# “Cool” things that can be done with computers and AV (for free!)

## I know computers do most things for us today – what’s new?

In truth – most software applications you might purchase today have significant open-source components inside them. This is actually a very good thing. I’m not “anti-commercial” I’m anti predatory. If someone bungs some open-source code in a package, markets it, and banks the cash they charged, they are behaving in a “caveat emptor” (buyer beware) mode. This is completely legal. They have spent time coding, promoting etc, so they are entitled to the profits. This is entirely fair. Indeed, in a B2B environment, this is the norm – the “contract of sale” is the thing that dictates the “deal”.

Often people, being people, are more familiar with “consumer” protection. This does not apply in the same way it does in a professional transaction. Time and time again people believe they are protected by “manufacturer’s warranty”. Sorry, even if you bought a domestic grade TV screen, and have used it in a “professional” context, consumer protection, manufacturer’s warranty, and support, don’t apply! There is not really such a thing as “buying in good faith” in a professional context, unless there are “terms of sale” that specify the limits, and benefits of the contract WITH THE PERSON (company) THAT SOLD IT TO YOU! This is rarely the manufacturer on the box!

Not being a lawyer, I should balance this with I believe that it is the case that nobody is allowed to lie concretely in any contractual negotiation. Whilst salespeople often rely on verbal presentation, and then follow up with a written “contract”, just remember the following: anything you may believe they said, and then you signed without carefully reading what you concretely agree to, evidences your acceptance of recorded terms, and by implication, anything they say is obsolete. This may not actually be legally, fully the case, but… it’s very difficult to “prove” after you have signed an agreement later!

I don’t intend to be gloomy. Outside the “hard” legal bit, consider that most salespeople you meet can only sell effectively if they believe in the product and that the company they represent will look after YOU the customer. This is largely the non-codified truth. IF you are not inclined to build your own software, then by all means, purchase a commercial solution that does what you need. (I do!). For things that pique my interest and after (substantial) testing, such a solution can deliver something that makes a workflow easier. For a niche application however,  open-source libraries (often the same contained in “proper products”,) can prove to be a powerful tool.

The value proposition of a product is “service and support”. This is what you are in 2025 paying for. We (hopefully) have met “manufacturers” that will move heaven and earth to help you. These are the “good” manufacturers that are worth their weight in gold.

It’s those “maybe” features that are “nice to have” but you can’t quite justify. Some wonderful libraries do amazing things.

## Example: Whisper API

Whisper (OpenAI's speech-to-text) is easy to install in a container or VM on premises. The download itself is only 800KB—though it does install dependencies once running.

But even with everything installed, it's tiny and blisteringly fast. It transcribes my strange accent perfectly to English with less than a second delay. A little depressingly, the entire practical English language is compressed down that small. And, there was me feeling sophisticated!

What would I do with something that can transcribe speech to text flawlessly and quickly? Well, I'm not about to give away my personal application… but I'm sure we all have had that feeling that "if only it wasn't in the cloud". Personally, I think the cloud will be wonderful, when it is finished. Today, I really don't think it is quite ready for prime time. In 2025 we have seen huge outages of Cloudflare; this crippled AI for the best part of a day TWICE in one month! The positive is hopefully a lot of tinfoil hats being abandoned, and hope being restored to humans.

Specifically, is this me showing off the greatest, most "human" Large Language Model ever written? Or is this some middle aged, balding geezer writing "stream of consciousness". Furthermore, would the AI gods try to mimic a geeky eccentric character? I think for a while, we are somewhat safe <<the user has identified…>> 😉.

I am however entirely happy to share how we can install it in about 5 minutes, play with it, and then realise; this is only deployable if we learn some simple coding. Spoiler alert- reading and following my text is about 100x more difficult than a software language (that's why we can use an LLM to do 90% of the typing for us!). Again, I "coded" this website using an AI, its very good. In truth, I can't think of anything more boring than coding websites! I don't let an LLM write my content (it would be wrong, and full of inaccurate nonsense fed to it by suspect training data!). It elevates me (without any justification) to being the architect, rather than the world's worst chippy!

In the pursuit of academic honesty, it's better to give credit (and shift blame) to the authors! So here is the install guide for whisper.  Have fun!

GitHub - innovatorved/whisper.api: This project provides an API with user level access support to transcribe speech to text using a finetuned and processed Whisper ASR model.

## So "What Language" do I choose?

Quite simply – how long is a piece of string (pardon the pun). If its "strings" you want – python is wonderful! Seriously. Python will likely have an excellent library for anything you might want to do. This is because it is used extensively in research. Libraries like numpy and scipy have extremely well-built maths engines, which will happily chew up a Fast Fourier Transform, calculate a biquad for a filter, and deliver and/or apply the output to your data. It is possible to do some very silly things – like a FFT of a 3 hour opera – guess what happens? Pretty much a flat line! Entirely predictable but processed in seconds! This puts a huge load on the CPU, but, well, its impressive if materially useless!

It was once considered "serious" only if using C, or C++. This was, once upon a time, probably due to a fire in the 1990s (Sumitomo chemical plant in 1993).  And you thought the pandemic was the first serious silicon supply chain disruption! RAM cost an absolute fortune, and this lasted for years (well, I was a lot younger then… and it felt like forever!) Going back (and forwards) I remember being warned that the "embedded" memory was to be treated carefully as it was "only" 1MB. To someone whose first computer was a BBC B+ with 32kB RAM this felt like "luxury" (apologies to the Monty Python's 4 Yorkshire men sketch!)

We can all brag about our “hardships” of the past (I bow to the recounting of Ferrite cored RAM which was literally “space-age” - but not in the way you might think!). Now, I can buy 2 Raspberry Pi 5’s for the price of 1MB of RAM in the 1990s (or a house… I know Mr Ferrite core!)

Whilst we can all pet our favourite dog in the race, there is one ineluctable truth, everything ends up in a compiler, and ends up as machine code! Yes even python… kinda… Python is a language that is often misrepresented/misunderstood. Ultimately, Scipy and Numpy are compiled C programs, with handles presented to the python interface, so through judicious “cheating” one gets the “speed” of C with the facility of Python. All’s fair in love, war, and computer programming!

In my perithoracic, or circumloquacious (if you prefer) way we finally reach the point! With the modern firepower of a Raspberry Pi 5 (I’ve heard described as a “toy”) one can consume several MADI streams (of 64 channels at 48kHz sample rate, each!). We can show a 4K video on such a “toy” (you might want to add the fan of course!) but its less than 10% of a proprietay MADI interface in cost! One can happily build and run a small piece of python to ingest the strings from the whisper api, look at the strings, and then from the text, get the Raspberry Pi 5 to do some stuff! For example, known speech could be ingested by a “listener” in python, and do something on a keyword. Maybe send a jsonb object containing some OSC to change an audio state, or start a DAW- the sky’s the limit! And while its wonderful, the Pascal based VHDL for an FPGA can stay untouched for another day!

## OK your dodgy dead language references, how about that “point”!

Fair – I like Jupyter labs or its smaller cousin Jupyter notebook for python. This is (of course) much better on Linux. I’d recommend setting the server up to accept connections from your preferred web browser (its 2025, we don’t need discomfort to do cool stuff!) again, credit (and blame) where its due!

Installation — JupyterLab 4.5.0 documentation – the official site

Getting Started with JupyterLab Tutorial: Installation, Extensions, and More | DataCamp an alternative guide!

## So- you have followed this “adventure” got a nice screen that you can type in-then what?

OK, this is maybe a bit of a cliff edge! Buttons and User Interfaces can be built using something like the wonderful tkinter library. You can, of course, do an entire website in Flask (although I wouldn’t) for “production” and to be more conventional! Sticking to javascript in html5, then piping the output as a json object into python “server side” gives you a centralised processing engine; this will be accessible by a “normal”, sane laptop or desktop, through a web browser. Using the standard, powerful webservers like Nginx or Apache will allow you to connect your web server to your python machine, and not anger your users with truly massive downloads (for 1 or 2 functions!). If all of this is opaque, I still remember why someone citing “function x” and “simple object” meant very little to me, consider it as learning musical notation vs TAB. OK, learning notation (especially also/viola clef, or Tenor the sometimes trombone clef) seems difficult. Yes, to this day I “cheat” at C clefs (viola and trombone) and yes, I make it hard for myself! I read them in transposition (a ridiculous thing for normal people to do, but horn players aren’t normal!). Suffice to say, I’m aware that anything I say to “simplify” this, will make it ridiculously complicated. My only advice is to find something you actually

want to do, and you will achieve it. When you have done one or two things, you will have learnt how you work, and then, you won't need someone telling you "how". Unless you feel like commercial "software engineering" is your thing. Enjoy this and the wonderful libraries available!

# The power of open communities:

This is personally something I am extremely conflicted with internally. Having spent decades falling in and out of various jobs that I always considered "way above me" (the hallmark of imposter syndrome). I have slowly come to the conclusion that maybe I just about got by for some time, and did some "mad" things. My irritating "perfectionist" tendencies are not a criticism of anyone else, they are the manifestation of my own inner voice. This is not particularly healthy, but, it can be a mask I have worn, and wrinkled! In reality, most things are probably "ok". At the same time, everything can be better. Of late, I have been trying some thoughts on open communities. It strikes me that probably 85% of people in these communities are good people, there are about 5% egomaniacs, and 10% of "imposter syndrome" people. Depending on the subject these proportions are highly dynamic!

I was an adherent to "proprietary is better" for decades! I'm not so sure now. One thing I am certain of is that this is not "simple" (binary). The reasons I am how I am are extremely complex, and I probably should get a therapist! But, onwards! If you take an open forum the way I think it should be, then putting an idea out there should be welcomed with either silence (nothing constructive to add, but silent acknowledgement) or a gentle, polite counter idea – I like these. Unfortunately, the things that totally make me run are – "praise", which I am immediately suspicious of! Or worse, savage, mis-placed criticism. There is a third kind, this is the person, that attempts to give a "pep talk". For personalities like mine, this is a complete disaster! The "problem" is entirely with me, and the well-intentioned person is without blame.

So, given the imperfection! It's a bit like the apocryphal "Churchill" quote – "Democracy is terrible, but it's the best choice when the other options are considered!"

# "Forums" and open-source development

There are many extremely beautiful examples of this. The giant (at present) is Github. A savage critique may be like dropping a bomb on the contributor. This can (in extremis) cause them to leave the thread, never to return. This is the "democratic damage" at work. In general, the process is – read, contribute, read some more, and then if its truly against you. And you didn't initiate – LEAVE! My instinct, with no quantitative analysis is that this is probably why "proprietary" solutions have (and likely will maintain) the "edge" over open-source. Most creative people suffer from some degree of imposter syndrome. Working within an organisation that may protect you, is extremely appealing. This may be a manufacturer, a university, or any other organisation. The fallacy that "everyone is entitled to an opinion" is utter nonsense! I would argue that "everyone is entitled to a point of view" if you think that is the same thing, I might suggest asking an academic or a judge for the distinction! In any case, apologies for the nearly whole page deviation! Let's look at some more "cool things".

## Lighting control

I have written about MQTT in another article. I worked hard to reframe the perception of mqtt being the domain of IoT hobbyists. In fact, I am not for one second suggesting that you allow Alexa or

google home to run your theatre lighting system! And no I'm not going to spring "smart life" on you either! Mission critical systems use the state driven concept to run. To define "mission critical" I'm talking about Nuclear reactors, oil pipelines, and transport systems. SCADA uses state driven systems, DMX is still using polled logic. Whilst I fully appreciate how it might feel like it, a lighting error in a theatre is not Chernobyl 2.0! The polled paradigm for lighting requires around 20,000 packets per second to (most of the time) maintain a relatively static state. Consider the contrast to midi – this is a state driven system. Its cruel, revised acronym – "marvellous idea dumb implementation" refers to the extremely low data rate chosen. In the 1980s, it was reliable. More recent experiments have shown that adsl can achieve 8Mbps over wet string! (This was actually a documented experiment!) Please avoid running DMX over wet string, it goes without saying – nobody will thank you for it!

Given ETC et al will not "bin" DMX based on a jumped-up network/noiseboy's say so overnight! We need to "deal with the world as we find it". Someone said this to me once upon a time (I may have ignored them… it did take a few years!). Back to what we can do RIGHT NOW. Let's unpack our cheap little Raspberry Pi5 "toy" and do some lighting! In fact we can do this on a Raspberry Pi nano… but I deviate more absurdly than normal! There is an SACN library for python! We can dup this out of a UART to an XLR – 5 pin or 3 pin – whichever you prefer (I know 5 pin prevents us dodgy noise boys nicking your lighting cables – we all know noise boys can't count to 3!)

## OK so some white numbers and text output on the screen?

Yes, not very exciting is it! Obviously, we want our code adventure to "do" stuff that might be useful. Traditionally, one would be exposed to semesters of theory, regurgitated "standard" teaching. Just think OSI model teaching, etc. As someone who may not have "normal" learning modes, I clearly remember having my ability to subtract BROKEN at the age of 8. This was a primary school teacher (not a mathematician… surprise!) teaching the rote learned method. Unfortunately, she managed to completely confuse me, and the thing I could already do (subtraction) was "wrong" and being "respectful of authority" I followed her explanation, and completely failed to execute.

Herein lies the problem. If you understand something, feel free to attempt to share your knowledge. There might be an Hippocratic concept – first do no harm! This would be wonderful – if anyone actually followed it perfectly! However, being human, we don't. We have no objective idea of our own limitations, so therefore, we have even less idea about the limitations of others! This is why my (slightly unhelpful) method advocates for – here is the library, here is the bookshelf, use a "toy" computer… here are some pages that will help you get going, and if it breaks… try again!

My later mathematics experiences have shown a repeating pattern. Mathematics – or in its "pure" form described as "religion" by some great minds in physics, is the abstraction of the concrete. The "religion" gentle critique is the acknowledgement that ultimately it doesn't provide any concrete "answers". Believe it or not, this is the entire purpose of mathematics!

On a section obsessed with practicality, why have I got so absurdly IMPRACTICAL! I think, this is a healthy way to "show the simple hack". This was revealed to me by a maths teacher, with a superb understanding (greater than mine) of the human condition, and Mathematics. I've through total coincidence, ran into him again, just over a quarter of a century later! His sage advice was, acknowledging the neuro mechanics of learning – go to the library, get this book, work through the exercises, and when you have done 200 or 300 exercises, it will get easier!

Just because this is how an organic neurological system works! The specific example involved logarithmic manipulation. Now, thanks to this gentleman, I cant remember how I ever found it hard, or got it wrong! The same could be said of my subtraction example, and, I would suggest this was a

“skimmed over” explanation of decimal place value. If we had spent just 1 day longer on this fundamental concept, the “method” would have worked, and saved three days of depression and head scratching. Symbol place value is so fundamental, that a good grasp makes “weird” symbols less intimidating. For 99% of the people you will ever meet, I’ll look (as normal) like I’m talking absolute nonsense! The next number after 9? In decimal its 10, in hexadecimal it’s A – different symbols used for different base. Its not “new” or “hard” just less familiar – this is why IPv6 seems “hard” – it’s not! In fact hexadecimal in ipv6 is a lot more sensible than decimal fallacy of ipv4, because numbers don’t stop at 255! The decimal notation of ipv4 is the “odd” “fraudulent” and actually as a result “hard” bodge, because it’s not telling the truth!

Being entirely aware of the natural tendency to “skim” , and I’m certainly no saint in this regard, I’m following the superior example from my anecdote! Because I start with the assumption that whoever I’m addressing is not “stupid”, but likely hitherto “ignorant” of what I’m enthusiastic about, I’ll take the criticism that I’ve exposed something 10 steps down the road! In acknowledgement of this, I’d prefer to step back gently, until we find ourselves on a mutually comfortable step. There is little point (for either of us) to go all the way back to the bottom. Hopefully, with conversation, we can progress together, more efficiently!

# Deviation over!

I thought it more helpful to put a big sign here! Hopefully, you have jupyter lab or notebook up. If you can’s see it in web browser – this will be the server not listening for a connection on the network – easily fixed. An AI will help in a heartbeat. Naturally, there will be myriad dire warnings about “hackers” – remember – hopefully you are using an RFC1918 address! And, hopefully you have deployed on a “disposable toy” (raspberry pi), so you are a lot less exposed.

Make something visual – look at the tkinter library, and maybe matplotlib to make a graph of your fft output.

There is absolutely nothing wrong with asking an AI (even Gemini) to create python code to test this (and its surprisingly very short).

Handling an audio interface or a DMX output is not that “hard”. After all the buffer for the interface is a file (in true unix style).

May I suggest for a quick start, using an already existing audio file, then its much easier to fiddle with the buttons on the fft.

Once again practice – you will install all manner of libraries, be confident that the flash card on the raspberry pi can be re formatted and start again, many times. Just remember to save anything of value elsewhere!

It wont be “right first, or second… time” this is the most dishonest statement made by salespeople to attempt to fool customers!

## OK – AI- they will steal my stuff!

As this is foundational, and based on first tentative steps; its unlikely you will expose cutting-edge code, or anything commercially useful in this endeavour! IF an AI tried to “steal” this stuff it’s a simple risk/reward calculation. Consider the egos involved – “my data is priceless” – yes it may well be (to you). Do not assume anything you say is immediately of value to anyone else! AIs work from

the base assumption that most data on the internet is garbage, and will not devote time and money to dead ends! Capitalism just doesn't work like that!

If you are on someone else's system, it is ESSENTIIAL to respect their position. To make this easy, the owner of the system should have policies and procedures concerning this kind of activity. You MUST respect these (no matter what your personal perspective may be). Personally, I do all my development on my own systems. This is the safest way. I go so far as to do this exclusively at home, no boundary blurring!

Personally, I'd advocate looking at stackoverflow. To get started – this is the kind of discussion that helps. Here is an excellent example of someone who has made some effort, posted, and look at the wonderful responses, and the evolution!

numpy - Plotting a fast Fourier transform in Python - Stack Overflow

The best thing- the search will help you find comments on many applications, and with a little practice you will find something of interest to you, and begin to develop quickly. This is the point where a revised security posture becomes worth considering!

For lighting – this is the sacn library

sacn · PyPI

for video – ffmpeg is the holy grail!

ffmpeg-python · PyPI

# Final Thoughts

If you have read this far, congratulations! I've quite deliberately excluded Computer Science jargon, in favour of providing a path to play. One day, you may well need the CS lexicon. This is only appropriate when needing to share ideas with excellent engineers, with no real practical experience! This is extremely common. Don't let people bully you with word salad! Personally, I know exactly how this document would land on a mid-level software engineer. If you feel this is "basic" or "over simplified" this document was not for you. If you have read this far and you feel this way, you are clearly looking for a fight – I'm NOT – go away!

For everyone else – I'm certain that this is not monolithic. There are inevitable sticking points. Kindly email me, and I will endeavour to help, and revise. In doing this, you have my thanks in advance.