M Michael Damkier Apr 19, 2022 5 min read

A new graphic display for a Fluke 8050A multimeter

Updated: Jul 2, 2022

In an earlier project, I replaced the LCD display in a Fluke 8050A digital multimeter with a LED display. While gathering information about that project, I ran across a really fine upgrade at Ken's Electronic Projects, https://sites.google.com/site/kenselectronicsprojects/fluke8050a_display, using an LCD graphic display. Well, I have since picked up another Fluke 8050A and, with Ken's work as inspiration, this time I thought I would give it a new graphic display.

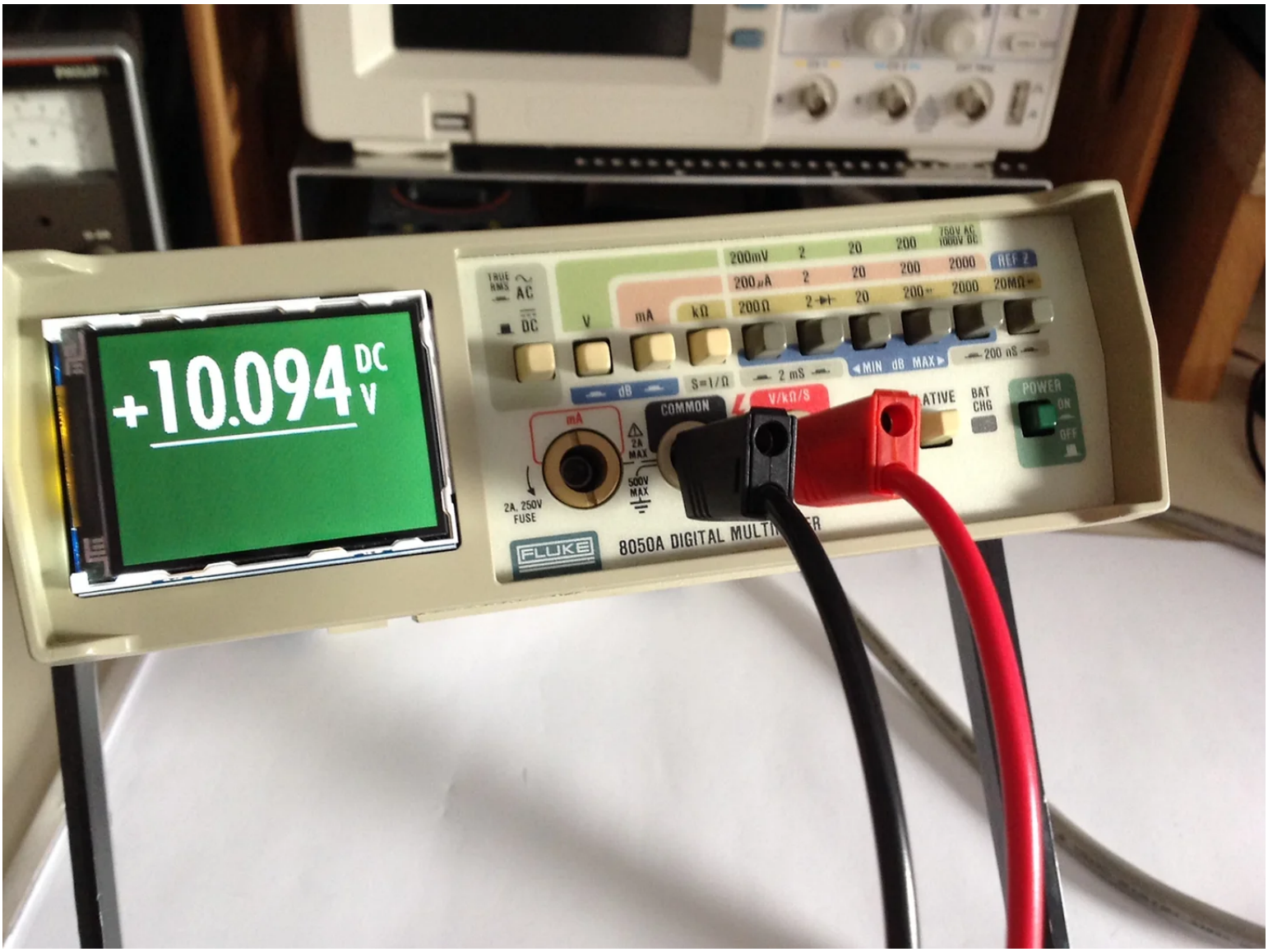
I'd like to give a shout out to YouTuber ToltecMerc who implemented this display upgrade on his 8050A and he made a video about it. Check it out at https://www.youtube.com/watch?v=rdQvFy0bY6Y

More than just 5 digits

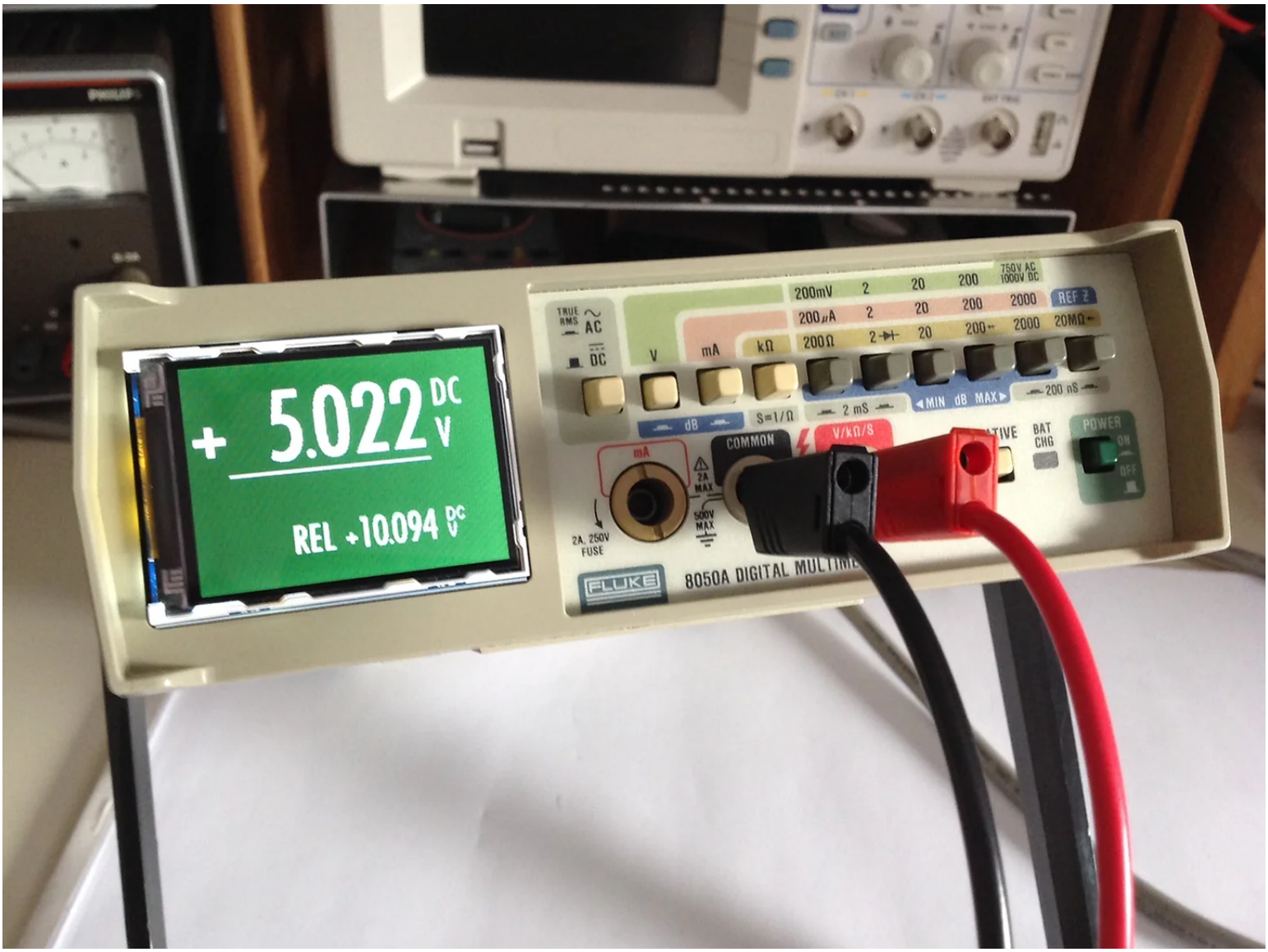
The Fluke 8050A is really quite a capable instrument. It measures not just voltage, current and resistance but also decibels (at various reference impedances) and conductance. And it has a relative mode. Unfortunately, there's only a little symbol for REL and dB, so you have no visual indication of the relative offset or the dB reference impedance or even what unit you're measuring. So, I went into this with the goal to display these bits of information as well the value being measured. Another goal was to leave the Fluke meter completely unmodified. I think I achieved these goals pretty well. I'll let the photos tell most of the story.

Pictures tell the story

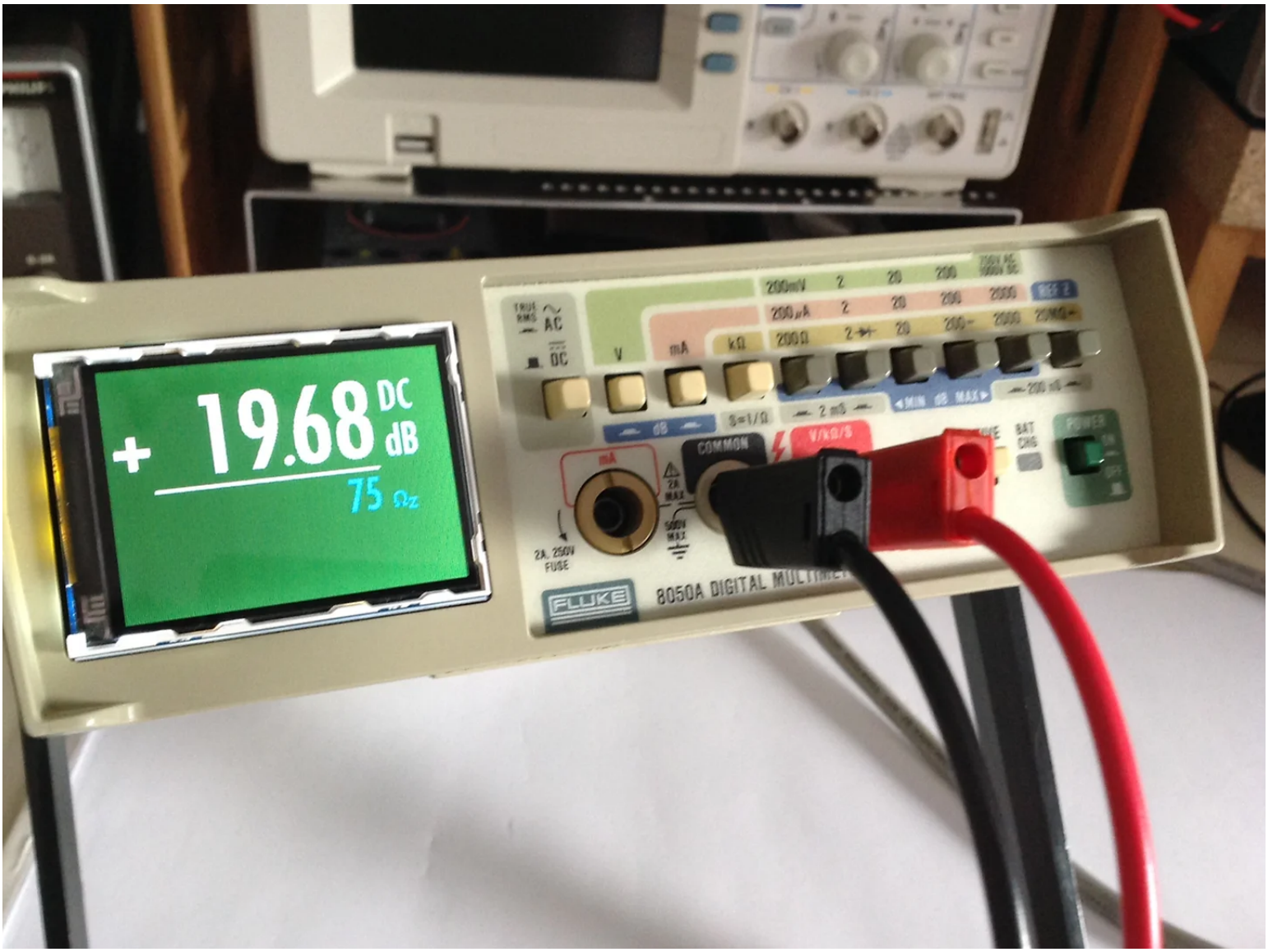
Here is the meter in simple voltage mode. The background is dark green and the digits are white. (I still need to come up with something to hide the edges of the display.)



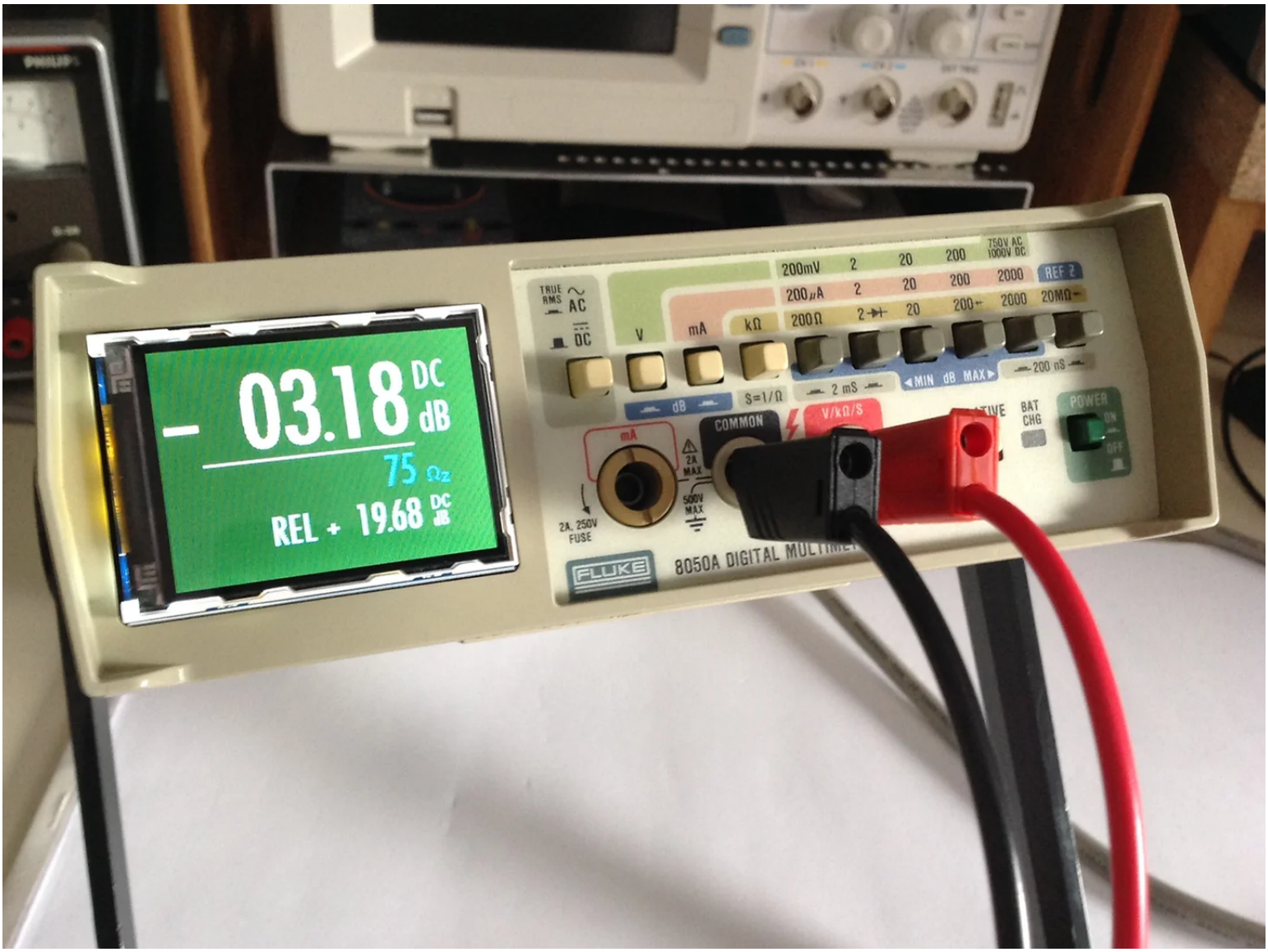
The next image shows the meter in relative mode. The relative offset voltage is displayed at the bottom. For all of the unit indicators and relative values, I used color-coding that follows from the Fluke panel colors. V is light green, A is light red, Ω is yellow, S is grey, dB is light blue and Z is a darker blue. (It's subtle and doesn't show up that well in some pictures.)



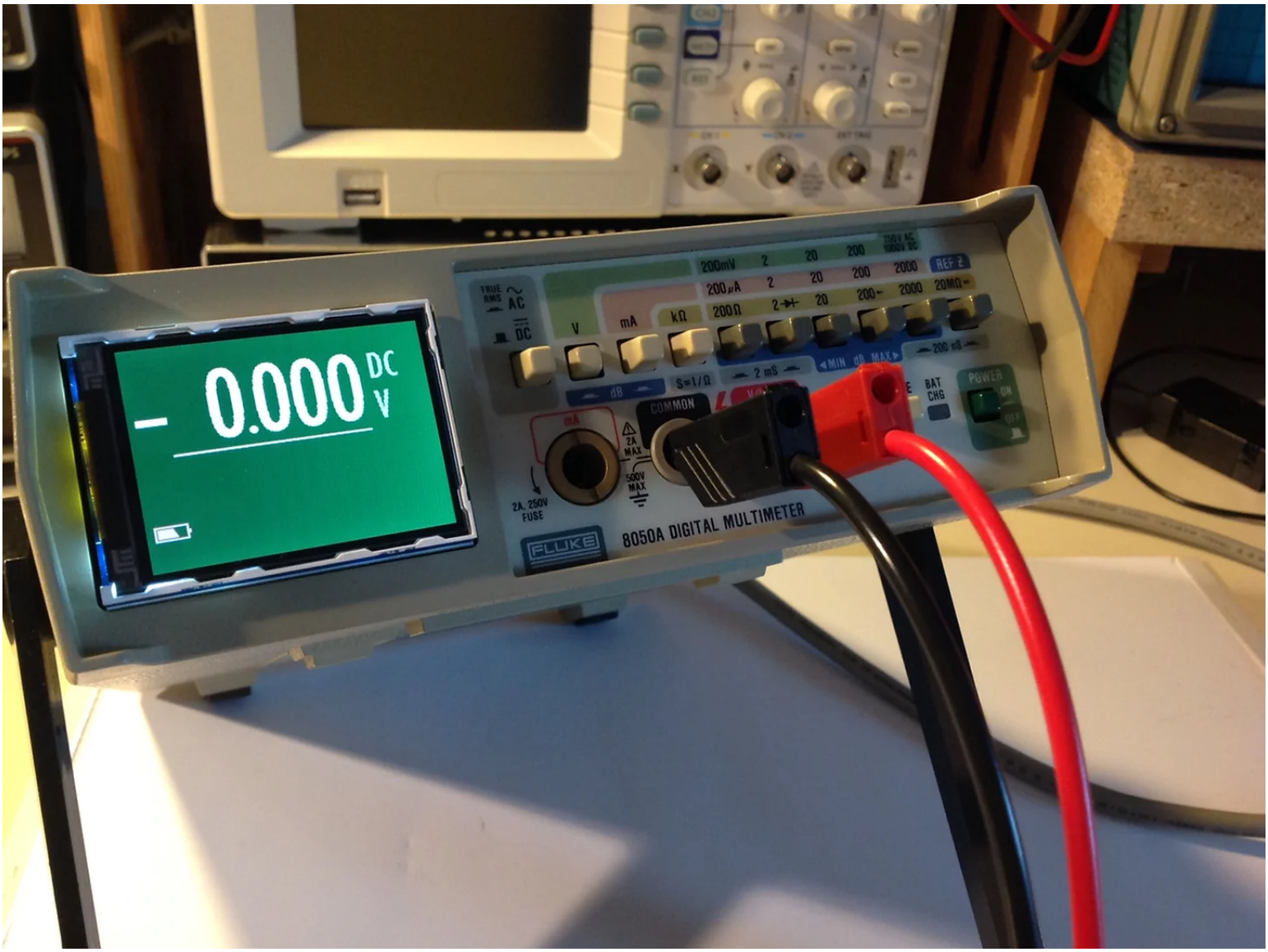
This is in dB mode. The reference impedance is also displayed.



Of course, dB works in relative mode as well.



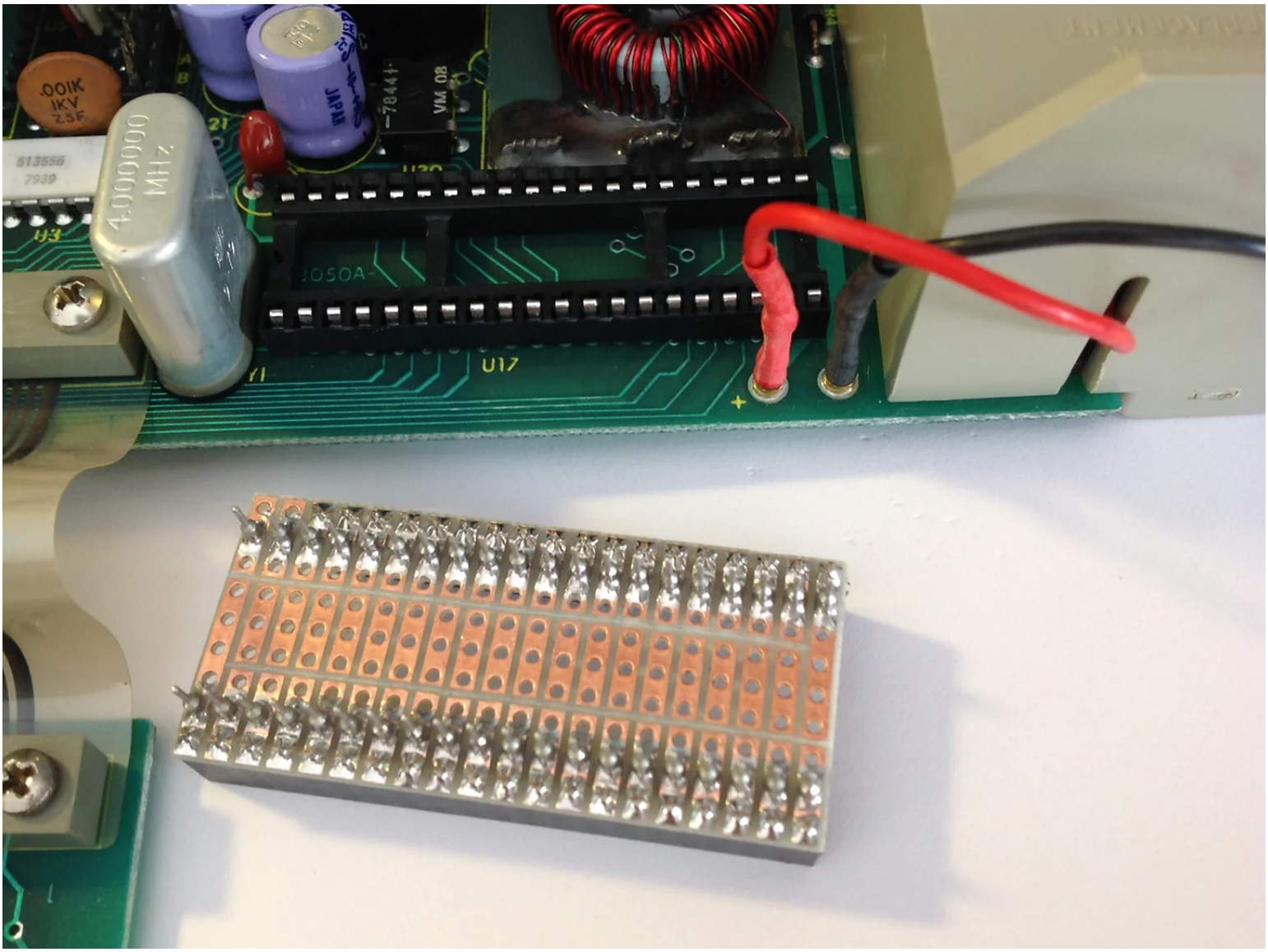
This shows the low-battery indicator. It is actually in red. By the way, the line under the main display turns red when the Fluke sets the high-voltage indicator.



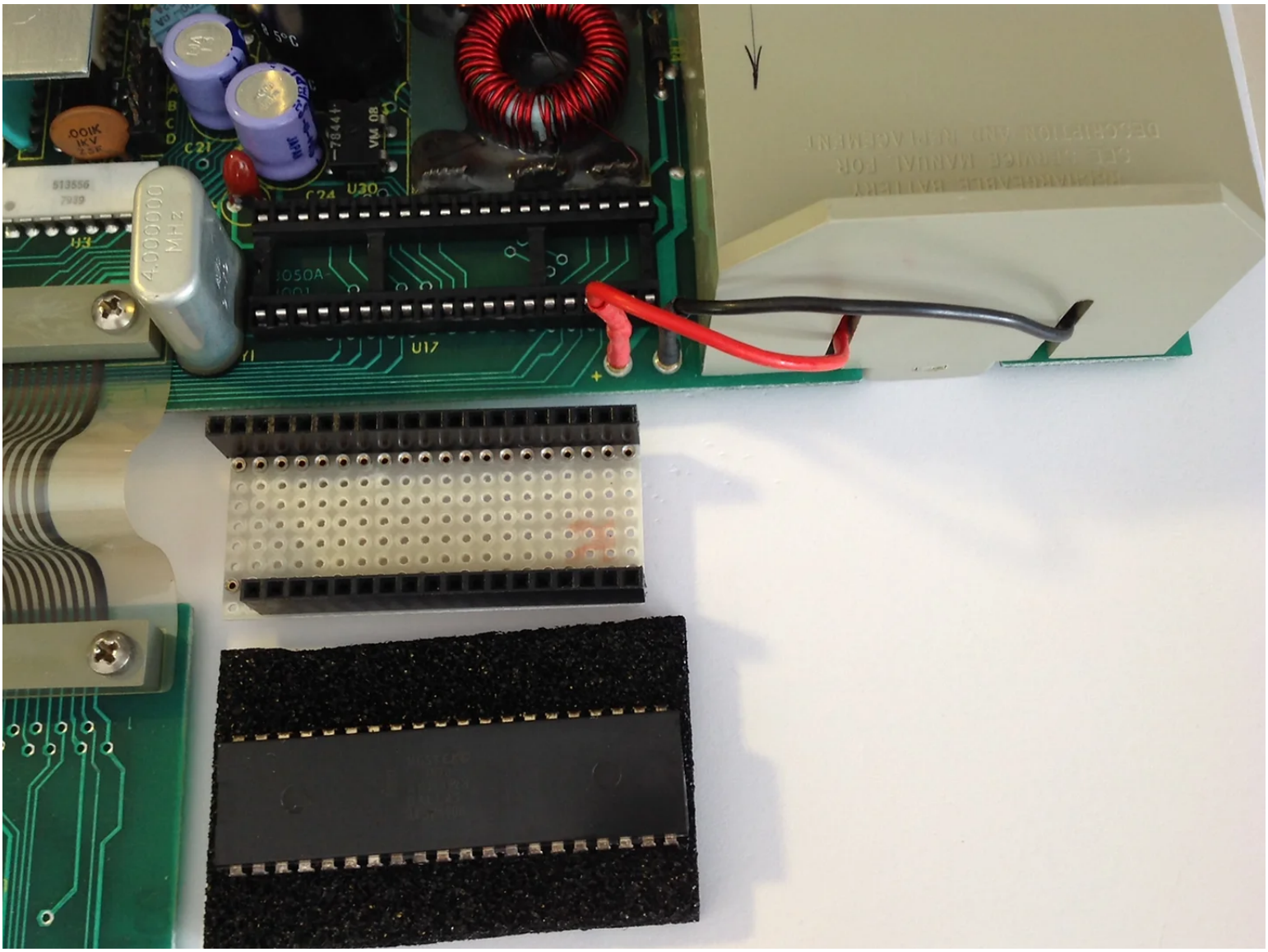
Construction

The 8050A is not modified in any way with one small exception (I'll point that out when we get to it). The LCD display module is removed intact. All of the signals needed (except one) are available at the microcontroller in the Fluke.

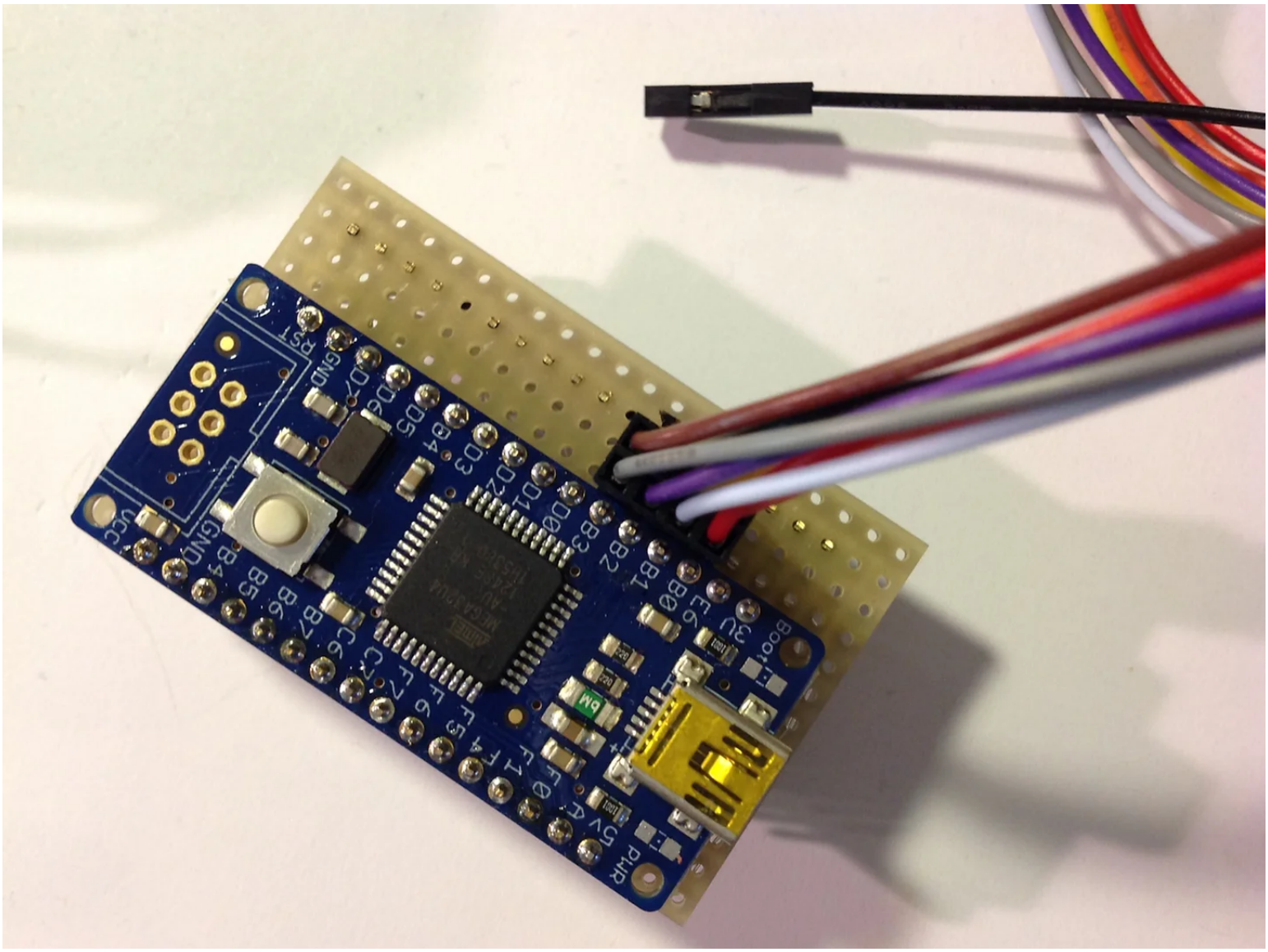
In order to not have to solder anything to the meter, I built this little piggyback carrier to bring out the Fluke's microcontroller pins.



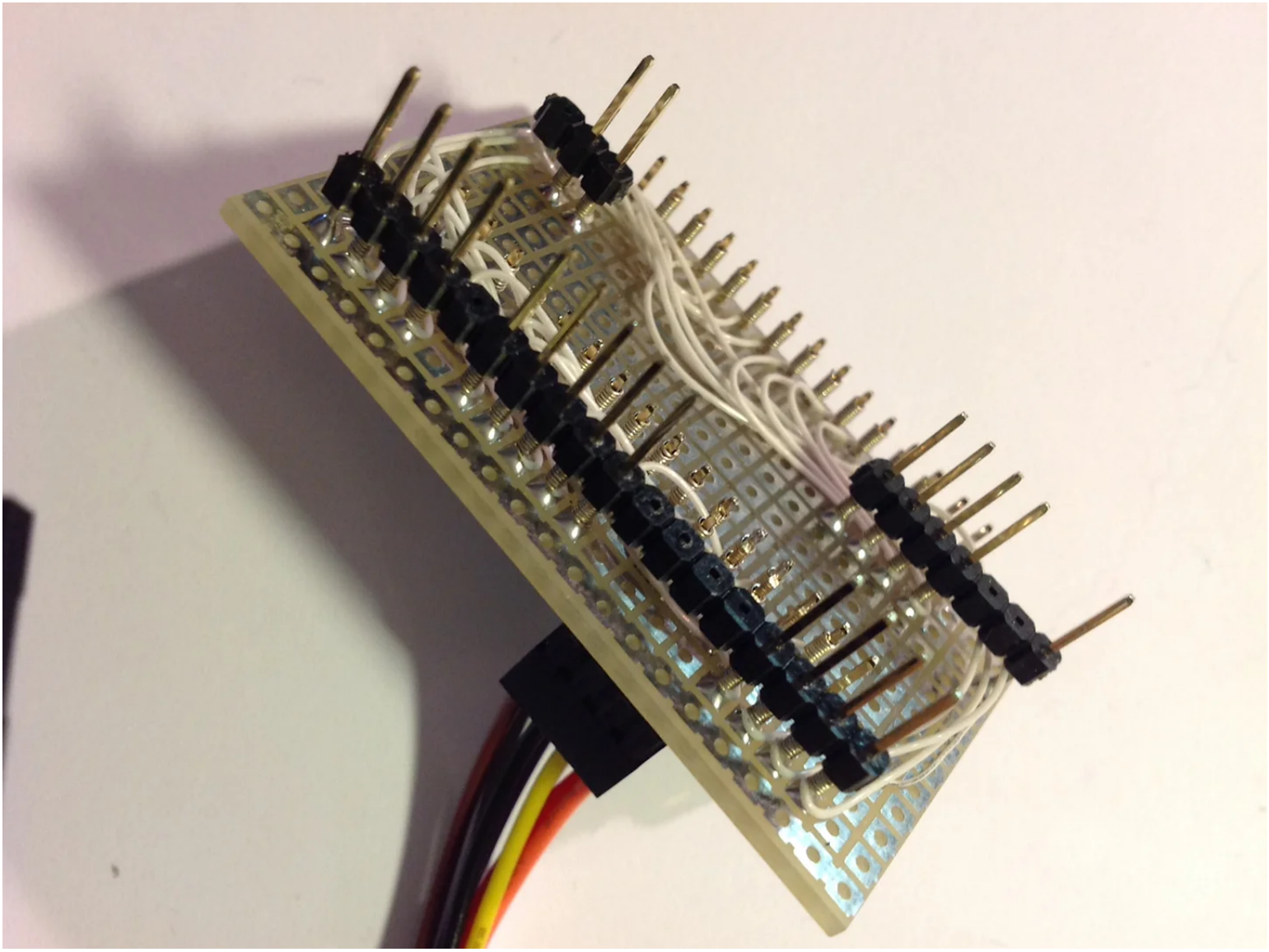
The Fluke processor now plugs into this board and the pin headers provide access to the processor pins (the Fluke processor sits down between the pin headers).



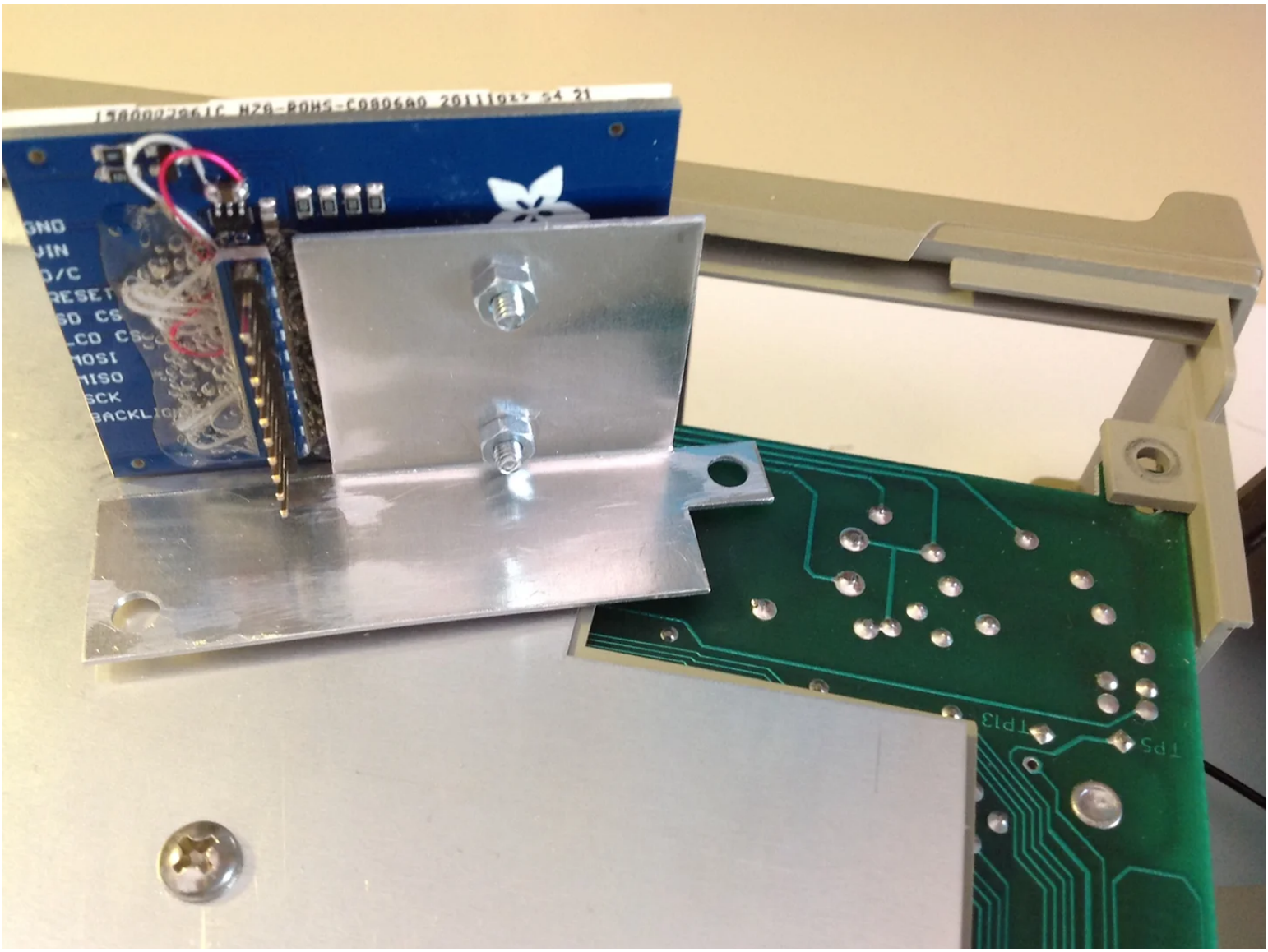
For the extended processing I used an Atmega32u4 breakout board from Adafruit, http://www.adafruit.com/product/296. (I removed the two LEDs from the board. First, because there's no need to light up the inside of the meter. Second, and more importantly, I needed the PE6 pin as an interrupt and not as a communication monitor.)



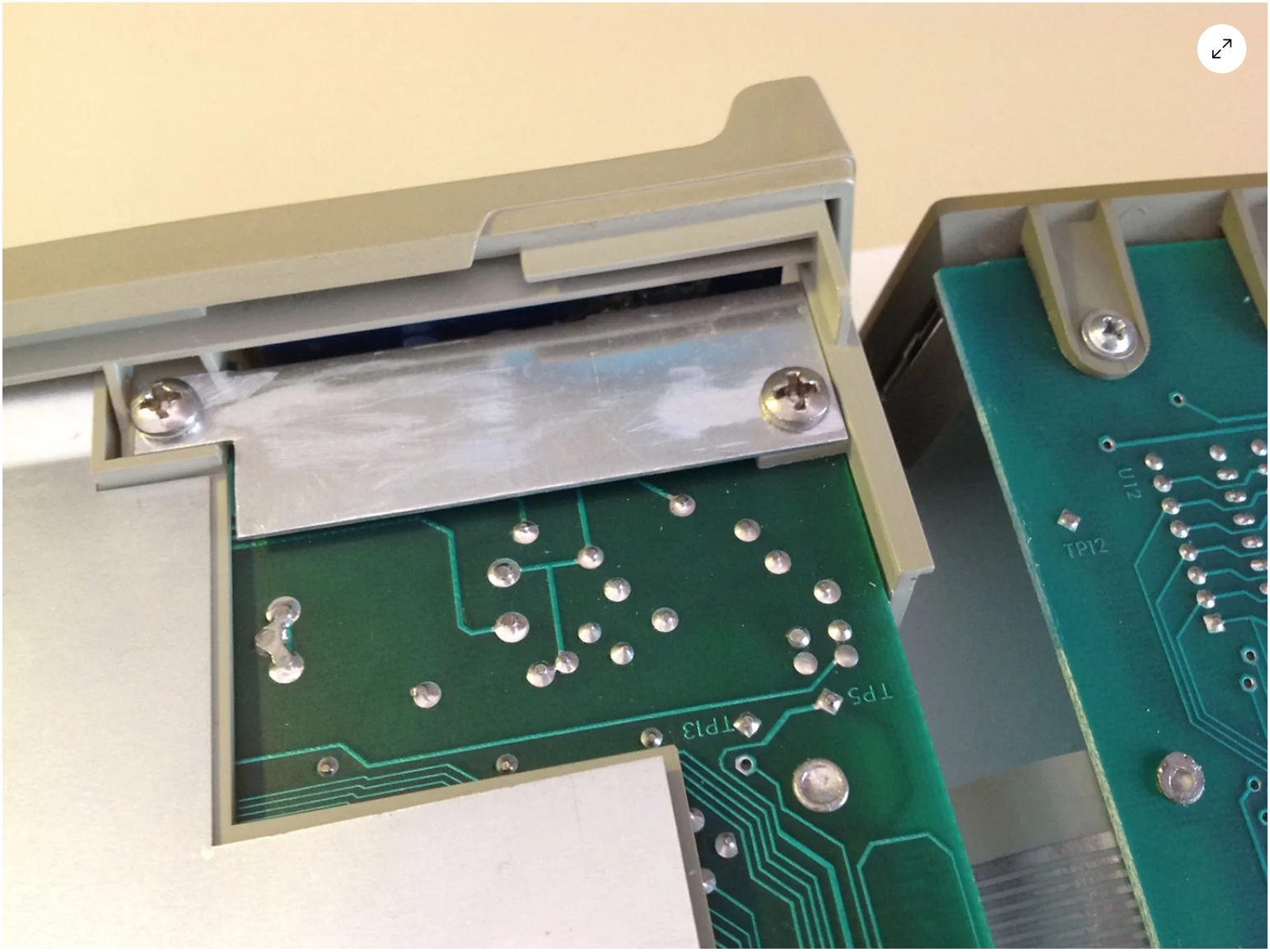
The breakout board is mounted to a bit of perf-board with pins to match the headers on the piggyback board. There's an 8-pin connector to the display and the low battery sense. Yes, that is actually wire-wrap.



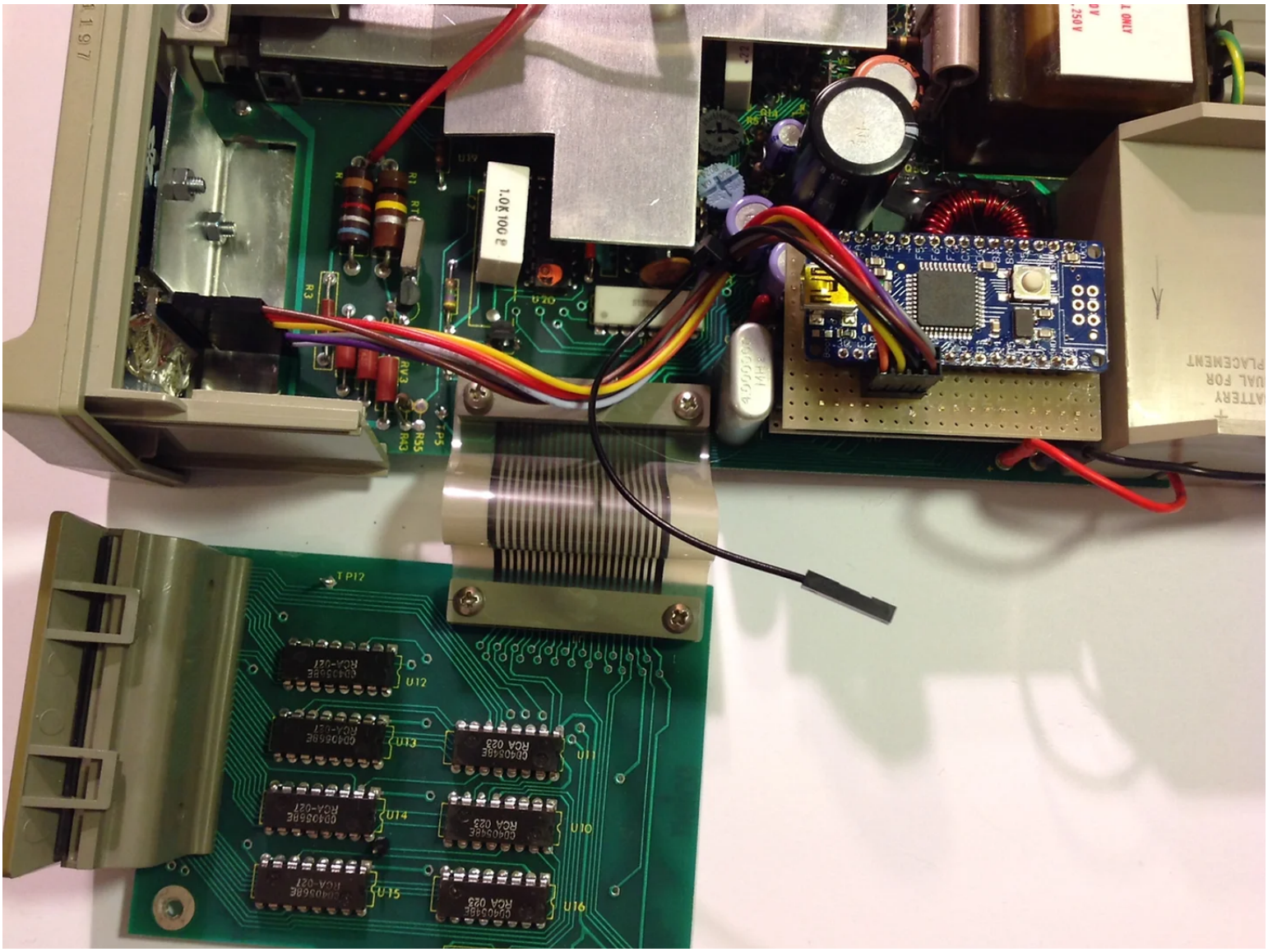
The display is also from Adafruit, their 2.2" TFT LCD display, http://www.adafruit.com/product/1480. As Ken mentions in his article, the breakout board for the display is too big for the opening in the meter. I carefully made the board the same size as the display. I glued a connector to the back of the board after wiring it to the appropriate locations. (The breakout board supplies 5V/3.3V level-shifting and a 3.3V regulator for the display power.) Also, the SD card reader was removed. The CS signal is hard-wired to ground on the LCD board. The backlight PWM control pin is not used and neither, of course, is the SD card CS pin. This leaves just 5 lines RST, DC, MOSI, MISO (not really needed) and SCK in addition to GND and 5V. These seven wires, together with the low battery signal, plug into the 8-pin header on the piggyback board.



The display is mounted to a bracket that fits the screw mounts for the main circuit board. (This is not the final bracket. I needed to move the display farther forward to avoid having to trim the Fluke display board.)



Here it is all together. Well, almost.



As I mentioned, the Fluke is unmodified. That is, except for the addition of one header pin soldered to the display board. My unit has the battery option and the low battery signal is generated on the display board. Fortunately, there is a via in the routing of this signal to the original display driver chip. You can see the pin in the picture above between U14 and U15. The black wire from the piggyback board attaches there.

Software

Download the zip file:

TFT_8050A.zip
Download ZIP • 24KB

(last updated 7. August 2015)

Unzip the file somewhere. Move the Modified_Adafruit_ILI9340 folder to your Arduino libraries folder. This is the Adafruit_ILI9340 display library but stripped down to just what is needed. I did this for size and speed. The Arduino sketch folder is named TFT_8050A. Just move this entire folder to your Arduino sketch directory. The folder contains the .ino file and the header files that define the character and symbol bitmaps used.
Be sure to select the Arduino Leonardo board in the Arduino environment if you are compiling the sketch there.
I have been using the meter with the new display for a little while now and feel pretty confident that the software is in order.
At some point, I'll get a description of the code here. If you have any questions, you can post a comment or send me an e-mail (the address is in the .ino file).

Conclusion

This is just intended as a better-than-original display replacement. The Fluke is a solid measurement front-end and with a bit more effort you could extend it, using added computation, even further. You could display a graph showing trends, for example. With USB support (the Atmega32u4 has it) you could provide logging. I have no need for these things and I've wasted, uh, *invested* enough time in this already. As good as it is, the 8050A is old-school, with big, physical switches, no auto-ranging, etc. Still, it does the job and this one is now back in service, maybe for another 30 years.
Thanks for your interest. Your comments are welcome.

UPDATE: 05. Feb. 2017
Thank you all very much for your interest in this project! Some of you have asked if a schematic is available and a couple have inquired about a kit. (I am sorry that for many of you, this update is rather late.) In fact I did look into developing an upgrade kit but thought it would be too expensive to really make it worthwhile. I figured it would cost upwards of $60 and that was with little or no profit. Oh well.
Anyway, at that time I did create a schematic (no PC layout, though!) but kind of resisted posting it. It does not exactly represent the project described in the post (uses 'raw' parts and not breakout boards). And I thought it was best to derive a schematic from the pin assignments in the code.
But anyway, here goes. This is the latest version of the schematic I could find.

I'm pretty sure that this is correct but I can't make any guarantees. I think I used the same pin assignments as in the code. If not, change either the schematic or code to conform.
Again, thank you all for your interest. It pleases me greatly that this project and my LED upgrade have inspired people all over the globe. Just to let you know, my meters are still going strong and get used regularly. I hope yours are too!
And if one of you ever does develop this into an upgrade kit, let me know. I have a couple more 8050As myself!