Transistor Curve Tracer

by Peter Balch

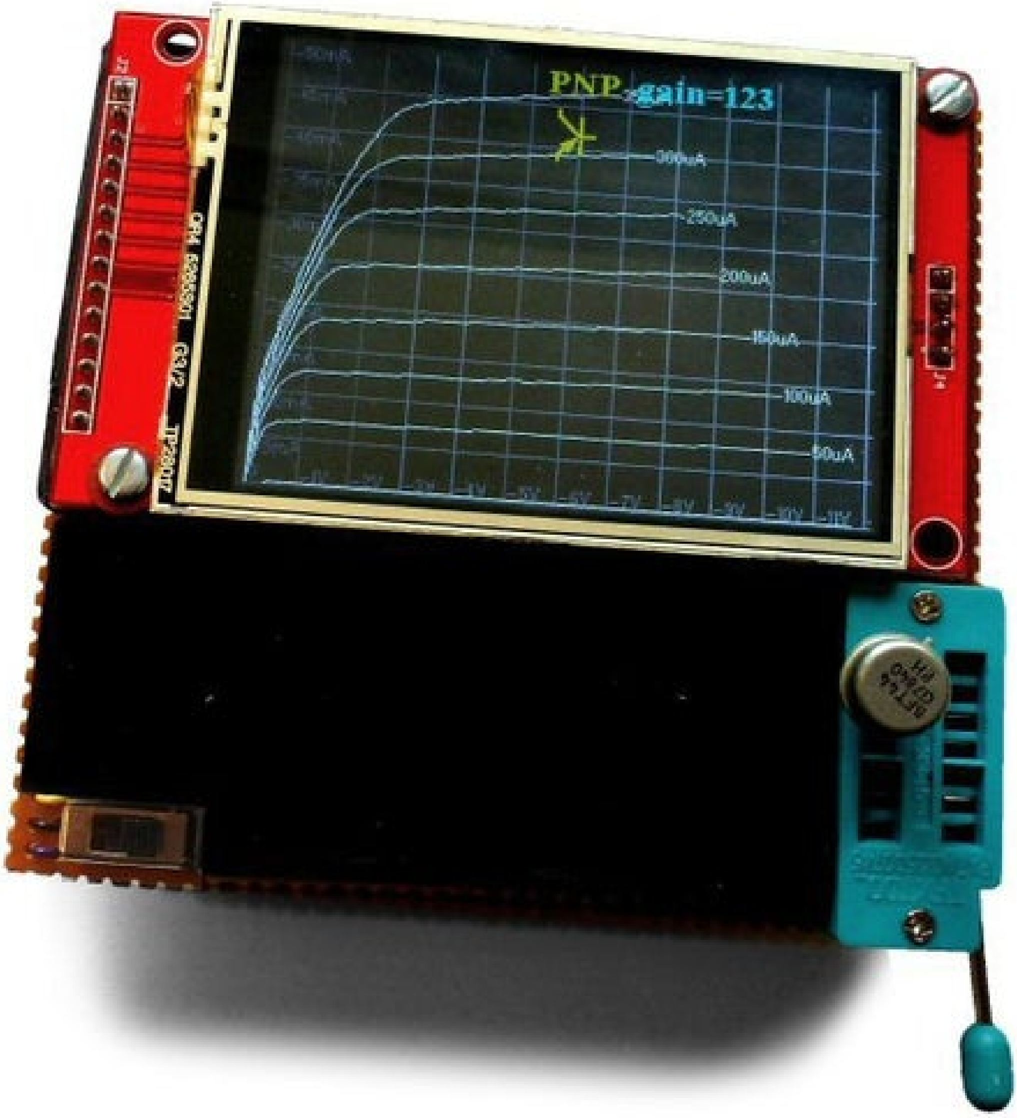
I've always wanted a transistor curve tracer. It's the best way of understanding what a device does.Having built and used this one, I finally understand the difference between the various flavours of FET.

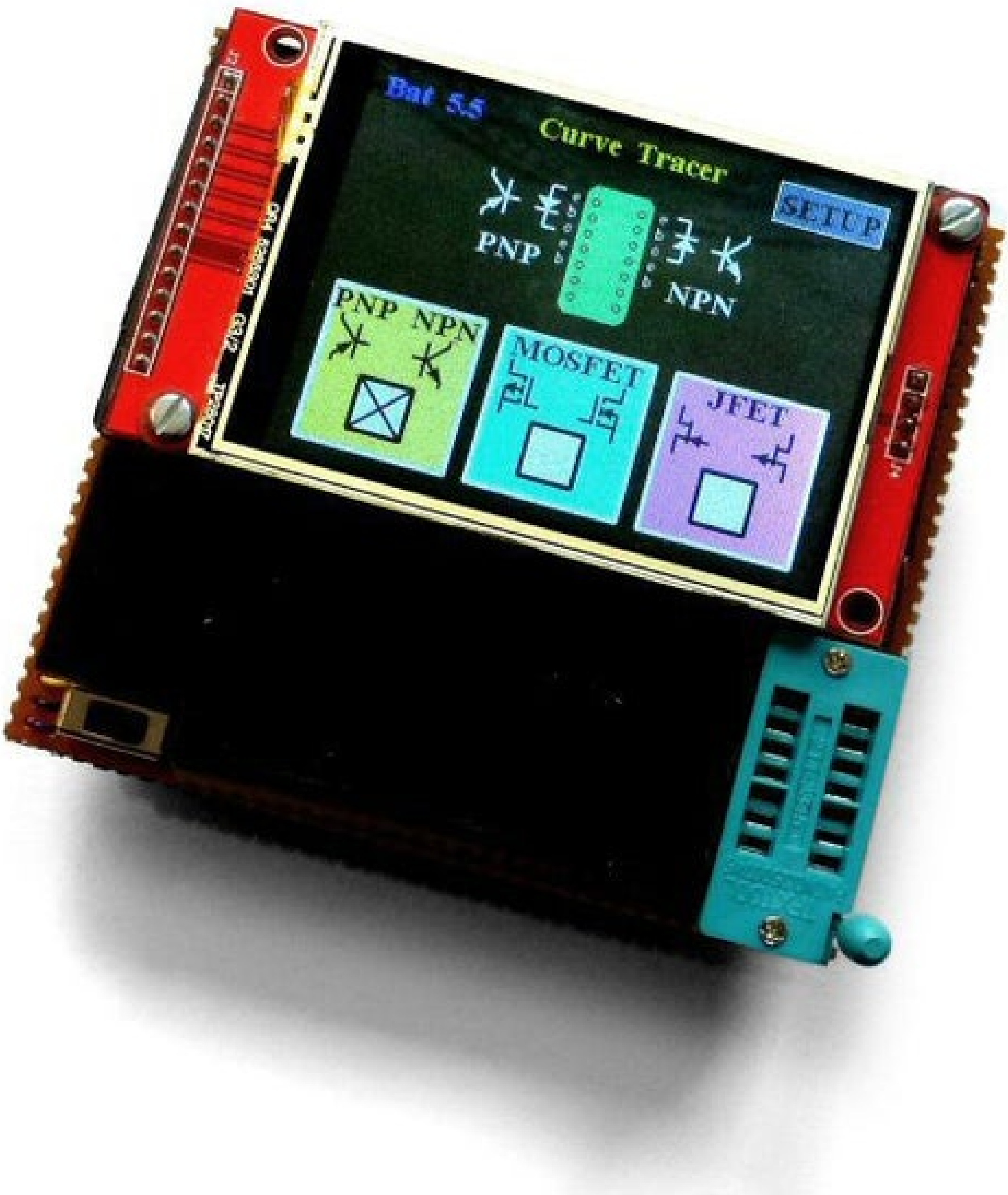
It's useful for

- matching transistors
- measuring the gain of bipolar transistors
- measuring the threshold of MOSFETs
- measuring the cutoff of JFETs
- measuring the forward voltage of diodes
- measuring the breakdown voltage of Zeners
- and so on.

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.


PNP gain=123
300uA
250uA
200uA
150uA
100uA
50uA



Step 1: How to Use It

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 Vce (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 Vds (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.

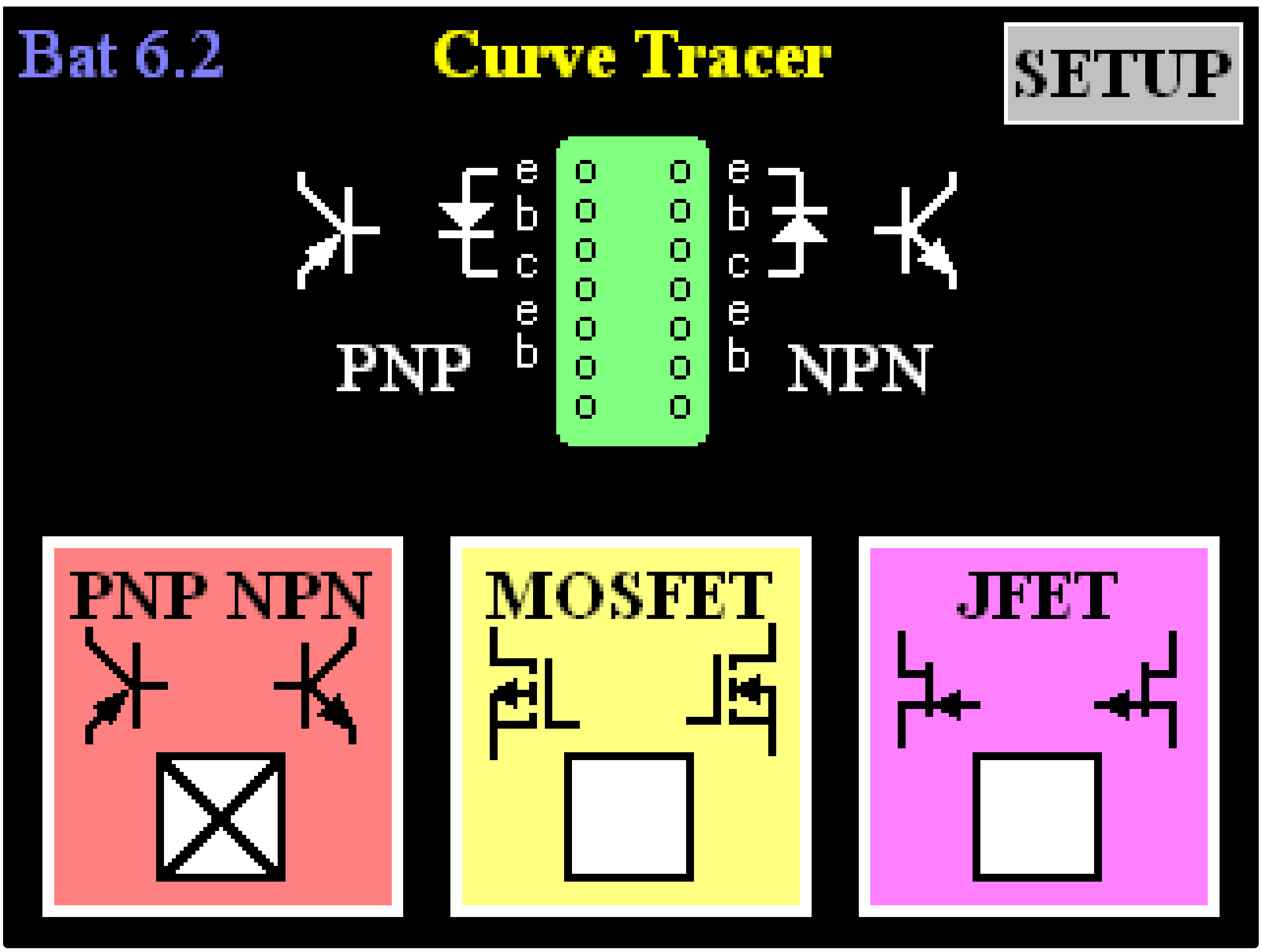
For a JFET it plots Vds (the voltage between the drain and source) versus the current flowing into the drain. A line is drawn for each different gate voltage - 0V, 1V, 2V, etc. With depletion JFETs, current flows when the gate voltage is equal to the source voltage. As the gate voltage is changed to 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.

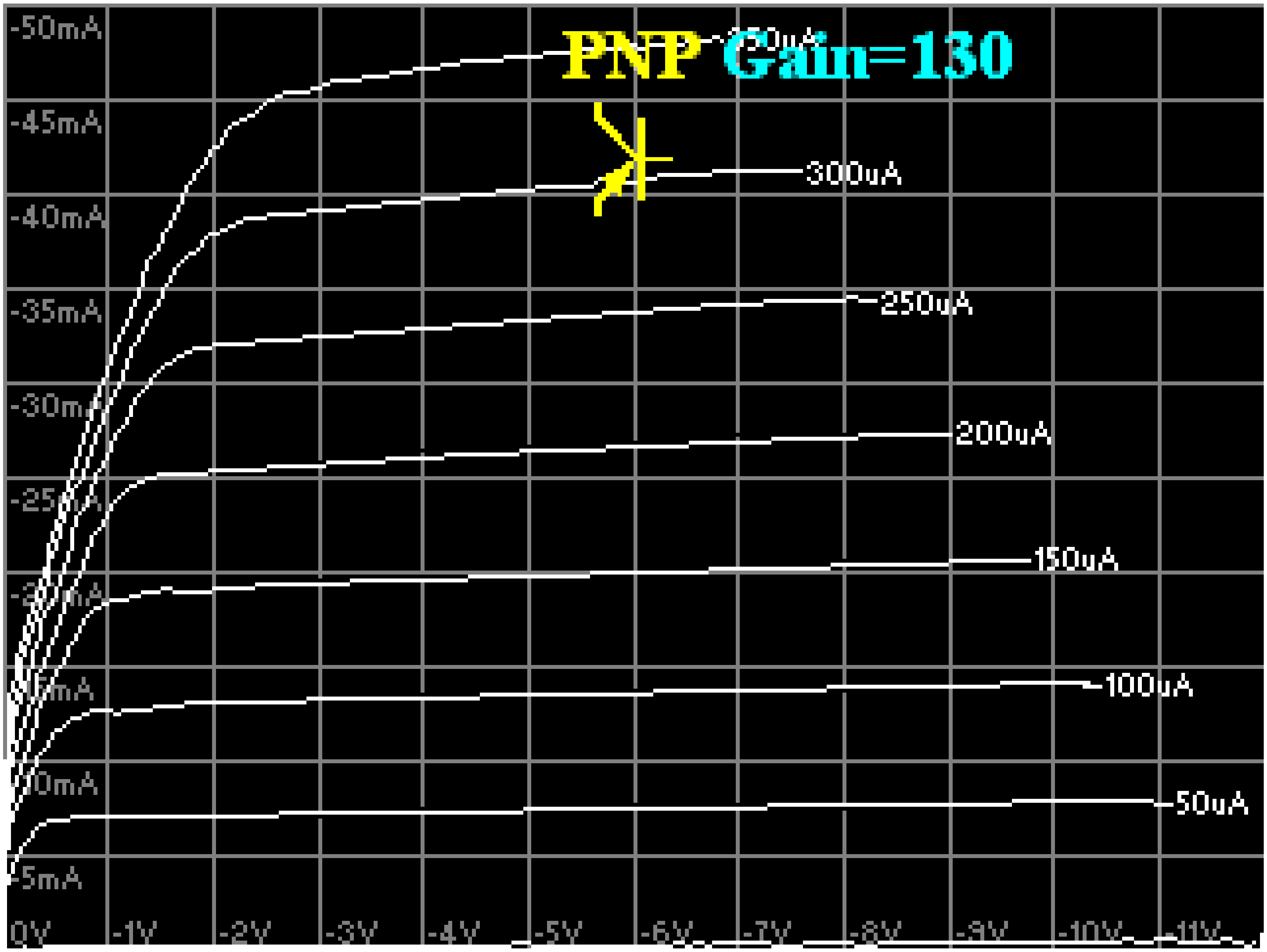
For a PNP or NPN transistor, the Setup screen allows you to select the minimum and maximum base current

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.

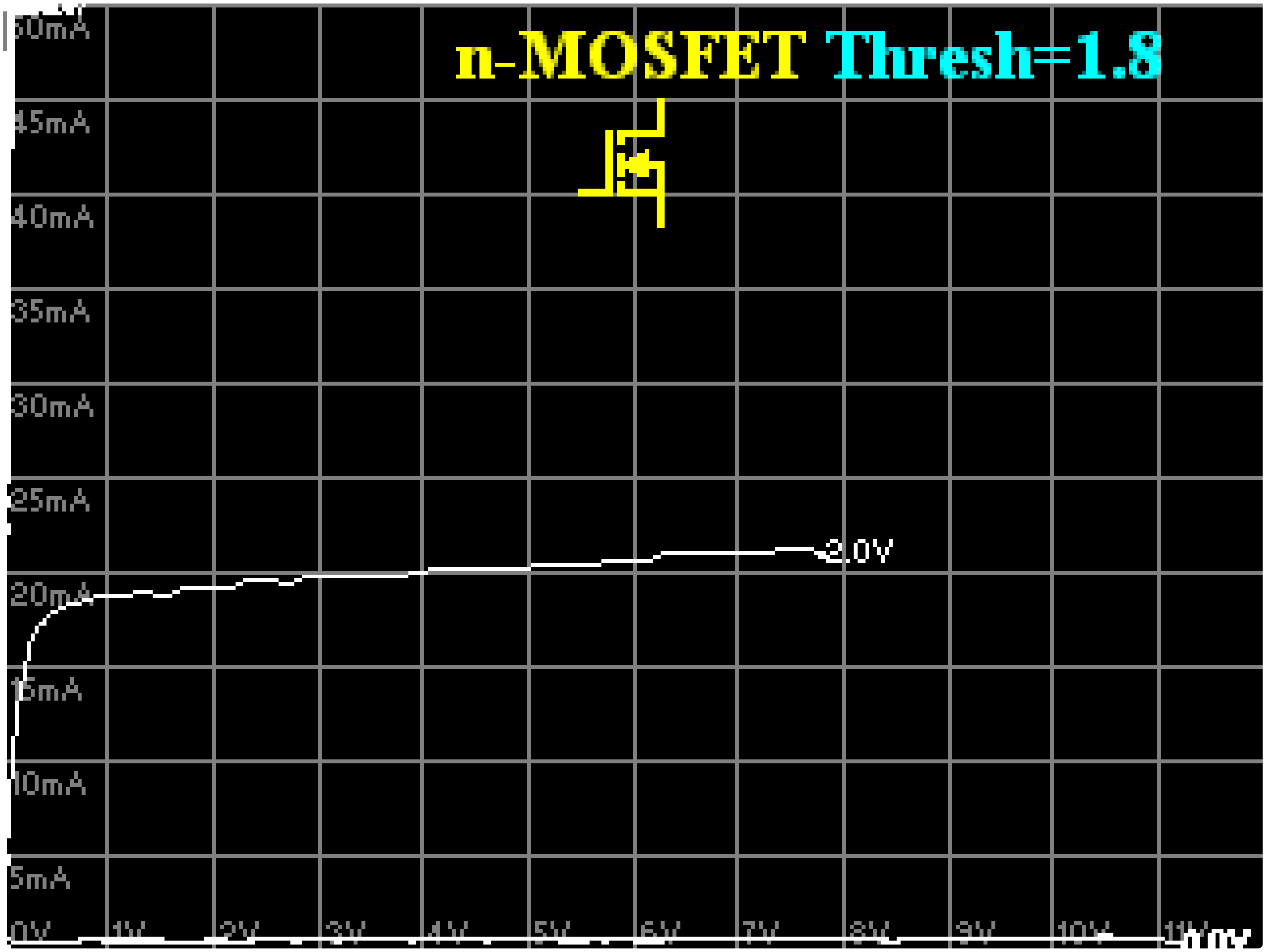
Main Menu



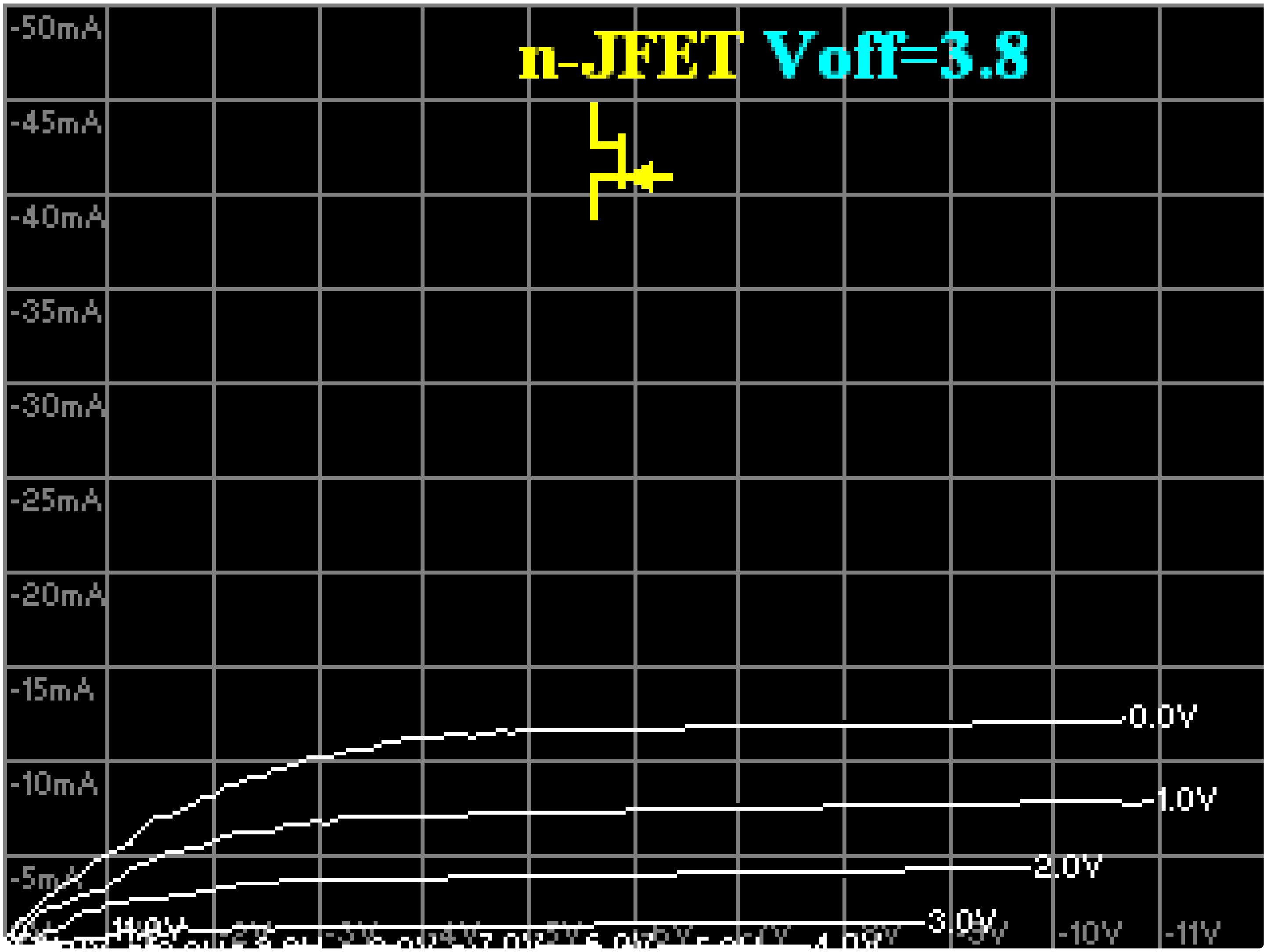
Curves for PNP



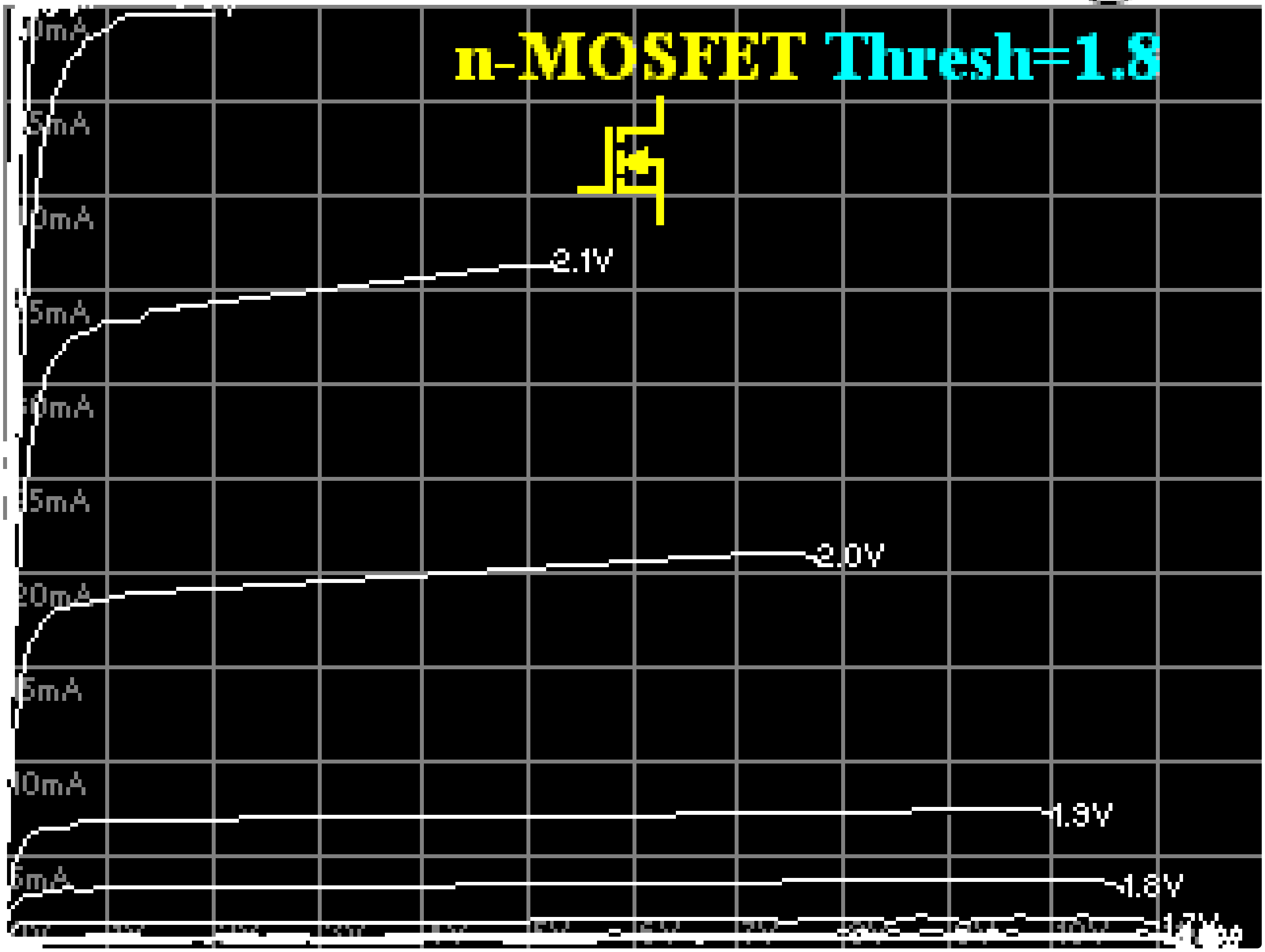
Curves for MOSFET



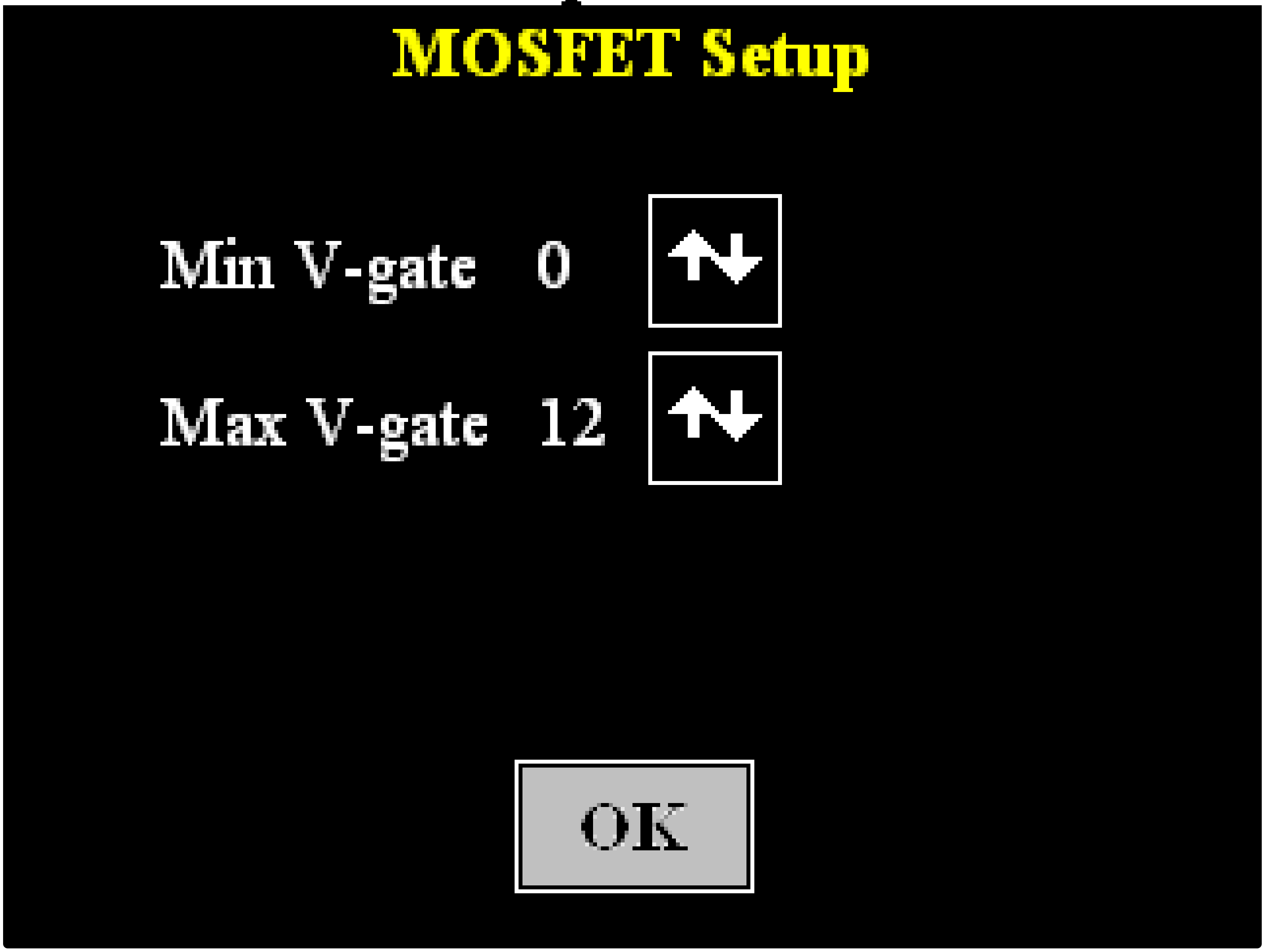
Curves for JFET

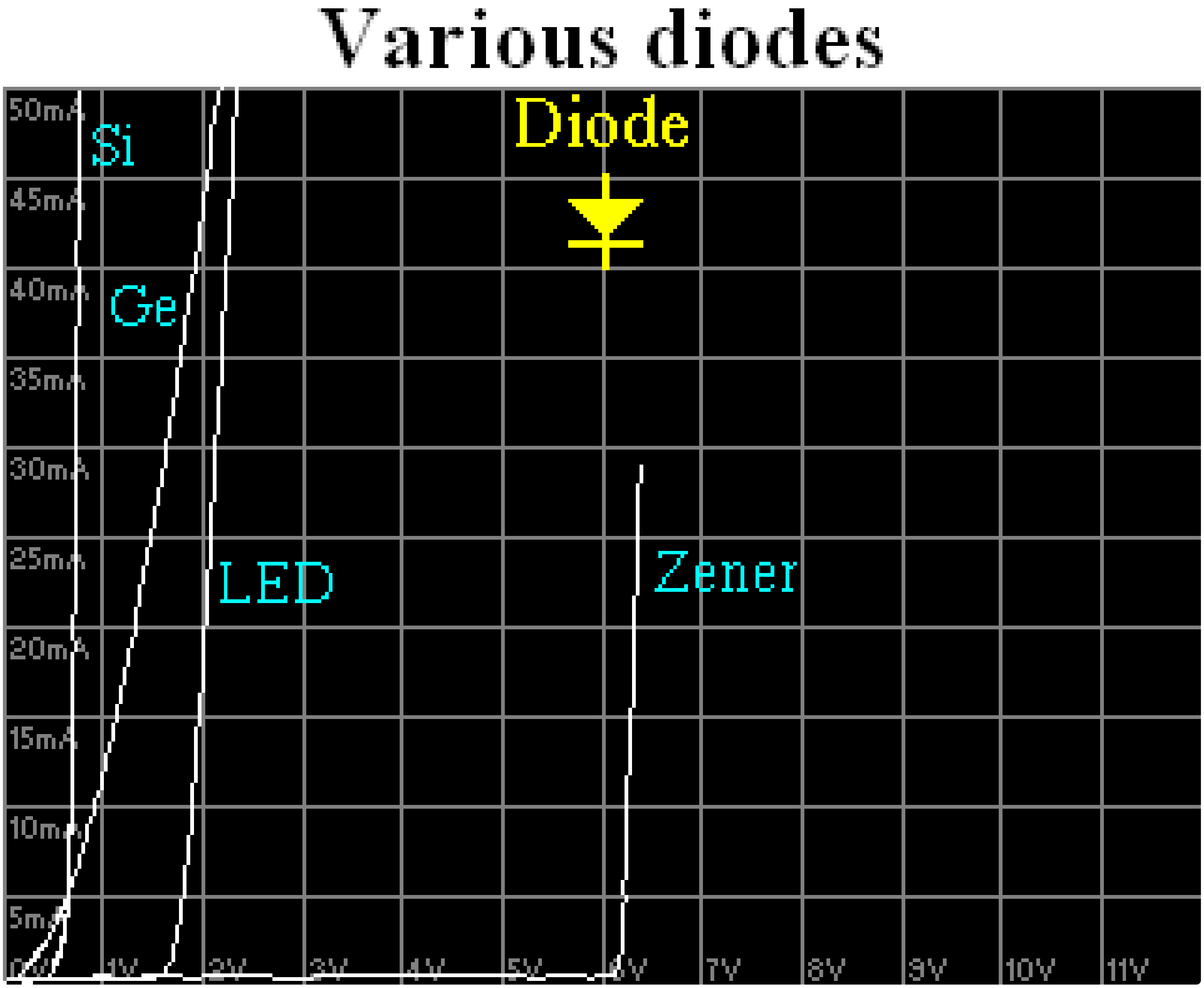


MOSFET in Threshold Region



Setup Menu





Step 2: How It Works

Let's consider an NPN transistor. We're going to draw a graph of the voltage between the collector and emitter (x-axis is Vce) versus the current flowing into the collector (y-axis is Ic). We'll draw one line for each different base current (Ib) - 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 Vbe - 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 an 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 Vgs (the gate-source voltage) to step from 0V, -1V, -2V, etc. We want Vgs to remain constant as Vds (the drain-source voltage) varies. So the Arduino sets the voltage at the load resistor then adjusts the gate voltage DAC until Vgs 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.)

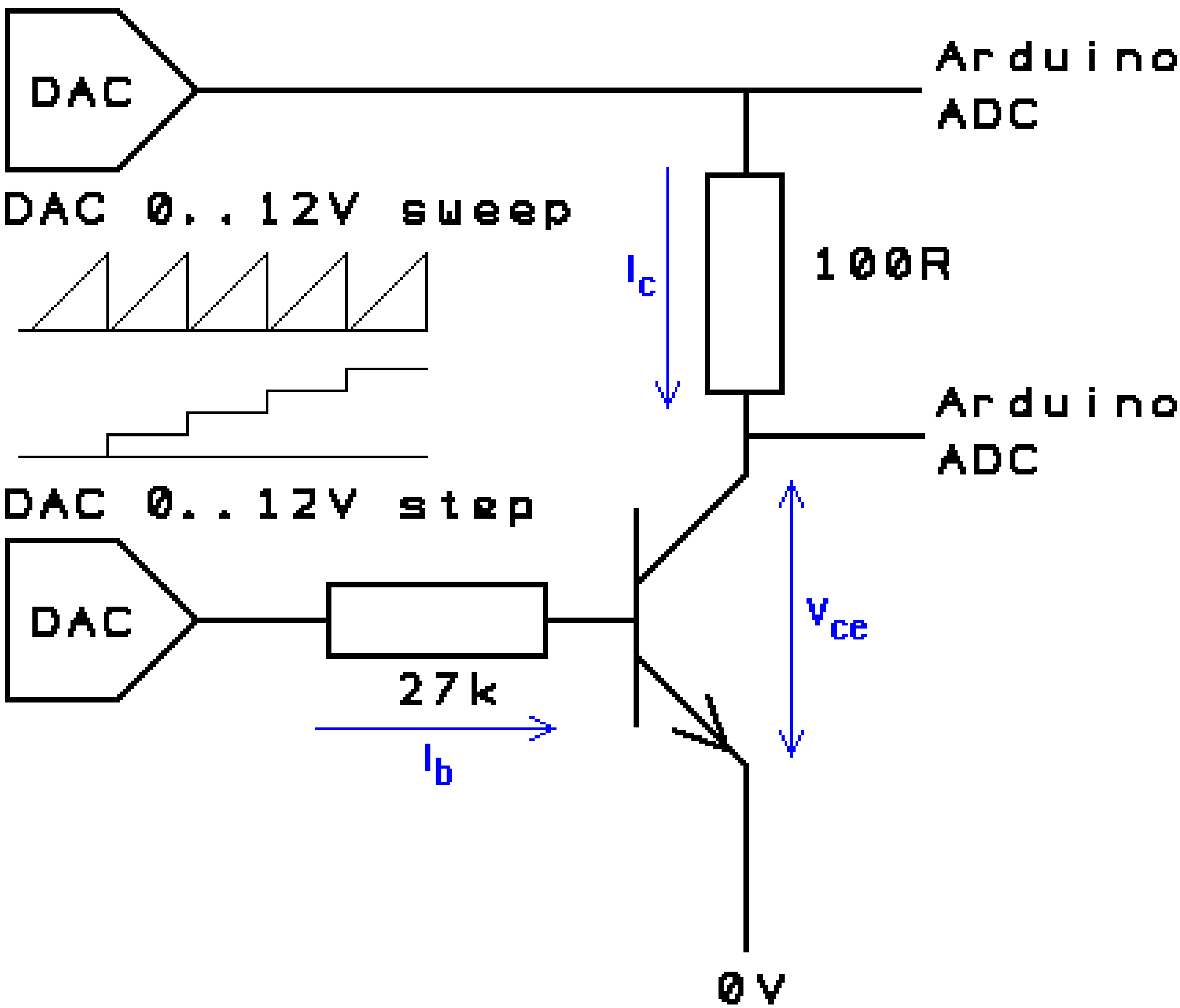
Once it has completed the graph the curve tracer calculates the gain, threshold or cut-off of the transistor.

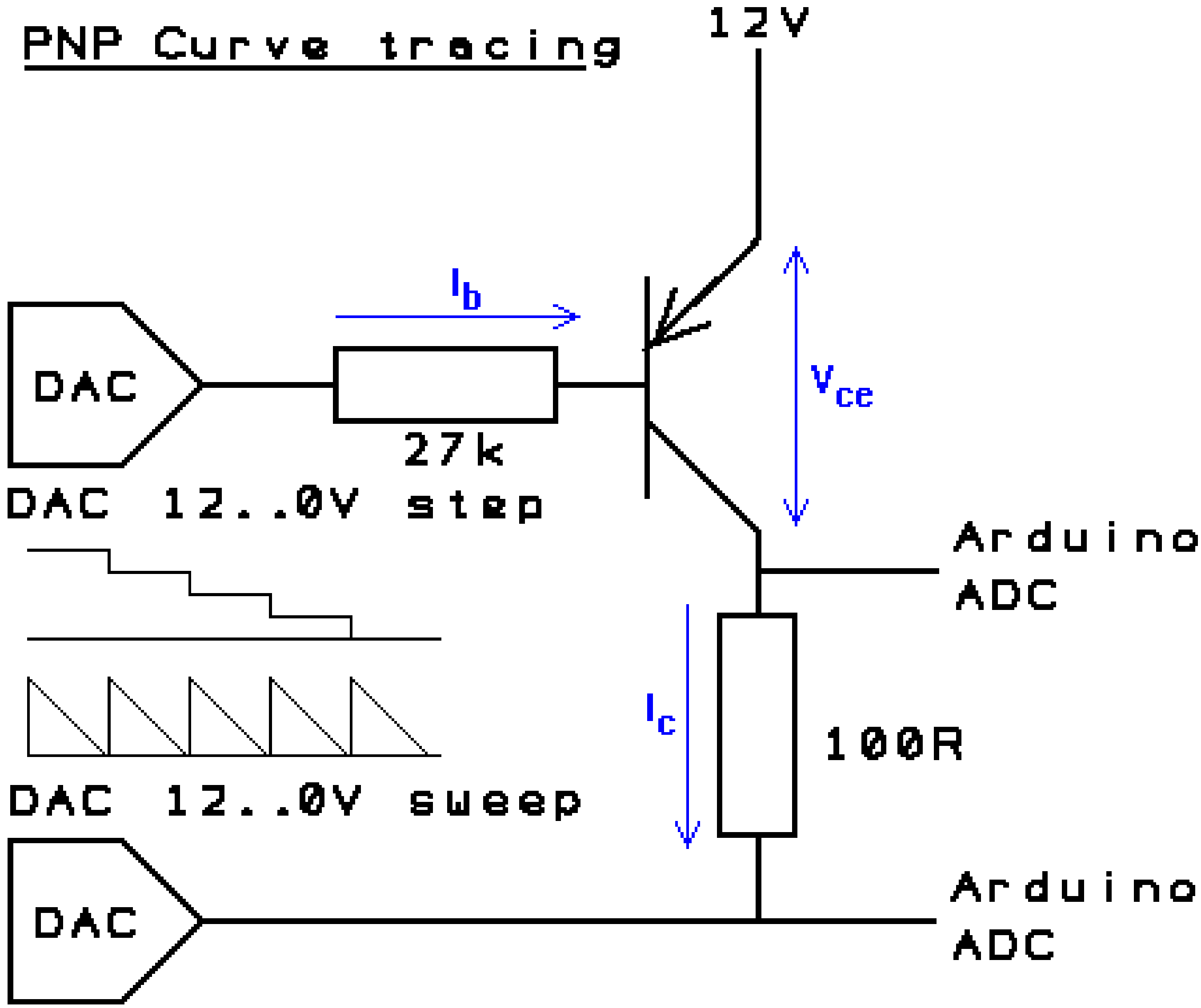
For bipolar transistors, the Arduino looks at the average spacing of the horizontal lines of the curves. As it draws the curve for base current, it notes the collector current when Vce is equal to 2V. The change in collector current is divided by the change in base current to give the gain. The gain 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 Vgs until the current through the load exceeds 5mA.

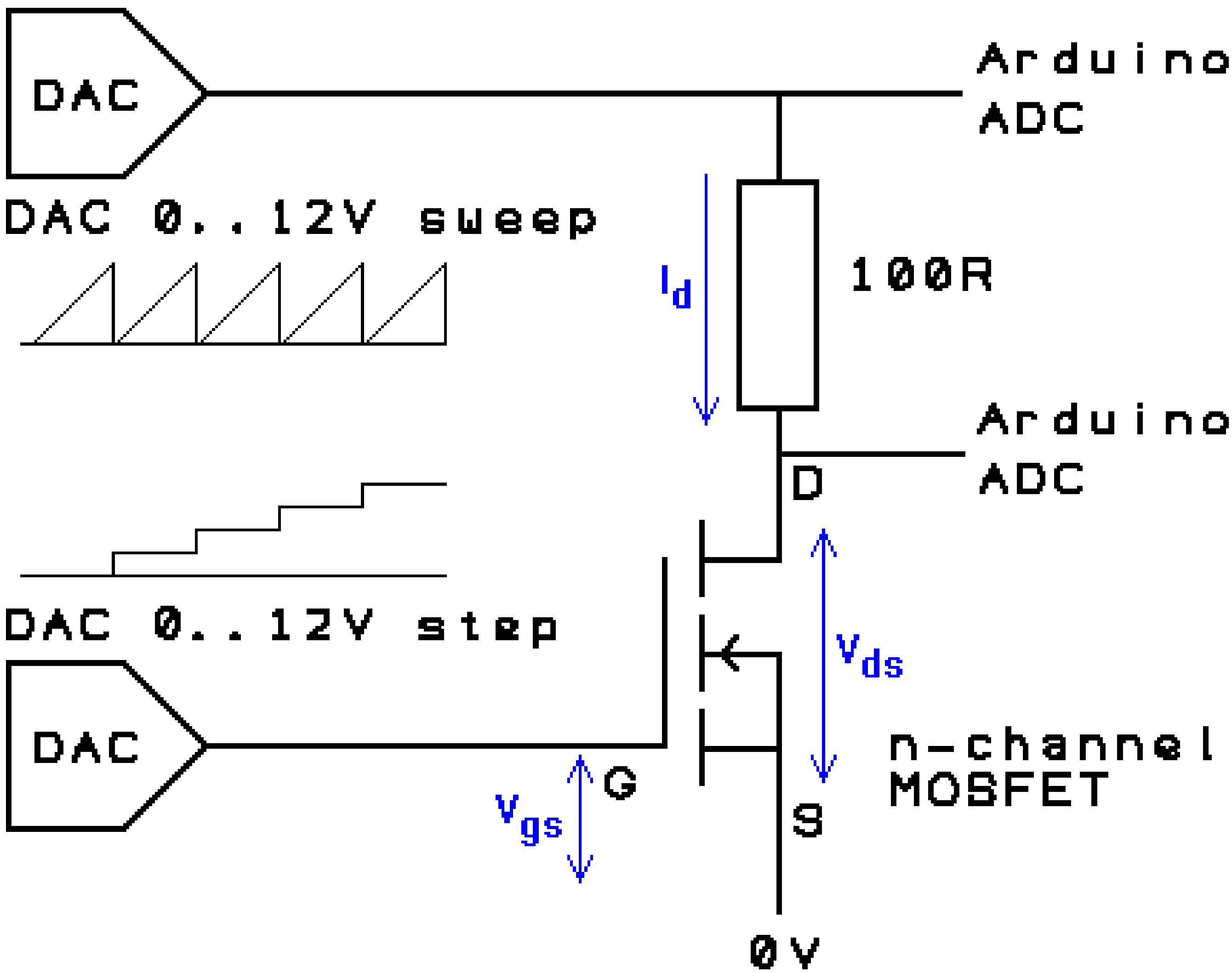
For JFETs, the Arduino measures the cut-off voltage. It sets the load voltage to 6V then gradually increases (negative) Vgs until the current through the load is less than 1mA.

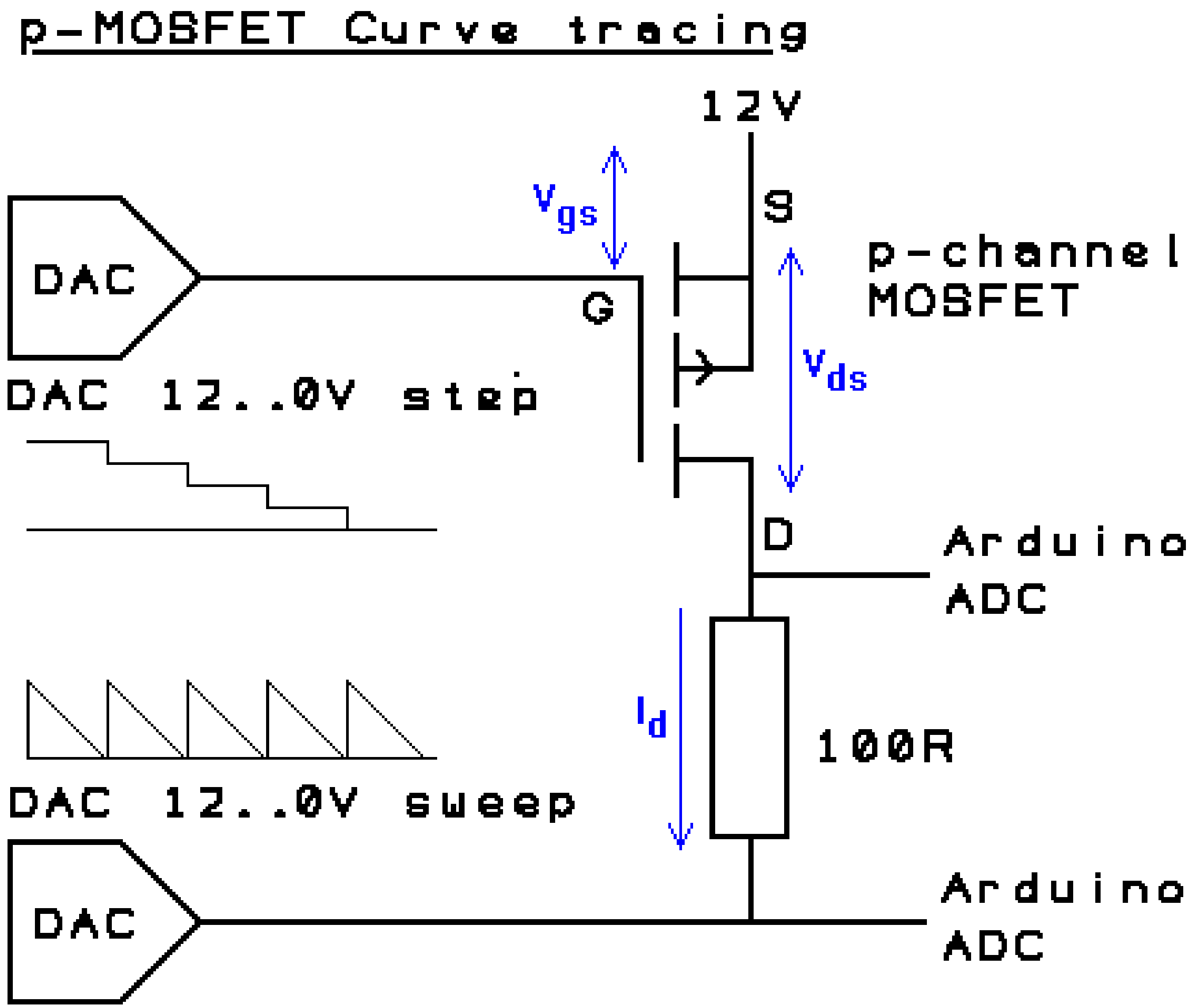
NPN Curve tracing

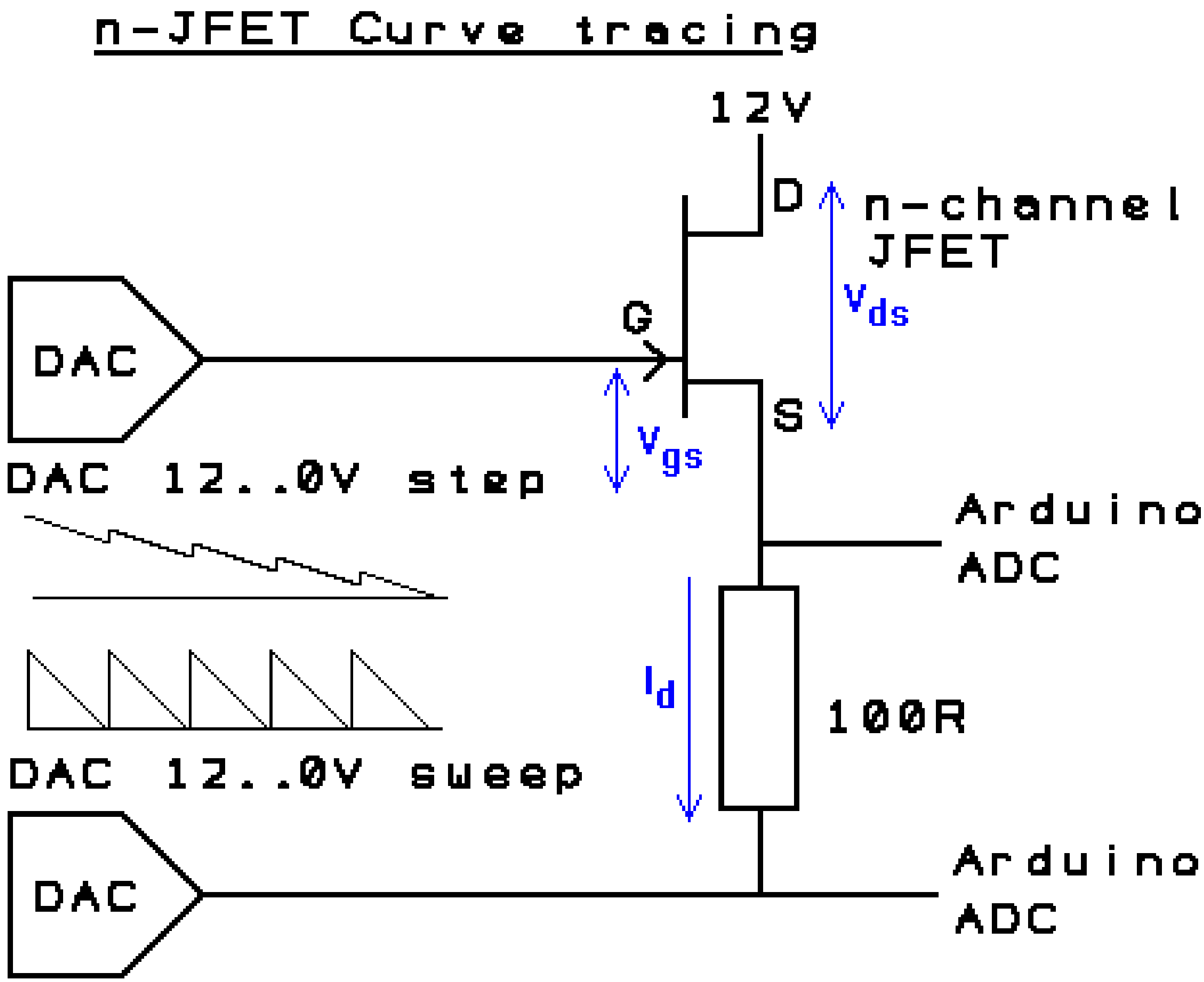


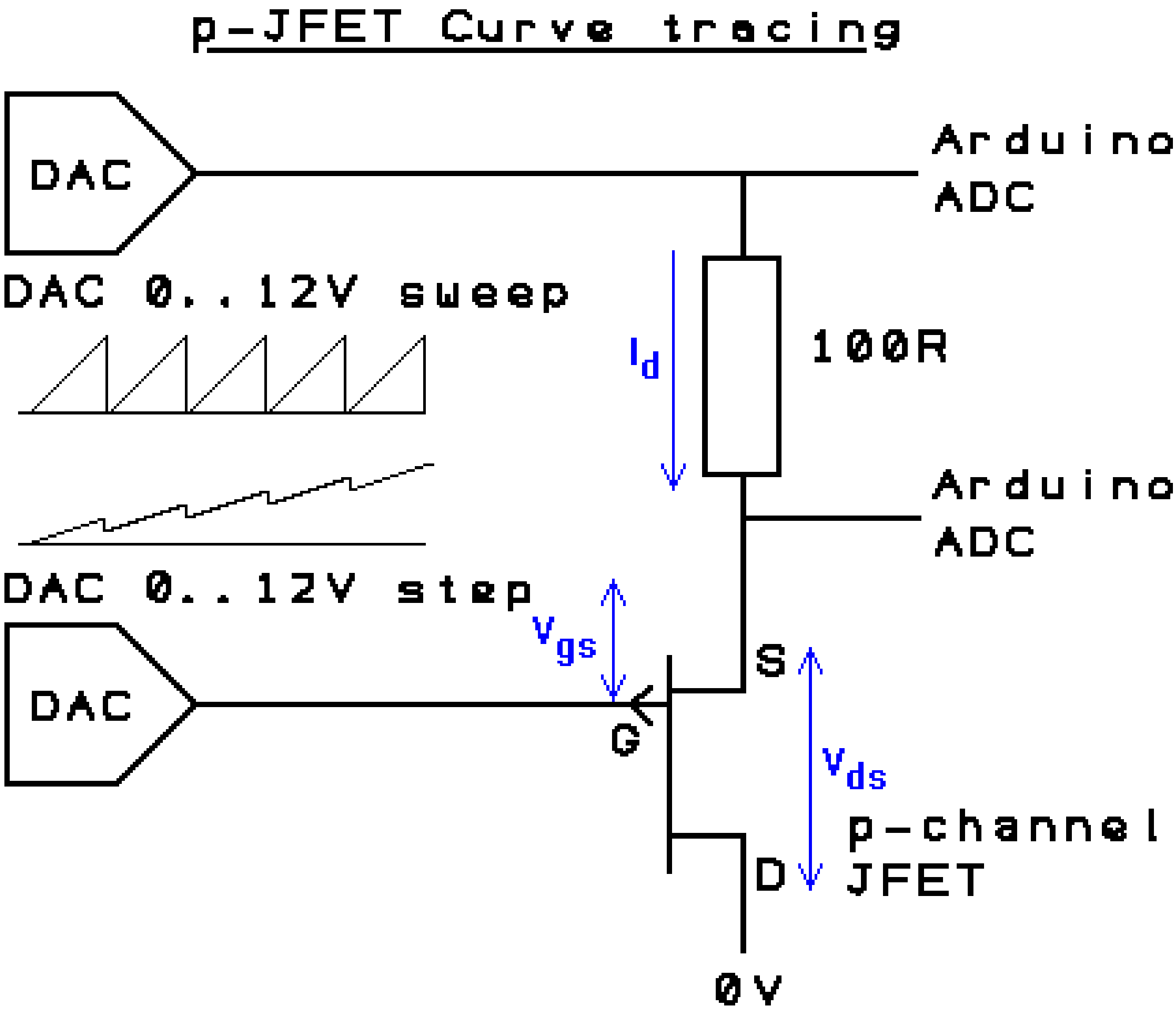

PNP Curve tracing
12V
I_b
DAC
27k
V_{ce}
DAC 12..0V step
Arduino
ADC
I_c
100R
DAC 12..0V sweep
Arduino
ADC
DAC

n-MOSFET Curve tracing

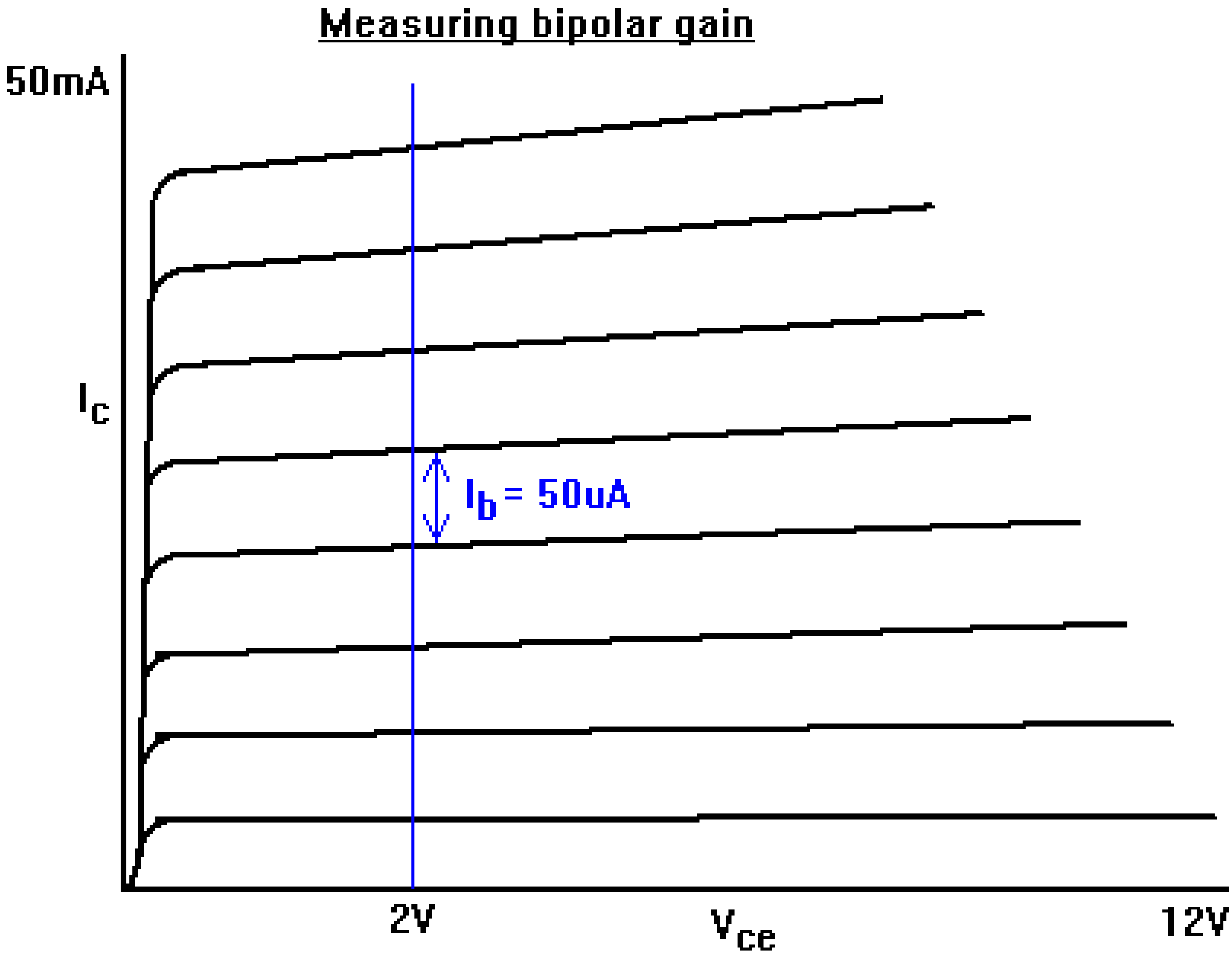



p-MOSFET Curve tracing
12V
V_{gs}
S
DAC
p-channel
MOSFET
G
V_{ds}
DAC 12..0V step
D
Arduino
ADC
I_d
100R
DAC 12..0V sweep
DAC
Arduino
ADC


n-JFET Curve tracing
12V
D
n-channel
JFET
V_{ds}
G
DAC
S
V_{gs}
DAC 12..0V step
Arduino
ADC
I_d
100R
DAC 12..0V sweep
DAC
Arduino
ADC



p-JFET Curve tracing
DAC
Arduino
ADC
DAC 0..12V sweep
I_d
100R
Arduino
ADC
DAC 0..12V step
V_{gs}
S
DAC
G
V_{ds}
p-channel
JFET
D
0V



Step 3: The Circuit

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 curve tracer is powered from 4 AA cells which give 6.5V when new and can be used down to around 5.3V.

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:

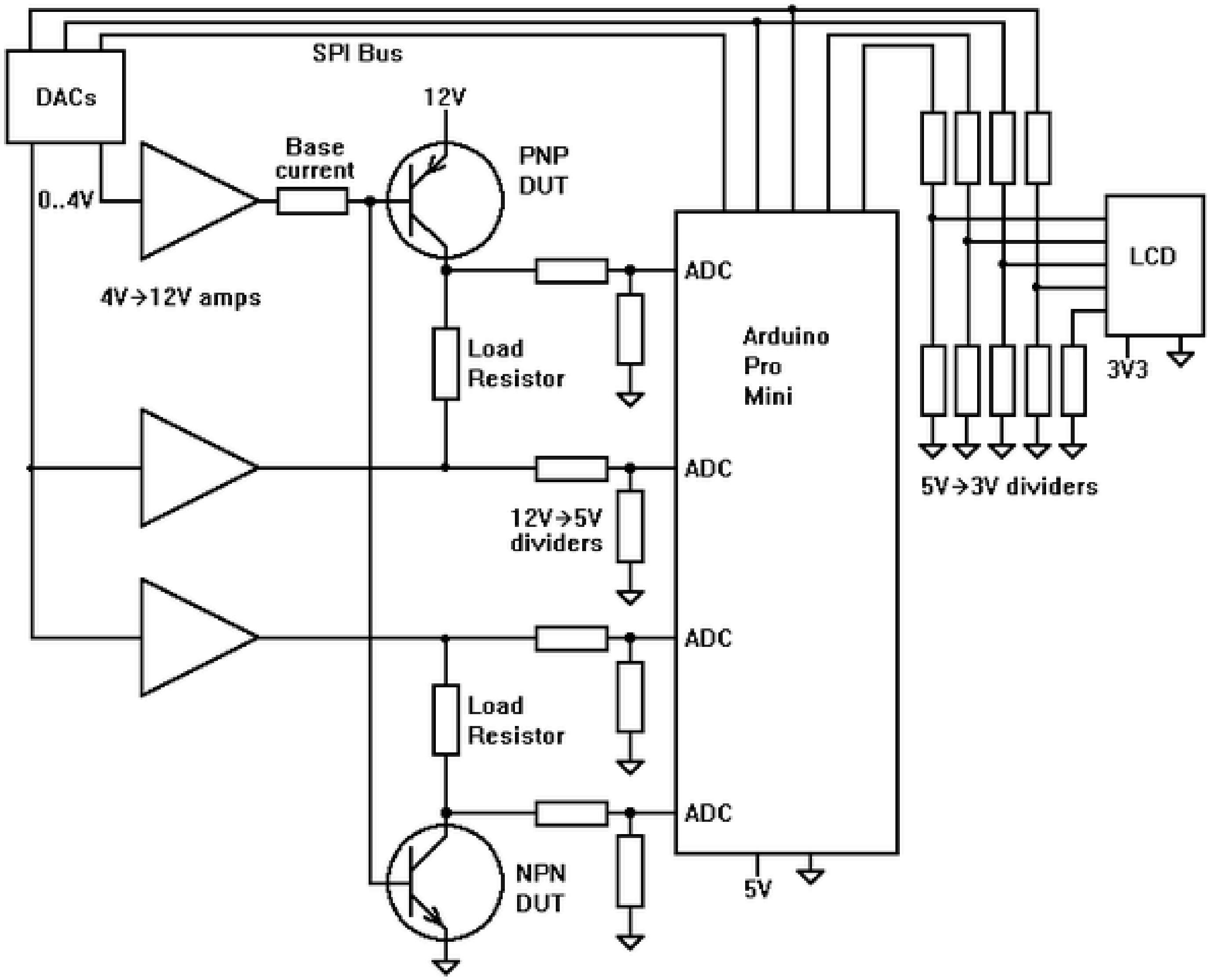
https://www.ebay.co.uk/itm/Boost-Voltage-Regulator-Converter-Step-up-Power-Supply-DC-3-3V-3-7V-5V-6V-to-12V/272666687043?hash=item3f7c337643%3Ag%3AwsMAAOSw7GRZE9um&_sacat=0&_nkw=DC+3.3V+3.7V+5V+6V+to+12V+Step-up+Power+Supply+Boost+Voltage+Regulator+Converter&_from=R40&rt=nc&_trksid=m570.l1313

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.

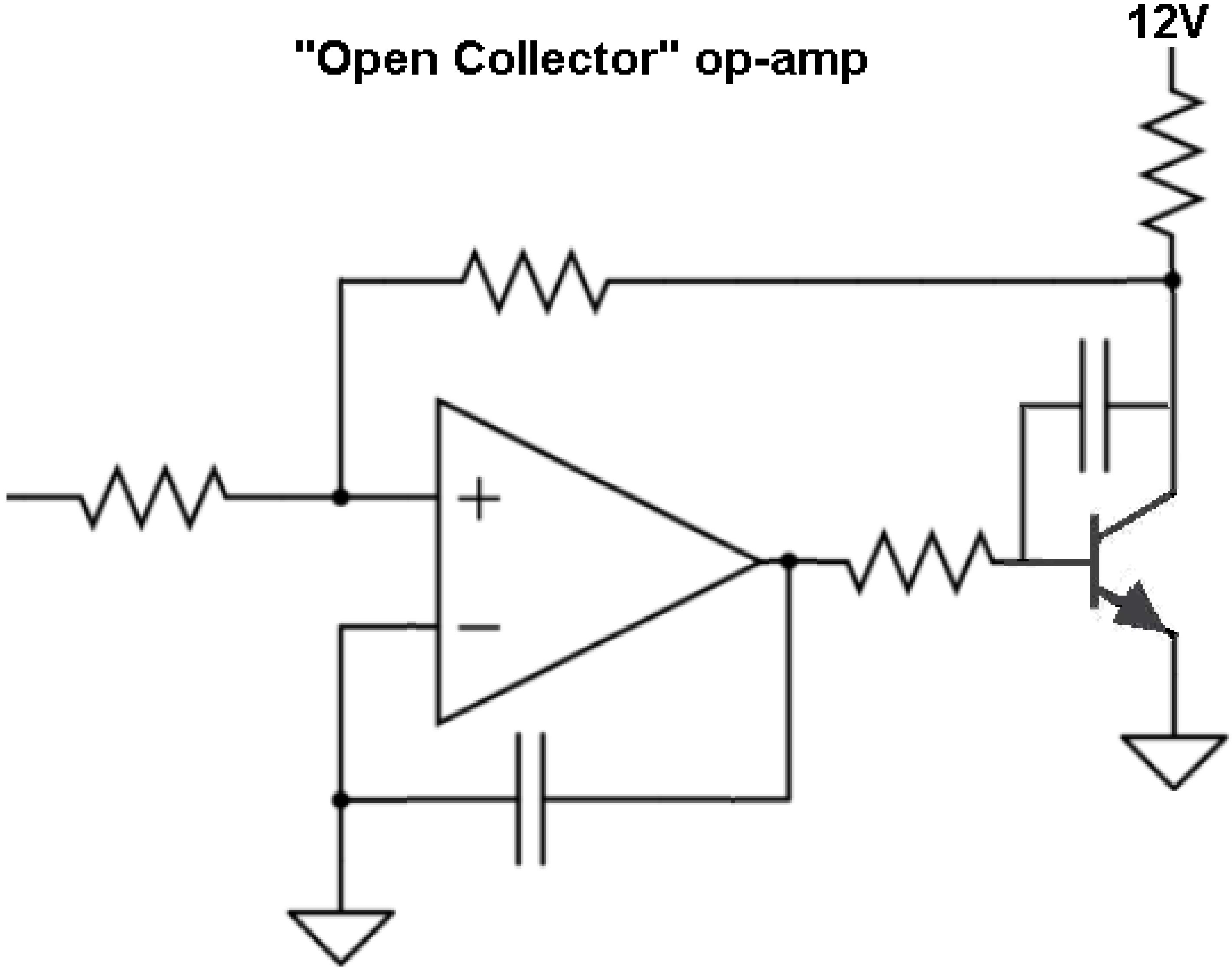
As the AA cells are discharged the voltages vary, the software compensates by using reference voltages. The Arduino measures the 12V supply. The Arduino ADC uses its "5V" supply as a reference voltage but that "5V" is calibrated accurately against the Arduino's internal 1.1V reference voltage. The DAC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

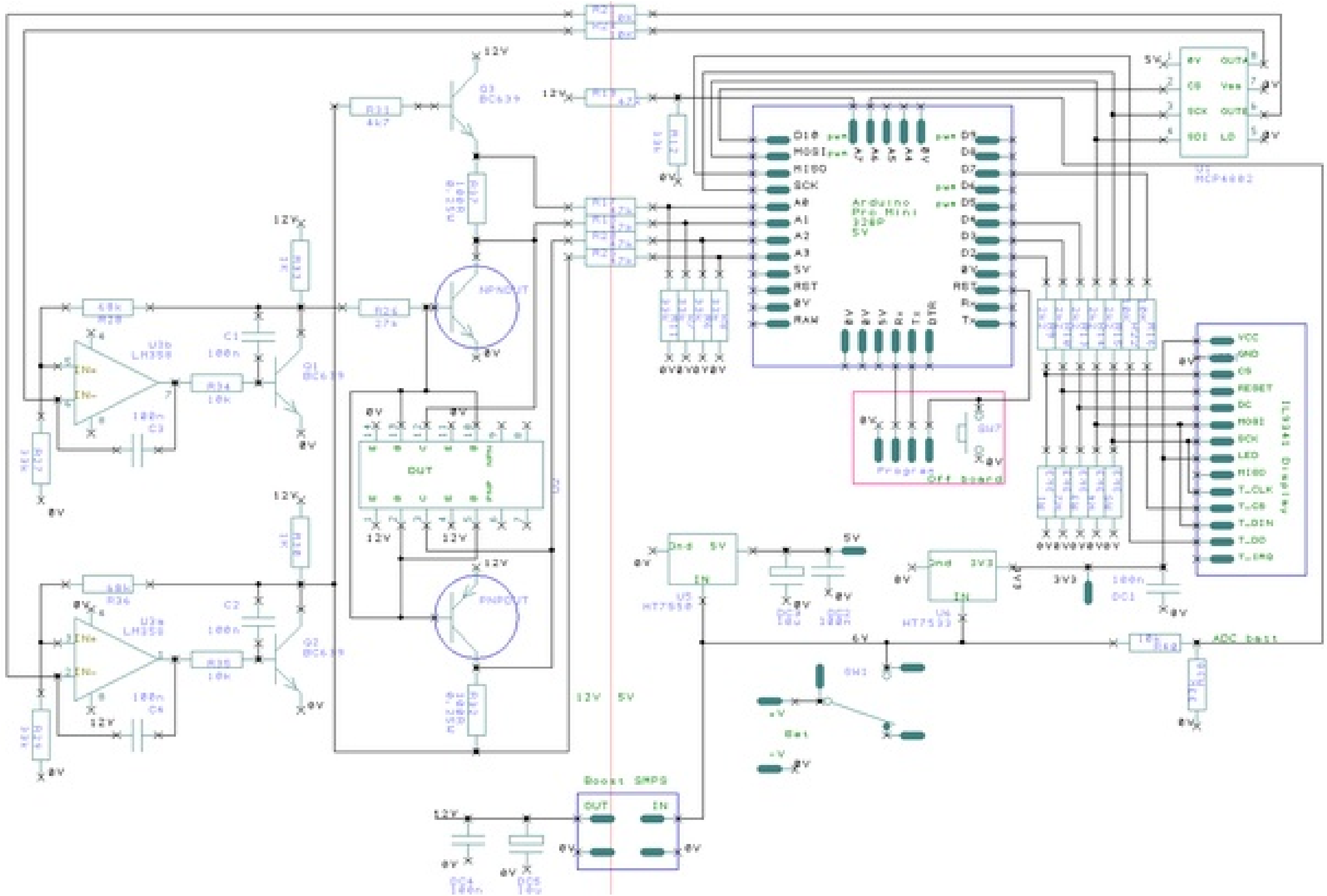
I like the way the LCR-T4 has a button to turn it on and turns itself off automatically with a timeout. Unfortunately, the circuit introduces a voltage drop which I can't afford when powering from 4 AA cells. Even re-designing the circuit to use a FET wasn't sufficient. So I'm using a simple on/off switch.

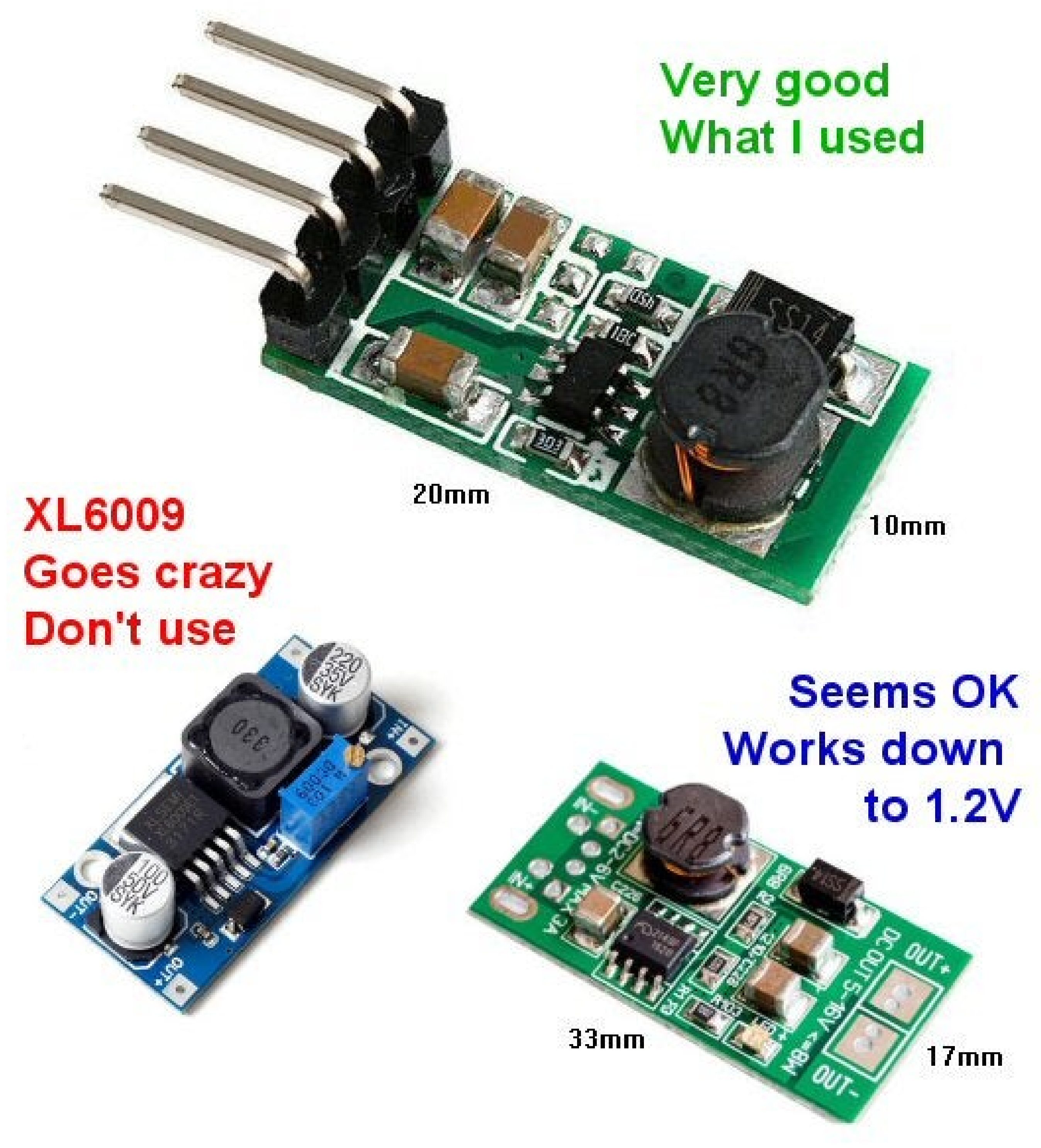
Block Diagram



"Open Collector" op-amp







Download

https://www.instructables.com/ORIG/FSY/QEAD/JL6UXLES/FSYQEADJL6UXLES.zip

Download

https://www.instructables.com/ORIG/FJA/H3N3/JL6UZT5N/FJAH3N3JL6UZT5N.pdf

Step 4: The Software

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, maye 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.

Download

https://www.instructables.com/ORIG/F80/OBXC/JJT6TALL/F80OBXCJJT6TALL.zip

Download

https://www.instructables.com/ORIG/FJZ/TPLY/KUQYGQ4D/FJZTPLYKUQYGQ4D.ino

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-A 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 avreraging; you should do the same.

When the PC sends an "X" command, The ADC values are returned as integers:

- "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.

Download

https://www.instructables.com/ORIG/FCF/0NK7/JJZN5QGI/FCF0NK7JJZN5QGI.zip

Step 6: Building the Curve Tracer

Here are the major components that you'll probably need to buy:

- 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

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

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 ballpoin 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.

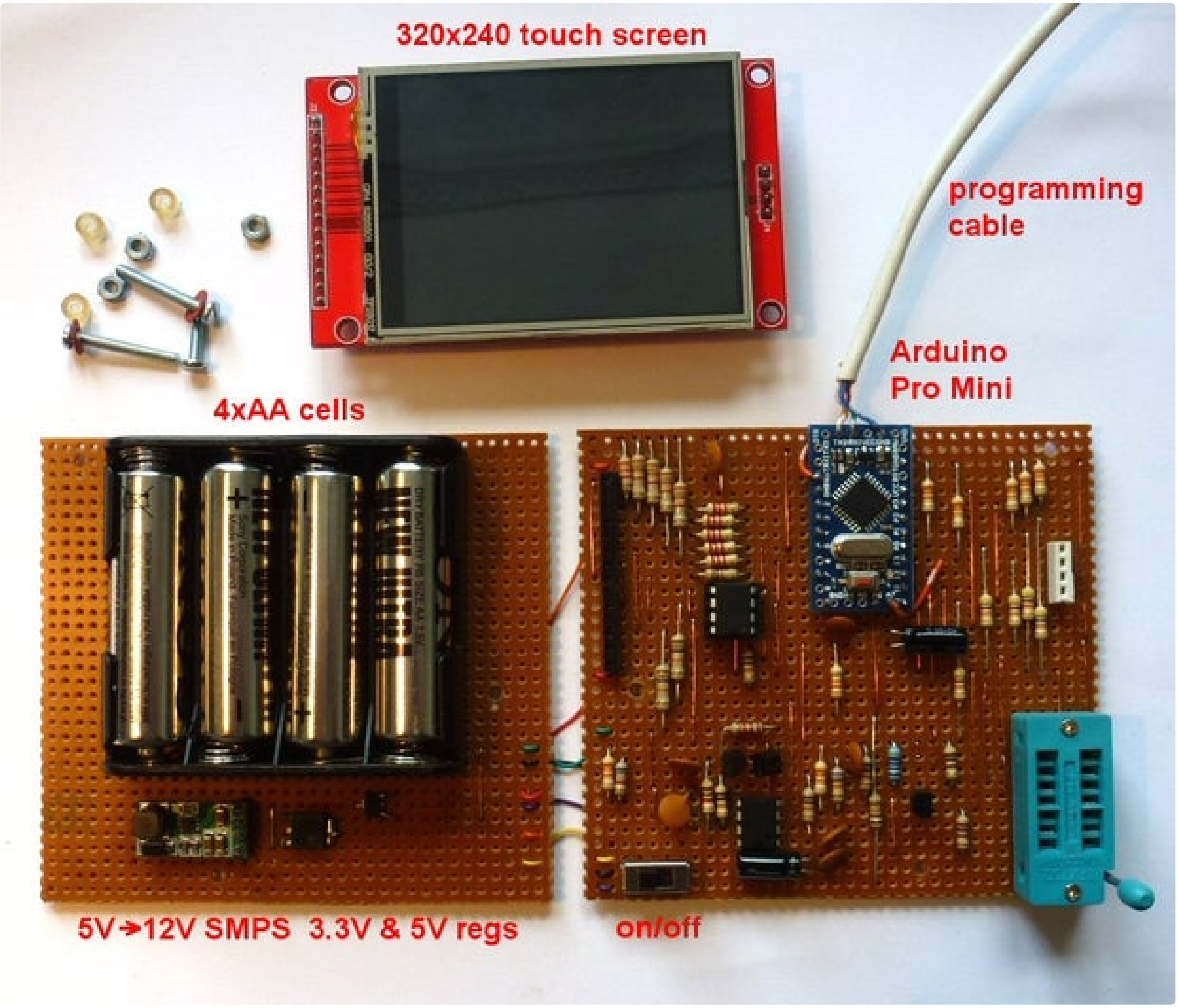
The LCD contains a socket for an SD card (which I didn't use). It is connected to 4 pins on the pcb. I used the pins and a piece of SIL header or IC socket to help support the LCD.

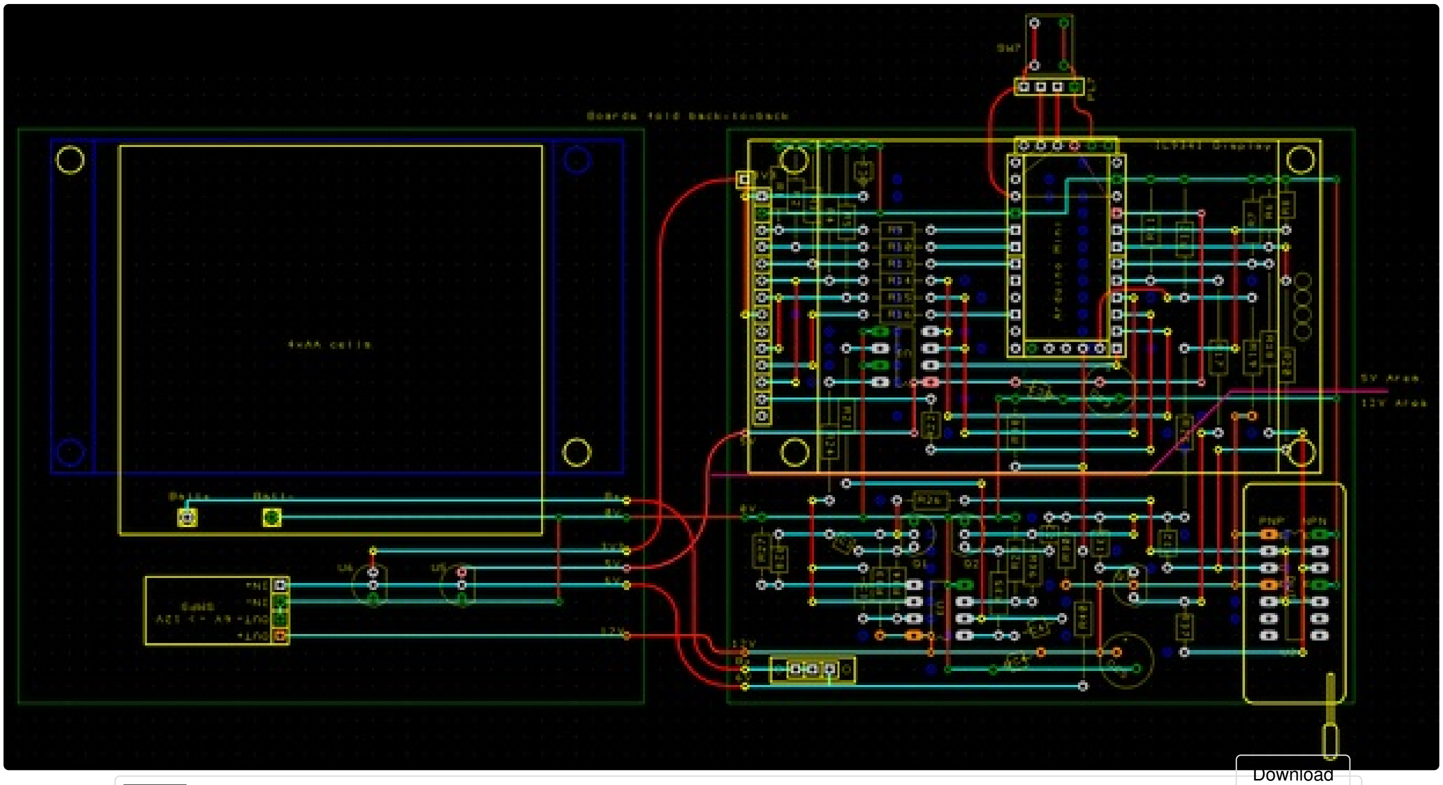
Notice that there are some links under the ZIF socket. Solder them in before you fit it.

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.


320x240 touch screen
programming
cable
Arduino
Pro Mini
4xAA cells
5V→12V SMPS 3.3V & 5V regs
on/off



https://www.instructables.com/ORIG/FMQ/5RKR/JJT6T88I/FMQ5RKRJJT6T88I.zip

Download

https://www.instructables.com/ORIG/FO5/XH22/JL6UZT94/FO5XH22JL6UZT94.pdf

Download

Step 7: Future Development

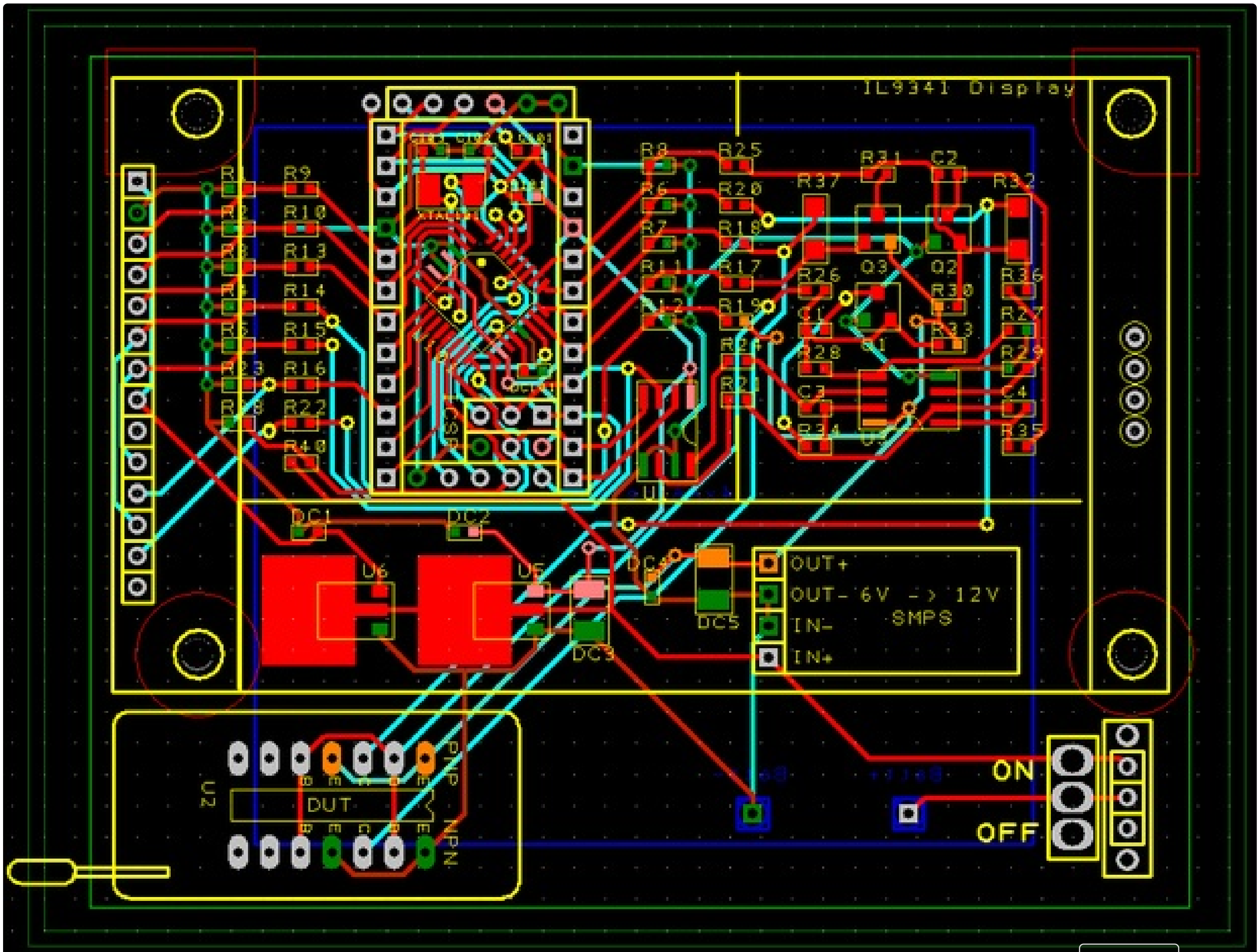
It might be nice to produce curves for other components but which? It's not clear to me what extra infomation the curve of a thyristor or triac would tell me beyod what the LCR-T4 tester does. The LCR-T4 tester can even be used with opto-isolators. I've never used a depletion MOSFET or an enhancement 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.



Download

https://www.instructables.com/ORIG/F8Q/W0DV/JJT6T82Y/F8QW0DVJJT6T82Y.zip

hi Peter

would really want to build this. need for matching jfets in phaser guitar pedal. only problem is I cant order online the mcp chip. I have a couple PCF8591 modules. can this be used here?

The MCP4802 is an 8-bit DAC controlled by SPI.

The PCF8591 contains a DAC and four ADCs and is controlled by I2C. I presume it uses completely different commands.

I've never used a PCF8591 but it looks like a nice chip. It looks like it would work well in the circuit.

Do you have the programming skills to make the changes?

so nice that it could be done.

I have no problem hacking hardware stuff but I suck at coding . lol.

would it be a lengthy code hack?

Yes probably a couple of days. Why can't you order the MCP4802 - there seem to be lots on eBay, Alibaba, Mouser, Digikey, etc.

Peter

Have another slight problem. I have tft 2.4" SPI, but no touch function. also have a bigger 3.2" (model:YX32B) has touch, but NOT SPI. can I still use these?

> I have no more credit card.

Maybe you can find a friend to buy stuff on your behalf.

> tft 2.4" SPI, but no touch function.
> 3.2" (model:YX32B) has touch, but NOT SPI. can I still use these?

The YX32B needs lots of pins - is it a 16-bit parallel interface? The Arduino just doesn't have enough pins.

It would be easier to change the program to use push-buttons rather than a touch screen. But if you're not a programmer then you're stuck with just copying the project as it is.

If all you are wanting is to match JFETS then google for "match jfets circuit".

There are lots of simple circuits that will do an approximate match.

Or you can google "matching jfets phaser guitar" for dicussions of how important matching is.

Good luck

Peter

yes I guess that lcd needs lots of pin connection. would better with using SPI lcd & physical buttons.

I have no more credit card. I now rely solesly on COD and sites like shopee(w/c is not really an electronic store) but some sellers there do sell arduino modules and the likes. sad but true

Very nice project Peter! Do you know why my antivirus detect something wrong in your " SimpleILI9341" library?

I had the same issue too, it's the font converter, presumed to be infected with some malware. Anyway if you want to make your own fonts and bitmaps you can use several converters. I'm using http://javl.github.io/image2cpp
, or : http://www.riuson.com/lcd-image-converter

The zip file contains an exe and several source files.

The source files are plain text so cannot contain a virus.

I've downloaded the zip from this site and compared the exe with my original and they're identical. My virus checker doesn't find any virus in either copy.

Is you anit-virus software just complaining that he zip file contains an exe? Not that the exe contains a virus?

Hey Peter, thank you for fast replay, you right!! I removed the exe from zip and all is fine except one more thing. I got an error as "Invalid library found in my directory : Missing 'url' from library after uploading the code. I searched for this error and i found that need insert an expression like url= "a website" but i didn't understand where i should insert it. Have any idea?

That doesn't happen with the version of the Arduino IDE I'm using (ver 1.8.9).

Googling for

arduino Missing "url" from library

gives this as the first result

https://forum.arduino.cc/t/what-does-missing-url-from-library-error-mean-adding-external-libraries/653913

It tells you how to fix the error. The other hits give the same advice.

Maybe it's a new problem with the IDE.

Thank you, Peter for your time! I followed the instruction with more attention from your link and i removed the caution.

Thanks again! :D

A.H.

Very Nice project. A few days ago I started looking at buy/build me a curve tracer. Found several nice simple ones, nothing wrong with any of designs. Your DYI design out classes all I have found so far. Interesting is I started a design few days back, would look very close to yours. Only main difference was choice of processor/display. I was thinking of using ESP32, Bagel Bone Black or Pie, mostly because I have several of each. Any thoughts on what you might change now.

The LM358 op-amps I used are pretty poor: they don't go rail-to-rail and they're low current. If you could find a 12V rail-to-rail op-amp that will supply 200mA that would be better. Maybe using audio amp (speaker driver) chips?

I'm not sure about the way I did the FET curve tracer. Maybe you can think of a better way.

And, as I say in Step7, it would be nice to incorporate an LCR-T4 tester.

Good luck (and write an Instructable when youre done)!

Peter

Thank you Peter, great project.
It works like a charm. I have made few adjustments to my needs: I feed it with 5V via a barrel connector (I have a lot of 5V battery packs hanging around) and I have used an arduino nano with the USB-to-serial interface on board, to avoid the FTI programmer (and again, to use what I had already at home).
I have 3 spare pcbs if anyone is interested but it was my FIRST pcb design so ... I made three

errors. Nothing big, but a bit annoying.

Again: great work !
Thank
Tiberio


That's great. It looks good. It's really nice, isn't it, when you see the curves just like in a textbook.

> I have used an arduino nano

Yes. I'd do that if I was starting today. I just happened to have a ProMini to hand.

Is it possible for you to post the pcb design and gerbers somewhere? People keep asking about pcbs.

Peter

Hi Peter.
I went for a GitHub repository. Here is a link to it: https://github.com/tiberiov/Curve-Tracer-pcb
First time I use GitHub, but it should be working ok.
I have 3 boards as well, if someone needs one.
Thank you for the design.
Tiberio

My name is James Ashley and I am very impressed by your curve tracer. How much in US Dollars would you charge me for a fully assembled and programed curve tracer.

Thank you very much
James Ashley

Hi. This is a hobby for me , I'm not really doing this for third parties . I have sent a PCB to other people that wanted to build one. I may still have a PCB (just the PCB). That is something I can ship to you if you're interested. Then you have to buy the components and get it assembled. Where are you based ?

Regards
Tiberio

Hi Tiberio,
Do you still have a spare PCB ?
I want to build one. Thanks in advance

Regards,
Abraham

Hello,
yes I still have one. 5 euro + shipping. One additional euro if you want also the ziff connector and the power supply barrel jack connector (I have them lying around). Depending on your choice postage cost can vary (from flat package to thick package) . Where do you need it shipped ? I can find out for you what would be the postage cost would be.

Keep in mind I'm currently travelling, I would not be able to ship it before end of August.

Regards
Tiberio

Hi,
Okay. How much would be the shipping cost? I'm from Melbourne Australia.

Regards,
Abraham

Hi. Australia! That's a long way! Have you considered downloading the Gerber (I uploaded mine on github) and get it done by one of the Chinese companies? It is likely to cost you the same and you get 5 boards....

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 = Udac/(27k+Rbe) so the current is not exact.

You should measure the basis current step by step.

Thanks
Kris

Maybe.

I don't think that Rbe is a useful concept. The base voltage/current curve is so non-linear you can't treat it as a resistor. As I said, you're better considering Vbe so your equation should be I = (Vdac-Vbe)/27k. Vbe depends on Ibe and also on Ice. It's traditional simply to assume that Vbe is 0.7V for silicon transistors (0.3V for germanium) and design your circuit around that.

You can measure the base current if you want but why do it? I think it's a mistake to base your circuit design on the exact shape of the Vbe curve - it varies too much between different transistors or with temperature changes or ageing. Design your circuit with plenty of negative feedback to compensate for differences between components or changes over time.

My curve tracer shows the approximate shape of the transistor's response - that's all any curve tracer does. Plug a transistor into different transistor-testors and you'll get very different values for the "gain". The only reason you might worry about the exact gain is to choose a "matched pair" and then you'll simply choose two transistors that give the same reading on the same tester. Even a "matched pair" bought from a reputable manufacturer might differ by 5%.

Don't design circuits that rely on exact values of any components - particularly the gain. Use negative feedback.

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

The ExecSetupMenu function executes one of the setup menus (for Bipolar or FET) then waits for either the OK button to be pressed or for a device to be inserted. (It also deals with pressing the up/down buttons)

It returns false if the OK button is pressed and returns true if a device is inserted.

The "if (millis() - time > 1000)" block tests whether a device has been inserted. It does so once every second.

I guess you don't have a millis() function in your 18f4620 software. Of course, it doesn't really matter that the test is done exactly once a second. I think I put the timer in there so that the test wasn't done every time round the loop - it slows down the response time of pressing the buttons so should only be done occasionally. Perhaps you could do the TestDeviceKind() test once every 100 or 1000 times the loop is executed.

The construction
if (millis() - time > 1000) {
// do something
time = millis();
}
is used in the Arduino to prevent problems with overflow:

https://www.norwegiancreations.com/2018/10/arduino...

> For current It has to divide by collector load

True. The collector load is 100 ohms. The Arduino uses the "5V" line as Vref for its 10-bit ADC so 5V is equivalent to an ADC value of 1023. (The Arduino compensates for changes in the "5V" line by measuring its internal band-gap voltage reference.) If you do the maths the equation is

i = ADC_value * 5 / 1023 / 100

So, for instance, a current of 10mA would give an ADC value of 205.

The line you refer to is in TestDeviceKind() when it is trying to decide what device is inserted. It compares the current through the collector load using
if (i > 50) {
}
A current of 2.5mA gives i = 51. So the program is testing whether the current through the device is greater than 2.5mA.

Peter

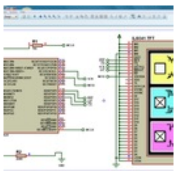
Thanks you sir a lot for your explanation For Touch portion I used external interrupt for sensing touch action. by connecting external interrupt pin RBO to interrupt pin of TFT display.The pressure sensing method Z1+Z2 which you followed not gives reliable result for me Thnking you once again

This is actual pic pf my prototype with pic 18f4620

actual pic is here

actual

Sir I have ported your project to pic 18f4620 and I have successfully completed TFT DISPLAY PORTION AND TOUCH PORTION GRAPH PORTION is pending After completing I will share fully code



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?

(Sorry for the delay - Instructables has only just forwarded your question to my email.)

I've not used a Due.

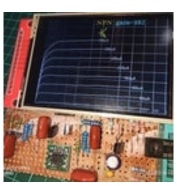
The Due runs at 3.3V while my circuit assumes the Pro Mini (or Nano/Uno if you prefer) runs at 5V. So you'll need to modify my circuit appropriately.

The Due analog output delivers 0.55V-2.75V while my circuit assumes DAC produces 0V-4V. Once again, you'll need to modify my circuit.

Peter

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

After long time hunting for dac ic and only found mcp4822 but it's finally working, layout by pierrotlalune67 stripboard, but don't know about it's accuracy with this dac, but thank you for sharing this little nice project



Great.

Did you need to make changes to the software?

Just use yours original software, don't know how to change software

What does work and what doesn't work?

when the t din conected to sck the display show nothing, when disconected the disply is looping screen menu

T_DIN should be connected to MOSI.

SCK should be connected to T_CLK.

Double check the wiring all correct, by the way i'm using stilus mode tft touchcreen, is it ok, and not build the whole project, just wiring betwen arduino and display

What do you mean "stilus mode tft touchcreen"? Is it a different screen from the 2.8" ILI9341?

> not build the whole project, just wiring betwen arduino and display

I don't know whether the INO file will work without the rest of the circuit. I would expect it to but I haven't tested it.

If you are just wanting to ise an ILI9341 with an Arduino without the rest of the Curve Tracer then there are several other Instructables. Search for

https://www.instructables.com/circuits/howto/ili9341/

Or use Google to search for "ili9341 arduino".

i'm intersting to build curve tester but i've just have this type display, this is a touchscreen with stylus, is it working just plug and play or need to make some adjustment with my display

A "touchscreen" like yours consists of a TFT display screen and a resistive touch controller.

The display screen is controlled by a chip such as the ILI9341 but several other controller chips are available. Each kind of chip needs its own Arduino library and each kind of chip might be connected to the screen in several different ways. So you need the right library and it must be configured to match how the controller chip has been used on that particulat board.

Similarly, the resistive touch pad is controlled by a chip such as the XPT2046 (several other controller chips are available). Once again, each kind of chip needs its own Arduino library and each kind of chip might be connected to the pad in several different ways. You need the right library and it must be configured to match how the controller chip has been used on that particular board.

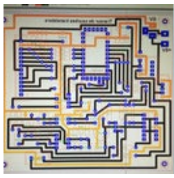
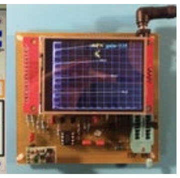
I believe that some people have found that their touch pad is "mirrored" relative to the library they are using. It depends how the board designer connected the chip.

My library is specific to the touchscreen I recommended. Your touchscreen looks similar but may not be exactly the same. Other people have built this project and have not had problems with the screen.

The first question you need to answer is: is the fault in the hardware or software.

As I suggested, you could search Instructables or the web for a project that uses exactly the same screen you have. If the project works then you will know your circuit is good and the problem is that my library is not suitable for your touch pad chip.

Thanks for your job! I was watching for a curve tracer for a long time.
Works perfectly!
I design the PCB with CIDESS, a french PCB design software. I make PCB with toner transfer.
Next step : connection with laptop.

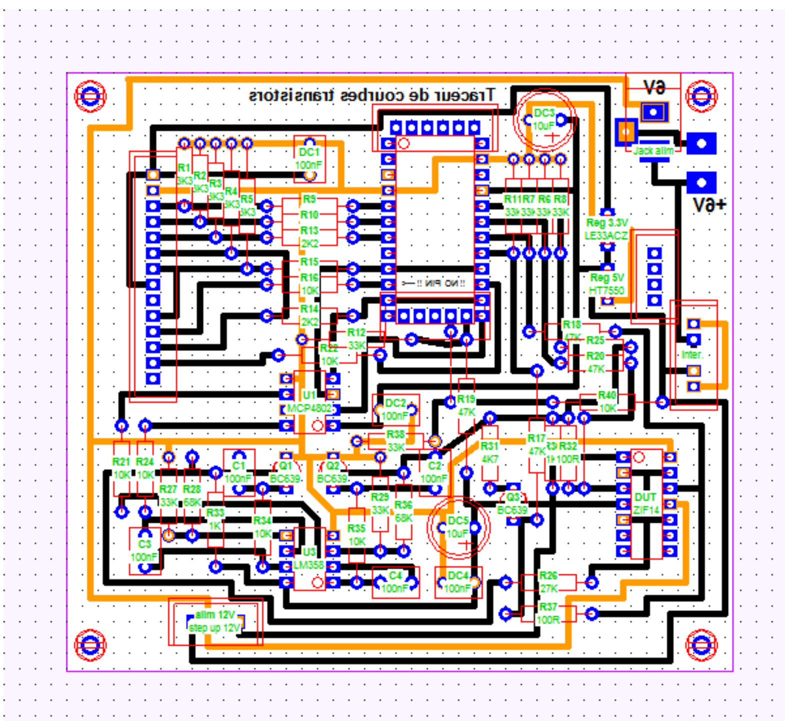



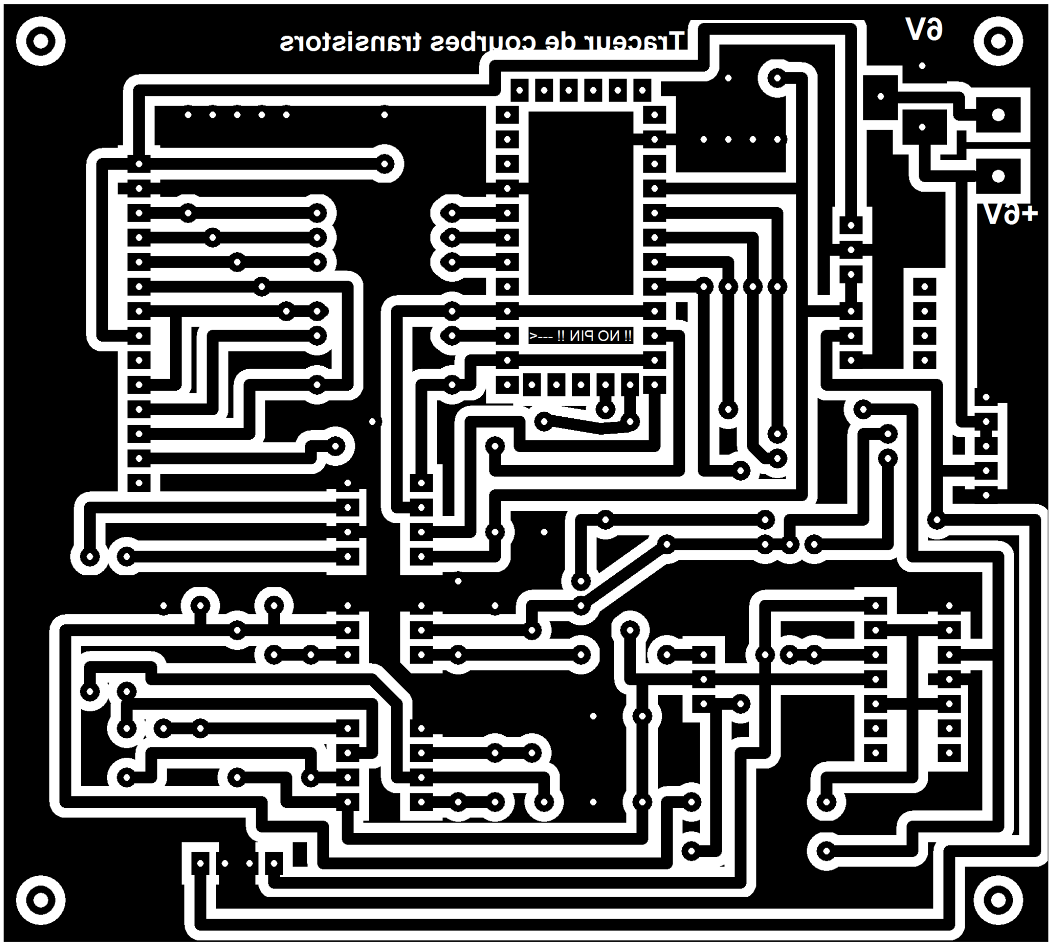

I'm glad it's working for you. "TiberioV" (see below) was asking last week for a through-hole PCB design. Could you send your gerbers to him? Could you post them here so they're available to other people?

CIDESS looks like an interesting PCB design package. I see that you've avoided wire links by varying the length of resistors. Is that built into CIDESS? The package I use (EasyPC) doesn't allow it so I've had to define lots of resistor packages of different lengths.

Peter

Hello, here it is :




Traceur de courbes transistors
6V
+6V
!! NO PIN !! --->