

Machina Speculatrix

Home

About

★ Member-only story

MICROCONTROLLERS

The joy of AVR 8-bit microcontrollers

Microcontrollers are becoming insanely fast and capable. But there's something about the simpler types that appeals.



Mansfield-Devine · Following

Published in Machina Speculatrix · 4 min read · Mar 1, 2025



101



Open in app ↗

Medium

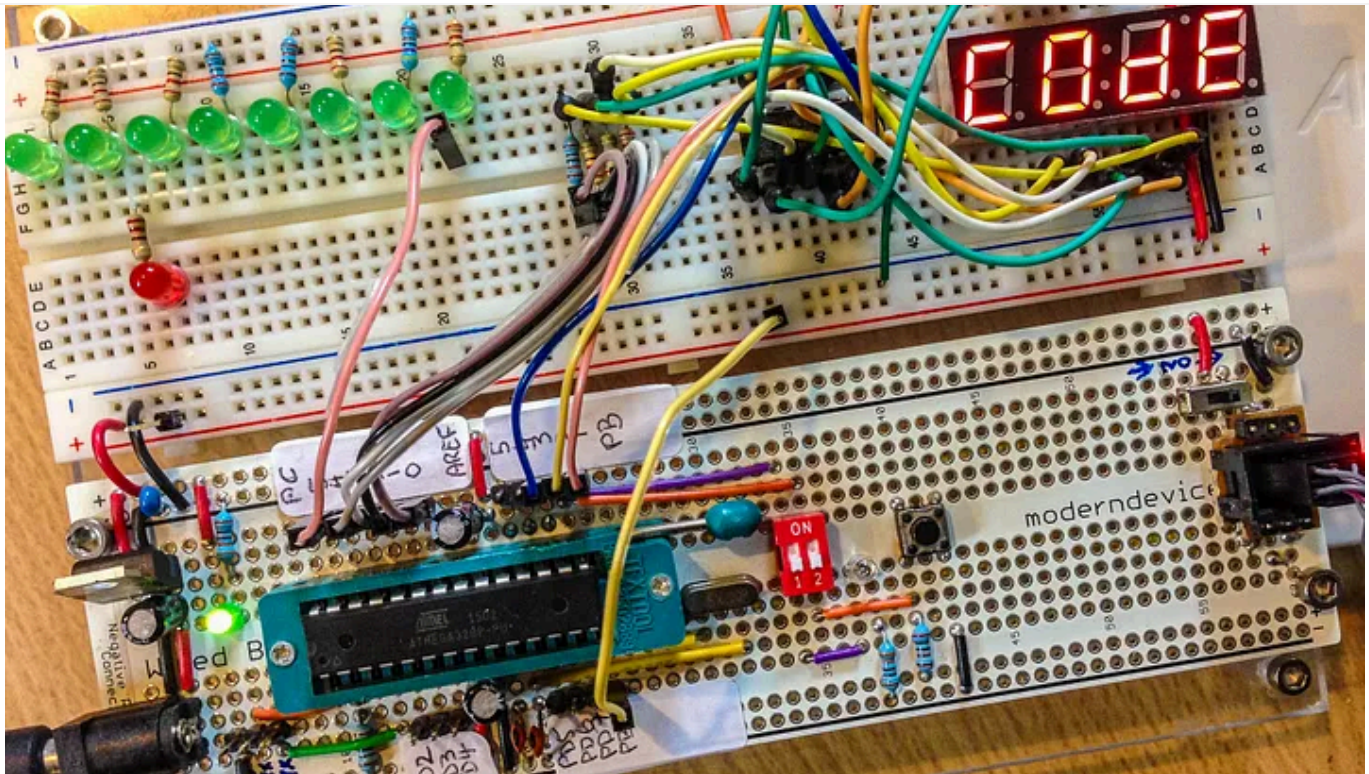


Search



Write





Keep it simple, stupid (KISS) is advice I live by.

We live in an age of immense computing power. Every day we carry in our pockets devices that would once have been considered supercomputers.

But if your passion lies in understanding technology — pulling it apart and hacking it — then there's much to be said for the smaller, cheaper, simpler end of the computing spectrum. And by that I mean microcontrollers.

Complete package

A microcontroller is a complete computer on a single chip. It has a processor (sometimes more than one), memory, storage and input/output (I/O) capabilities. Once programmed, it stores its code in non-volatile memory so that, when powered up, it jumps straight into performing its task.

They are everywhere. If you drive a modern car you're constantly accompanied by countless microcontrollers. They are in ovens, dishwashers, lightbulbs, fridges, alarm clocks, christmas tree lights — pretty much anything electronic that does more than simply switch on and off and isn't a full-blown computer.

And some microcontrollers are so powerful now that you can emulate entire computers of yesteryear on them. The justly popular ESP range even gives you built-in wifi.

But even within the microcontroller domain, I'm drawn towards the more basic devices — specifically AVR's 8-bit ATMEGA and ATTINY devices. And I'm not alone. The Arduino achieved its huge popularity using the ATMEGA328 microcontroller (although the range has moved on to more complex and capable chips since).

So what's the attraction?

Down to the basics

Even if you're writing code for microcontrollers in a relatively high-level language like C (and I say 'relatively' because we all know C is really just assembly language in a pretty frock), coding AVR microcontrollers encourages you deal more directly with the architecture, such as understanding registers.

You can get an Arduino and use its fine integrated development environment (IDE) and dazzling array of libraries. But I would strongly encourage you to dig deeper, learn to access the hardware more directly. Perhaps even delve into assembly language.

Digging deep into the machine

I lost my computing virginity to a Sinclair ZX Spectrum but quickly graduated to the much more capable BBC Micro. This had a powerful Basic — one that I personally think was never bettered on 8-bit machines and even on any 16-bit computers. But the venerable old Beeb also made it easy to try your hand at assembly. You could include 6502 assembly code inside Basic programs, to handle just the bits of the program that needed to go fast.

The Raspberry Pi is widely regarded as the spiritual descendent of the BBC Micro — not least because Eben Upton conceived it that way. It's no coincidence that both machines originally came out in 'A' and 'B' models. And the appeal to the Beeb's heritage is not without justification. Like the Beeb, the Raspberry Pi has encouraged many people — and especially children — to get their coding hands dirty.

However, with its Arm processor and Linux operating system (and yes, I know other OSes are available), the Raspberry Pi doesn't have the same accessibility as the old 8-bit machines.

But 8-bit microcontrollers do. Indeed, my experiences with them vividly brought back my time learning about computers in the early 1980s. And my love of these chips is not without a nostalgic dimension.

Native state

When you're toiling away with the AVR in its native state, without the benefit of the Arduino IDE's abstractions, you find you have to do a lot more work just to enable features you take for granted on the Arduino. And yet, being forced to confront registers, interrupts, timers and other low-level stuff also instils a sense of power and possibility. And in burrowing so deeply into the inner workings of the microprocessor, you learn there's so much more you

can do — and that you can often do it a lot faster than you can in the conventional Arduino world, let alone the Linux environment.

And so, in the coming weeks and months, I'll be posting a number of articles on Machina Speculatrix about programming AVR chips, initially in C but we'll also be dipping our toes into AVR assembly. And we'll be learning about technologies such as SPI, I2C, serial communications and more.

You can find all the AVR-related articles here.

I've created a GitHub repo for supporting files to accompany this AVR series of articles. You can find it here: https://github.com/mspeculatrix/AVR_8bit_Basics/

Steve Mansfield-Devine is a freelance writer and photographer. You can find photography portfolio at Zolachrome, buy his books and e-books, or follow him on Bluesky or Mastodon.

You can also buy Steve a coffee. He'd like that.

[Electronics](#)[Technology](#)[Microcontrollers](#)[Programming](#)[Avr](#)



Published in Machina Speculatrix

Following

25 Followers · Last published 2 days ago

Electronics, robotics, home automation, hacking and more. The lab notebook of an amateur meddler who likes playing with things until they work—or blow up.



Written by Mansfield-Devine

Following

42 Followers · 2 Following

Freelance writer & photographer, tech journalist and electronics botherer.

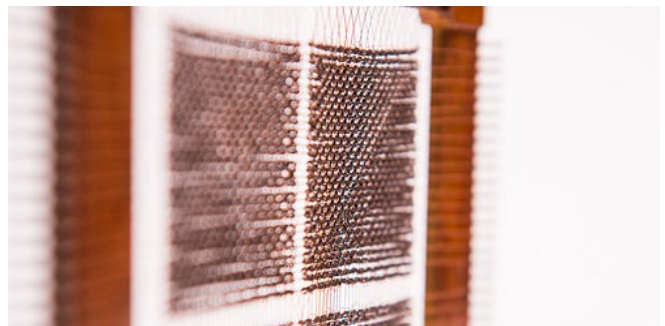
No responses yet



Alex Mylnikov

What are your thoughts?

More from Mansfield-Devine and Machina Speculatrix





In Machina Speculatrix by Mansfield-Devine

About me

You might be wondering, who the hell is this?
Let me explain.



Mar 3



24



2



In Machina Speculatrix by Mansfield-Devine

Retro to the core

Today, computer memory comes buried
inside anonymous chips. But there was a tim...



Mar 4



414



10



In Machina Speculatrix by Mansfield-Devine

AVR basics: reading analogue input

Turning a microcontroller's I/O ping on and off
is all very clever, but what about when things...



5d ago



35



1



In Machina Speculatrix by Mansfield-Devine

Getting to grips with the parallel interface

Putting an old printer back into use meant
talking the language of its now (mostly)...



Feb 28



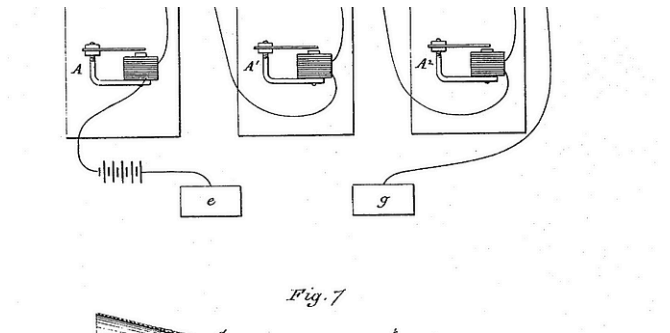
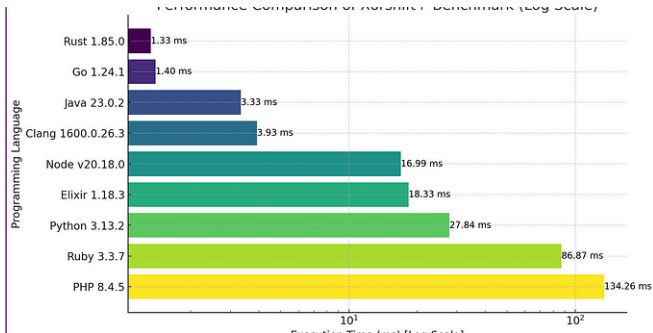
11



See all from Mansfield-Devine

See all from Machina Speculatrix

Recommended from Medium



 In Effective Programmer by Naveed Khan

So, How Fast is Rust Anyway

Well, its pretty darn fast

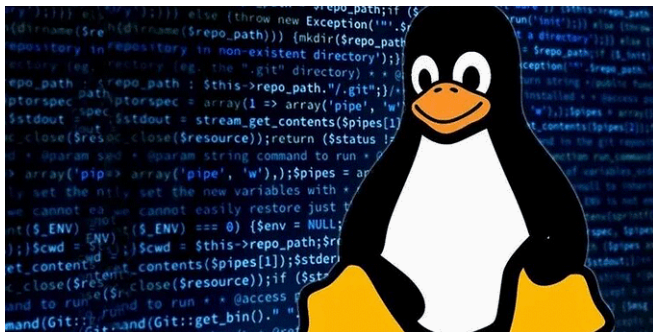
🌟 5d ago 🖱️ 292 💬 21 📌⁺ ⋮

 In Radio Hackers by Simon Quellen Field 

Python Radio 35: You've got mail...

Get a Wi-Fi notification of snail-mail

🌟 3d ago 🖱️ 140 💬 5 📌⁺ ⋮

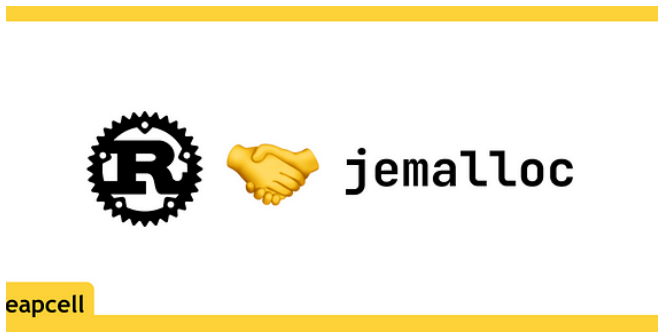


 In InfoSec Write-ups by Frost

Powerful Linux Tricks That Will Change Your Life

If you've ever worked in a Linux environment, you know how powerful and versatile it can...

🌟 Mar 12 🖱️ 234 💬 3 📌⁺ ⋮



 Leapcell

Optimizing Rust Performance with jemalloc

Learn how jemalloc optimizes memory management

4d ago 🖱️ 70 📌⁺ ⋮




 Martin Ayvazyan

Advanced C++ Optimization Techniques for High-Performanc...

Performance is a critical feature. In domains like game development, high-performance...

Mar 9  64  1



 Giorgio Provinciali

Ukrainians Were Never Encircled In Kursk

Live report

 3d ago  1.7K  7



See more recommendations