computers are bad

You are receiving this facsimile because you signed up for fax delivery of this newsletter. To stop delivery, contact Computers Are Bad by email or fax.

https://computer.rip - me@computer.rip - fax: +1 (505) 926-5492

2023-01-29 the parallel port

A few days ago, on a certain orange website, I came across an article about an improvised parallel printer capture device. This contains the line:

> There are other projects out there, but when you google for terms such as "parallel port to usb", they drown in a sea of "USB to parallel port" results!

While the author came up with a perfectly elegant and working solution, on reading that article I immediately thought "aren't they just being an idiot? why not just use a USB parallel port controller?" Well, this spurred me to do some further reading on the humble parallel port, and it turns out that it is possible, although not certain, that *I* am in fact the idiot. What I immediately assumed---that you could use a USB parallel controller to receive the bytes sent on a parallel printer interface---is probably actually true, but it would depend on the specific configuration of the parallel controller in question and it seems likely that inexpensive USB parallel adapters may not be capable. I think there's a good chance that the author's approach was in fact the easier one.

I wrote a popular post about serial ports once, and serial ports are something I think about, worry about, and dream about with some regularity. Yet I have never really devoted that much attention to the serial port's awkward sibling, always assuming that it was a fundamentally similar design employing either 8 data pins each way or 8 bidirectional data pins. It turns out that the truth is a lot more complicated. And it all starts with printers. You see, I have written here before that parallel ports are popular with printers because they avoid the need to buffer bits to assemble bytes, allowing the printer to operate on entire characters at a time in a fashion similar to the electromechanical Baudot teleprinters that early computer printers were based on. This isn't wrong, it's actually more correct than I had realized---the computer parallel port as we know it today was in fact designed entirely for printers, at least if you take the most straightforward historical lineage.

Let's start back at the beginning of the modern parallel port: the dot matrix printer.

There's some curious confusion over the first dot matrix printer, with some Wikipedia articles disagreeing internally within the same sentence. In 1968, Oki introduced the Wiredot. The Wiredot is probably the first "dot matrix impact printer" (dot matrix thermal and dyesub printers had existed for about a decade before), but the title of "first dot matrix impact printer" is still often given to the Centronics 101 of 1970. I have a hard time telling exactly why this is, but I can offer a few theories. First, the Oki Wiredot was far less successful on the market, seemingly only released in Japan, and so some people writing about printer history may just not have heard of it. Second, the Wiredot made use of an electromechanical character generator based on a permanent metal punched card, so it's arguably a different "type" of machine than the Centronics. Third, the Wiredot may have actually been more of a typewriter than a printer in the modern sense. The only photos I have seen (I think all of the same specimen in a Japanese computer history museum) show it mounted directly on a keyboard, and I can't find any mention of the interface used to drive it.

In any case, while the Centronics 101 is unlikely to be the *very first,* it is almost certainly the first widely successful example of dot-matrix impact printing technology. Prior to the Centronics 101, the most common types of computer printers had used either typewriter mechanisms or chain-driven line-printer mechanisms. These machines were expensive (I have heard as much as $30,000 in 1970 money for IBM line printers), physically large, and very tightly integrated with the computers they were designed for. With the inexpensive and small Centronics 101 a printer was, for the first time, a peripheral you could buy at the store and connect to your existing computer. The dot matrix printer was to the line printer somewhat as the PC was to the mainframe, and a similar (albeit smaller) "dot matrix printer revolution" occurred.

This posed a challenge, though. Earlier printers had either relied on the computer to perform all control functions (somewhat like the inexpensive inkjet printers that became ubiquitous in the '00s) or had a high-level interface to the computer (such as an IBM mainframe I/O channel) and a large internal controller. These approaches kept printers some combination of computer-specific and expensive, usually both. The new generation of desktop printers needed a standardized interface that was inexpensive to implement in both the printer and the computer.

For this purpose, Centronics seems to have turned towards the established technology of teletypewriters, where the bits of each character were used to select an appropriate type matrix. The dot matrix printer would work very similarly, with the bits of each character directly selecting an impact pattern template from a simple character generator (unlike the Oki's metal plates, the Centronics seems to have used some type of ROM for this purpose). This was far easier if all of the bits of each character were available at the same time. While computers mostly used serial data interfaces for components, the extra hardware to buffer the bits into each byte (given that the printer had no other reason to need this kind of storage) would have been an expensive addition in 1970. Since ASCII was a 7-bit standard, there was an implication that any interface to the printer should be at least 7 bits wide.

Centronics was an interesting company to make a printing breakthrough. Effectively a division of Wang (best known for their CRT word processors), Centronics had originally built special terminals for casino cashiers. A major component of these terminals was a slip printer for receipts and account statements, and the Centronics dot-matrix impact head was originally developed as a faster, smaller print head for these slip printers. As sometimes happens with innovations, this new form of high-speed printer with simplified head (consisting of one solenoid for each pin and only 7 pins needed for acceptable quality) became a more interesting idea than the rest of the terminal.

Centronics was not a printer company, and did not have the expertise to develop a complete printer. To close this gap they entered a partnership with Brother International, the US arm of Japanese typewriter (and sewing machine) maker Brother, whose president incidentally lived next door to the president of Centronics. And thus came about the Centronics 101, with the print head and control electronics by Centronics and basically all other components by Brother based on their typewriter designs.

I go into this history of Centronics as a company both because it is mildly interesting (Brother printers today are more the descendent of the original Centronics than the hollow shell of Centronics that survives as Printronix) and because the general flavor of Centronics going into the printer business with a next-door neighbor makes an important property of the Centronics 101 more explicable: the choice of interface. Centronics was getting this printer together quickly, with an aim towards low cost, and so they stuck to what was convenient... and incidentally, they had a significant backstock of 36-pin micro-ribbon connectors on hand. This style of connector was mostly used in the telecom industry (for the RJ21 connector often used by key telephones for example), but not unheard of in the computer world. In any case, since there was no established standard for printer interfaces at the time, it was just about as good as anything else.

And so, largely by coincidence, the Centronics 101 was fitted with the 36-pin micro-ribbon (also called Amphenol or CHAMP after telecom manufacturers) connector that we now call the Centronics interface.

The pinout was kept very simple, consisting mostly of a clock (called Strobe), 8 data pins, and a half dozen pins used for printer-specific status information and control. For example, a Busy pin indicated when the printer was still printing the character, and a Paper Out pin indicated that, well, the paper was out. Control pins also existed in the forward direction, like the Autofeed pin that indicated that the printer should linefeed whenever it reached the end of the line (useful when the software generating output was not capable of inserting correct linebreaks). Since 36 pins is a whole lot of pins, Centronics provided generous features like a pull-up reference and a separate ground for every data pin, besides chassis ground and shield.

Let's take a look at the details of the Centronics protocol, since it's fairly simple and interesting. Centronics printers received data one character at a time, and so I will use the term character, although the same applies to any arbitrary byte. To print a character, the computer first checks the "busy" pin to determine if the printer is capable of accepting a new character to print. It then sets the character values on the data 0 through data 7 pins, and produces a 1uS (or longer) pulse on the "strobe" pin. This pulse instructs the printer to read out the values of the data pins, and the printer places a voltage on the "busy" pin to indicate that it is printing the character. Once completed, the printer pulses the "acknowledge" pin and resets the "busy" pin. At this point, the computer can send another character. A couple of sources suggest that the "acknowledge" signal is largely vestigial as most computer implementations ignore it and send another character as soon as "busy" is reset.

Critically, the Centronics interface was completely directional. From the computer to the printer were the data bus and several control pins. From the printer to the computer were four control pins but no data bus at all. Returning to my original inspiration to read about the parallel port, this suggests that a parallel controller actually can't be used to receive data from a printer output, because it is only capable of transmitting on the data bus.

Of course, reality is yet more complicated.

The Centronics interface we're speaking of here is strictly the interface on the *printer*, and especially in the '70s that bore only a loose relation to the interface on the computer. Most computers would still need a dedicated printer controller card and a variety of proprietary interfaces were in use on the computer side, with various cables and adapters available to make a Centronics printer work.

As with many things in the world of microcomputers, there wasn't really any standardized concept of a computer parallel interface until the IBM PC in 1981. The IBM PC included a parallel printer controller, but not one intended for use with Centronics printers---the IBM PC was supposed to be used with the IBM printers, rebadged from Epson, and the IBM PC's printer port wasn't actually compatible with Centronics. IBM chose a more common connector in the computer world, the D-shell connector, and specifically the DB25. Because there are only a limited number of practical ways to implement this type of printer interface, though, the IBM proprietary parallel printer protocol was substantially similar to Centronics, enough so that an appropriately wired cable could be used to connect an IBM PC to a Centronics printer.

And that pretty much tells you how we arrived at the printer interface that remained ubiquitous into the late '90s: an adapter cable from the IBM printer controller to the Centronics printer interface.

The thing is, IBM had bigger ambitions than just printers. The printer port on the original IBM PC was actually bidirectional, with four pins from the printer (used to implement the four printer control pins of the Centronics standard) available as a general-purpose 4-bit data bus. The IBM printers seem to have used the interface this way, ignoring the Centronics assignments of these pins. Even better, the original IBM PC was capable of a "direction-switching" handshake that allowed the peripheral to use the full 8 bit data bus to return data. This feature was apparently unused and disappeared from subsequent PCs, which is ironic considering later events.

The four printer control pins provided by Centronics were rather limited and more sophisticated printers had more complex information to report. To address this need, HP (which had adapted the Centronics interface nearly from the start of their desktop printers) re-invented IBM's simpler bidirectional arrangement in 1993. The HP "bi-tronics" standard once again made the four printer pins a general purpose data bus. Since the computer already needed the hardware to monitor these four pins, bi-tronics was compatible with existing Centronics-style printer controllers. In other words, it could be implemented on the PC end as a software feature only. All parallel printer interfaces capable of the Centronics standard are also capable of bi-tronics, as long as suitable software is available.

The lopsided nature of this arrangement, with 8 bits forward and 4 bits reverse, became an irritation to the new generation of computer peripherals in the '90s that demanded high bandwidths. Scanners and ethernet interfaces were made with parallel ports, among many other examples. To address this need, Intel, Xircom, and Zenith introduced the Extended Parallel Port or EPP standard in 1991. I would love to know which product from Zenith motivated this effort, but I haven't been able to figure that out. In any case, EPP was basically a minor evolution of the original 1981 IBM PC parallel port, with a more efficient handshaking mechanism that allowed for rapid switching of the 8-bit data bus between the two directions.

Because standards need friends, Hewlett-Packard and Microsoft designed the Extended Capability Port or ECP just a year later. ECP was actually quite a bit more sophisticated than EPP and resembled later developments like Firewire in some ways. ECP parallel ports were not only fully bidirectional but supported DMA (basically by extending the ISA bus over the parallel port) and simple compression via run-length encoding. This might sound overambitious for the '90s but it found plenty of applications, particularly for storage devices. Perhaps one of the most notable examples is the first generation of Iomega Zip drives. They were internally all SCSI devices, but SCSI controllers were not common on consumer machines. Iomega offered a parallel port version of the drives, which relied on a proprietary ASIC to convey SCSI commands over an ECP parallel port. ECP's DMA feature allowed for appreciably better transfer speeds.

This setup is very similar to the modern UASP protocol used by many higher-end USB storage devices... SCSI is one of those protocols that is just complex enough that it will probably never die, just adapt to new media.

The parallel printer port remained only loosely standardized, with most supporting some combination of "standard" mode (also called SPP), EPP, and ECP, until, surprisingly, 1994. IEEE 1284 aimed to nail down the mess of parallel ports and modes, and while 1284 is superficially viewed as a standardization of the Centronics connector, it really does take on the whole issue. IEEE 1284 specifies five modes:

- Compatibility Mode, which behaves identically to the original Centronics specification. This is also called SPP for Standard Parallel Port.
- Nibble Mode, which uses the four printer control lines of the Centronics standard as a 4-bit data bus in the reverse direction, like the original IBM PC and printers.
- Byte Mode, which uses the 8-bit data bus for both directions with a simple handshaking arrangement to switch directions. Byte Mode is largely similar to, although not compatible with, the original IBM PC bidirectional feature.
- EPP, as specified by Intel/Xircom/Zenith
- ECP, as specified by HP/Microsoft

Because EPP and ECP both specify distinct handshake protocols, it’s mostly possible for a parallel controller to automatically detect which mode a device is attempting to use. This has some rough edges though and becomes more unreliable when dealing with devices that might use the pre-EPP byte mode, and so most parallel controllers provided some way to select a mode. In modern motherboards with parallel ports, this is usually in the form of a BIOS or EFI configuration option that allows the parallel port to be set to SPP only, EPP, or EPP/ECP automatic detection. Usually this can be left on EPP/ECP, but some legacy devices that do not implement normal IEEE 1284 (and legacy EPP/ECP) negotiation may not be able to set up bidirectional communications without manually putting the controller in a more basic mode.

And that brings us to the modern age, with parallel ports via EPP or ECP capable of bidirectional operation at around 2Mbps. But what of my original reaction to the parallel printer emulation article? Given what we know now, could the author have just used a USB parallel controller? I suspect it wouldn’t have worked, at least not easily. From my brief research, common USB parallel adapters seem to only implement the the unidirectional "compatibility mode" of IEEE 1284 and actually appear as character printers. More expensive USB devices with ECP support are available (routinely $50 and up!) as well as PCI cards. The IEEE 1284 spec, though, requires one of a few handshake processes before the port enters a bidirectional mode. I suspect it would require modification to the drivers or worse to read the presented byte without completing a negotiation to byte mode or EPP, which of course an older device intended only for printers wouldn’t support.

There’s a bit of a postscript about Centronics. As with many early technology companies, Centronics made a few missteps in a quickly moving market and lost their edge. Centronics partnership with Brother may have proven fatal in this regard, as Brother introduced their own print head designs and rapidly became one of Centronics chief competitors. Epson, Citizen, and Oki gained significant ground in dot matrix printers at around the same time. Centronics, no longer a market leader, was purchased by Control Data (CDC) to be merged with their printer business, CPI, itself formed through CDC’s previous purchase of slip and receipt printer businesses including that of National Cash Register (NCR).

This was in the early '80s, and while dot matrix printers were still very common in that time period it was becoming clear that business and accounting users would be one of the dominant ongoing markets: perhaps the greatest single advantage of dot-matrix printers is that they were the superior choice for filling out pre-printed forms, for several reasons. First, dot-matrix print heads could be made very small and with an impact ink-transfer design could print on just about any material. This was highly advantageous for slip printers that printed on small-format preprinted media. These types of printers are no longer especially common except at grocery stores where they are often integrated into the thermal receipt printer and referred to as check endorsers, because they are used to print a "deposit only" endorsement and transaction reference number on the back of check payments.

Second, impact dot matrix printers use a formidable impact force and can effectively print on several layers of carbon or carbonless multi-part forms. This was very convenient in applications where a multi-part form was needed to transfer handwritten elements like signatures, although it has to some extend faded away as it has become more common to use a laser printer to simply print each part of a carbonless form individually.

Third, dot-matrix printers were so popular for form-filling applications that many featured built-in registration mechanisms such as an optical sensor to detect notches and, later, black bars printed on the edges of forms. This allowed the "vertical tab" character to be used to advance the form to a known position, ensuring good alignment of printed information with preprinted fields even with a feed mechanism that was prone to slippage. Of course many dot-matrix printers used non-slipping drive mechanisms such as perforated edge tapes, but these drove up the cost of the paper stock appreciably and users found tearing them off to be pretty annoying. Unfortunately, as dot matrix printers have faded from use, so too has software and printed media support for registered vertical tab. Purchasing a car a year or two ago, I chuckled at the dealership staff using the jog buttons on an Oki printer to adjust the form registration part way through a particularly long preprinted contract. The form had black bars for vertical registration and the printer model was capable, but it seemed the software they were using hadn't implemented it. The Oki printer in question was of the true "form filler" type and both fed and returned media from the front, making it convenient to insert preprinted forms to be filled.

Centronics did not last especially long under CDC ownership, and met a particularly odd fate. In 1987, Centronics, then under ownership of an investment bank, sold its printing business to competitor Genicom (now part of Printronix). The remainder of Centronics, essentially just an empty shell of a corporation flush with cash from the sale of its former operation, purchased kitchen goods and housewares manufacturer Ekco and renamed itself to that brand. Ekco, in some sense once a leading innovator in computer printing, is now a brand of measuring spoons manufactured by Corelle.