

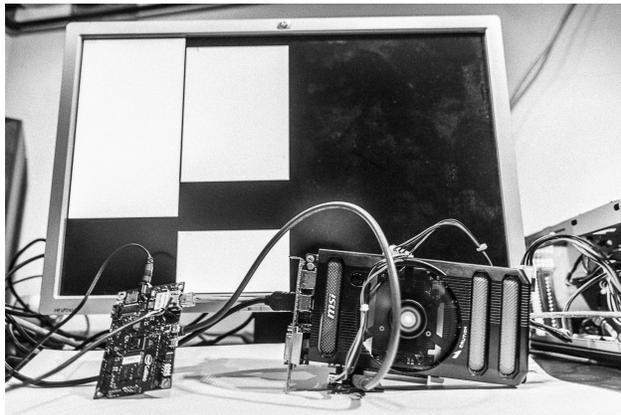
7 A Breakout Board for Mini-PCIe; or, My Intel Galileo has less RAM than its Video Card!

by Joe FitzPatrick

Dear Acolytes of Electricity, let us spend a moment remembering the daily struggles from a time before enlightenment. For let us not forget that there was a time that even the most modest system upgrade required a screwdriver. And let us recall the dark moments when we were alone with DIP switches, not knowing what to set or where to seek divine guidance.

Alas, device enumeration has come and we are saved. An I for an O is no longer the rule of the land, but devices now merely ask and they shall receive. The bounty of interrupts and fruitfulness of MMIO are gifts granted upon enumeration, a baptism into a new order of hardware that Just Works.

Beware, friends. There are those that would have us believe that life is not easy. For we may still find need to open cases with screwdrivers, align cards in slots, and insert cables with retention clips. But this is merely a ruse! Deep down inside, it is new and enlightened, but still lives and acts as it has since the unenlightened times. Verily I tell you: there is a better way. Let us liberate this hardware!



7.1 PCIe is as easy as USB

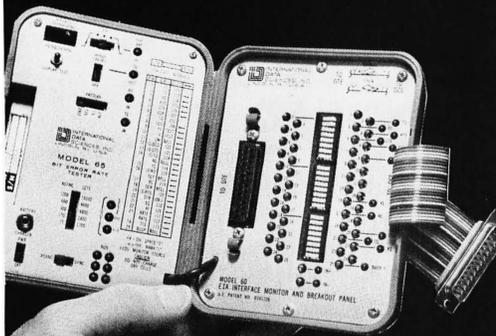
USB is great. We can plug stuff in, and it just works. If we need more ports, we can use a hub. Down below there's differential signaling. There's automatic speed negotiation. At the higher layers there are standardized structures that report all the INs and OUTs of the device. And these help software know exactly which drivers to load when the device is attached and identified.

PCIe is more similar than you might imagine. You plug stuff in and it just works, though it sometimes requires a shutdown. If you need more slots, you can use a switch. There's differential signaling automatic detection, and automatic speed

The 'Red Box'. Our dynamite duo!

Bit Error Rate Test Set – EIA Interface Breakout Panel in pocket size package.

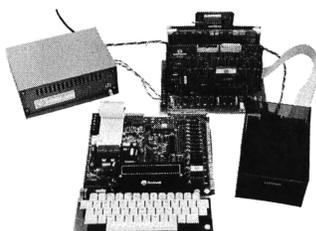
IDS'S MODEL 65/60 lets you both analyze and test at the EIA interface between a modem and terminal. Combines our popular "Blue Box" model 60 with a new bit error rate test set. All in one light, portable, hard plastic case. Works on rechargeable batteries. Available now.



INTERNATIONAL DATA SCIENCES, INC.
7 Wellington Rd., Lincoln, R.I. 02865
Tel. 401-333-6200 TWX 710-384-1911
Export: EMEC, Box 1285,
Hallandale, Fla. 33009 Telex 51-43-32

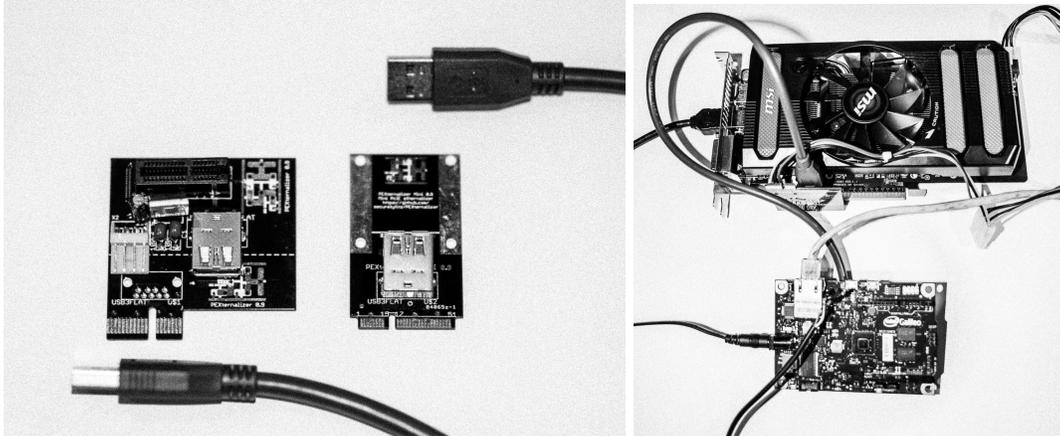
compas
microsystems

There is nothing like a
DAIM



A complete disk system for the Rockwell Aim 65. Uses the Rockwell Expansion Motherboard. Base price of \$850 (U.S.) includes controller with software in Eprom, disk power supply and one packaged Shugart SA400 Drive.

224 SE 16th St. AMES, IA 50010
P.O. BOX 687 (515) 232-8187



and width negotiation. Standardized structures report the details of the device, and allow software to know exactly which drivers to load.

The PCI SIG actually did a pretty darn good job with PCIe. They made it so that even if you screw everything up with your hardware design, it'll still probably work. Which also means we can screw around with it, hack things together and it'll still probably work too.

I have a divine vision I would like to share. I believe with all of my soul that, as long as we can get a couple wires hooked up properly, we can bring any PCIe host and PCIe device together.

Before you all tell me to GTFO, I'll get on with the PoC. Galileo is a board with a 400 MHz Pentium-class processor that has been kluged into an Arduino form factor. It has a MiniPCIe slot on the bottom which is *supposed* to only be used for Wifi adapters. But if I just stuck to what I was supposed to do I'd still be flashing LEDs and saving my graphics cards for real computers.

7.2 An Incongruous Fornication of Hardware

So, the PoC is to get this Arduino working with a Geforce GTX 650 Ti Boost. Because a 1.1 GHz, 768-core gpu with 2 GB of memory is a good mate to a 400 MHz single core CPU. First we'll talk hardware, then we'll gloss over the software.

We've got a PCIe 3.0 x16 device—sixteen TX pairs and sixteen RX pairs that run up to 8 GHz on a 164 pin connector. When the device first connects, the physical layer figures out how wide the link is and scales it down as necessary. In addition, the link starts at PCIe 1.0 speeds of 2.5 GHz and only 'retrains' to a higher speed if both ends support and the error rate stays low. Even at 2.5 GHz, we can do a crappy job wiring it and our data rate might suck—but thanks to fancy protocols and error detection it will probably still work.

So really, we only need four wires—two for TX and two for RX. Many devices work fine without a reference clock, but we'll throw in those extra 2 pins for good measure. The Galileo board has a MiniPCIe slot, and we've got a full size PCIe card that's five times the size of and twenty times the weight of the Galileo itself. We need some way of cabling them together.

The PCI SIG actually defines external cables for PCIe, but they're really expensive. Let's brainstorm. We need a cheap cable that can carry two 2.5 GHz pairs and one 100 MHz clock pair. hmmm. USB 3 cables! So, I threw together a couple boards—one to plug in the MiniPCIe slot, the other to plug the graphics card into, and USB 3 sockets to connect them. The slot-end board also has a 12 V/5 V power header and voltage regulator—MiniPCIe only supplies a little juice at 3.3 V while PCIe requires 12 V and 3.3 V. Pirate the board files by unzipping this PDF.¹⁰ You can get premade PCIe extenders/adapters like these on eBay or elsewhere, but what's the fun in that?

¹⁰`git clone https://github.com/securelyfitz/PEXternalizer`

```

1 root@clanton:~# lspci -k
00:00.0 Class 0600: 8086:0958 intel_qrk_sb
3 00:14.0 Class 0805: 8086:08a7 sdhci-pci
00:14.1 Class 0700: 8086:0936 serial
5 00:14.2 Class 0c03: 8086:0939
00:14.3 Class 0c03: 8086:0939 ehci-pci
7 00:14.4 Class 0c03: 8086:093a ohci_hcd
00:14.5 Class 0700: 8086:0936 serial
9 00:14.6 Class 0200: 8086:0937 stmmaceth
00:14.7 Class 0200: 8086:0937
11 00:15.0 Class 0c80: 8086:0935
00:15.1 Class 0c80: 8086:0935
13 00:15.2 Class 0c80: 8086:0934
00:17.0 Class 0604: 8086:11c3 pcieport
15 00:17.1 Class 0604: 8086:11c4 pcieport
00:1f.0 Class 0601: 8086:095e lpc_sch
17 01:00.0 Class 0300: 10de:11c2 nouveau
01:00.1 Class 0403: 10de:0e0b

```

So, plug everything in, attach an external power supply to the graphics card, power it up, and... nothing. Or so it would seem. But, we've got a serial console on the Galileo, so we can check it out by running `lspci`.

And there we have it! An Nvidia `0x10de` standing out in a sea of Intel `0x8086`. Our graphics card is connected, enumerated, and waiting for drivers.

7.3 Solemnization through Software

On a normal desktop, the BIOS starts up, runs the video BIOS that initializes the display, and gets on with things. But this is supposed to be a tiny embedded system. While it does boot via EFI, it doesn't run video BIOS or any option ROMs. We'll have to that by hand.

There's already great instructions by Sergey Kiselev on how to build your own Linux for Galileo available.¹¹ I mostly followed those to get a standard install working, but I had to make two changes between steps 7 and 8 of Kiselev's tutorial. We need to add all the X11 related packages, and we need to enable `nouveau`, the open-source Nvidia drivers, in our kernel configuration.

```

7.1. Add 'x11' to the DISTRO_FEATURES line in
2 meta-clanton/_vxxx/meta-clanton-distro/conf/distro/clanton-tiny.conf
7.2. Configure the kernel by running 'bitbake linux-yocto-clanton -c
4 menuconfig' and enabling nouveau under drivers->graphics->nouveau

```

Copy the resulting files to a MicroSD card, pop it in your Galileo, and you are a `modprobe nouveau` && `startx` away from what might be the most inefficient way to drive a display ever devised. Of course, there's no window manager or input devices yet configured, so you can't do much, but that's just a software problem, right?

¹¹<http://www.malinov.com/Home/sergey-s-blog/intelgalileo-buildinglinuximage>

