

Blog

624 Besucher haben diese Seite aufgerufen

Offtopic: Ich nutze jetzt NixOS und weiß noch nicht, ob ich es lieben oder hassen soll

Vorweg: Ich bin Laie und berichte hier von persönlichen Erfahrungen und meinem eigenen Wissen, das ich mir über die Zeit angeeignet habe. Faktische Korrektheit nicht garantiert!

Als ich im Mai auf einem Event eines befreundeten Hackspaces war, wurde mir erstmals von NixOS erzählt – einer Linux-Distribution, die anders funktioniert, als ich es bisher jemals gesehen habe. Denn anstelle der üblichen Paketmanager, wie beispielsweise apt bei Debian oder pacman bei Arch Linux, werden alle Apps, Programme und sogar Einstellungen in einer Konfigurationsdatei, der **configuration.nix**, deklariert. Dadurch bezeichnet man NixOS auch gerne als „deklaratives Betriebssystem“.

Das heißt: Wenn ich 5 Apps installieren will, schreibe ich sie einfach in die configuration.nix. Möchte ich das Tastaturlayout ändern, geht das ebenfalls dort.

Ein Beispiel, wie eine configuration.nix aufgebaut sein könnte:

```
{ config, pkgs, ... }:

{
  imports = [
    ./hardware-configuration.nix
    ./my-dev-tools.nix
    ./my-desktop-env.nix
    ./etc.nix
  ];

  networking.hostName = "mymachine";
  time.timeZone = "Europe/Utrecht";

  services.xserver.layout = "us";
  services.xserver.xkbVariant = "altgr-intl";

  users.users.myuser = {
    extraGroups = [ "wheel" "networkmanager" ];
    isNormalUser = true;
  };

  environment.systemPackages = with pkgs; [
    ddate
    testdisk
  ];
}
```

```
services.openssh.enable = true;  
}
```

Will man das noch etwas weiter treiben, wie es die meisten NixOS-User tun, kann man seine Konfiguration strukturieren und in mehrere Dateien und Ordner aufteilen. So lassen sich beispielsweise die Bluetooth-Einstellungen in einer separaten Datei konfigurieren.

Aber wofür das Ganze?

Stellen wir uns vor, man hat zuhause einen Laptop, einen PC und einen Homeserver. Tatsächlich lassen sich in einer Config auch mehrere Systeme definieren. Diese Config liegt dann beispielsweise in einem Git-Repository und ist für alle Geräte abrufbar.

Geht mir nun mein Laptop kaputt, lade ich mir meine Config einfach herunter, wende sie auf meinem neuen Laptop an und habe alle Einstellungen und Programme wieder installiert. Andersherum kann ich auch an meinem Laptop die Config für meinen Homeserver bearbeiten. Wenn ich meinen Homeserver so eingestellt habe, dass er sich regelmäßig die neueste Config aus dem Git-Repo zieht, lassen sich sämtliche Einstellungen und Programme aus der Ferne verwalten.

Und hier wieder dasselbe: Wenn mir der Server stirbt, setze ich einfach einen neuen auf, lade meine Config – und schon habe ich alles wieder.

Nun, ich bin noch ziemlich am Anfang und habe nicht wirklich viel Erfahrung im Umgang mit NixOS. Ehrlich gesagt habe ich generell wenig Erfahrung mit Software. Glücklicherweise haben mir Freunde bei der Einrichtung geholfen, und langsam fange ich an, Dinge selbst zu verstehen.

Oft stehe ich aber vor seltsamen Fehlermeldungen, wenn ich die Config bearbeitet habe. Manchmal funktionieren Dinge einfach nicht, wie sie sollen, und manchmal braucht das System einfach eine Weile. Aber das ist alles ein sogenanntes „Skill Issue“, wie es die Jugend zu sagen pflegt.

Ich lerne stetig dazu und kann mir sogar vorstellen, irgendwann auch mit meinem Hauptrechner von Linux Mint auf NixOS umzusteigen.

Übrigens, wen es interessiert: Lange Zeit habe ich KDE Plasma als Desktop verwendet. Plasma ist ein sehr umfangreiches, im Aussehen stark konfigurierbares und modernes Desktop Environment, welches aber leider auch zu den leistungshungrigeren gehört. Insbesondere auf meinem relativ alten Laptop fing es recht schnell an zu ruckeln und beeinträchtigte die Performance spürbar.

Inzwischen bin ich bei **niri** hängen geblieben, einem Scrolling-Window-Manager. Den Unterschied zwischen einem Window Manager und einem Desktop Environment sauber zu erklären, liegt außerhalb meiner fachlichen Kompetenzen – aber im Grunde ist ein Window Manager deutlich minimalistischer aufgebaut.

In Verbindung mit der **Noctalia Shell** bin ich mit dem Aussehen und der Bedienung meines Laptops sehr zufrieden. Alles geht plötzlich schneller und einfacher mit ein paar Tastenkombinationen.

Ich werde mir mal überlegen, mehr solcher Offtopic-Blogeinträge zu verfassen, falls Interesse besteht. (Oder ich mache dafür einen separaten Blog – aber aktuell habe ich schon zu viele andere Projekte offen.)

Bis demnächst und noch viel Spaß beim Durchstöbern meiner Seiten!

— Niklas Poteczin, 12/04/2026 23:55

12/04/2026 23:51 · Niklas Poteczin · 0 Kommentare · 0 Linkbacks

[linux](#), [nixos](#), [software](#)

Casio CP-1000 - Soll man das jetzt Kamera nennen?



Zwischen einem Haufen anderer Kameras liegt das gute Stück vergraben

Dieses exotische Gerät habe ich freundlicherweise von Boris (<https://www.digicammuseum.de/>) bekommen - vielen Dank!

Ehrlich gesagt weiß ich bis heute nicht, wie ich dieses „Ding“ einordnen soll. Eine Kamera? Ein Drucker? Oder doch ein Scanner? Denn irgendwie ist es alles gleichzeitig und doch nichts davon so richtig. Boris nannte die CP-1000 eine Art Schwarzweiß-Sofortbildkamera - und in gewissem Sinne ist sie genau das.

Die Kamera aus dem Jahr 1993 hat keinen normalen Bildsensor, sondern - wie bei einem Scanner - einen linearen Sensor, der das Bild seitwärts abfährt. Während das Foto aufgenommen wird (was übrigens einige Sekunden dauern kann), wird das Bild parallel direkt auf Thermopapier ausgedruckt. Da die CP-1000 handelsübliche 210 mm breite und 15 m lange Faxrollen benötigt, kann man diese einfach und günstig online bestellen.

Das Einzige, was bei mir nicht funktionierte, war der NiCd-Akku (wen wundert's). Glücklicherweise kann man den einfach entnehmen und entweder durch einen 14,4 V NiMH-Akku ersetzen oder die Kamera ganz ohne Akku betreiben.

Erster Test



Erster Test mit Standardeinstellungen aus der Kamera

Ich habe also den Akku entfernt und die Anschlusskabel mit Krokodilklemmen an mein altes 13,8 V-Netzteil angeschlossen. Der erste Test konnte beginnen. In der Kamera war noch ein Reststück Faxrolle, womit ich dann gleich ein Bild von meinem Zimmer machen konnte.

Die gute Nachricht: Die Kamera druckt ohne Probleme. Langsam, aber sicher spuckt sie ein Bild aus. Die schlechte Nachricht: Die Kamera ist sehr wählerisch, was die Belichtung angeht. Etwas zu wenig Licht und das Bild rauscht extrem. Etwas zu viel Licht – und es kommt ein komplett weißes Blatt heraus. Ohne einen selbst gebauten ND-Filter wird das also nichts mit DIN-A4-Landschaftsaufnahmen. ;)

Tatsächlich hat die Kamera mehrere Modi und Möglichkeiten, das Bild auszugeben – entweder in reinem Schwarzweiß (z. B. für schwarze Schrift auf weißer Tafel) oder in Graustufen.

Leider reichten meine Reststücke der Thermorolle nur für etwa drei Aufnahmen. Deshalb werde ich den Artikel fortführen, sobald ich weitere Rollen und Testaufnahmen gemacht habe. **Update (29.05.2025):** Hier geht es zur [Archivseite](#) mit weiteren Tests.

Akkuproblem

Das Letzte, was es jetzt noch zu beheben gilt, ist das Akkuproblem. Die CP-1000 kommt mit einem 14,4 V (12S) Nickel-Cadmium-Akku aus dem Hause Sanyo mit den Maßen 10,5 cm × 3,5 cm × 4,5 cm. Sollte ich einen Akku gefunden haben, der den Maßen entspricht, werde ich den Beitrag hier auf dem Laufenden halten.

NiMH-Akkus zu bekommen ist grundsätzlich kein Problem, da diese z. B. in Werkzeugen oder im Modellbaubereich – neben Lithium-Akkus – immer noch eingesetzt werden. Da ich aktuell allerdings keine passenden Akkus zur Hand habe, betreibe ich die Kamera momentan nur am Netzteil.

Eine weitere Möglichkeit wäre jedoch, die Kamera via USB-C zu betreiben. Der Power-Delivery-Standard unterstützt schließlich auch 15 V Spannung. Das sind zwar etwas mehr als die angegebenen 14,4 V eines NiMH-Akkus, aber was auf Akkus draufsteht, ist in der Regel die sogenannte Nennspannung. Die Ladeschlussspannung (die Spannung, bei der ein Akku wirklich voll ist) liegt meist etwas höher. Beispiel Autobatterie: Nennspannung 12 V, Spannung im voll aufgeladenen Zustand:

14,4 V. Also ist die leichte Abweichung hier gar kein Problem.

Nur sollte man darauf achten, dass die Spannungsquelle genug Strom liefern kann – die Kamera verlangt während des Scanvorgangs gerne mal über 1,5 A. Einige Spannungsquellen gehen dabei etwas in die Knie. Mein altes 1,5 A-Labornetzteil von Monacor z. B. funktioniert nicht mit der Kamera. Es liefert zwar locker 15 V, allerdings fällt die Spannung bei voller Last ab – genau das, was auch bei leeren Batterien passiert. Deshalb quittiert die CP-1000 beim Versuch, ein Bild aufzunehmen, mit einem blinkenden Batteriesymbol den Dienst.

Jetzt werden aber erst mal Faxrollen bestellt – dann geht's ans Testen. Sobald ich alles Nötige zusammen habe, geht's hier weiter.

— Niklas Poteczin, 23/05/2025 22:11

Update (29.05.2025): Die Faxrollen sind da und es existiert jetzt auch eine [Archivseite](#) zu dem guten Stück.

23/05/2025 22:21 · Niklas Poteczin · 0 Kommentare · 0 Linkbacks

Zwei Symbol LS-7000 Barcode-Scanner der 80er mit echter Laserröhre!



So kamen die Barcode Scanner bei mir an

Ich habe mal wieder Post bekommen – diesmal von einem Bekannten, der mir im Vorhinein erzählt hatte, er hätte ein paar alte Barcode-Scanner übrig. Ich wollte fast schon ablehnen, als er mir sagte, dass diese Scanner angeblich eine echte Helium-Neon-Laserröhre verbaut haben.

Kurz zur Klärung: Eine Helium-Neon-Laserröhre (HeNe-Laser) erzeugt einen kohärenten Lichtstrahl durch die Anregung eines Gasgemischs aus Helium und Neon in einer Entladungsröhre. Dabei wird mithilfe einer Hochspannung ein stabiler, sehr feiner roter Laserstrahl mit einer Wellenlänge von typischerweise 632,8 nm erzeugt. Der Vorteil solcher Laser ist ihre hohe Strahlqualität, Stabilität und Sichtbarkeit – allerdings benötigen sie eine relativ hohe Betriebsspannung und sind empfindlich gegenüber Erschütterungen, weshalb sie eher selten in kompakten, mobilen Geräten wie Barcode-Scannern zu finden sind.

Und so funktioniert der Barcode-Scanner: Wenn man die rote Taste drückt, tritt ein Laserstrahl aus dem oberen Fenster des Scanners aus, der mithilfe eines Spiegelgalvanometers schnell hin und her bewegt wird. Ein Spiegelgalvanometer ist ein elektromechanisches Bauteil, das kleine Spiegel mithilfe von Magnetfeldern sehr präzise bewegen kann – ideal, um Lichtstrahlen kontrolliert zu lenken. Der Laser fährt dann zeilenweise über den Barcode, wobei die schwarzen und weißen Streifen unterschiedliche Lichtmengen reflektieren. Diese Helligkeitsunterschiede werden von einem optischen Sensor, typischerweise einer Fotodiode oder einem Fototransistor im Gerät erfasst und anschließend decodiert.

Wer jetzt allerdings denkt: „USB dran und fertig!“ – der hat sich geirrt. Zumal meine Scanner aus dem Jahr 1987 stammen – und USB fand erst Ende der 1990er Jahre so wirklich als einheitlicher Anschluss für Peripheriegeräte Verbreitung.

In diesem Fall funktioniert der Scanner nicht ohne ein separates Netzteil und eine Empfängerbox, an die der Scanner selbst via DB9-Kabel angeschlossen wird. Der DB25-Port ist dann für die Verbindung mit dem Rechner gedacht. Ob es sich dabei um ein proprietäres Protokoll handelt, gilt es noch herauszufinden.



Schon besser, wenn auch nicht perfekt

Vor der ersten Inbetriebnahme habe ich die Scanner alle gründlich gereinigt – mit mehr oder weniger Erfolg, trotz Schmutzradierer und Kunststoffreiniger. Leider habe ich dadurch von einem der Scanner den leider nur relativ schwach aufgedruckten Symbol Schriftzug und das ebenfalls nur aufgedruckte Typenschild weggeputzt, welche beide nur eine spiegelnde Folie hinterlassen haben. Aber zumindest sieht er jetzt deutlich besser aus. Die Etiketten vom anderen Scanner waren deutlich hochwertiger, dieser wird vermutlich zu einer anderen Revision gehören. Ansonsten sind beide Scanner zumindest äußerlich absolut baugleich.

Sobald ich die Scanner ausführlich getestet und auseinandergenommen habe, schreibe ich den Beitrag weiter. Allerdings bin ich aktuell aufgrund eines bevorstehenden Umzugs etwas ausgelastet.

23/05/2025 19:28 · Niklas Poteczin · [0 Kommentare](#) · [0 Linkbacks](#)

Modifizieren einer IBM Model M als USB-Tastatur



Die Model M auf dem Schreibtisch des Autors

Die IBM Model M ist sicherlich kein unbekannter Kandidat, wenn es um klassische Tastaturen geht. Unter Liebhabern ist sie insbesondere aufgrund ihres legendären Tippgefühls und des unverwechselbaren Klickgeräuschs sehr beliebt und genießt dadurch inzwischen auch einen gewissen Sammlerwert.

Meine Model M habe ich vor einer Weile von einem guten Bekannten bekommen – leider ohne den Keycap für die „4“ auf dem Numpad und mit einem defekten Kabel.

Ich hätte mir natürlich einfach ein neues Kabel kaufen können, allerdings sind diese allein schon relativ selten und schwer zu bekommen. Dann stieß ich auf einen Online-Shop aus Australien, der angeblich eine Platine für die Model M entwickelt hat, die die Originalplatine ersetzt und die Tastatur somit USB-fähig macht.

Somit hätte ich gleich zwei Fliegen mit einer Klappe geschlagen: Einerseits müsste ich mich nicht mehr um ein passendes Kabel kümmern, andererseits bin ich mit USB einfach deutlich flexibler – besonders, wenn ich die Tastatur mal an einem Gerät verwenden möchte, das entweder keinen PS/2-Anschluss oder keinen passenden Adapter hat.

Ich habe also die Platine bestellt und etwa zwei Wochen später landete ein Zettel im Briefkasten, dass ich „nicht angetroffen werden konnte“. (Klassiker.) Mich wunderte nur, dass der Paketbote die Platine nicht mit den anderen Paketen einfach vor der Tür abgelegt hat. Etwas nerviger fand ich dann die Tatsache, dass ich das Paket nun ausgerechnet in dem einen Paketshop im Dorf abholen durfte, der wirklich die bescheidensten Öffnungszeiten hat.

Drei weitere Tage später und ich bekomme von der Frau am Schalter gesagt, dass ich noch 13,75 € Zoll nachzahlen muss. Dachte ich mir fast schon.

Nachdem ich also nun knapp 55 € für die Platine blechen durfte, lag sie endlich auf meinem Tisch.



Der nachgerüstete USB-Anschluss

Der Umbauprozess ist eigentlich super einfach und weitgehend selbsterklärend. Tastatur auf – dazu die vier Schrauben auf der Unterseite lösen und die Oberseite vorsichtig abheben. Als Nächstes: die Flachbandkabel vorsichtig aus der alten Platine ziehen. Vorher muss allerdings noch das Erdungskabel abgeschraubt werden. Dann nur noch das neue Board einsetzen, die Flachbandkabel wieder an die gleichen Stellen anschließen – und fertig!

Die 3D-gedruckte Blende ist etwas fummelig einzusetzen, ohne dass sie sofort wieder herausfällt, aber sobald der obere Teil der Tastatur wieder auf dem unteren Gehäuse sitzt, hält alles bombenfest.

Jetzt nur noch die Schrauben wieder eindrehen – und fertig!

Es bleibt noch zu erwähnen, dass ein USB-B-Kabel, das für die Verwendung der Tastatur benötigt wird, nicht mitgeliefert wird. Allerdings findet sich so ein „Druckerkabel“ mit hoher Wahrscheinlichkeit in jeder gut sortierten Kabelkiste.



Die Model M im Vergleich zur Corsair K95 Platinum RGB

Einmal eingesteckt, funktioniert die Tastatur sowohl unter Linux als auch unter Windows einwandfrei – ganz ohne zusätzliche Treiber oder Konfiguration.

Den gesamten Beitrag habe ich übrigens mit der modifizierten Model M getippt. Ich muss sagen: Der etwas härtere Anschlag ist anfangs wirklich gewöhnungsbedürftig – insbesondere, wenn man zuvor eine Corsair K95 Platinum RGB mit optischen Switches benutzt hat. Aber mit der Zeit gewöhnt man sich daran, und inzwischen tippe ich sehr gerne mit dieser Tastatur. Das Einzige, was manchmal wirklich nervt, ist die fehlende „Windows“-Taste. Aber für die meisten Schreibanwendungen ist diese sowieso nicht so wichtig.

Jetzt werde ich die Model M erstmal für einige Zeit im Alltag ausprobieren. Sollte dieser Beitrag keine Fortsetzung oder Ergänzung bekommen, dann können Sie davon ausgehen, dass die Tastatur bis dahin keinerlei Probleme gemacht hat.

— Niklas Poteczin, 23/05/2025 18:48

23/05/2025 18:49 · Niklas Poteczin · 0 Kommentare · 0 Linkbacks

Und mal wieder Probleme mit der Website.

Wer in der letzten Zeit im Archiv unterwegs war, der fand und findet teilweise immer noch kaputte Seiten, fehlerhafte Darstellungen etc. Das liegt daran, dass das verwendete CMS Dokuwiki ein wichtiges Update bekommen hat, welches zwar für einige unserer Plugins sehr wichtig ist, allerdings nicht mehr mit unserem bis dato verwendeten Template (also der grafischen Oberfläche), welches auf Bootstrap 3 basiert kompatibel ist. Zeitgleich gab es dann noch für mehrere Tage immer wieder Schwierigkeiten bei unserem Host, was das Arbeiten an der Website nochmal erschwert hat. Inzwischen läuft die Seite wieder, diesmal unter einem auf Bootstrap 5 aufgebautem Template. Leider ist die korrekte Konfiguration und Einrichtung des Templates sowie das perfekte Zusammenspiel mit den anderen grafischen Elementen ein Haufen Arbeit. Da ich im August jetzt auch noch ausziehe, bin ich aktuell leider so beschäftigt, dass das Nebenprojekt „Broadcast Archiv“ aktuell erstmal hinten angestellt werden muss. Außerdem sind wir hier weder IT Spezialisten noch verstehen wir sonderlich viel von PHP, Websites und CMS im Allgemeinen. Trotzdem versuchen wir Ihnen das Surf-Erlebnis so angenehm, wie möglich zu gestalten.

Was es auch noch zu überarbeiten gilt ist die mobile Ansicht für diese Seiten. Diese ist nämlich aktuell quasi unbrauchbar. Bis dahin kann es allerdings noch etwas dauern.

— Niklas Poteczin, 23/05/2025 17:53

23/05/2025 18:03 · Niklas Poteczin · 0 Kommentare · 0 Linkbacks

[Ältere Einträge >>](#)

From:
<https://broadcastarchiv.de/> - **broadcastarchiv.de**

Permanent link:
<https://broadcastarchiv.de/blog>

Last update: **24/05/2025 03:40**

