Channel your inner Wolverine with these 3D-printed, muscle-controlled bionic claws | Arduino Blog

In the fictional Marvel Universe, Wolverine has sets of claws that pop out of his hands as if they were natural parts of his body. While a seemingly fantastic concept, myoelectric sensors are able to pick up on muscle movements in order to illicit a response. YouTuber MERT Arduino & Tech decided to take this concept and build a pair of forearm-mounted claws.

The wearable device senses muscle activation via a MyoWare muscle sensor, which sends information on to an Arduino Nano on a custom carrier board. Depending on the signal, it's able to extend or retract claws, with the help of a servo motor and linkage system.

It's Been 10 +1 Years Now! [3]

Today, while I was wearing it, I realized that tomorrow is the anniversary of my blog.

So, after 11 years, I am still a Linux user.

Who would have thought?

My walk started almost by accident, with a modest Asus Eee PC with Xandros preinstalled (The legendary MiniMe!) I never thought that such a humble machine with change the way I use computers in such a profound way.

I was re-reading the first entry that I posted on this blog and a bitter-sweet wave of memories made me smile.

So much has changed since then!
Yet, I am glad that I can rely on my Linux computers, just in the same way like the day I quit Windows. Linux has truly made my work easier and less worrisome... (Boy! I've been online without an antivirus for 11 years!)

- **Imperator: Rome from Paradox is put on hold to focus on other projects**

  Paradox Interactive have announced that they've put Imperator: Rome effectively on hold for all of 2021.

  As probably their least popular modern full game, Imperator: Rome struggled from the moment it was released. Paradox have tried to overhaul parts of it with some free updates, and some DLC for true fans but it appears all their work so far just hasn't been enough to turn it around in players eyes and in a business sense.

- **How to Secure WordPress Installation with Bedrock ? Google Cloud**

  How to Secure WordPress Installation with Bedrock on Google Cloud Platform. Bedrock is a WordPress boilerplate with a different improved directory structure and easier configuration.

  It is more secure by isolating the web root and limit access to non-web files and more secure passwords using wp-password-bcrypt which replaces the MD5 hashing with modern bcrypt method.

- **EasyOS: Recompile in OE for aarch64 target**

  Have done a complete cross-compile in OpenEmbedded, host system EasyOS Dunfell version 2.7.2 x86_64 CPU, target aarch64 (ARM64). These are the same packages as in Easy 2.7.x on a x86_64 PC.

  Great, it went right through, except for 'flsynclient' package -- will tackle that later.

- **Nouveau Working On Bringing Up Some OpenGL Compute Shader Support For NV50 Era GPUs**
Open-source "Nouveau" driver developers have been working on at least partial support for OpenGL compute within the NV50 Gallium3D driver that is used by the NVIDIA GeForce 8 series through GeForce 300 series graphics cards.

Longtime Nouveau contributor Ilia Mirkin has been working recently on compute support for NV50 Gallium3D driver for pre-Fermi GPUs. The focus with this compute support seems to be about working towards OpenGL ES 3.1 requirements where ARB_compute_shader is mandated.

- **AMD Begins Prototyping CRIU Support For ROCm Compute - Phoronix** [8]

  As part of AMD's growing HPC focus and maturing of their Radeon Open eCosystem GPU compute stack, they ended out this week by making public a prototype implementation of CRIU support for ROCm.

  AMD's Radeon open-source graphics software developers are working on Checkpoint/Restore In Userspace (CRIU) handling for ROCm. CRIU allows the ability to freeze a running process and archiving it to disk that can then be thawed/restored later on. This user-space-based solution is, of course, much more tricky when it comes to handling processes interacting with the GPU.

- **Perl Weekly Challenge 110: Valid Phone Numbers and Transposed File** [9]

  These are some answers to the Week 110 of the Perl Weekly Challenge organized by Mohammad S. Anwar.

- **EV charge pricing per State in US** [10]

  Continuing from my previous post (http://blogs.perl.org/users/itcharlie/2021/04/calculating-ev-battery-charge-with-perl.html) I learned that residential electricity charges are calculated using an electricity supply rate per kWh and a electric delivery rate per kWh which actually increases the total EV charge prices that I have calculated in my previous post. My current bill states that my electric supply rate is at 6.9057 cents per kWh and my delivery rate is 11.1785 cents per kWh making it a total of 18.0842 cents per kWh which is 5 cents more than my original post at $0.13 cents per kWh.

The Async Vision Doc effort has been going now for about 6 weeks. It’s been a fun ride, and I’ve learned a lot. It seems like a good time to take a step back and start talking a bit about the vision doc structure and the process. In this post, I’m going to focus on the role that I see vision docs playing in Rust’s planning and decision making, particularly as compared to RFCs.

**Vision docs frame RFCs**

If you look at a description of the design process for a new Rust feature, it usually starts with “write an RFC”. After all, before we start work on something, we begin with an RFC that both motivates and details the idea. We then proceed to implementation and stabilization.

But the RFC process isn’t really the beginning. The process really begins with identifying some sort of problem1 ? something that doesn’t work, or which doesn’t work as well as it could. The next step is imagining what you would like it to be like, and then thinking about how you could make that future into reality.

We’ve always done this sort of framing? when we work on RFCs. In fact, RFCs are often just one small piece of a larger picture. Think about something like impl Trait, which began with an intentionally conservative step (RFC #1522) and has been gradually extended. Async Rust started the same way; in that case, though, even the first RFC was split into two, which together described a complete first step (RFC #2394 and RFC #2592).

The role of a vision doc is to take that implicit framing and make it explicit. Vision docs capture both the problem and the end-state that we hope to reach, and they describe the first steps we plan to take towards that end-state.

---

**Misc**

**Source URL:** [http://www.tuxmachines.org/node/150698](http://www.tuxmachines.org/node/150698)

**Links:**

[1] [http://www.tuxmachines.org/taxonomy/term/78](http://www.tuxmachines.org/taxonomy/term/78)
[2] [https://blog.arduino.cc/2021/05/01/channel-your-inner-wolverine-with-these-3d-printed-muscle-controlled-bionic-claws/](https://blog.arduino.cc/2021/05/01/channel-your-inner-wolverine-with-these-3d-printed-muscle-controlled-bionic-claws/)
[3] [http://mandrivachronicles.blogspot.com/2021/05/its-been-10-1-years-now.html](http://mandrivachronicles.blogspot.com/2021/05/its-been-10-1-years-now.html)
[4] [https://www.gamingonlinux.com/2021/05/imperator-rome-from-paradox-is-put-on-hold-to-focus-on-other-projects](https://www.gamingonlinux.com/2021/05/imperator-rome-from-paradox-is-put-on-hold-to-focus-on-other-projects)
[5] [https://www.cloudbooklet.com/how-to-secure-wordpress-installation-with-bedrock-google-cloud/](https://www.cloudbooklet.com/how-to-secure-wordpress-installation-with-bedrock-google-cloud/)
[11] [http://smallcultfollowing.com/babysteps/blog/2021/05/01/aic-vision-docs/](http://smallcultfollowing.com/babysteps/blog/2021/05/01/aic-vision-docs/)