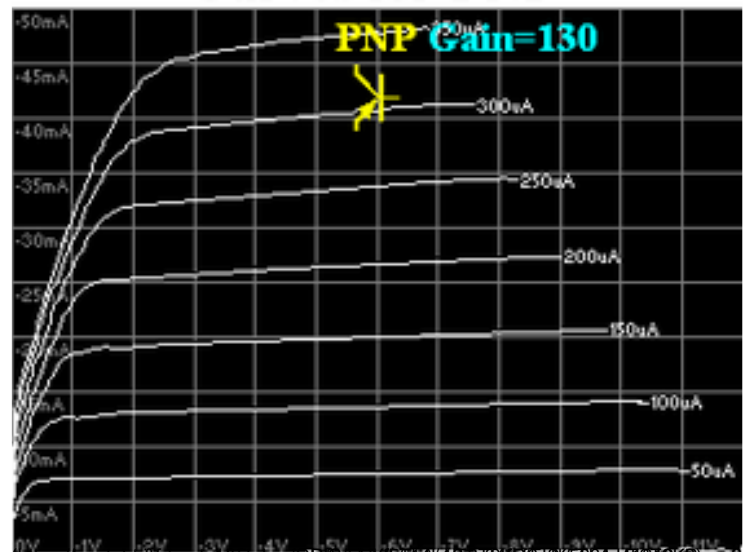
I was very impressed when I bought one of the wonderful LCR-T4 testers by Markus Frejek and others but I wanted it to tell me more about the components so I started to design my own tester.

I began by using the same screen as the LCR-T4 but it doesn't have a high enough resolution so I changed to a 320x240 2.8" LCD. It happens to be a colour touch-screen which is nice. The curve tracer runs on an Arduino Pro Mini 5V Atmega328p 16MHz and is powered by 4 AA cells.

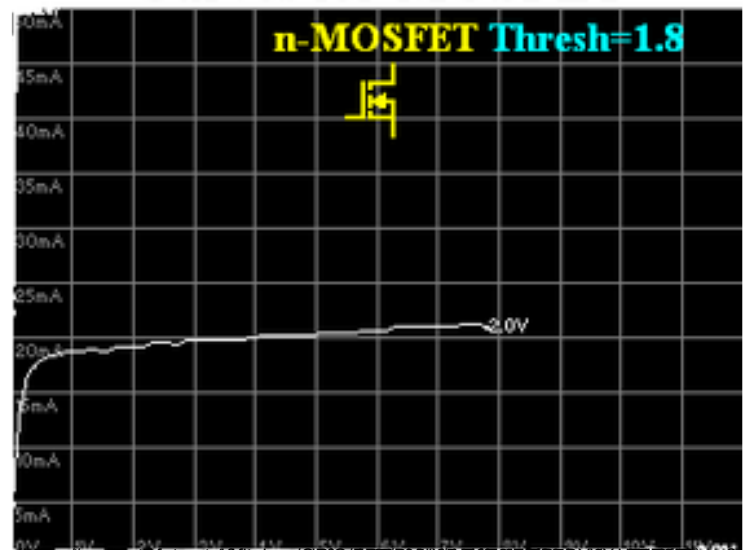
[Add Tip](#)[Ask Question](#)[Comment](#)[Download](#)

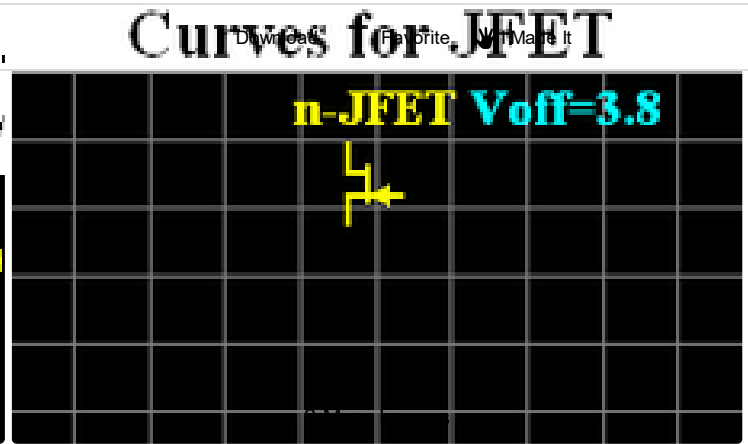
Step 1: How to Use It

Curves for PNP



Curves for MOSFET





When you switch on the curve tracer, the main menu screen is displayed.

Select the kind of device by touching one of "PNP NPN", "MOSFET" or "JFET". You can test diodes in the "PNP NPN" mode.

Put the Device Under Test (DUT) into the ZIF socket. The menu screen shows you which pins to use. PNPs, p-channel MOSFETs and n-channel JFETs go in the left side of the socket. NPNs, n-channel MOSFETs and p-channel JFETs go in the right side of the socket. Close the ZIF socket.

After a second or so, the tester will realise that it has a component and will start to draw the curves.

For a PNP or NPN transistor it plots V_{ce} (the voltage between the collector and emitter) versus the current flowing into the collector. A line is drawn for each different base current - e.g. 0uA, 50uA, 100uA, etc. The gain of the transistor is shown at the top of the screen.

For a MOSFET it plots V_{ds} (the voltage between the drain and source) versus the current flowing into the drain. A line is drawn for each different gate voltage - 0V, 1V, 2V, etc. The turn-on threshold of the FET is shown at the top of the screen.

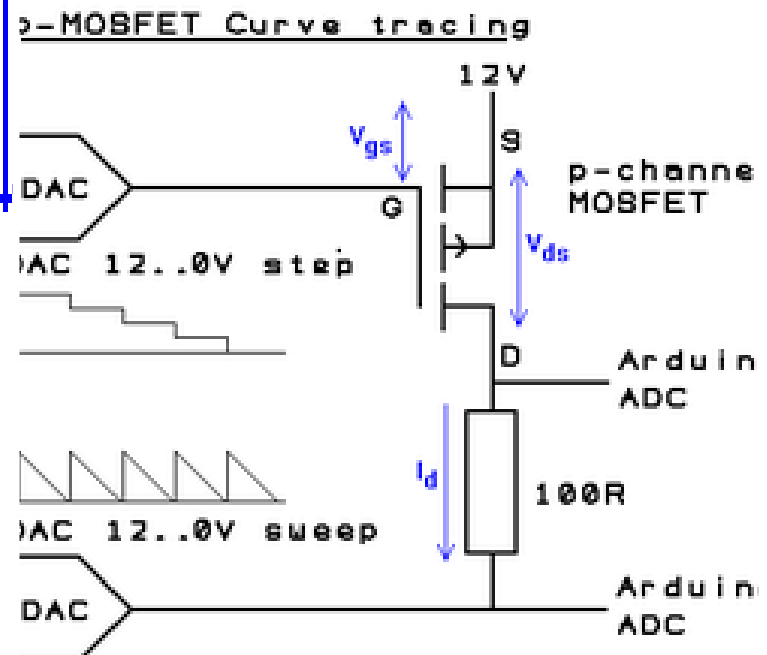
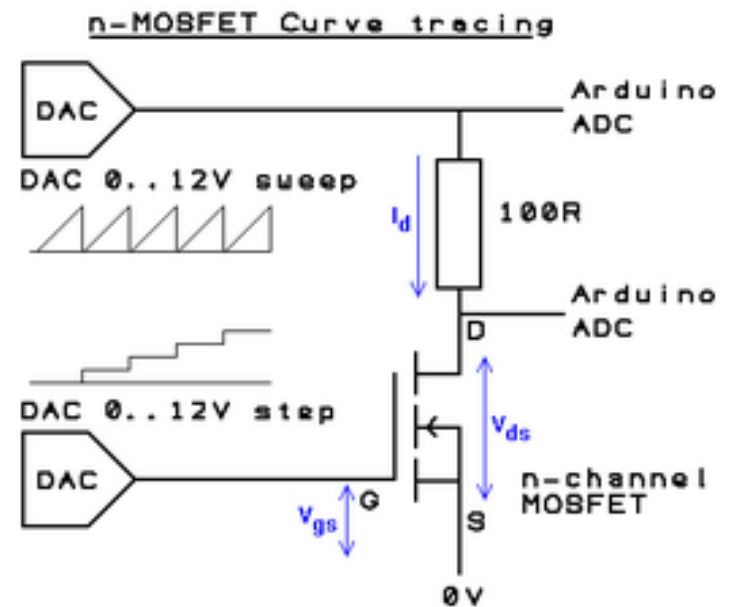
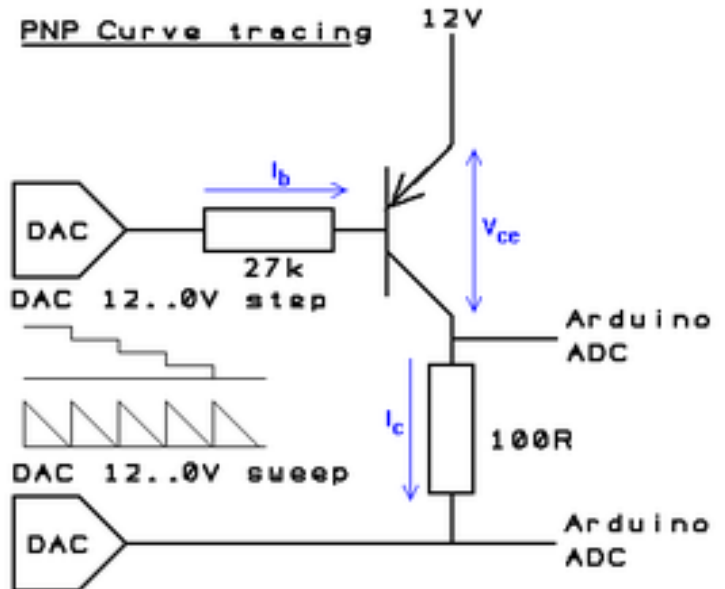
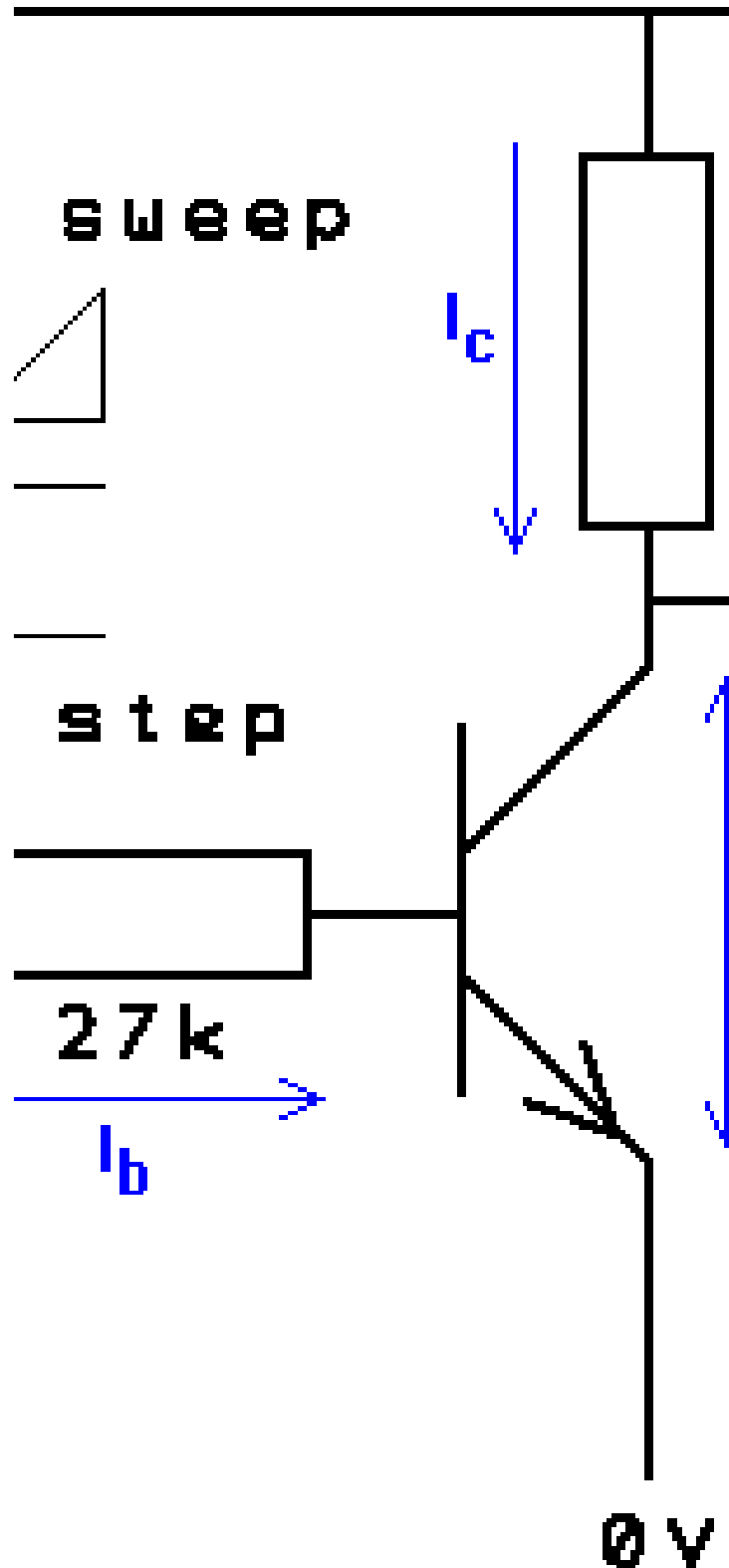
For a JFET it plots V_{ds} (the voltage between the drain and source) versus the current flowing into the drain. A line is drawn for each different gate voltage - 0V, 1V, 2V, etc. With depletion JFETs, current flows when the gate voltage is equal to the source voltage. As the gate voltage is changed to be further from the drain voltage, the JFET turns off. The cut-off threshold of the FET is shown at the top of the screen.

The most interesting part of a MOSFET or JFET curve is around the turn-on or cut-off voltage plus or minus a few hundred mV. In the main menu, touch the Setup button and the Setup screen will be shown. You can select the minimum and maximum gate voltage: more curves will be drawn in that region.

For a PNP or NPN transistor, the Setup screen allows you to select the minimum and maximum base current

With diodes, you can see the forward voltage and with Zeners, the reverse breakdown voltage. In the image above, I've combined the curves of several diodes.

Curve tracing



Let's consider an NPN transistor. We're going to draw a graph of the voltage between the collector and emitter (x-axis is V_{ce}) versus the current flowing into the collector (y-axis is I_c). We'll draw one line for each different base current (I_b) - e.g. 0uA, 50uA, 100uA, etc.

The emitter of the NPN is connected to 0V and the collector is connected to a 100ohm "load resistor" and then to a voltage that slowly increases. A DAC controlled by the Arduino sweeps that test voltage from 0V to 12V (or until the current through the load resistor reaches 50mA). The Arduino measures the voltage between the collector and emitter and the voltage across the load resistor and draws a graph.

This is repeated for each base current. The base current is generated by a second 0V-to-12V DAC and a 27k resistor. The DAC produces 0V, 1.35V (50uA), 2.7V (100uA), 4.05V (150uA), etc. (Actually, the voltage has to be a little higher because of V_{be} - assumed to be 0.7V.)

For a PNP transistor, the emitter is connected to 12V and the collector is connected to a 100ohm load resistor and then to a voltage that slowly decreases from 12V to 0V. The base current DAC steps down from 12V.

An n-channel enhancement MOSFET is similar to an NPN. The source is connected to 0V, the load resistor is connected to the drain and to a voltage sweeping from 0V to 12V. The DAC that was controlling the base current now controls the gate voltage and steps 0V, 1V, 2V, etc.

A p-channel enhancement MOSFET is similar to a PNP. The source is connected to 12V, the load resistor is connected to the drain and to a voltage sweeping from 12V to 0V. The gate voltage steps 12V, 11V, 10V, etc.

An n-channel depletion JFET is slightly more difficult. You would normally imagine the source connected to 0V, the drain connected to a varying positive voltage and the gate connected to a varying negative voltage. A JFET normally conducts and is turned off by a negative gate voltage.

The curve tracer can't generate negative voltages so the n-JFET drain is connected to 12V, the source is connected to a 100ohm load resistor and then to a voltage that slowly decreases from 12V to 0V. We want V_{gs} (the gate-source voltage) to step from 0V, -1V, -2V, etc. We want V_{gs} to remain constant as V_{ds} (the drain-source voltage) varies. So the Arduino sets the voltage at the load resistor then adjusts the gate voltage DAC until V_{gs} is the required value. It then sets a new voltage at the load resistor and again adjusts the gate voltage, etc.

(The curve tracer can't measure the voltage applied to the gate but it knows what it's told the DAC to do and that's accurate enough. Of course, this only measures the negative-gate part of the JFET response; if you want to see the positive-gate part, treat it as a MOSFET.)

A p-channel depletion JFET is treated similarly but the 0-to-12V values are all inverted.

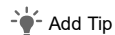
(The curve tracer does not specifically deal with depletion MOSFETs or enhancement JFETs but you could treat them as depletion JFETs and enhancement MOSFETs.)

Once it has completed the graph the curve tracer calculates the gain, threshold or cut-off of the transistor.

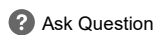
For bipolar transistors, the Arduino looks at the average spacing of the horizontal lines of the curves. As it draws the curve for base current, it notes the collector current when V_{ce} is equal to 2V. The change in collector current is divided by the change in base current to give the gain. The gain of a bipolar is a vague concept. It depends on how you measure it. No two makes of multimeter will give the same answer. Generally, all you're asking is "is the gain high?" or "are these two transistors the same?"

For MOSFETs, the Arduino measures the turn-on threshold. It sets the load voltage to 6V then gradually increases V_{gs} until the current through the load exceeds 5mA.

For JFETs, the Arduino measures the cut-off voltage. It sets the load voltage to 6V then gradually increases (negative) V_{gs} until the current through the load is less than 1mA.



Add Tip



Ask Question

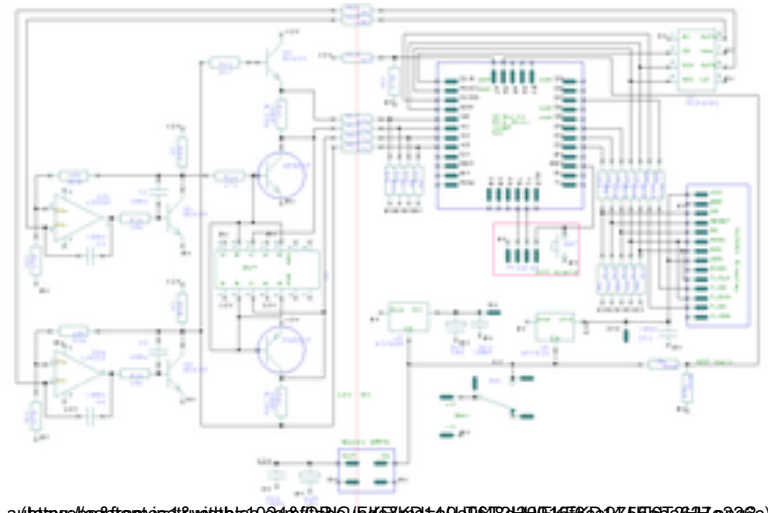
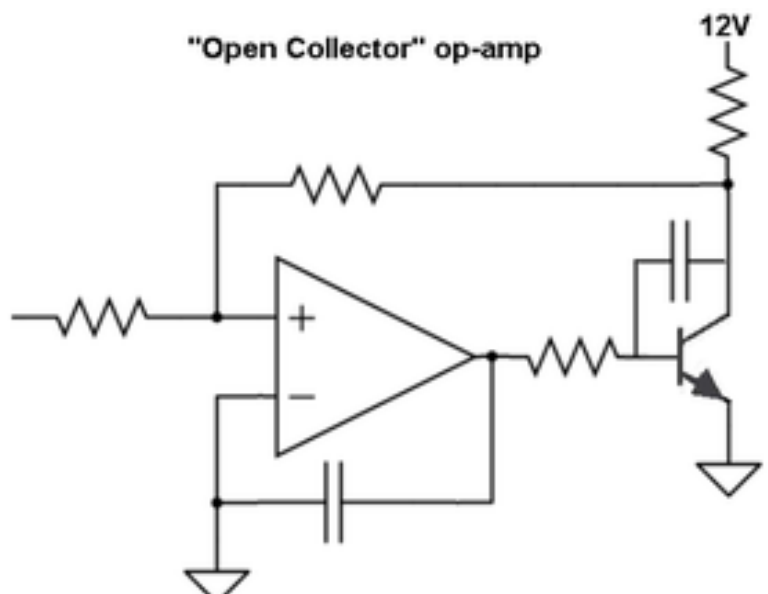
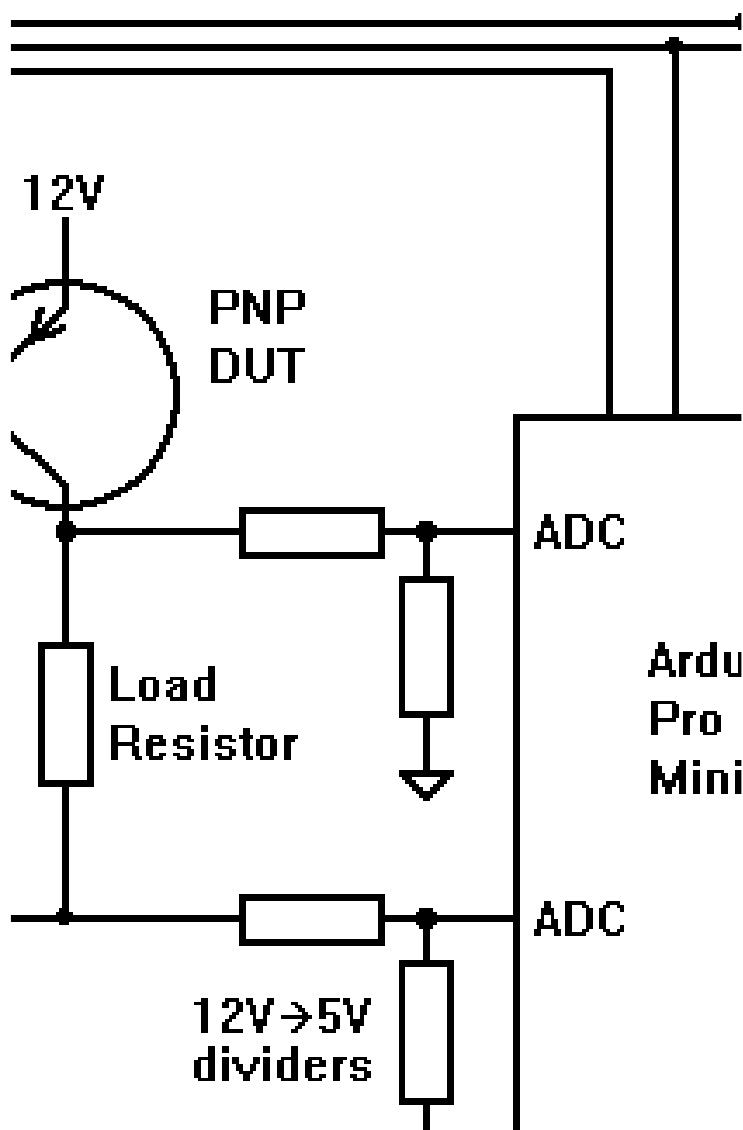


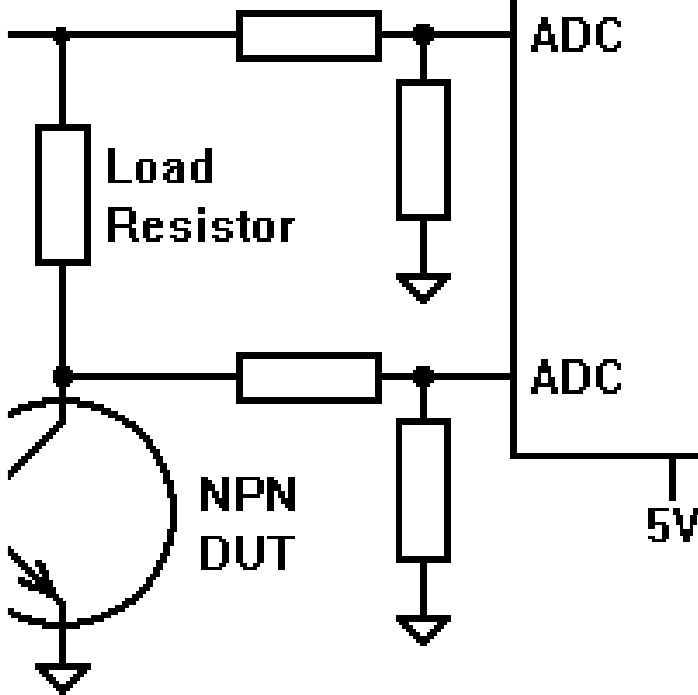
Comment

Download

Step 3: The Circuit

Block Diagram





6009
ies crazy
n't use

Very good
What I used

Seems C
Works dow
to 1.2

Here is a brief description of the circuit. A more complete description is in the attached RTF file.

The curve tracer needs three voltages:

- 5V for the Arduino
- 3.3V for the LCD
- 12V for the test circuit

The circuit must convert make these different voltages from the 4 AA cells.

The Arduino is connected to a 2-channel DAC to produce the various test voltages. (I tried using the Arduino PWM as a DAC but it was too noisy.)

The DAC produces voltages in the range 0V to 4.096V. These are converted into 0V to 12V by op-amps. I couldn't find any through-hole rail to rail op-amps that can source/sink 50mA, so I used an LM358. The output of an LM358 op-amp cannot go higher than 1.5V below its supply voltage (i.e. 10.5V). But we need the full range of 0-12V.

So we use an NPN as an open-collector inverter for the output of the op-amp.

The advantage is that this home-made "open-collector op-amp" output can go right up to 12V. Feedback resistors around the op-amp amplify the 0V to 4V from the DAC to 0V to 12V.

The voltages at the Device-Under-Test (DUT) vary between 0V and 12V. The Arduino ADCs are limited to 0V to 5V. Potential dividers do the conversion.

Between the Arduino and the LCD are potential dividers that drop 5V to 3V. The LCD, the touch screen and the DAC are controlled by the SPI bus.

The 6V from the cells is dropped to 5V with a very low dropout regulator - a HT7550 (if you don't have one then a 5V zener and a 22ohm resistor is not too much worse). The current consumption of the 5V supply is around 26mA.

The 6V from the cells is dropped to 3.3V with a low-dropout regulator - the HT7533. The current consumption of the 3.3V supply is around 42mA. (A standard 78L33 would work but it has a 2V dropout so you'd have to throw away your AA cells sooner.)

The 6V from the cells is boosted to 12V with a SMPS (Switched Mode Power Supply). I simply bought a module from eBay. I had real trouble finding a decent converter. The bottom line is, don't use an XL6009 converter, it's an absolute menace. As the battery goes flat and falls below 4V the XL6009 goes crazy and produces up to 50V which would fry everything. The good one I used is:

[https://www.ebay.co.uk/itm/Boost-Voltage-Regulator-Converter-Step-up-Power-Supply-DC-3-3V-3-7V-5V-6V-to-12V/272666687043?](https://www.ebay.co.uk/itm/Boost-Voltage-Regulator-Converter-Step-up-Power-Supply-DC-3-3V-3-7V-5V-6V-to-12V/272666687043?hash=item3f7c337643%3Ag%3AawsMAAOSw7GRZE9um&_sacat=0&_nkw=DC+3.3V+3.7V+5V+6V+to+12V+Step-up+Power+Supply+Boost+Voltage+Regulator+Converter&_from=R40&rt=nc&_trksid=m570.11313)

[hash=item3f7c337643%3Ag%3AawsMAAOSw7GRZE9um&_sacat=0&_nkw=DC+3.3V+3.7V+5V+6V+to+12V+Step-up+Power+Supply+Boost+Voltage+Regulator+Converter&_from=R40&rt=nc&_trksid=m570.11313](https://www.ebay.co.uk/itm/Boost-Voltage-Regulator-Converter-Step-up-Power-Supply-DC-3-3V-3-7V-5V-6V-to-12V/272666687043?hash=item3f7c337643%3Ag%3AawsMAAOSw7GRZE9um&_sacat=0&_nkw=DC+3.3V+3.7V+5V+6V+to+12V+Step-up+Power+Supply+Boost+Voltage+Regulator+Converter&_from=R40&rt=nc&_trksid=m570.11313)

It's tiny and is about 80% efficient. Its input current consumption is around 5mA when waiting for a DUT to be inserted and momentarily up to 160mA when drawing the curves.

As the AA cells are discharged the voltages vary, the software compensates by using reference voltages. The Arduino measures the 12V supply. The Arduino ADC uses its "5V" supply as a reference voltage but that "5V" is calibrated accurately against the Arduino's internal 1.1V reference voltage. The DAC has an accurate internal reference voltage.

I like the way the LCR-T4 has a button to turn it on and turns itself off automatically with a timeout. Unfortunately, the circuit introduces a voltage drop which I can't afford when powering from 4 AA cells. Even re-designing the circuit to use a FET wasn't sufficient. So I'm using a simple on/off switch.



CurveTracerCircuit.zip (<https://content.instructables.com/ORIG/FJA/H3N3/JL6UZT5N/FJAH3N3JL6UZT5N.pdf>)



ArdCurve (Schematic).pdf

Download (<https://content.instructables.com/ORIG/FJA/H3N3/JL6UZT5N/FJAH3N3JL6UZT5N.pdf>)
(<https://content.instructables.com/ORIG/FJA/H3N3/JL6UZT5N/FJAH3N3JL6UZT5N.pdf>)



Add Tip



Ask Question



Comment

Download

Step 4: The Software



<https://www.instructables.com/Arduino-Transistor-Curve-Tracer/>

The Arduino sketch is attached here. Compile and upload it to the Pro Mini in the usual way. There are lots of descriptions of how to upload programs on the web and in other Instructables.

The sketch starts by drawing the main menu then waits for you to insert a component or touch one of the buttons (or send a command from the PC). It tests for component insertion once a second.

It knows you have inserted a component because, with the base/gate voltage set to half way (DAC = 128) and the load resistor voltage set to 0V or 12V, a current of several mA flows through one or other of the load resistors. It knows when the device is a diode because changing the base/gate voltage does not change the load current.

It then draws the appropriate curves and switches the base and load currents off. It then tests once a second until the component is unplugged. It knows the component is unplugged because the load current falls to zero.

The ILI9341 LCD is driven by my own library called "SimpleILI9341". The library is attached here. It has a standard set of drawing commands very similar to all such libraries. Its advantages over other libraries are that it works (some don't!) and it shares the SPI bus politely with other devices. Some of the "fast" libraries you can download use special timing loops and are upset when other, maybe slower, devices are used on the same bus. It's written in plain C and so has smaller overheads than some libraries. A Windows program is attached which allows you to make your own fonts and icons.

[SimpleILI9341.zip](#) (<https://www.instructables.com/Arduino-Transistor-Curve-Tracer/>)



Add Tip



Ask Question



Comment

Download

Step 5: Serial Comms to PC

The curve tracer can communicate with a PC via a serial link (9600bps, 8-bit, no parity). You will need a suitable USB-to-serial convertor.

The following commands can be sent from the PC to the curve tracer:

- Command 'N': trace the curves of an NPN transistor.
- Command 'P': trace the curves of a PNP transistor.
- Command 'F': trace the curves of an n-MOSFET.
- Command 'f': trace the curves of a p-MOSFET.
- Command 'J': trace the curves of an n-JFET.
- Command 'j': trace the curves of a p-JFET.
- Command 'D': trace the curves of a diode on the NPN side of the socket.
- Command 'd': trace the curves of a diode on the PNP side of the socket.
- Command 'A' nn: set DAC-A to the value nn (nn is a single byte) then return an 'A' to the PC. DAC-A controls the load voltage.
- Command 'B' nn: set DAC-B to the value nn then return a 'B' to the PC. DAC-B controls the base/gate voltage.
- Command 'X': continuously send ADC values back to the PC.
- Command 'M': show the main menu.

When the curves are traced following one of the commands, the results of the curve are transmitted back to the PC. The format is:

- "n": start a new plot, draw the axes, etc.
- "m (x),(y),(b)": move the pen to (x),(y).
 - (x) is Vce in integer mV.
 - (y) is Ic in integer hundreds on uA (e.g. 123 means 12.3mA).
 - (b) is the base current in integer uA
 - or (b) is 50 times the gate voltage in integer mV
- "l (x),(y),(b)": draw a line to pen to (x),(y).
- "z": the end of this line
- "g (g)": the end of the scan;
 - (g) is the gain, threshold voltage (x10) or the cut-off voltage (x10)

The values sent to the PC are the raw measured values. The Arduino smooths the values before drawing them by averaging; you should do the same.

- "x (p),(q),(r),(s),(t),(u)"
 - (p) the voltage at the load resistor of the PNP DUT
 - (q) the voltage at the collector of the PNP DUT
 - (r) the voltage at the load resistor of the NPN DUT
 - (s) the voltage at the collector of the NPN DUT
 - (t) the voltage of the "12V" supply
 - (u) the voltage of the "5V" supply in mV

You could write a PC program to test other devices. Set the DACs to test voltages (using 'A' and 'B' commands) then see what the ADCs report.

The curve tracer only sends data to the PC after it has received a command as sending data slows down the scan. It also no longer tests for the presence/absence of a component. The only way to turn off the curve tracer is to send an 'O' command (or to remove the battery).

A Windows program is attached which demonstrates sending commands to the curve tracer.

CurveTracer.zip(<https://www.instructables.com/files/orig/FYJN8Q/GIZNSQBN7QJZNIQTGIZNSQGI.zip>)



 Add Tip



 Ask Question



 Comment

Download

Step 6: Building the Curve Tracer

- Arduino Pro Mini 5V 16MHz Atmel328p (£1.30)
- 14pin Zif Socket (£1)
- MCP4802 (£2.50)
- HT7533 (£1)
- LE33CZ (£1)
- IL9341 2.8" Display (£6)
- 5V to 12V boost Power Supply (£1)
- 4xAA cell battery holder (£0.30)

Search eBay or your favourite supplier. That's a total of around £14.

I got my display here:

<https://www.ebay.co.uk/itm/2-8-TFT-LCD-Display-Touch-Panel-SPI-Serial-ILI9341-5V-3-3V-STM32/202004189628?hash=item2f086351bc:g:5TsAAOSwp1RZfIO5>
(<https://www.ebay.co.uk/itm/2-8-TFT-LCD-Display-Touch-Panel-SPI-Serial-ILI9341-5V-3-3V-STM32/202004189628?hash=item2f086351bc:g:5TsAAOSwp1RZfIO5>).

And the boost SMPS here:

https://www.ebay.co.uk/itm/DC-3-3V-3-7V-5V-6V-to-12V-Step-up-Power-Supply-Boost-Voltage-Regulator-Converter/192271588572?hash=item2cc4479cdc%3Ag%3AJsUAAOSw8IJZinGw&_sacat=0&_nkw=DC-3-3V-3-7V-5V-6V-to-12V-Step-up-Power-Supply-Boost-Voltage-Regulator-Converter&_from=R40&rt=nc&_trksid=m570.l1313 (https://www.ebay.co.uk/itm/DC-3-3V-3-7V-5V-6V-to-12V-Step-up-Power-Supply-Boost-Voltage-Regulator-Converter/192271588572?hash=item2cc4479cdc%3Ag%3AJsUAAOSw8IJZinGw&_sacat=0&_nkw=DC-3-3V-3-7V-5V-6V-to-12V-Step-up-Power-Supply-Boost-Voltage-Regulator-Converter&_from=R40&rt=nc&_trksid=m570.l1313).

The remaining components are things you probably have already:

- BC639 (3 off)
- 100nF (7 off)
- 10uF (2 off)
- 1k (2 off)
- 2k2 (5 off)
- 3k3 (5 off)
- 4k7 (1 off)
- 10k (7 off)
- 27k (1 off)
- 33k (8 off)
- 47k (5 off)
- 68k (2 off)
- 100R (2 off)

• Slide Switch (1 off)

- LM358 (1 off)
- stripboard
- 28-pin IC socket or SIL header
- nuts and bolts

You will need the usual electronics tools - soldering iron, cutters, solder, odd pieces of wire, etc. - and a USB-to-serial convertor to program the Arduino.

The curve tracer is built on stripboard. If you're the sort of person who wants a curve tracer, you will already know how to lay out stripboard.

The layout I used is shown above. Cyan lines are copper on the back of the stripboard. Red lines are links on the component side or are the extra-long leads of the component. Curved Red lines are flexible wire. Dark blue circles are breaks in the stripboard.

I built it on two boards, each 3.7" by 3.4". One board contains the display and the tester circuit; the other board has the battery holder and the 3.3V, 5V and 12V supplies. I kept the low-voltage ("5V") and high-voltage ("12V") parts of the tester circuit separate with only high-value resistors crossing the frontier.

The two boards and the display form a triple-decker sandwich held together with M2 screws. I cut lengths of plastic tube to act as spacers or you could use ballpoint pen tubes, etc.

I only connected the Arduino Mini pins that I needed and only the ones on the sides (not at the top and bottom ends of the Mini PCB). I used short lengths of wire rather than the usual row of square pins that Arduinos are supplied with (the pins soldered to the PCB are square in the drawing). I wanted the Arduino to be flush against the stripboard because there's not a lot of height under the display.

The Arduino ProMini pinout is rather variable. The pins on the long edges of the board are fixed but the pins on the short edges differ between suppliers. The layout above assumes a board with the 6 programming pins with Gnd next to the Raw pin and with DTR next to Tx on the long edge. At the other end of the board are a row of 5 pins with 0V next to D9 and A7 next to D10. None of the short-edge pins are soldered into the stripboard so you can use loose wires if your ProMini is different.

Use a SIL header socket to hold the display. Or cut a 28-pin IC socket in half and use the pieces to make a socket for the display. Solder the square pins that are supplied with the display (or came with the Arduino) into the display. They're too fat to plug into a turned-pin socket - choose a socket that has the "spring clip" kind of pins. Some "spring clip" kind of IC sockets can only withstand half a dozen insertions/removals of the LCD so try to find good ones in your component drawer.

The LCD contains a socket for an SD card (which I didn't use). It is connected to 4 pins on the pcb. I used the pins and a piece of SIL header or IC socket to help support the LCD.

Notice that there are some links under the ZIF socket. Solder them in before you fit it.

I added a programming connector with Tx, Rx, Gnd and a reset button. (My USB-to-serial convertor doesn't have a DTR pin so I have to reset the Arduino manually.) I unsoldered the programming connector when the project was finished.

To protect the electronics, I made a cover from polystyrene sheet.

Files for the circuit in EasyPC format are attached.

ArdCurve.zip (<https://content.instructables.com/GR9/5G5D84221/6L6T2744/5G5D842216L6T2744.zip>)

ArdCurve (PCB).pdf (<https://content.instructables.com/GR9/5G5D84221/6L6T2744/5G5D842216L6T2744.pdf>)



Add Tip



Ask Question



Comment

Download

Step 7: Future Development

It might be nice to produce curves for other components but which? It's not clear to me what extra information the curve of a thyristor or triac would tell me beyond what the LCR-T4 tester does. The LCR-T4 tester can even be used with opto-isolators. I've never used a depletion MOSFET or an enhancement JFET or a unijunction transistor and don't own any. I presume the curve tracer could treat an IGBT as a MOSFET.

It would be nice if the curve tracer could recognise a component automatically and say which pin is which. Ideally, it would then go on to produce the curves. Unfortunately, the way the DUT pins are driven and measured, that would require a lot of extra components and complexity.

A simpler solution is to copy the existing LCR-T4 tester circuit (it's open source and very simple) with a second Atmega processor. Extend the ZIF socket to 16-pin to give three extra pins into which the unknown component can be plugged. The new Atmega acts as a slave on the SPI bus and reports to the main Arduino Mini what it sees. (SPI slave sketches are available on the web.) The software of the LCR-T4 tester is available and looks well documented. There's nothing inherently difficult there.

The main Arduino displays the component type and a diagram of how to plug the component in to the curve tracer part of the ZIF socket.

I've attached a surface-mount layout which can be used with an Arduino ProMini or with a naked Atmega328p (in EasyPC format). If there is sufficient demand (and orders with money) I could produce a batch of SM PCBs Could you buy one from me ready built? Well yes, of course, but the price would be silly. The advantage of dealing with China is that so many nifty electronic modules can be bought so cheaply. The disadvantage is that it's not worth developing anything: if it's a success, it will be cloned. Nice as this curve tracer is, I don't see it as a viable business opportunity.

ArdCurve SMD Zip (<http://www.instructables.com/ARD/ARD-CURVE-TRACER/ARD-CURVE-TRACER-006182122/>)



Add Tip



Ask Question



Comment

Download

4 People Made This Project!

TiberioV (/member/TiberioV/) made it!

pierrotlune67 (/member/pierrotlune67/) made it!

bemarios77 (/member/bemarios77/) made it!

StefanoP65 (/member/StefanoP65/) made it!


Did you make this project? Share it with us!


I Made It!


Recommendations


	Transistor Curve Tracer by Peter Balch (/member/Peter Balch/) Follow	Download	Favorite	👤 I Made It
	(/Easy-Zoom-Quit-Button/) Easy Zoom Quit Button (/Easy-Zoom-Quit-Button/) by randofa (/member/randofa/) in Arduino (/circuits/arduino/projects/) 12 1.2K	(/Light-Up-Clock-20/) Light Up Clock 2.0 (/Light-Up-Clock-20/) by JonathonT (/member/JonathonT/) in Microcontrollers 15 829		

(/contest/microcontroller2021/)
 (/contest/automation2021/)




Add Tip


Ask Question


Post Comment

We have a **be nice** policy.
Please be positive and constructive.

Add Images
Post

188 Comments

(/member/krisz18/) krisz18 (/member/krisz18/) 5 weeks ago

Reply
 ▲ Upvote

Hi !

I have a question, you wrote :

"This is repeated for each base current. The base current is generated by a second 0V-to-12V DAC and a 27k resistor. The DAC produces 0V, 1.35V (50uA), 2.7V (100uA), 4.05V (150uA), etc. (Actually, the voltage has to be a little higher because of Vbe - assumed to be 0.7V.)"

But it somehow not true. You assume you have a voltage generator with ONLY 27k.
But in fact, you have $I = U_{dac}/(27k+R_{be})$ so the current is not exact.

You should measure the basis current step by step.

Thanks
Kris

Reply

▲ Upvote

In ExecSetupMenu function for what purpose you have used

```
if (millis() - time > 1000) {  
  if (TestDeviceKind(tkNothing, false) != tkNothing)  
    return true;  
  time = millis();  
  Can you explain if possible thanking you sir
```

Further it was defined

```
i = analogRead(pin_ADC_NPN_Vcc) - analogRead(pin_ADC_NPN_Vce);  
It is voltage difference only For current It has to divide by collector load  
Pl clarify that point also
```

2 replies ▼

(/member/kathiresanmuthiah2001/) kathiresanmuthiah2001 (/member/kathiresanmuthiah2001/)
2 months ago

Reply

▲ Upvote

This is actual pic pf my prototype with pic 18f4620

(<https://content.instructables.com/ORIG/FQE/HQFM/KJ2UVBSG/FQEHQFMKJ2UVBSG.jpg?auto=webp&fit=bounds&frame=1&height=1024&width=1024>)

(/member/kathiresanmuthiah2001/) kathiresanmuthiah2001 (/member/kathiresanmuthiah2001/)
2 months ago

Reply

▲ Upvote

actual pic is here

(/member/kathiresanmuthiah2001/) kathiresanmuthiah2001 (/member/kathiresanmuthiah2001/)
2 months ago

Reply

▲ Upvote

actual

(/member/kathiresanmuthiah2001/) kathiresanmuthiah2001 (/member/kathiresanmuthiah2001/)
2 months ago

Reply

▲ Upvote

(<https://content.instructables.com/ORIG/FOO/7UBU/KJ2UVBLW/FOO7UBUKJ2UVBLW.png?auto=webp&fit=bounds&frame=1&width=1024>)

(/member/lironaviv28/) lironaviv28 (/member/lironaviv28/) Question 2 months ago

Answer

▲ Upvote

hello
I've been studying the circuit of the curve tracer for a days and i don't understand what is Q3 doing, why cant i just connect the NPN/MOSFET-N/JFET-N straight to the collector ?
1 answer ▼

(/member/lironaviv28/) lironaviv28 (/member/lironaviv28/) 2 months ago

Reply

▲ Upvote

hi

ty looks great.
i want to modify it so it will work with arduino due. if i use arduino due that all ready have DAC's and ADC's build in to it, i still need to use the MCP4802 chip?
1 reply ▼

(/member/LA+Audio/) LA Audio (/member/LA+Audio/) 1 year ago

Reply


▲ Upvote


i have some trouble on wiring touchscreen, not working when touchscreen conected
6 replies ▼


More Comments


Post Comment


Categories


-  Circuits


[\(/circuits/\)](/circuits/)
-  Living

[\(/living/\)](/living/)
-  Workshop

[\(/workshop/\)](/workshop/)
-  Outside

[\(/outside/\)](/outside/)
-  Craft

[\(/craft/\)](/craft/)
-  Teachers

[\(/teachers/\)](/teachers/)
-  Cooking

[\(/cooking/\)](/cooking/)

About Us

- Who We Are

[\(/about/\)](/about/)
- Why Publish?

[\(/create/\)](/create/)
- Jobs

[\(/jobs/\)](/jobs/)

Resources

- Sitemap

[\(/sitemap/\)](/sitemap/)
- Help

[\(/how-to-write-a-great-instructable/\)](/how-to-write-a-great-instructable/)
- Contact

[\(/contact/\)](/contact/)

[\(https://www.instagram.com/instructables/\)](https://www.instagram.com/instructables/) [\(https://www.pinterest.com/instructables\)](https://www.pinterest.com/instructables/) [\(https://www.facebook.com/instructables\)](https://www.facebook.com/instructables/) [\(https://www.twitter.com/instructables\)](https://www.twitter.com/instructables/)