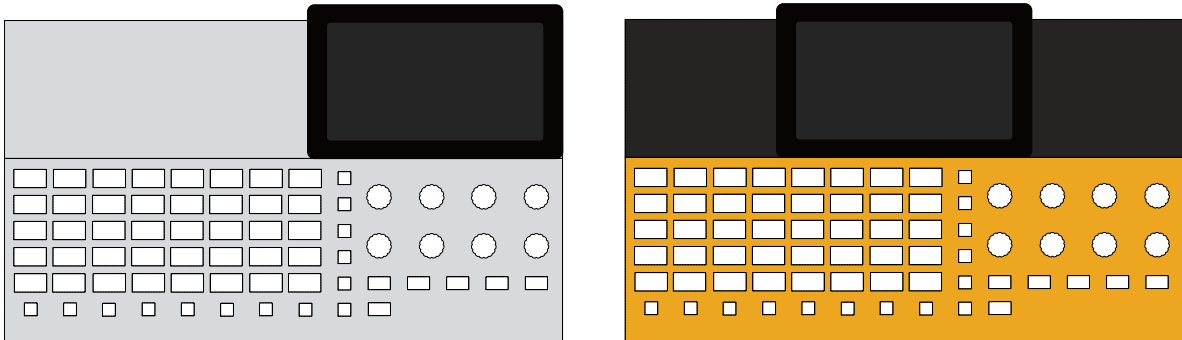




WUESTE MLP

“ ... is a call to work creatively TOGETHER and to take a stand—without startup posturing, without funding programs, and without military budgets.”

Dieser Text ist eine überarbeitete Form eines Transkripts eines Vortrags, der in Potsdam am 26.06.2026 gehalten wurde. Ziel des Vortrags war es, einen Überblick über die Entstehungsgeschichte von Wueste MLP zu geben, dessen technische Grundstruktur zu erläutern und noch offene Fragestellungen anzusprechen.

1. Wueste.info (Wer sind wir?)

Dieses Instrument ist nicht in einem luftleeren Raum entstanden. Es hat eine lange Vorgeschichte, die auf verschiedene Linien und Geschichten zurückgeht. Entscheidend im Hintergrund ist, dass wir zusammen eine Webseite betreiben und seit etwa zehn Jahren gemeinsam Musik machen. Julian hat mich 2014 in Uganda besucht, wo ich ein Tonstudio hatte. Es gab dort verschiedene Konzerte, er war dabei, und danach lud er mich zu Sessions ein. Das hat sich verstetigt und sich mit der Zeit immer stärker miteinander verwoben – bis er schließlich die Webseite aufmachte.

Vor fünf, sechs Jahren haben wir eine Schallplatte gemacht und angefangen, mit

Synthesizern live improvisierte Musik aufzunehmen. Julian hatte davor viele andere Projekte verfolgt – unter anderem in einer Sandgrube ein kleines DIY-Festival für live improvisierte Musik organisiert. Daher kommt die Domain wueste.info: Es war eine Sandgrube, die eigentlich eine Wüste sein sollte. Diese Domain haben wir dann weitergenutzt. Anfangs war es ein kleiner Blog, dann haben wir gemeinsam angefangen, daran zu arbeiten. Als meine Arbeitsstelle gehackt wurde und wir einen Monat lang arbeitsunfähig waren, nutzte ich diese Zeit, um die Webseite neu zu bauen.

Die Diskussionen und Gespräche, die wir zu Musik und anderen Themen geführt haben, haben sich ständig weiterentwickelt. Zunächst ging es uns darum, eine andere Form von Musik zu machen. Nur Techno aufzulegen, nur Krautrock oder nur Stonerrock – das erschien uns zu eindimensional. Wir wollten aus diesen geraden Linien, aus diesen vorgefertigten Schubladen ausbrechen. Je tiefer wir uns damit beschäftigt haben, desto klarer wurde: Wir wollen diesen Anspruch nicht allein auf die Musik beschränken, sondern auch auf die Art übertragen, wie wir Musik zugänglich machen. Wir hatten keine Lust mehr, unsere Musik auf Bandcamp oder Spotify hochzuladen – nicht nur weil das Plattformen sind, die kommerzielle Interessen verfolgen, sondern weil sie die Art, wie Menschen Musik hören und wie Musik gemacht wird, grundlegend verändern. So sind wir Schritt für Schritt dazu übergegangen, eine eigene Infrastruktur aufzubauen. Dabei war uns klar, dass das ein Mammutprojekt ist – Spotify lässt sich nicht ersetzen, man kann nicht von heute auf morgen eigene Musikinstrumente bauen, man kann nicht einfach so seine eigenen Schallplatten pressen. Aber wir wollten nicht dabei bleiben, den Status quo zu beklagen, wie es letztlich alle tun, sondern mit einem Experiment, einem Prototyp beginnen und sehen, wie weit wir kommen.

Im Zuge dieser Diskussionen sind uns momentan absolut gängige Praktiken des Marketings und der endlosen Selbstvermarktung immer mehr aufgestoßen. Wir stellten fest, dass selbst Technopartys, bei denen früher selbstverständlich war, dass man nicht fotografiert oder unbedingt gesehen sein wollte mittlerweile nach den Metriken sozialer Medien und Plattformen bewertet werden. Sichtbarkeit, Likes, permanente Präsenz, das Aufspringen auf jeden Trend: Das erscheint uns als eine grundlegende Entfremdung. Wir haben selbst versucht, einen Instagram-Account zu führen, haben es gelassen. Spotify verweigern wir uns. Wir haben ein Buch darüber gelesen und mit verschiedensten Leuten Diskussionen über deren Spotify-Nutzung als Hörende und Schaffende gehabt. Wir hatten auch einen Bandcamp-Account, eine Plattform, die gemeinhin als die bessere Alternative gilt, die aber inzwischen aufgekauft wurde.

Was wir beobachten, lässt sich als symptomatisch für die gesamte Musikindustrie beschreiben: massiver Kapitalzufluss auf der einen Seite, auf der anderen Seite eine Welle von Insolvenzen. Firmen, die ich seit meiner Kindheit kenne und die wegweisende Synthesizer und Instrumente gebaut haben, sind aus diesem Geschäft verschwunden – weil sie Teams entlassen haben, um anderswo rentabler zu werden. Dieser Ausverkauf zieht sich durch die ganze Branche.

Auf der Webseite gibt es neben der Selbstdarstellung und Kritik eine Kategorie „Work in Progress“. Dort haben wir zum Beispiel eine Tour angekündigt, die wir im August machen werden – die Details sind noch offen, aber wir werden mit einem Lkw aufbrechen und an verschiedenen Orten kleine Workshops anbieten, bei denen gemeinsam Sessions gespielt werden. Wir sind derzeit dabei, Anfragen zu verschicken.

Letztes Jahr haben wir eine Interviewserie mit befreundeten Künstlerinnen und

Künstlern durchgeführt und sie gefragt, wie sie Plattformen nutzen, wie sie Musik hören und hören wollen, wie sich das Verhältnis von Lohnarbeit und kreativem Schaffen für sie darstellt und welche Rolle Geld dabei spielt. Die Webseite versammelt außerdem unsere Performances, eine Kategorie „Releases“ mit fertigen Interviews, Alben und Aufnahmen – darunter Sessions aus Bulgarien und ältere Aufnahmen vom Olgashof. Zuletzt haben wir noch eine weitere Kategorie hinzugefügt: „Tools“.



2. Was ist MLP?

Dieses Tool, an dem ich seit anderthalb Jahren arbeite, steht heute im Mittelpunkt. MLP steht für Modular Live Performer. Modular, weil die interne Struktur modular aufgebaut ist. Live, weil es ein Instrument ist – kein Kompositionswerkzeug. Es ist nicht dafür gedacht, zu Hause Musikstücke von Anfang bis Ende durchzukomponieren; dafür gibt es bereits viele Tools, dafür sind Laptops unschlagbar effizient. Das MLP ist vielmehr dafür da, wie ein Kontrabass, ein Schlagzeug oder eine Trompete in eine Session zu gehen und live Musik zu machen. Deshalb gibt es keinen Song-Modus, keine Möglichkeit, ganze Stücke durchzuplanen. Bestimmte Entscheidungen werden bewusst offen gelassen, damit man sich während des Spielens damit auseinandersetzen muss. Manche Funktionen sind schnell zugänglich, andere verlangen konstante Aufmerksamkeit – das Gerät soll nicht einen Knopfdruck und eine fertige Show liefern, sondern dauerhaft zur Interaktion herausfordern. Intern ist das System modular und flexibel, aber diese Flexibilität ist ausschließlich für die Vorbereitungsphase gedacht. Während der Performance lässt sie sich nicht nutzen.

Das Gerät enthält 64 Module, verteilt auf acht Tracks mit je acht Modulen. Diese Module sind frei austauschbar: Sequencer, Effekte, Looper – musikalische Fachbegriffe für sehr unterschiedliche Funktionen. Der Grundgedanke ist, das zu überwinden, was den kommerziellen Markt dominiert: Entweder kauft man einen Drumcomputer, der Schlagzeugklänge erzeugt, oder einen Synthesizer für Bässe und Akkorde, oder einen Looper für die eigene Stimme – immer separate Geräte für separate Aufgaben. Das einzige Gerät, das diese Funktionen wirklich vereint, ist der Laptop, den die meisten Live-Elektroniker auf der Bühne haben, manchmal hinter einem Controller versteckt. Ein eigenständiges Gerät, das das leistet, ohne ein Laptop zu sein, existiert nach meiner Kenntnis bislang nicht.

Die Idee ist: Wer live Musik machen möchte, kann sich im Vorfeld – mit Unterstützung durch uns und ein ausführliches Handbuch – ein Gerät so konfigurieren, dass es der geplanten Performance entspricht. Das kann eine Kombination aus Loops, Sequenzen und live gespieltem Piano sein. Diese Konfiguration wird vorbereitet und auf das Gerät geladen; während der Performance selbst können die Module nicht mehr umgebaut werden. Das hat einen persönlichen Grund: Ich selbst neige dazu, in letzter Minute alles noch einmal umzustellen – Julian weiß das und leidet gelegentlich darunter, weil er eher konsequent plant. Ich komme zu jeder Session mit neuen Ideen. Wenn dann live etwas nicht funktioniert, bin ich manchmal versucht, auf das Vertraute zurückzugreifen, meistens die Klaviatur. Ein festgelegtes System zwingt einen, diesen Moment auszuhalten und sich durch die Live-Situation hindurchzuarbeiten. Daraus entsteht oft etwas Unerwartetes: eine Facette, die in der Vorbereitung nicht absehbar war, die sich aber genau in dieser Konstellation – mit dem anderen, unter dieser Einschränkung – öffnet.

Modular-Synthesizer erfreuen sich gerade großer Beliebtheit, und das Prinzip ist dem, was wir bauen, nicht unähnlich – nur wird es dort mit physischen Metallmodulen umgesetzt, die über Patchkabel verbunden werden. Viele Menschen beschreiben die Arbeit mit einem Modular-Synthesizer als Befreiung, weil sie aus einer Softwarewelt kommen, in der sich Effekte, Instrumente und hunderte Spuren beliebig stapeln lassen – bis der kreative Fluss irgendwann versiegt, weil die Reduktion fehlt, die einen zwingt, sich mit dem Einzelnen zu befassen: mit einer Stimme, einem Filter, einer Melodielinie. Modular-Synthesizer sind grundsätzlich monophon, ohne Akkorde – jeder Ton muss einzeln erzeugt werden. Das kann sehr komplex werden, aber selbst ein überschaubares System kann bereits etwas Befreiendes haben. Die Idee war, dieses Prinzip zu übertragen und gleichzeitig auf digitaler Ebene die Möglichkeit zu

erhalten, die Konfiguration in der Vorbereitungsphase zu verändern und Ideen aus Projekten anderer einzubinden.

Hinzu kommt ein Aspekt, der für mich zentral ist: Ich habe mich intensiv damit beschäftigt, wie elektronische Musik historisch entstanden ist und welche Formen ihrer Live-Performance sich herausgebildet haben – Looper, Sequenzer und anderes mehr. Diese Kategorien haben sich über die letzten hundert Jahre entwickelt, aber ich bin überzeugt, dass wir uns in dieser neuen Musik noch in einem sehr frühen Stadium befinden. Die ersten Tonaufnahmen entstanden Mitte des 19. Jahrhunderts auf Wachszylindern – davor gab es keine aufgezeichneten Klänge, nur Notation, und Musik musste immer live gespielt werden. Das ist ein kulturhistorischer Bruch ohne Beispiel. Mit der Aufzeichnung wurden neue Kategorien möglich und notwendig; neue Klangwerkzeuge entstanden neben den alten Streich- und Blasinstrumenten. Mein Anspruch war, diese Kategorien nicht einfach fortzuschreiben – nicht einen Looper, einen Sequenzer oder einen Synthesizer zu bauen –, sondern sie in einem Gerät zusammenzuführen, das verschiedene Spielweisen gleichzeitig ermöglicht. Ich glaube, dass das zu interessanterer Musik führt.

Das MLP ist vollständig quelloffen, alles ist online verfügbar – auf Codeberg, nicht auf GitHub.

3. Warum gibt es MLP?

Wie kam es dazu, dass ich mit dem Bau begann? Im Januar des vorletzten Jahres besuchte mich Julian in Siegen. Ich war gerade arbeitslos geworden, weil ein Projektjob ausgelaufen war, und hatte Zeit. Wir fuhren nach Köln und trafen uns mit einem Journalisten, den Julian kannte, in einer Szenekneipe. Das ist meine, zugegebenermaßen etwas einseitige Erinnerung an dieses Treffen – Julian und der

Journalist selbst würden es sicher anders beschreiben. Wenn wir erzählten, was uns an den Plattformen störte und welche Fragen uns umtrieben, antwortete er immer wieder, dass es auf diese Fragen doch bereits Antworten gebe, dass alles schon gestellt, schon gedacht worden sei. Dabei klang gelegentlich eine kölsche Arroganz durch: In Siegen gebe es das eben nicht, aber in Köln, dem Mekka der elektronischen Musik, habe man das alles – Festivals, Szene, Infrastruktur. Gleichzeitig erzählte er, dass er als Kulturjournalist ungefähr so viel verdiene wie ein Postbote. Ich fragte ihn, wie er sich das Wohnen in Köln leisten könne. Die Antwort: Er sei früh genug in seine Wohnung gezogen, und sein Vermieter sei SPD-Mitglied. Kurz: Er hatte Glück gehabt, zur richtigen Zeit am richtigen Ort zu sein. Heute nach Köln zu ziehen, ist kaum weniger schwierig als nach Berlin – mit Kindern und einem begrenzten Budget nahezu unmöglich. Er setzte diese kulturelle Infrastruktur als selbstverständlich voraus, ohne zu hinterfragen, auf welchen materiellen Bedingungen sie beruht.

Er schwärmte auch von einem Avantgarde-Musikfestival in Monheim, einer kleinen Stadt nahe Köln. Der dortige Bürgermeister hatte die Gewerbesteuer stark gesenkt, woraufhin Unternehmen aus den Nachbarkommunen nach Monheim zogen und die Steuereinnahmen explodierten. Da Kommunen in Deutschland wesentlich über Gewerbesteuern finanziert werden, verfügte Monheim plötzlich über erhebliche Mittel: kostenlosen Musikunterricht für Kinder und Jugendliche, ein großes Musikfestival, umfangreiche Kunstinstallationen. Das war für den Journalisten ein Beispiel dafür, dass man eben dorthin gehen müsse, wo die Bedingungen stimmen. Einige Monate später, auf einer Autofahrt, hörte ich im Deutschlandfunk einen Bericht: Monheim ist genau das passiert, was fast allen NRW-Kommunen widerfahren ist. Die Gewerbeeinnahmen sind eingebrochen, Monheim ist hochverschuldet. Und damit ändert sich auch die Kulturpolitik der Stadt.

All das sind zentrale Ankerpunkte für mich: Ich kann nicht darauf setzen, lange genug in einer gut versorgten Stadt zu leben und darauf zu hoffen, dass Unternehmen genug Steuern zahlen, damit ich mein Kulturprojekt daran andocken kann. Wenn mir etwas wirklich wichtig ist – und das ist es mir –, dann will ich Instrumente bauen, die sich nicht von solchen Abhängigkeiten nähren. Es gibt Klaviere und Orgeln, weil Menschen über Generationen in diese Technologien investiert haben. Wenn man kulturelle Praxis dauerhaft daran knüpft, dass das richtige Unternehmen in die richtige Stadt zieht oder dass man früh genug in die richtige Wohnung eingezogen ist, sind das keine tragfähigen Grundlagen. Ich will, dass eine Sache durch die Begeisterung von Menschen getragen wird, nicht als Nebenprodukt von Wirtschaftspolitik.

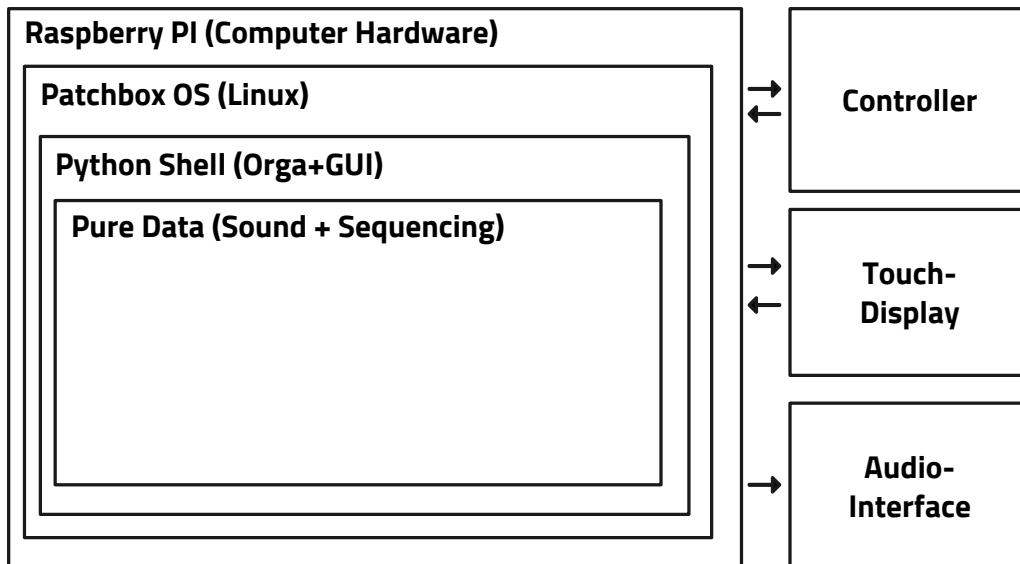
Daraufhin habe ich angefangen, das MLP zu bauen. Ich habe mich – auch auf Drängen von Julian – für eine vollständig quelloffene Lösung entschieden. Anfangs hatte ich das anders geplant, aber die Entscheidung war richtig. Ich sollte dazu sagen: Ich habe Programmieren nie gelernt. Ich hatte zwischenzeitlich Berührungspunkte mit Linux, bin aber kein Linux-Enthusiast und komme aus einem ganz anderen Bereich – ich habe Medienwissenschaft studiert und mache seit meiner Kindheit Musik. In den letzten anderthalb Jahren habe ich mich vollständig in dieses Projekt eingearbeitet, in den vergangenen acht, neun Monaten zunehmend mit Unterstützung von Claude, einer KI, die sich gut dazu eignet, Skripte zu schreiben und technische Lösungen zu entwickeln. Das betrifft allerdings ausschließlich die technische Außenhülle des Systems; die Musik selbst entsteht auf einer anderen Ebene.

3. Technisches

Technisch besteht das Gerät im Kern aus einem Raspberry Pi – einem sehr kleinen Linux-Computer, der ein Betriebssystem von einer SD-Karte lädt. Darauf aufbauend kommt Python, eine Skriptsprache, die nicht kompiliert werden muss, zum Einsatz. Das Herzstück ist Pure Data, aus meiner Sicht die bemerkenswerteste Open-Source-Software im Bereich der Klangerzeugung. Sie existiert seit den 1990er-Jahren, entstanden am IRCAM (Institut de recherche et coordination acoustique/musique) in Paris, einem der wichtigsten Zentren für elektronische Musik seit mehreren Jahrzehnten. Der Grundgedanke war, Komponistinnen und Komponisten sowie Klangkünstlern ein Werkzeug zu geben, mit dem sie selbst Synthesizer bauen können – nicht durch das Schreiben von Code, sondern durch das visuelle Verbinden von Objekten mit Kabeln. Um Pure Data existiert seit Mitte der 1990er-Jahre ein gepflegtes Forum mit umfangreichem Archiv, in das Hochschulen und Privatpersonen gleichermaßen ihre Projekte einstellen – eine bedeutende Ressource, weil alles offen und zugänglich ist.

Als Controller dient ein umgebauter MIDI-Controller: Ich habe das zugehörige Keyboard, dessen Tasten mir zu klein und unspielbar waren, kurzerhand abgesägt. Den Controller auf diese Weise herzurichten war die pragmatische Lösung, weil ich noch nicht in der Lage bin, einen eigenen Controller zu bauen; in Zukunft soll das anders werden. Dazu kommen ein Touchscreen zur Navigation – Projekte laden, Einstellungen öffnen, Updates einspielen – sowie ein Audio-Interface mit acht Ausgängen, was für ein Gerät dieser Größe beachtlich ist. Das Gehäuse besteht aus einfachem Sperrholz, zusammengesägt und verschraubt, nicht geklebt. Ergänzt wird das durch Farbe, einige Abstandshalter, einen kleinen Lüfter, der selten anspringt, weil das System bislang wenig Rechenleistung beansprucht, und ein Netzteil. Die

Materialkosten liegen bei etwa 350 bis 400 Euro, zuletzt gestiegen, weil RAM-Preise stark angezogen haben – Raspberry Pis kosten inzwischen etwa das Doppelte gegenüber vor einem Jahr. Eines der Geräte läuft mit 4 GB RAM, das andere mit 2 GB; 2 GB erweisen sich als vollkommen ausreichend.



Die Architektur des Systems lässt sich in Ebenen beschreiben: Controller, Touchscreen und Audio-Interface bilden die Schnittstelle zur Nutzerin und zum Nutzer; der Raspberry Pi ist der Rechenkern, prinzipiell durch jeden anderen Computer ersetzbar – das System läuft auf alten wie auf neuen Laptops, da Debian nahezu überall lauffähig ist. Auf der Hardware läuft Patchbox OS, ein auf Audio spezialisiertes Linux mit sehr niedriger Latenz: kein Echtzeitsystem, aber nah daran. Darauf liegt eine Python-Shell, die die gesamte Verwaltung übernimmt: grafische Oberfläche, das Laden, Duplizieren und Löschen von Projekten, MIDI-Verwaltung, Einstellungen – und den Start von Pure Data, das den Klang erzeugt.

Der wesentliche Vorteil dieser Schichtung besteht darin, dass jede Ebene unabhängig ausgetauscht werden kann. Ich entwickle alles auf meinem MacBook; wenn etwas

funktioniert, lade ich es über Codeberg (einer Open-Source-Alternative zu Github) hoch, und das Gerät zieht sich automatisch die aktuelle Version. Ich muss nichts manuell übertragen; das System ist immer auf dem aktuellen Stand. Ein weiterer Vorteil: Sollte Pure Data abstürzen, läuft die Python-Shell auf einem anderen Prozessorkern weiter, unabhängig davon. Ein Neustart des Projekts dauert maximal zehn Sekunden – für Live-Musik eine akzeptable Unterbrechung, selbst wenn man weiß, wie unangenehm es ist, wenn ein System auf der Bühne hängt.

Pure Data hat eine visuelle Logik, die für jemanden, der aus der Programmierung kommt, vielleicht befremdlich wirkt, für jemanden aus dem gestalterischen oder visuellen Bereich aber intuitiv ist. Es senkt die Einstiegshürde erheblich und macht es Menschen ohne Programmierkenntnisse möglich, eigene Instrumente und Projekte zu entwickeln. Ich habe mich aus zwei Gründen dafür entschieden: Es läuft auf nahezu jeder Hardware, und es trägt der Hoffnung Rechnung, dass ein Projekt wie dieses sich irgendwann verselbstständigt und andere Menschen eigene Module beisteuern. Pure Data ist vollständig in C geschrieben und lässt sich deshalb auf verschiedensten Plattformen kompilieren.

Die Verbindung zwischen Hardware und Software erfolgt über klassische Linux-Treiber: Die Controller-Knöpfe sind per USB angeschlossen, Pure Data liest die MIDI-Signale direkt aus, ihre Belegung ist im Projekt hinterlegt. Audio läuft über Jack und Alsa-MIDI.

4. Vision + Herausforderung

Nach anderthalb Jahren Entwicklung laufen zwei Prototypen stabil. Mit beiden haben wir bereits gespielt; Julian hat einen davon bei einer Zirkusvorstellung eingesetzt und dabei live Musik gemacht. Ich hatte immer damit gerechnet, irgendwann auf ein

Problem zu stoßen, das mich aufhält – ich komme schließlich nicht aus der Programmierung. Aber es ging immer weiter. Das hat nicht zuletzt damit zu tun, dass KI-Werkzeuge diese Art von Arbeit erheblich zugänglicher gemacht haben: Wo ich früher bei Linux-Konfigurationen oder Audiotreibern feststeckte, finde ich heute über Claude sehr schnell Lösungen. Das nächste unlösbare Problem kommt bestimmt – bisher ist es aber immer doch gelöst worden.

Wir haben jetzt einen Punkt erreicht, an dem die technischen Hürden der ersten Phase überwunden sind und sich die Fragen verschieben: Ich möchte historisch entwickelte Formen elektronischer Musik einbinden, eigene Hardware entwerfen statt vorhandene zu zerlegen, und ich möchte, dass andere Menschen das Gerät nutzen – es in Workshops selbst bauen oder, wenn sie das nicht möchten, erwerben. Zwischen dieser Vision und dem, was wir haben, liegt eine Reihe offener Fragen: Wie produziert man so etwas in kleiner Stückzahl? Wie organisiert man das? Wie unterstützt man die Nutzerinnen und Nutzer? Wie finanziert man es? Im Moment finanziere ich alles selbst, wenn es geht – aber zehn Geräte auf einmal zu bauen, bei 400 Euro Materialkosten pro Stück, ist ohne weiteres nicht möglich. Hinzu kommt die Frage der Zeit: Ich arbeite vierzig Stunden die Woche, habe vier Kinder und versuche, dieses Projekt in Randzeiten unterzubringen. Genau deshalb sind wir heute hier – um aus der eigenen Musikblase herauszutreten und diese Fragen mit anderen zu diskutieren.

Wie geht man mit einem solchen Projekt auf Open-Source-Basis um, wie entwickelt man es weiter? Die übliche Startup-Logik – Businessplan, Skalierung, Rendite – scheidet aus. Aber das Projekt kann auch nicht dauerhaft allein auf Überzeugung laufen; es muss sich irgendwann selbst tragen. Ist es eine Genossenschaft, ein Produkt, eine Anleitung? Gibt es Förderungen? Wie organisiert man den Support – Forum, direkte Erreichbarkeit? Das sind die Fragen, mit denen wir gerade ringen.