You never forget a great final project!

Here's my first one:

I built this robot for a year-long science fair in South Carolina. I was twelve, in seventh grade in public school in a town called Summerville. Microcontrollers and memory were astronomically priced for someone on a quarter-per-lawn-mow budget in 1979...

... so I built it using logic gates and transistors and cardboard and pieces from an Erector set (google it). Hence its name, SEPTOR: Self-programming transistorized robot. The two "eyes" are scraps of solar cells that could detect light. It would follow a flashlight, turning left or right to keep the brightness of the beam centered on the "nose" of its "face". It would back away from a loud noise. It could be "driven around" using the remote control (made from a dead calculator) at the very bottom of the picture. It had an entertaining "autonomous" mode where it would cruise around the house, banging into things, sensing the collisions with bumper switches, and backing away, attempting to "map" the room by keeping track of the times between collisions and the wheel speeds (the "mapping" was the "self-programming" part).

To build it, I figured I needed a decent multimeter. I scrounged everything else (motors from toys, IC's pulled from surplus junk, dead calculator from the garbage, etc.), but I needed some way to make believable measurements. I saved quarters for a summer and finally decided that I could afford a "bottom-of-the-line" model for $7.95 plus tax, the "Micronta 22-027". So:

My Dad...


Dr. Leeb directs Naval pathology labs photos by Mort Fryman

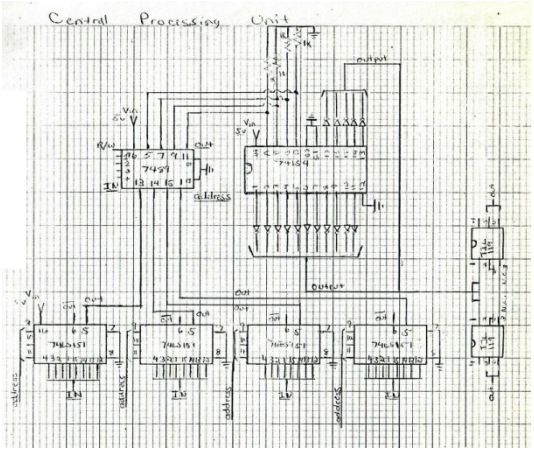
...drove me to RadioShack to buy this 22-027:

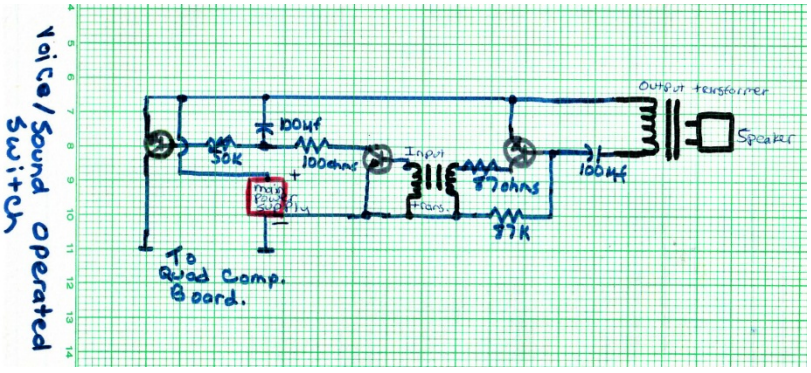
...but while I was salivating over the display cabinet, he put the 22-027 back and bought me this 22-208 with high impedance FET inputs:

I still have the 22-208 sitting on my desk as I type.
It's the single greatest piece of test equipment I've ever owned.
Now aren't you glad that we sent you home with solid test equipment?
You might be amazed what you can do with your kit.

P.S., check out this documentation:

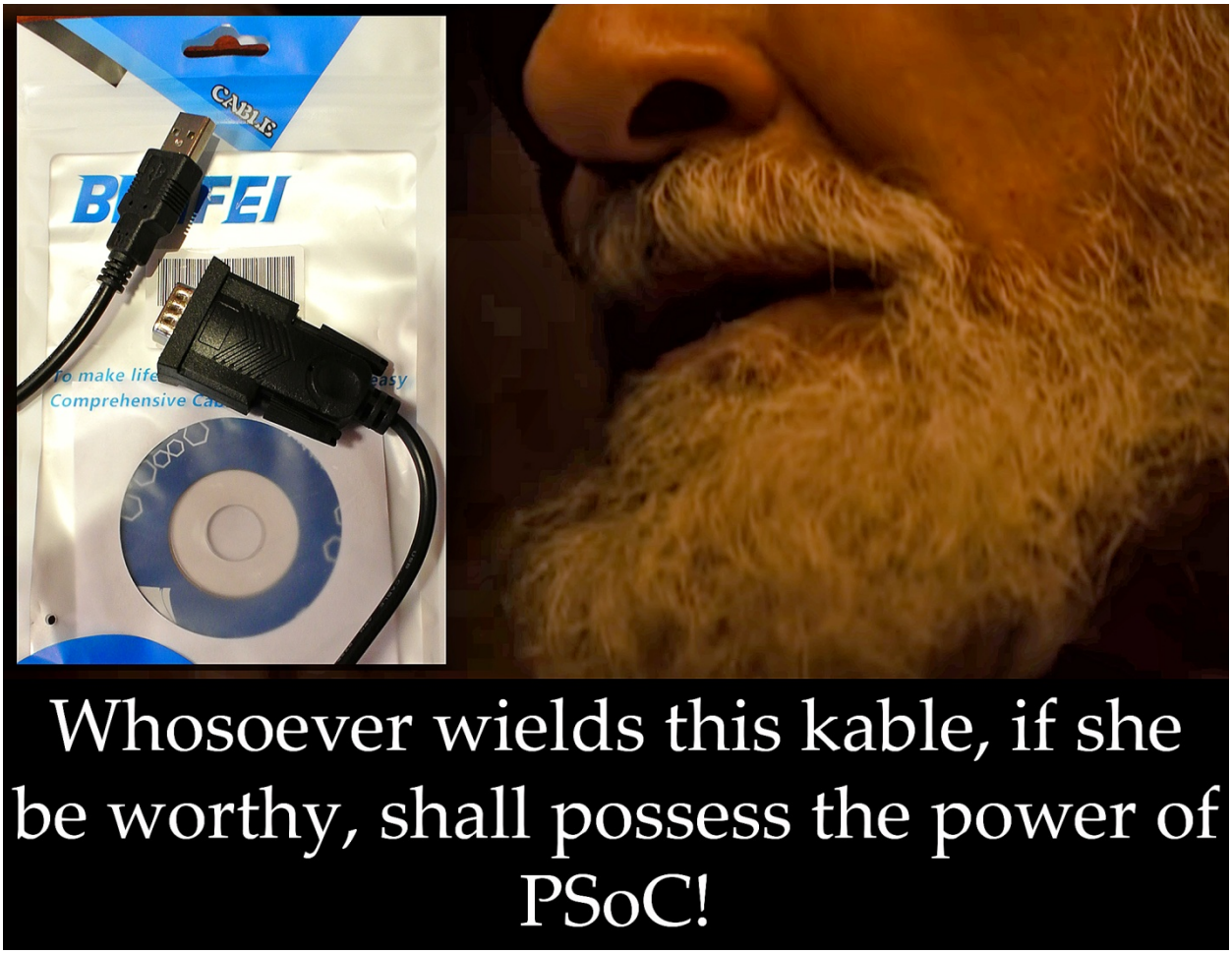
Not bad for seventh grade, eh?





FOR EXAMPLE: SERIAL COMMUNICATIONS!
Excalibur, Mjolnir, and Kable

You have heard from Odin previously, but it never hurts to hear again:

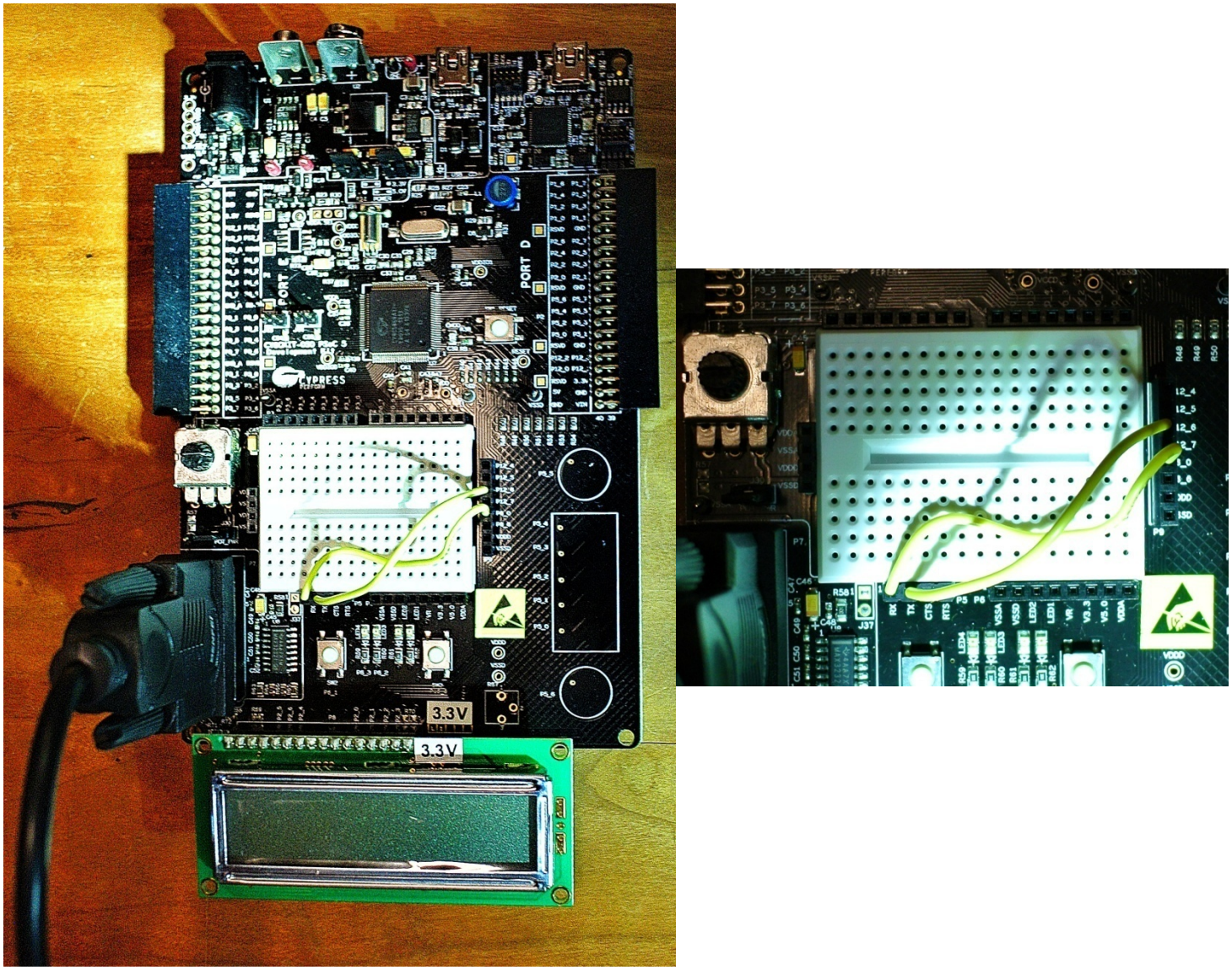


Like the rainbow bridge Bifrost that connects Asgard to Midgard (Earth), the Kable (shown above) from your Kovid Kit will permit you to connect your PSoC 5LP "Big Board" to your Windows 10 PC. Just as the ever-vigilant Heimdall guarded the Bifrost, you too must impress Odin by configuring your PC, Kable, and "Big Board" to permit just and righteous communications to flow between your systems.

This tech and skill set may be enormously useful for your final project. Your Kable grants you the equivalent of Thor's hammer for crafting this type of interaction. Let's see how!

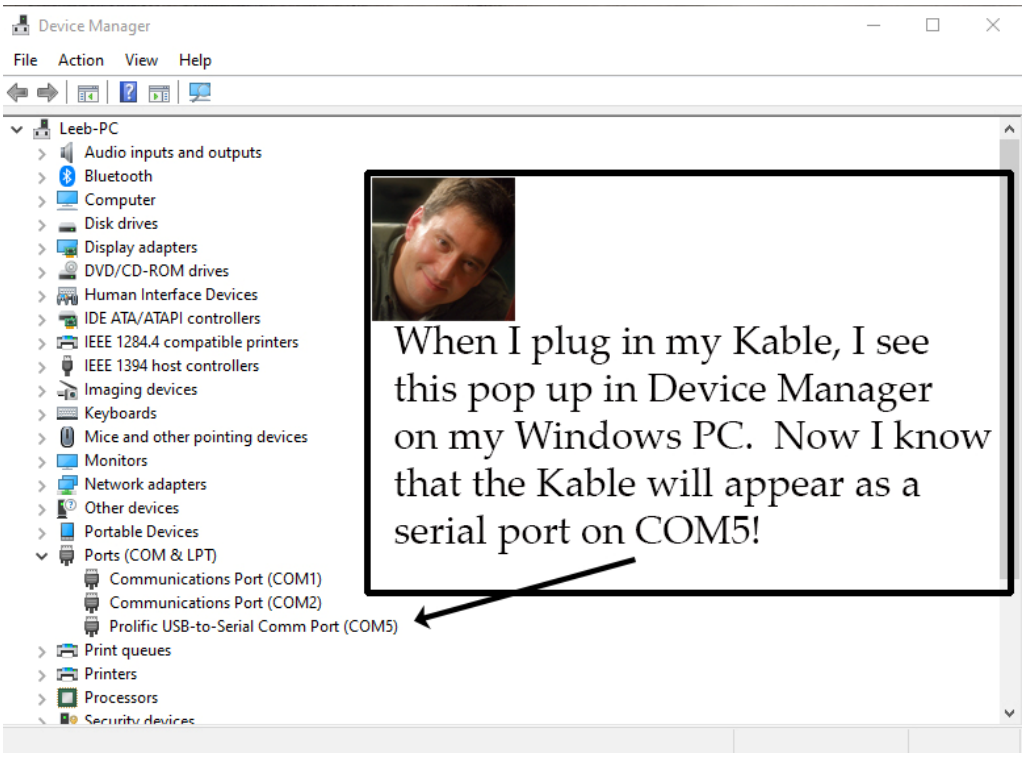
To establish SERIAL connections between your Big Board and your PC, you would do the following:

1. With the power OFF, add a mighty wire on your Big Board that connects "RX" to "P6_0" near the breadboard. Add another wire that connect "TX" to "P12_6". This connects the RX and TX pins of the RS232 line driver IC (like the MAX232 on the R31JP) to two pins that are connected to a serial port UART on the PSoC project. Be careful! Also connect the RS232 cable to the Big Board. Here's what it looks like on my kit:



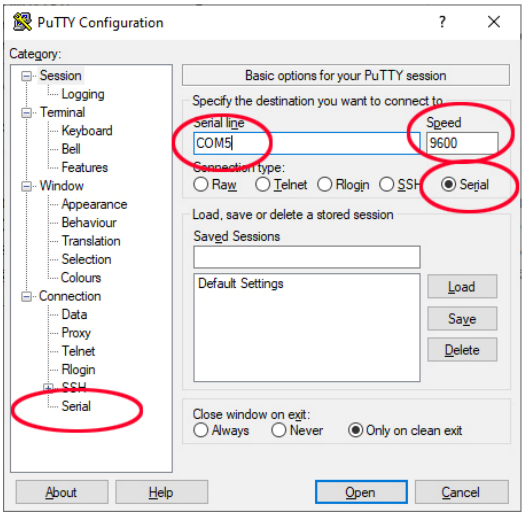
2. Go to the "PSoC" page on the 6.115 course website: http://web.mit.edu/6.115/www/page/psoc-information.html
3. Under the "Blackbird Kit Interface" section, you'll find a PSoC project called "RS232_PSoC", it's the first menu item in the section, under the "Blackbird Kit Interface" title. Download this project, and program it into your PSoC 5LP Big Board.
4. You should have your USB-to-mini-USB cable plugged in to PC and PSoC, powering up your PSoC. You should also have your Kable plugged into the PC USB Port and the PSoC serial connector. Now, you can configure a terminal program like Putty to talk to your PSoC:

On the staff Windows 10 PC (Uncle Steve's desktop computer), the Kable is "plug and play". Simply inserting it into a free USB port resulted in the Kable appearing as a serial "COM" port in the Windows Device Manager. If you need to, you can use the Kable package CD to install drivers, but we have not found this to be necessary. When I plug the Kable into a USB port on my Windows 10 PC, here's what I see in Windows "Device Manager" under "Ports (COM & LPT)":

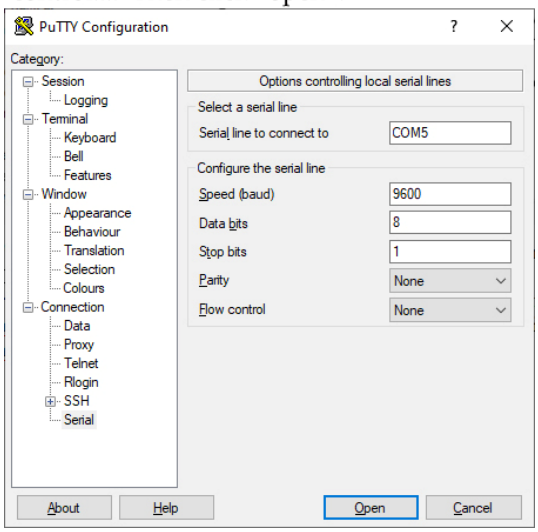


Now, use a "terminal" program on your Windows 10 PC to configure a terminal window to talk on the COM port (COM5 in my case) at 9600 baud, 8 data bits, 1 stop bit, "none" for parity, and "none" for flow control. Here's an example interaction using "Putty" from the software downloads (but you can use TeraTerm, RealTERM, etc….):

Set Serial, COM (COM5 in my case), 9600 Baud, and then click "Serial"...

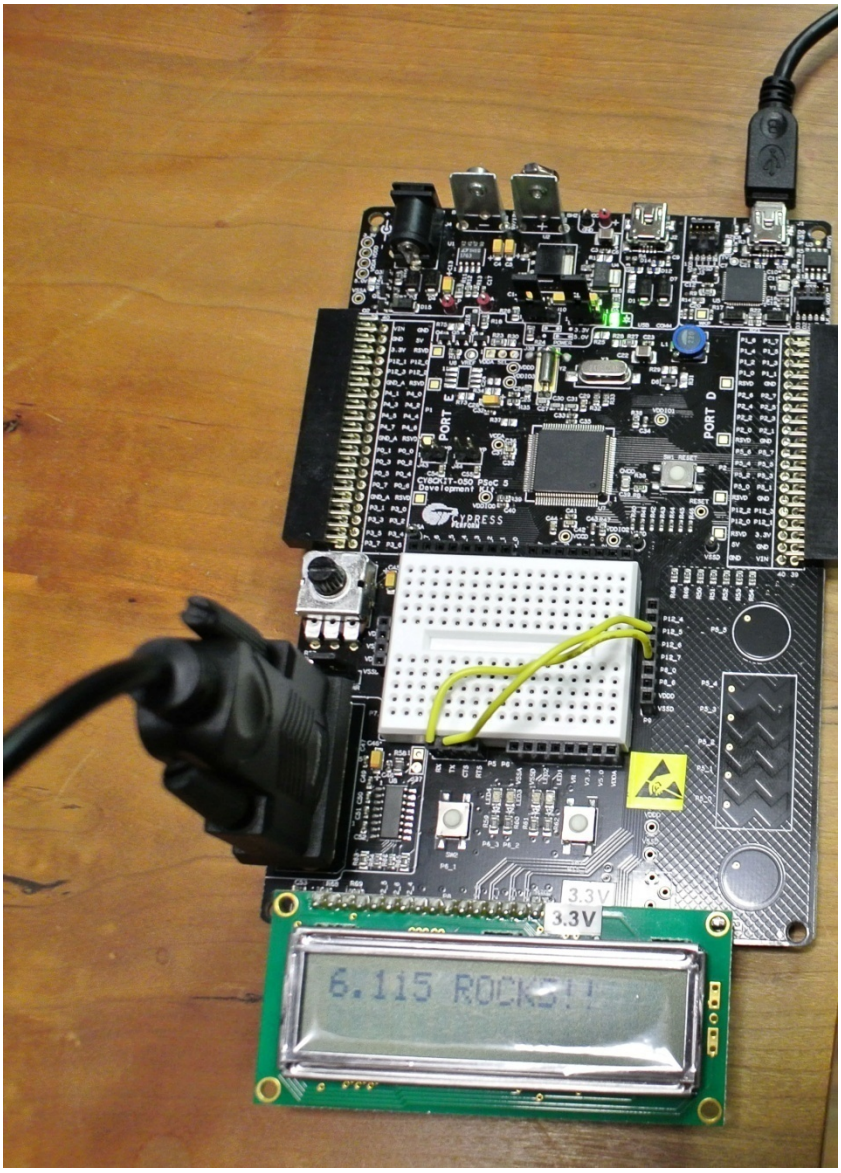


Selecting "serial" brings you to this window in Putty, where you can ensure that you are set for 9600 baud, 8 data bits, 1 stop bit, "none" for parity, and "none" for flow control.... Then click "open"!



... which leaves you with this terminal window. Typing in this window sends characters over to whatever is connected to the "other end" (RS232 end) of the serial Kable....

If you type a happy message on the PC in your terminal window, the characters should appear on your PSoC LCD screen, like this example when I typed "6.115 ROCKS!!" in the Putty Terminal:



Notice that I have both cables plugged in (serial Kable and USB programming cable). Kable provides the communications, and the USB programming cable powers up the PSoC. In this case, you get a common ground (which is required!) between the PC and the Big Board through the USB cable ☺!

P.S. - In case you doubt the veracity of any statement in this document - this project tip is endorsed by my buddy, Loki:
(Yes, he is that tall...)

(and Captain Kirk also)

(and of course, Dr. Who...)