TLDR: Free software/open hardware advocates are hypocritical if they are using proprietary hardware and software to design free software/hardware while making the cost of hardware more expensive /cost prohibitive to those in the third world by purposefully making hardware that does not fit like Legos under a laptop hood.

10/14/22: A short video I made on the topic. The speed can be set to 2x with nothing lost ;)

10/15/22: George Banks is saying “No!” (Do not try this IRL)

Update: 10/6/2022 If one truly wants to develop interoperable computer hardware that can be built for a laptop (not referring to microcontrollers), one would have to design a mobile motherboard that fits other hardware- monitor, keyboard, trackpad, power, and not try to control anything else about the design. I would call it something like the SBC-ITX form factor, or mobile ITX-SBC. Motherboard manufacturers like Gigabyte, Biostar, Asus/ASRock have a major product line just for motherboards! They do not need to know what size of the displays or laptop chassis you will use, just the connector and mounting holes! The connector changes over time- the VGA/DSI/HDMI/DP, etc, but the PC/104 form factor (later iterations like ATX) does not! They also expect builders to put it together! Even though not everyone can build a computer from scratch, designing and manufacturing a more accessible and modular product greatly *increases* the number of people who can, by demystifying all-in-one designs like iMacs. But if you look at open source designs like like MNT reform and repairable phones like Fairphone, they designed the entire chassis/enclosure, which limits the modularity of it. By contrast, a motherboard architect designing a board for a laptop can reuse the chassis if only they have a connector (and ribbon cable) that can reach the display, such as DSI or eDP. The intrinsic flexibility of physical space (air!) and physical ribbon cables is what makes ATX cases so *flexible*. SBCS are largely, by design, or unwittingly, overstepping the boundaries of a flexible form factor by forcing an orientation to not become future proof. By designing an entirely unique laptop, one has created an ecosystem.

10/15/22: It could be said that both designer and consumer engage in a form of cognitive dissonance. They are unwilling or unable to determine if that their design or purchase is an aesthetic preference, or a rational byproduct of an impulse decision. In the absence of market research, or community consultation, one might very well make a unilateral or peer-based decision that is somewhat limited. That said, it could also be a bias that fewer designers seek to develop a standard than designers who prefer less standardization. Some designers are brilliant engineers, but not all engineers are brilliant architects. The difference between architecture and engineering is like the relationship between an individual and urban architecture. A city is more modular than the country side, in that apartments/condos comprise the majority of housing. The individual may have the choice to walk, use a train, or use a bike/car, even in a city. But if everyone in a city chose to use a car, there would be gridlock. In a way, an engineer can design an elegant electric SUV, but city adoption is going to be limited by the number of users willing to buy that SUV. Thus the designer may have a city in mind, but may find their market is actually a suburban or rural area.

It is possible to make two factual statements that may appear to be cognitively dissonant. For example, if I state it is arbitrary to design a board with 4 mounting holes on a 3”x3” PCB in 4 positions different from its 2nd generation PCB, I could also state that it is less arbitrary to place them in the same location for each new generation of PCB/SoC. The interpretation of the 2nd statement can appear to contradict the first, when it fact it uses the qualifier “less” in “less arbitrary” to modify the generalization that standardization is absolutely arbitrary. Thus, design considerations could be non-existent, if the critical questions on the positioning of mounting holes are antagonistic to the designer.

Back to form factors, more often than not, more will not agree with or be missing at least one component of that ecosystem- -but modifying it to fit what they want is either cost-prohibitive such as requiring a new laptop or too time consuming to modify- the size of it, the color, the texture. That is why open laptop designers (especially competing designers) first should focus on just agreeing to standardizing the most core component- the motherboard dimension. Users may not like one manufacturer's choice of display but they may like the chassis. So they can be sold separately! The more modular it is, the more combinations that can be used- like Legos! Even though not all makers will use it for a portable laptop, they may see it as an SBC, so why make it so hard to carry around, requiring the use of a portable monitor and USB keyboard, when it could all be integrated in a standardized, mobile interoperable form factor that encompasses a very genericized laptop shape? I compare it to the greatest common divisor - CPU, keyboard, trackpad and display. Everyone who buys a laptop uses those features.

Yet SBCs fail to make a form factor that can serve those basic functions. This approach should be examined more in the laptop design market. There's actually a word for it in physical chemistry: degrees of freedom. When an object needs to connect another component, especially in a space-constrained laptop, as soon as a design decision is made for one orientation or component, it limits the degrees of "freedom" of another component to be used, or where those components can be placed. Some placements of components are arbitrary, and some are not. If standardization is not important, placing an HDMI connector near a plug may be non-consequential, if the HDMI is practically near where the power cord is (i.e. the back of the laptop- which is convenient from the user experience perspective). However, if there is a specific need to fit a new display connector on top of the SBC (such as eDP in the Pine64's Quartz 64), since one of the popular uses are to *build* a laptop, not sell a turnkey product such as a fully manufactured laptop with windows or linux preinstalled, then it benefits the Maker community, or PC repair person, to connect a new laptop display without needing to change out the chassis- which would be a waste of plastic and money.

This would (gradually) shift the laptop hardware market to a component market, just like Gigabyte and Asus have done with desktop PCs for years. So what? Is the open hardware community too good for that level of industriousness? I understand my words can seem harsh, and they aren't meant to denigrate anyone's efforts. Rather it is to determine whether the open hardware designers are aware of this alternative hardware design, or whether it is a preference to design something to their own preferences. When designs are submitted to a community, it is possible that one may become more popular, and one may be designed more sleek than others. But it doesn't mean it will necessarily appeal to a large enough audience to be adopted the world over. It could also be that some are afraid of an ambitious project- I am too, at times. Consider this:

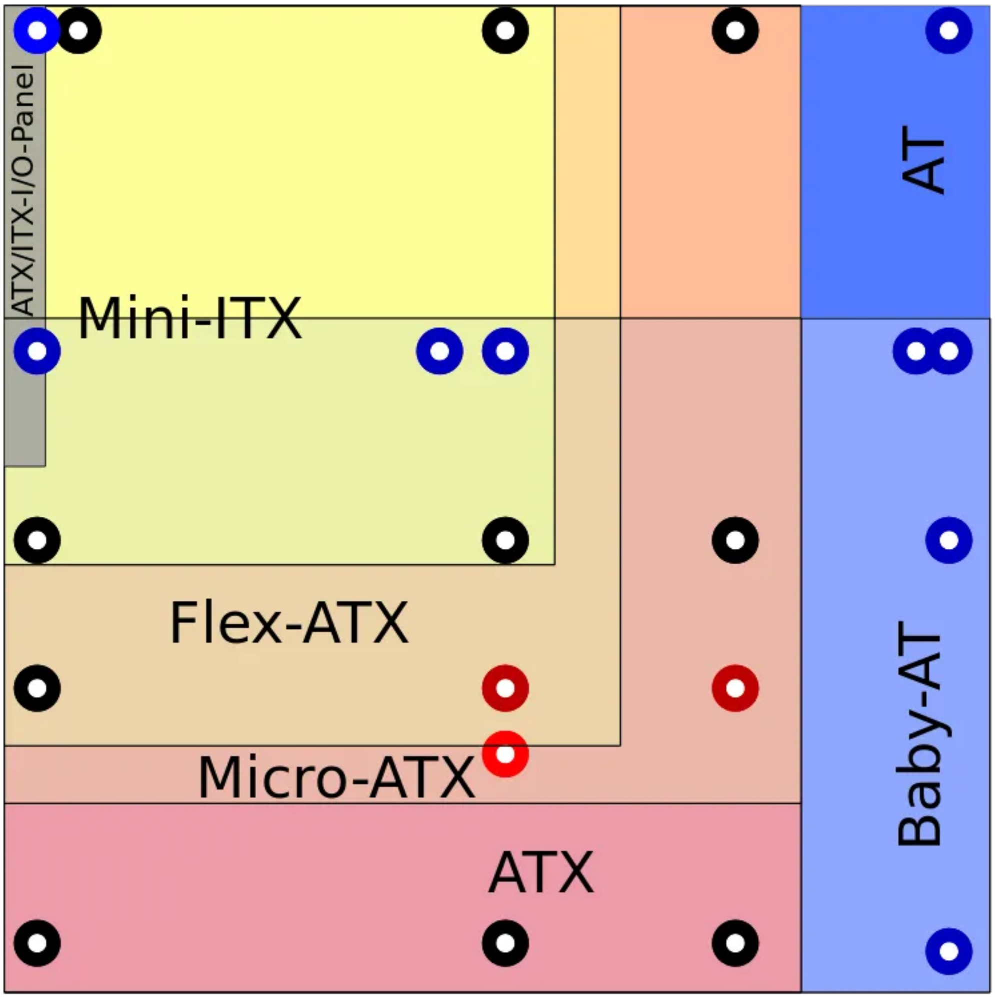
A software expert (their name will be withheld since it is not relevant) has stated something that I agree with: "People don't buy hardware or software. They buy solutions." I do not disagree with everything that people say. I may disagree with an approach to how they try to match the solutions of consumers. There is one group of consumers, called the "end user." The end user can be someone who buys a laptop to just do online banking, or it could be a power user who programs a lot. So, the word end user can be somewhat broad because they may both use Macbooks, and there aren't many Macbooks to choose from.

Now, lets consider a different end user. In the Maker community, websites like Tindie, Adafruit, Hackaday, sell components and hardware from Digikey, Mouser. The target end user here is not a novice who necessarily needs a pre-built laptop. It may be someone who is building a digital picture frame for their relative, or a laptop.

Let's call them the "intermediate end user"- not a power user, but not a novice either. They also comprise a fairly large part of the market- not a majority, but a large, double digit percentage.

This end user can put together a laptop. They don't want a blue laptop- maybe they want a pink one, or an orange one. They need a motherboard, and they can build the rest. What this end user is buying- the components- is actually a solution, just like the novice and power user. But their solution, is only the middle step of their project. They know how to put together a laptop if is designed like a mini-ITX form factor, except even smaller. They can't build the components, as they do not have the expertise to build something more advanced. That is the only thing they can't manufacture. But they can put it together. Motherboard manufacturers like Gigabyte, Asus/ASRock and Biostar understand that.

So if one is going to design a laptop- their Legos should be compatible with other laptop designers' Legos.



If one examines the MicroATX and Mini-ITX form factors above, you'll notice that the corner mounting holes overlap on the top and left (the black ones). So a form factor may use 6 or 9 screws, depending on the size. SBCs do not even have to be the same size to match. They just need to overlap in 4 areas, so that the mounting screws can be used in the same location for the smaller boards, and left empty for larger boards. A larger SBC could use 6 mini screws (not the size as the ones on the ATX cases).

One difference is that SBCs do not typically have PCI-Express, but may have M.2. This could be added to a board, but I do not think it is necessary. The form factor itself is sometimes more about providing access to a platform, rather than dictating what must be on the board. Hence, not all mini-ITX motherboards have PCI-express, even though they have enough space for one. The same concept could be applied to SBCs.

Original 9/30 post below:

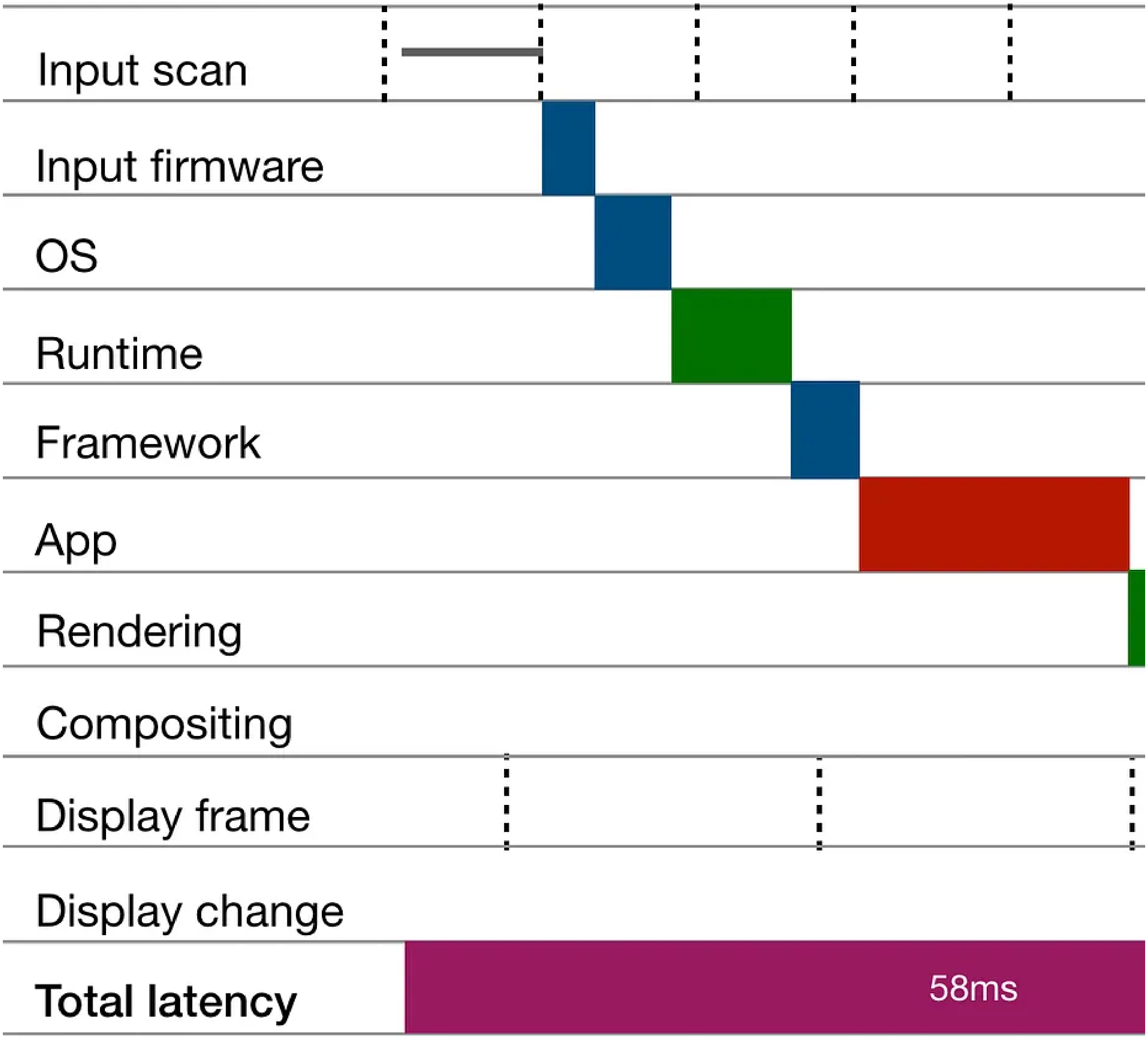
Dear Lord, what happened to the mobile ITX standard?! The year was 2007, before the housing crash caused a financial meltdown across the world. The housing bubble was making people feel….bubbly, for a while. At Computex 2007, a grainy demo on youtube shows, the mobile ITX made its world premiere in Taiwan. The red carpet was rolled out (no it was not). Yet this product never made its way into portable laptops, and the builder community was stuck at home, building desktop PCs in oversized ATX cases. Woe is the portable builder. Maybe some manufacturers didn't like the competition, and the standard quickly disappeared. Who knows? I don't.

Fast forward to 2012, and the $25 Raspberry Pi is released. Are you kidding me? $25? We don't need no stinking mobile-ITX standard. For that price, heck, I'll tip the the Rpi an extra $3, just for how inexpensive it is (Ok, I haven't, but I own a Rpi Zero and 3B+). But now System on a Chip development has become forked design ecosystem, much like Linux distributions and every new system on a chip that aims to be a single board computer with HDMI, USB, ethernet is either unaware of the era when motherboards had standard form factors, or is blasé to any shared standard. Freedom above all, the FSF philosophy insists.

But what is lost, when this "freedom" to be unique is dogmatically pursued? This stratification is not unlike social movements in the past. The libertarian streak in the free software community essentially believes along the likes of Ben Franklin, who states, "Those who would give up essential Liberty, to purchase a little temporary Safety, deserve neither Liberty nor Safety." Do Linux users feel safe under the protectorate of kernel maintainers? Have they given up their liberty? Maybe they traded their liberty long ago, by refusing to question the monolithic status quo of operating system designs found in Android, Smart TVs, Chromebooks, and Desktop PCs. There's enough silicon to entertain the whims of every device driver under the sun. Software development has been surfing on the undulating waves of shrunk process nodes thanks to Moore's Law for over 50 years. The linux kernel of today is an aircraft carrier, when in many cases, just a bike is needed.

It is no surprise the veterans of open source are comfortable with more easier software development platforms, because when they were working on Unix, things were hard. They built the operating systems we take for granted. So give that generation a break. Let them enjoy their golden years. It is up to the geriatric millennials like me to pick up where limited memory software development encouraged innovation in the mid 1990s. After the 1990s, there was much more RAM available to developers, which loosened the compactness of code needed to run cell phone apps efficiently. The early Nokia phones had robust user interfaces, and after the Symbian OS became discontinued, the smartphone era led to increasing latencies.

As their illustration points out, every software layer above the firmware or kernel adds additional complexity to the responsiveness of the interface:



source: https://www.inkandswitch.com/slow-software/static/input-latency-cascade.png

Back to the mobile ITX, it becomes clear that modularity in both software and hardware development helps reduce complexity in design and development. Portability of code is one thing, but modularity is an equally instrumental component of software and hardware solutions. Software and hardware need not be viewed as separate. It is obvious that software should be platform portable, and porting is a routine effort. But little consideration is made to hardware's portability.

It may be easy to suggest that I am comparing "modern" smartphone and system on a chip hardware platforms to some reification of an antiquated or presumably incompatible form factor, the ATX line of motherboards.

"*Reification* is when you think of or treat something abstract as a physical thing."

When I'm referring to an ATX standard, I'm not referring to the promotion of a consortium of (established) brands that profit from a de-facto standard. I'm referring to the abstract adoption of a voluntary standard- a form factor- that has not yet been determined for a mobile platform. A form factor, whether it is the Raspberry Pi orientation, or the BeagleBoard, or the Orange Pi, can be a de-facto standard- if enough users want to use it. It makes little sense to encourage users to switch form factors if there is already a ubiquitous one. However, that is not the case with the Raspberry Pi or Orange Pi. Each generation changes where the ports are oriented for mini HDMI, USB and the likes. This may be a moot point, for those who are not using the system on a chip for a laptop or portability. But how much of an inconvenience would it be for there to be a standard form factor in the open source community, one that could reuse chassis designs from 10 years ago?

System on a chip designs have several different ways to attach CPU modules to I/O. Some use a daughter board, as in the mobile ITX, MikroElectronika's SiBRAIN Card Sockets and SparkFun's MicroMod line of microcontrollers. But general purpose CPU chips (application processors such as ARM Cortex-A series, RISC-V, x86, and MIPS and sockets have not yet been developed or commercialized. KiCAD is a free software that helps designers build PCBs. The tools to make a standard laptop appear to be somewhat more democratized than ever before. Much like the early linux kernel developers, the first PCB maker to develop an open-ended PCB, one that is multi purpose enough for others to use, would appear to be a more widely adopted form factor than one that only serves their interests. How can I make such a statement? Am I not aware that PCB design is extremely difficult and is it true I don't even know how to design a PCB? Yes. It's true. But I'm not opposed to independent or unique designs. I am just writing because this issue seems to be an esoteric concept understood by just a few developers.

Imagine you have a system on a chip that gets fried because you plugged in the wrong power adapter (I've done that before with an SSD on an ATX motherboard, using the wrong SATA power cable from a different brand of modular ATX PSU). So now you're out of a chip. You spent money on a custom 3D printed case that you ordered from Etsy that is designed for your single board computer. Now that the CPU is fried, you can't use it anymore and decide to toss the case, since a new model is released with HDMI out on the opposite side of the board, and it no longer fits. You decide selling it on eBay isn't worth the shipping or time to list it, as no one would pay an extra $5, especially when they can get buy a new one for less if they get free shipping with some other items. One option is to donate it.

That is the cycle of consumerism. Liberty encourages the idea of disposable electronics. Yet the paradox is, the more developers support the concept of a walled garden of repair services (often due to the fact of proprietary repair tools being the only way one can properly repair a computer), the less this tendency supports the right to repair. After all, it is not just a legislative issue, but a technical issue. The design of hardware that cannot be re-used or salvaged makes it less valuable to those who value longevity and durability. The competitors to value-based products base their business model on planned obsolescence. The assumption that one cannot have repeat customers, or not earn enough profit from a product designed to last as long as possible appears to be one of the greatest mysteries of Keynesian economics. Ever since the Great Depression, supply and demand has been the focus (or lack thereof) of economists and believed to impact recessions as well as inflationary periods. Output sometimes increases if there is a fear of a recession. If the cost to produce something is low, and the margin is high, why not produce more? Thus, the abundance and inexpensiveness of a good in an era shapes that generation's awareness of the value of a product (or lack thereof), whereas, the scarceness of a good in an era, likewise, shapes the awareness of that product's rarity.

Hence I conclude this post. Plastic and microprocessors may be cheap today, but as many have seen with shifts in the supply chain, we can limit our impact on the increasing waste in environment if we could appreciate a little more the amount of effort it took to mine, manufacture, package, ship and deliver that product.

Thanks for reading IoT's Newsletter! Subscribe for free to receive new posts and support my work.

Type your email… Subscribe