GAYPYRUS THE GREAT NECROPOSTER RETURNS

A Preservation Notice, Incident Report, Firmware Postmortem, Personal History, and Aggressively Overengineered Forum Reply

```
CASE: EAGLEMOSS VECTOR 3
STATUS: PHYSICALLY EXTANT / PRACTICALLY DEAD
PRIMARY ARTIFACT: setup (2).exe
PRESERVATION STRATEGY: HASH IT / MIRROR IT / DO NOT TRUST ME
BLINDLY
ATTACK SURFACE: APPROXIMATELY TWO PEOPLE PER YEAR
CURRENT ARCHIVIST: GAYPYRUS
```

Prepared in 2026 while shopping for a replacement 3D printer and remembering, with increasing disbelief, how any of this happened in the first place.

NYEH HEH HEH.

Reader’s Warning

Security warning

This document contains links and placeholders for an old Windows executable belonging to a defunct commercial 3D-printer ecosystem. A colorful provenance story is *not* a cryptographic proof of safety. Treat every binary as untrusted until you have verified what you reasonably can. Compare independent hashes where possible, scan the file, and use an appropriately isolated environment if you are uncomfortable running it on a host system.

Field note

The chronology below is autobiographical and intentionally written in the style intended for the forum post. Dates and company-status timing are stated according to recollection where exact dates were not independently checked. The technical sequence of the repair attempt is preserved as accurately as possible: successful USB flash of Repetier, incorrect endstop/Z-axis behavior, corrected recompile, EEPROM trouble, loss of USB flashing, Arduino-as-ISP, dead Arduino, wrong rescue tool, eventual USBasp, and ultimate failure to revive the printer.

NYEH HEH HEH

The governing preservation principle of this document is simple: if a software ecosystem can disappear once, it can disappear again. Therefore the correct number of mirrors is not “one sensible link.” It is “enough links that somebody eventually asks whether this has become a CDN.”

Contents

Reader's Warning 2

1. The Return 5
- 1.1. Case Metadata . . . 6

2. How I Acquired This Cursed Machine 7
- 2.1. The Five-Year Preparation Phase . . . 7
- 2.2. Approximate Timeline of the Curse . . . 9

3. Slight Problem 10

4. The Great Vector 3 Resurrection Attempt 11
- 4.1. Attempt One: Replace the Firmware . . . 11
- 4.2. Attempt Two: EEPROM Is a Social Construct . . . 12
- 4.3. Attempt Three: Enter the Arduino . . . 13
  - 4.3.1. Repair Dependency Graph . . . 14
- 4.4. Attempt Four: The Wrong Fucking Tool . . . 15
- 4.5. Attempt Five: Professional Help . . . 16

5. The Current State of the Patient 19

6. How This Led Me to This Thread 20

7. Why Am I Back? 22

8. Provenance and Security 23
- 8.1. Threat Model, Such as It Is . . . 24
- 8.2. Canonical Artifact Hashes . . . 24
- 8.3. Optional Scanner / Archive References . . . 24

9. Downloads: The Gaypyrus CDN 25

10. Photo Archive 27
- 10.1. Public Photo Links . . . 28

11. For the Future Archaeologist 29
- 11.1. Trust vs. Availability . . . 30

12. In Conclusion 31

A. Artifact Metadata Template 33

B. Minimal Forum-Safe Security Note 34

C. Colophon 35

Field Index **36**

1. The Return

HELLO AGAIN.

@ateliernab, you were completely reasonable.

For reference, this is the thread: https://forums.sketchup.com/t/vector-3-firmwaresoftware/137936.

A couple of years ago, a brand-new account appeared in this already ancient thread, uploaded a ZIP containing an executable called `setup (2).exe`, provided approximately no provenance, and then disappeared.

You looked at this and, quite reasonably, went:

forum-reply threat model

```
NEW ACCOUNT + NECROPOST + RANDOM ZIP + setup (2).exe
```

That was perhaps not the world's most compelling software supply-chain security model.

The post was eventually removed.

Unfortunately, this may have been devastating to the approximately four people who have needed Eaglemoss Vector 3 software in the intervening two years. **NYEH HEH HEH.**

Anyway.

I recently remembered this thread while looking for a new 3D printer and reminiscing about the utterly absurd circumstances under which I got my first one. I came back to see whether my old link was still here.

It wasn't.

My post had been removed.

And you know what?

Verdict

FAIR.

If I were moderating a forum and some new account dropped `setup (2).exe` into a necro'd thread with essentially no explanation, I would also look at that and think:

console

```
ABSOLUTELY FUCKING NOT.
```

So I have returned, this time with an explanation, provenance, hashes, photographs, and enough redundant download links to make killing this installer slightly more difficult than killing Eaglemoss itself.

1.1. Case Metadata

Artifact record

Subject	Eaglemoss Vector 3 3D printer
Acquisition	Roughly eight years before this document, purchased by my mother from her friend at approximately then-market value
Approximate cost	About 2000 pre-inflation Polish złoty
Initial containment period	Roughly three years inside the shipping box
Secondary containment period	Roughly two additional years in the branded box under a table
Software strategy	Download official installer in advance and wait for civilization to permit ten screws
Later firmware strategy	Community Repetier, then direct ISP flashing
Current state	Dead enough that repair has become a philosophical Ship-of-Theseus problem
Preserved artifact	`setup (2).exe`
Archivist status	Studying cybersecurity; still somehow responsible for a dead printer company's installer

2. How I Acquired This Cursed Machine

Roughly eight years ago, my mother bought an Eaglemoss Vector 3 from her friend for approximately its market value at the time: around **2000 pre-inflation Polish złoty**.

This was, to me, incredible.

Initial condition

WE HAD A FUCKING 3D PRINTER.

There was one problem.

I wasn't allowed to use it because I would break it.

Not "don't use it unsupervised."

Not "wait until somebody shows you how."

I was not even allowed to take the fucking thing **out of its shipping box**.

And there it remained for approximately three years, perfectly protected from wear, damage, usefulness, and technological relevance.

Eventually I was granted a major privilege: I could take it out of the shipping box. **NYEH HEH HEH.**, technological progress!

I removed the branded Vector box from its shipping box and placed it underneath the table.

I still wasn't allowed to assemble it.

My mother was going to get somebody who knew what they were doing to come over and assemble it properly. This was promised repeatedly.

For another **two fucking years**.

The Vector, for reference, required roughly **ten screws** to assemble.

2.1. The Five-Year Preparation Phase

Despite apparently lacking the qualifications necessary to operate ten screws, I spent those years reading about the printer and preparing for the glorious hypothetical future in which I might actually be allowed to use the thing we owned.

Most importantly, **I downloaded the official software in advance**.

This turned out to be one of the better decisions younger me made, because apparently I had a stronger disaster-recovery strategy than the company selling the printer.

I downloaded:

console

```
setup.exe
```

Apparently more than once.

Then I waited.

And waited.

And waited.

For about five years. **NYEH HEH HEH.**

Eventually I got fed up with promises that somebody would someday be summoned to perform the arcane ritual of screwing the printer together.

So while my mother was away, I secretly started assembling it myself.

She returned while I was halfway through.

After five years of "don't touch it, you'll break it," she saw what I was doing and essentially went:

Authorization, revised

"GO AHEAD. BREAK IT."

So I finished assembling it.

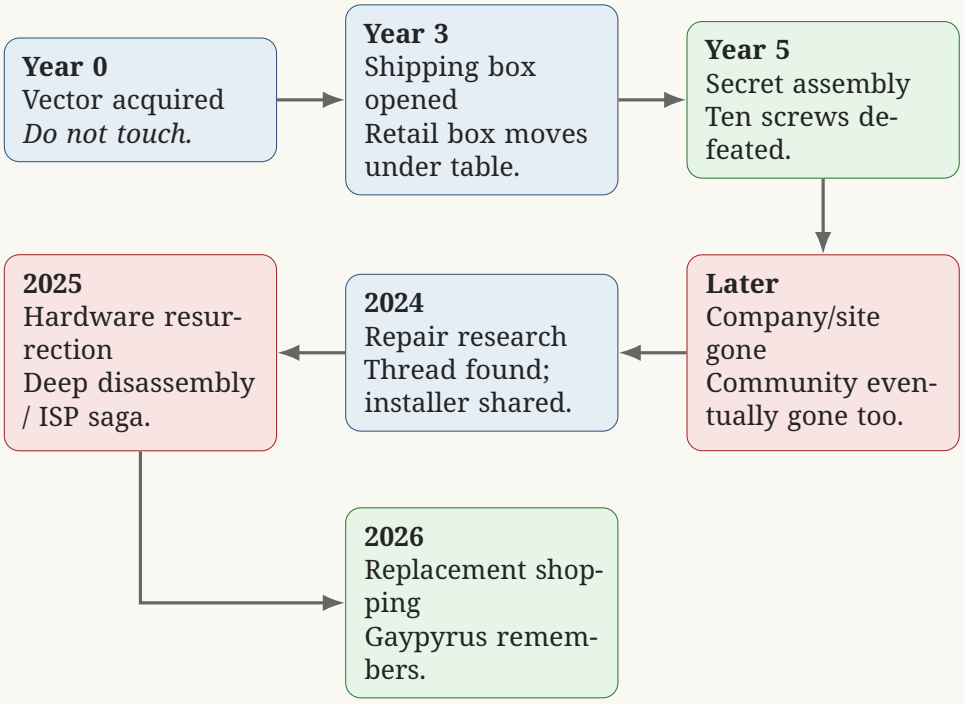
And then something completely unexpected happened.

It fucking worked.

It wasn't even difficult.

I had successfully defeated the screws.

2.2. Approximate Timeline of the Curse



3. Slight Problem

By the time this glorious moment occurred, the Eaglemoss website was already dead and, to the best of my recollection, Eaglemoss itself was already dead too.

Fortunately, the independent Vector 3 community forum **was still alive**.

And because younger me had spent five years anticipating the possibility of someday actually using my own printer, I already had the official software sitting in my files.

So the Vector worked.

It printed.

It wasn't particularly good, especially by then, but after half a decade of waiting I finally had my fucking 3D printer. **NYEH HEH HEH.**, VICTORY.

Eventually the community forum disappeared too.

And the years were not particularly kind to a proprietary printer that wasn't exactly cutting-edge to begin with.

But it kept going.

Eventually, by the 2024–2025 period, it broke badly enough that the resurrection saga began.

4. The Great Vector 3 Resurrection Attempt

By this point, troubleshooting the Vector was less “consult the manufacturer’s documentation” and more **perform digital archaeology on the remains of a vanished civilization**.

Component of civilization	Status
Eaglemoss	dead
Official website	dead
Official downloads	dead
Dedicated Vector 3 community forum	dead
Replacement parts	good fucking luck
Printer	increasingly dead

NYEH HEH HEH.

So I did the reasonable thing.

I attempted to resurrect it myself.

And, importantly, I actually got **surprisingly far**.

4.1. Attempt One: Replace the Firmware

I found the community Repetier setup for the Vector and successfully flashed it to the printer **over USB**.

That part actually worked.

The printer booted the firmware.

Success!

Sort of.

There were some minor configuration problems.

Such as:

⚠ Minor configuration issues

- the endstops being incorrect;
- **THE ENTIRE FUCKING VERTICAL AXIS MOVING IN THE WRONG DIRECTION.**

This is generally considered undesirable behaviour in a 3D printer. **NYEH HEH HEH.**, MOTION CONTROL.

Fortunately, that looked fixable. The configuration could be corrected. Repetier could be recompiled.

So that's what I did.

I corrected the configuration, rebuilt the firmware, and flashed it again.

And then the Vector decided that cooperation had been a temporary promotional feature.

4.2. Attempt Two: EEPROM Is a Social Construct

The next problem appeared to involve the EEPROM.

Settings that were supposed to be stored there were not behaving properly. Values weren't updating the way they should. The EEPROM wasn't properly editable.

It was not, in any useful sense of the word, **EEPROMing**.

Apparently *Electrically Erasable Programmable Read-Only Memory* had interpreted its name as an aspirational job description rather than a technical specification.

So I tried flashing again.

And again.

And again.

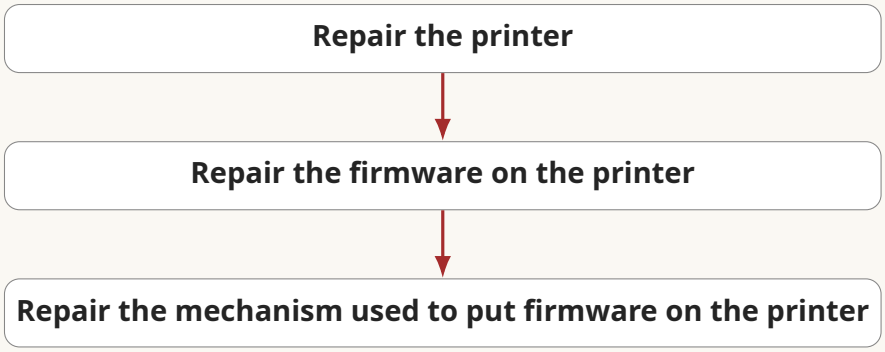
Roughly three more attempts later, I still did not have a working printer. **NYEH HEH HEH.**, NON-VOLATILE STORAGE.

And then another problem appeared.

The board stopped responding to firmware flashing over USB.

Excellent.

At this point the repair had progressed from:



This is approximately the point where the photographs begin becoming increasingly concerning.

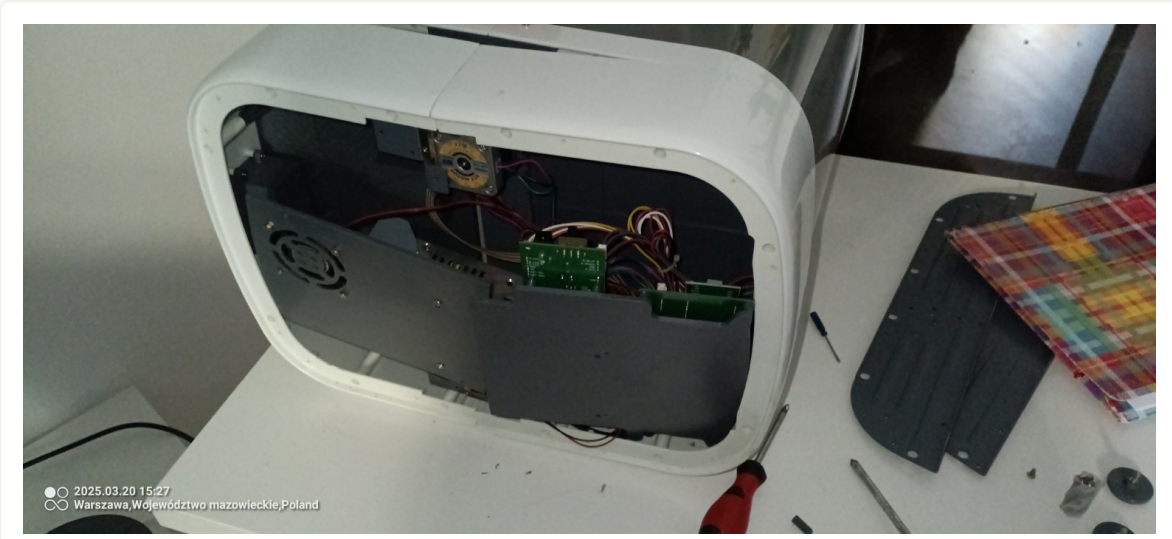


Figure 4.1.: **Exhibit A: first major disassembly.** Rear panel removed, metal electronics housing exposed. The patient is still recognizably a printer.

Public photo URL: `[INSERT LINK]`

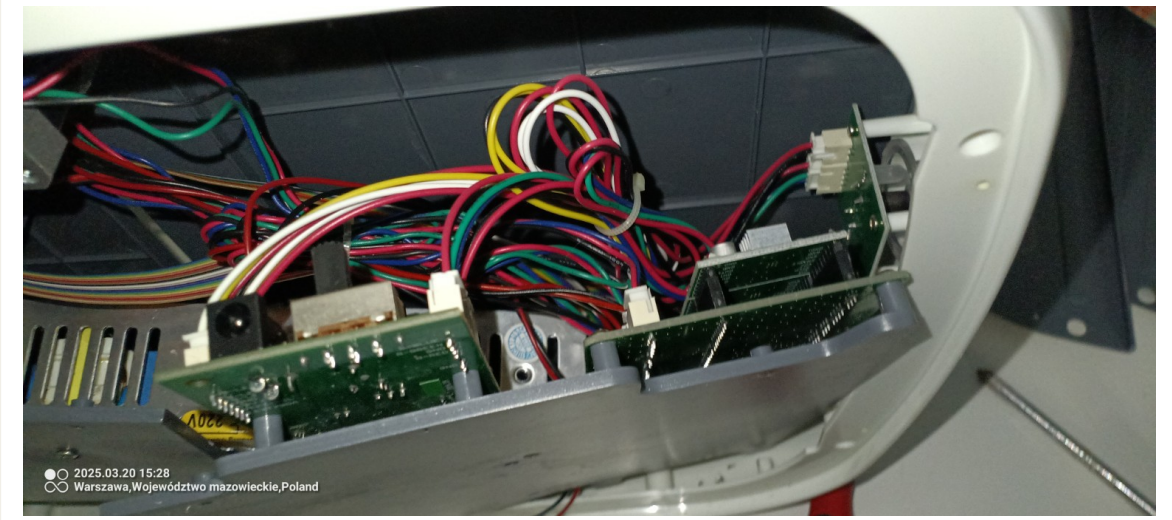


Figure 4.2.: **Exhibit B: electronics and wiring exposed.** The number of visible conductors has increased considerably. This is rarely emotionally reassuring.

Public photo URL: `[INSERT LINK]`

4.3. Attempt Three: Enter the Arduino

Since USB flashing had stopped working, the next obvious step was to bypass it and program the controller directly.

I did not yet own a USBasp.

But I did own an Arduino.

An Arduino can be used as an ISP programmer.

Problem solved. **NYEH HEH HEH.**

I configured the Arduino as a programmer and started using it to try to reflash the Vector's controller directly.

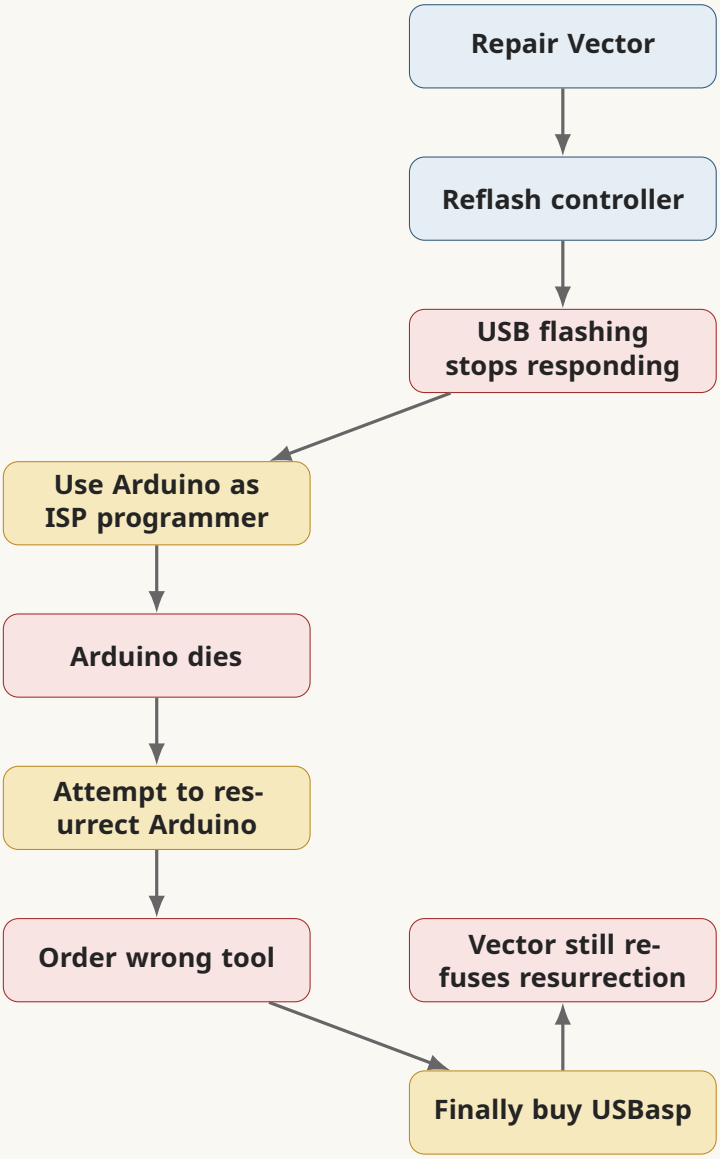
And then:

Escalation event

THE ARDUINO FUCKING DIED.

The device I was using to resurrect the device I was trying to resurrect now itself required resurrection.

4.3.1. Repair Dependency Graph



I had successfully reached the stage where I was **repairing the device I was using to repair the device I was trying to repair**. **NYEH HEH HEH.**, RECURSION.



Figure 4.3.: **Exhibit C: deeper controller access.** The lower bay is open and the controller boards are accessible. The repair has officially left the “consumer appliance” phase.

Public photo URL: `[INSERT LINK]`

4.4. Attempt Four: The Wrong Fucking Tool

Naturally, I attempted to resurrect the Arduino too.

So I ordered a tool for that.

It was the wrong fucking tool.

At some point I finally accepted that constructing increasingly elaborate programmer-programming-programmer chains might not be the optimal approach.

So I bought the thing I probably should have bought in the first place:

Acquired specifically because of this printer

USBasp

A proper dedicated AVR ISP programmer.

I bought that USBasp **specifically because of this fucking Eaglemoss Vector 3**.

I still own it.

Armed with the actual correct programming tool, direct access to the electronics, community firmware, a corrected Repetier build, several increasingly disassembled circuit boards, and significantly more determination than this printer deserved, I attempted the resurrection again.

It still didn’t fucking work. **NYEH HEH HEH.**



Figure 4.4.: **Exhibit D: electronics extraction.** By this point the electronics are no longer merely visible. They are becoming geographically independent of the chassis.

Public photo URL: `[INSERT LINK]`

4.5. Attempt Five: Professional Help

At some point during this saga, I considered the radical possibility that perhaps somebody who repaired 3D printers professionally might be willing to deal with this.

So I contacted a 3D-printer repair service.

Apparently my initial description did not adequately communicate the exact condition of the patient.

So I sent them a helpful follow-up email:

Helpful clarification

PS zdjęcie drukarki w aktualnym stanie

Or, approximately:

> *P.S. photo of the printer in its current state.*

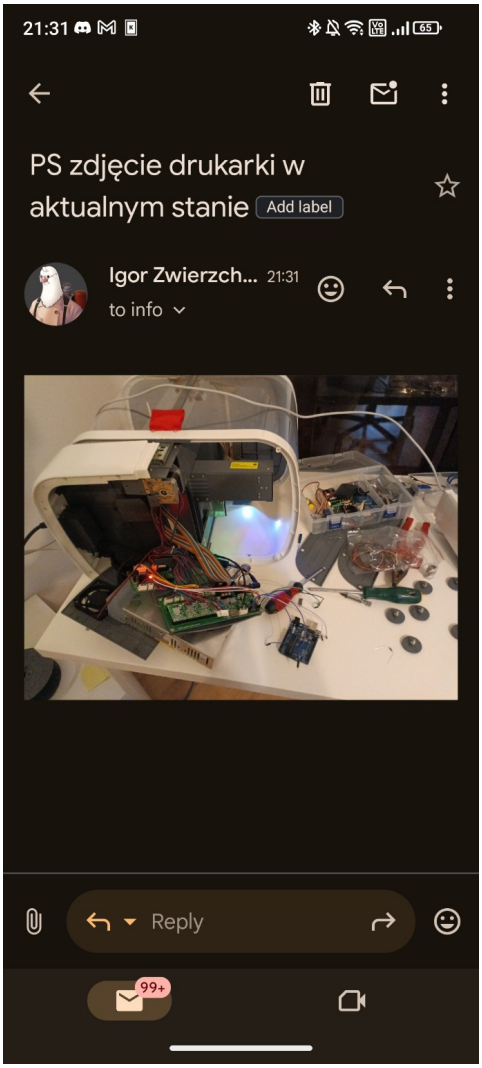


Figure 4.5.: **Exhibit E: the clarification email.** The embedded photograph shows the current state of the printer. This was apparently insufficient to inspire commercial enthusiasm.

Public photo URL: `[INSERT LINK]`

The photograph showed the Vector with significant portions of its electronics removed from the chassis and spread across the table, wires everywhere, tools surrounding it, and an Arduino involved.

It looked less like *consumer 3D printer requiring repair* and more like *aftermath of an unsuccessful autopsy*.

They declined the repair.

I cannot imagine why. **NYEH HEH HEH.**



Figure 4.6.: **Exhibit F: the full fucking autopsy.** Wide view of the Vector during the repair attempt, surrounded by tools, components, loose electronics, and the general atmosphere of a project that has developed lore.

Public photo URL: `[INSERT LINK]`

5. The Current State of the Patient

Eventually I had to accept that I had crossed the boundary between **repairing an old 3D printer** and **beginning an involuntary electrical-engineering research project**.

Could the Vector theoretically be made to print again?

Probably.

The motors are motors. The heaters are heaters. The thermistors are thermistors.

At some sufficiently extreme level of Ship-of-Theseus reconstruction, I could replace the controller, redo the electronics, configure modern firmware around whatever mechanical components remain useful, and make the thing move again.

But at that point:

Philosophical maintenance question

How much of the Eaglemoss Vector 3 am I actually repairing?

And perhaps more importantly:

Resource exhaustion

HOW MANY MORE PROGRAMMERS MUST DIE BEFORE THE GODS ARE SATISFIED?

So, for now, the Vector remains dead.

The USBasp remains alive.

```
>_ console
THE USBasp WON.
```

NYEH HEH HEH.

6. How This Led Me to This Thread

While performing all of the above digital archaeology—searching for firmware, old discussions, documentation, dead community resources, GitHub repositories, and anything else that might help me resurrect the machine—I stumbled across **this thread**.

Someone here was looking for the original Vector software because the official download had disappeared.

And I thought:

NYEH HEH HEH

WAIT.

I HAVE THAT FUCKER.

So I went through my archive and found the installer I'd downloaded years earlier.

The original installer was called:

```
>_ console
setup.exe
```

However, because I had apparently downloaded it multiple times, the particular copy I found was:

```
>_ console
setup (2).exe
```

I couldn't be fucked to rename it because I was busy getting fucked elsewhere.

So I created an account, necroposted on the thread, uploaded my copy, and provided approximately **none** of the provenance I have just spent several chapters explaining.

NYEH HEH HEH., CYBERSECURITY.

Somebody understandably responded along the lines of:

> *Why has a brand-new account just appeared in a years-old thread and given us a ZIP containing a suspiciously named executable?*

Excellent question.

Particularly because I am now studying cybersecurity.

In retrospect:

Post-incident review

I AGREE WITH YOU COMPLETELY.

And to the person whose 2024 reply I am specifically responding to now: **your reaction was correct**.

You had absolutely no reason to trust that upload.

My only criticism is directed at the universe for arranging things such that completely sensible forum skepticism, later post removal, and catastrophic software preservation happened to collide.

To anyone who arrived here during the roughly two years between my original post disappearing and this one appearing: I apologize.

You were collateral damage of **entirely sensible security practice and forum housekeeping**. **NYEH HEH HEH.**

7. Why Am I Back?

Fast-forward to 2026.

I'm finally looking at buying a replacement 3D printer.

While comparing modern Prusas and Bambu Labs and contemplating just how hilariously primitive the Vector has become by comparison, I started reminiscing about the absurd sequence of events through which it took me **five years to gain access to my own 3D printer**.

That reminded me of this thread.

So I came back.

And discovered that my original post containing the installer had been removed.

Again: **entirely reasonable**.



This time we're doing it properly.

8. Provenance and Security

As far as I know, the executable below is an original copy of the official Eaglemoss Vector 3 software installer, downloaded by me from the official source while it was still available.

The original filename was `setup.exe`. The `(2)` suffix was added automatically because I downloaded the same file multiple times.

I have not intentionally modified the executable.

Do not replace verification with lore

Do not blindly trust that statement.
I am still a random person on the internet distributing an old Windows executable for a bankrupt company. Yes, I now have an absurdly detailed history explaining why I possess it. Yes, I have photographs of the actual Vector progressively undergoing electronic disembowelment. Yes, I have the USBasp I bought specifically because of this machine. Yes, fabricating this entire saga purely to compromise the approximately two people per year still searching for Vector 3 software would represent an impressively poor return on investment.
None of that cryptographically proves the executable is safe.

Treat the story as **provenance**, not as proof.

Verify the hashes. Scan the file yourself. If you do not trust it on your host system, inspect and/or run it in an appropriately isolated VM or sandbox.

If anyone has another independently preserved copy of the original installer, **please hash it and compare it with mine**.

That would be enormously useful. An independently sourced matching copy is substantially better evidence than me repeatedly assuring everyone that *the suspicious EXE is totally fine bro*. **NYEH HEH HEH.**

8.1. Threat Model, Such as It Is

Question	Assessment	Consequence
Could the story be fabricated?	Of course. A sufficiently motivated attacker can write lore.	Story alone proves nothing.
Could the photos be staged?	In principle, yes. The commitment-to-the-bit cost would merely be hilarious.	Still not proof.
Is the target population enormous?	No. The plausible attack surface is approximately "a couple of Vector owners per year."	Extremely poor malware campaign economics.
Does low attacker ROI imply safety?	No. Weird attacks and compromised archives exist.	Verify anyway.
What materially improves confidence?	Independently sourced identical binaries/hashes, reputable scanning, controlled execution.	This is what future archaeologists should contribute.

Field note

The ideal corroboration is not twenty copies of *my* file. It is one or more independently preserved historical copies from other Vector owners whose cryptographic hashes match mine. Mirroring improves availability; independent provenance improves confidence. Those are different properties.

8.2. Canonical Artifact Hashes

Artifact record

Artifact	SHA-256
`setup (2).exe`	`[INSERT SHA-256]`
`Vector3-Eaglemoss-Software.zip`	`[INSERT SHA-256]`

8.3. Optional Scanner / Archive References

Virus scanning/report URL	`[INSERT LINK]`
Internet Archive item	`[INSERT LINK]`
Independent hash confirmation 1	`[INSERT LINK OR HASH]`
Independent hash confirmation 2	`[INSERT LINK OR HASH]`

9. Downloads: The Gaypyrus CDN

Last time I provided one link.

Apparently I underestimated the mortality rate of websites associated with the Eaglemoss Vector 3.

I have learned.

Below are mirrors of the **same canonical archive** across as many independent file hosts as I can reasonably be bothered to upload it to.

All mirrors should contain **identical bytes** and therefore match the SHA-256 hash above.

Node	URL
Mirror 01	[INSERT LINK]
Mirror 02	[INSERT LINK]
Mirror 03	[INSERT LINK]
Mirror 04	[INSERT LINK]
Mirror 05	[INSERT LINK]
Mirror 06	[INSERT LINK]
Mirror 07	[INSERT LINK]
Mirror 08	[INSERT LINK]
Mirror 09	[INSERT LINK]
Mirror 10	[INSERT LINK]
Mirror 11	[INSERT LINK]
Mirror 12	[INSERT LINK]
Mirror 13	[INSERT LINK]
Mirror 14	[INSERT LINK]
Mirror 15	[INSERT LINK]
Mirror 16	[INSERT LINK]
Mirror 17	[INSERT LINK]
Mirror 18	[INSERT LINK]
Mirror 19	[INSERT LINK]
Mirror 20	[INSERT LINK]

Yes.

Twenty.

If you're wondering why the same obscure 3D-printer installer requires twenty redundant mirrors, I invite you to gesture broadly at the preceding history. **NYEH HEH HEH.**, REDUNDANCY.

Please make another copy yourself. Put it on the Internet Archive if they'll take it. Preserve the hashes. If you discover useful Vector documentation, firmware, schematics,

community software, or old forum archives that are not already safely preserved, archive those too.

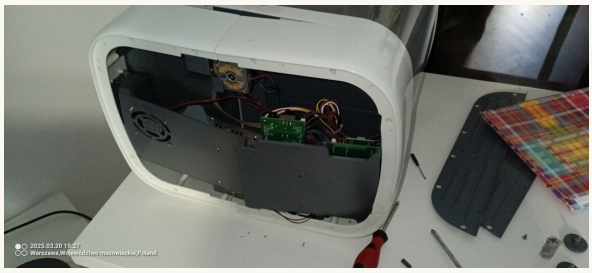
Architectural objective

I REFUSE TO REMAIN A CRITICAL COMPONENT OF THE DISASTER-RECOVERY INFRASTRUCTURE FOR EAGLEMOSS.

That is an absolutely catastrophic architecture.

10. Photo Archive

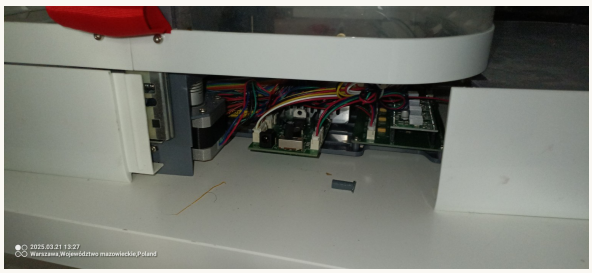
Since apparently I have also accidentally become a historian of my own failed printer resurrection, here is the photographic record in one place.



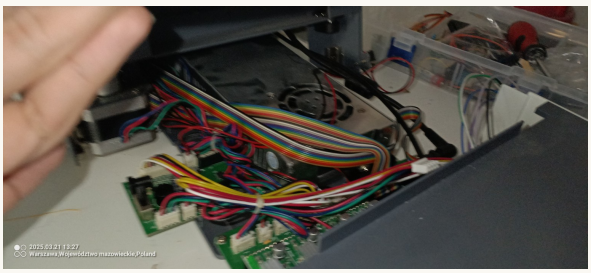
(a) Rear open / initial disassembly.



(b) Internal boards and wiring.



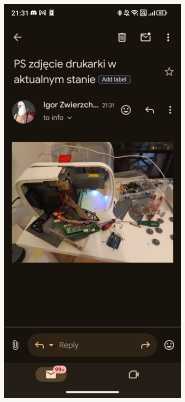
(c) Controller/electronics access.



(d) Electronics removed during repair.



(e) Wide shot of the full autopsy.



(f) The repair-service clarification email.

Figure 10.1.: Selected evidence from the Great Vector 3 Resurrection Attempt.

10.1. Public Photo Links

Rear open / initial disassembly	[INSERT LINK]
Wiring close-up	[INSERT LINK]
Lower bay / controller access	[INSERT LINK]
Electronics extraction	[INSERT LINK]
Full autopsy	[INSERT LINK]
Repair-service email screenshot	[INSERT LINK]

If any future Vector owner reaches this stage:

NYEH HEH HEH

I am sorry.

NYEH HEH HEH.

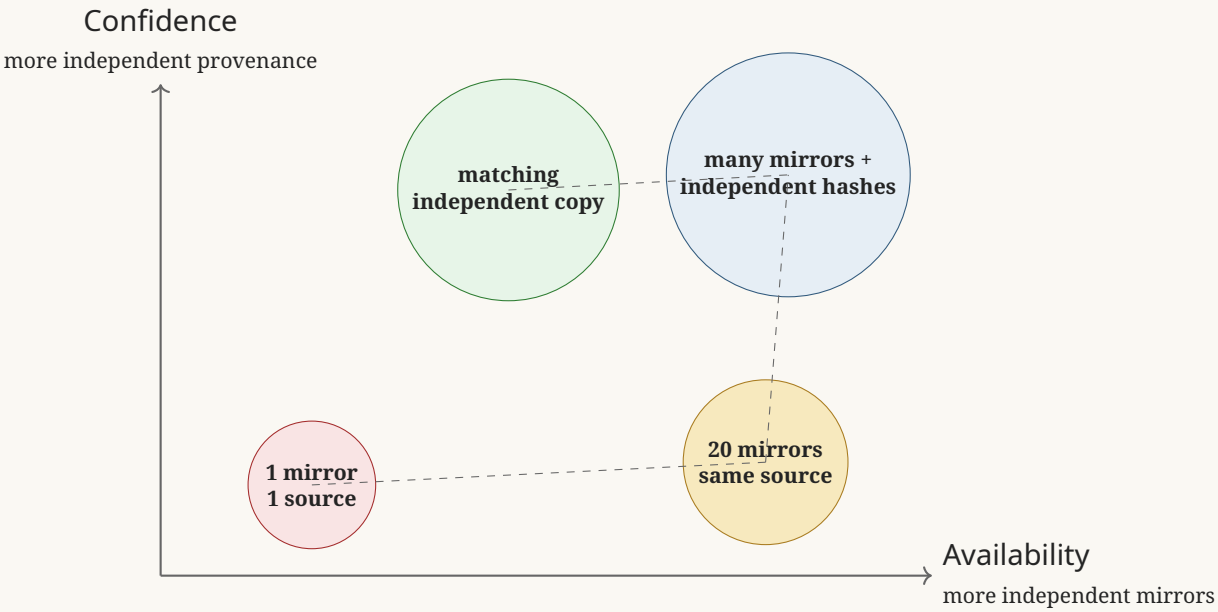
11. For the Future Archaeologist

If somebody finds this document years from now while trying to resurrect another Vector, the point is not merely to preserve one installer. Preserve the *context* needed to establish what the artifact is and how to use it safely.

Preservation checklist

1. Download the canonical archive from any surviving mirror.
2. Compare its SHA-256 against this document.
3. Extract it without overwriting your only archival copy.
4. Verify the executable hash separately.
5. Scan it using whatever reputable tooling is current in your era.
6. Prefer a disposable or isolated environment if you are uncertain about old software.
7. If you possess an independent historical copy, publish its hash even if you do not want to redistribute the binary.
8. Mirror manuals, configuration files, firmware, forum exports, schematics, and repair notes as well as the installer.
9. Preserve this document or its essential provenance alongside the binary so the next person does not encounter an unexplained `setup (2).exe` floating in the void.
10. If you create another mirror, add it to the thread.

11.1. Trust vs. Availability



Twenty mirrors solve **availability**. Another old owner producing a matching independently preserved copy improves **provenance**. Ideally, future Vector archaeology gets both.

12. In Conclusion

My mother bought me a 3D printer.

I wasn't allowed to remove it from the shipping box for three years.

Then I was allowed to put the retail box underneath a table.

I spent approximately five years anticipating being allowed to use it and, fortunately, downloaded the official software during that period.

Eventually I clandestinely assembled the approximately ten screws myself.

It worked.

By then Eaglemoss and its website were dead, but the independent community survived long enough for me to actually use the printer.

Eventually the community died too.

Then the printer broke.

I successfully replaced its firmware with Repetier.

The endstops were wrong.

The vertical axis moved backwards.

I corrected and recompiled the firmware.

The EEPROM decided it had no contractual obligation to function.

I reflashed it repeatedly.

USB flashing died.

I used an Arduino as an ISP programmer.

The Arduino died.

I tried to resurrect the Arduino.

I bought the wrong tool.

I bought a USBasp.

The USBasp worked.

The printer still didn't.

I disassembled increasing amounts of the machine.

I sent a photograph of the resulting electronic autopsy to a professional repair service.

They declined.

While trying to resurrect the printer, I discovered this thread and became an accidental software archivist.

I distributed the original installer in approximately the most malware-looking way imaginable.

A forum user quite reasonably looked at it and went:

>_ console

```
ABSOLUTELY FUCKING NOT.
```

That post was later removed.

Anyone who came here looking for the software during the intervening two years was, regrettably, collateral damage of sensible security practice and forum housekeeping. **NYEH HEH HEH.**

Two years later, while shopping for the Vector's replacement, I remembered all of this.

And now I've returned with:

Second necropost release notes

- provenance;
- SHA-256 hashes;
- photographs;
- a repair dependency graph;
- a threat model;
- twenty fucking mirrors;
- and considerably more cybersecurity education than I possessed the first time.

If somebody finds this document ten years from now while trying to resurrect another Vector:

Message to future archaeologists

Godspeed, future archaeologist.

Download everything.
Verify the hashes.
Make another mirror.
Preserve whatever is left.

And if the EEPROM starts acting strange: run.

NYEH HEH HEH.

A. Artifact Metadata Template

Use this section to replace placeholders before posting the document or converting its contents back into the forum reply.

Field	Value
Forum thread URL	https://forums.sketchup.com/t/vector-3-firmwaresoftware/137936
2024 skeptical reply	@ateliernab
Original executable filename	setup.exe
Preserved local filename	setup (2).exe
Executable SHA-256	[INSERT HASH]
Canonical archive filename	Vector3-Eaglemoss-Software.zip
Archive SHA-256	[INSERT HASH]
Virus scan/report	[INSERT LINK]
Internet Archive mirror	[INSERT LINK]
Independent corroborating hash	[INSERT HASH / LINK]
Date of Gaypyrus return post	[INSERT DATE]

B. Minimal Forum-Safe Security Note

If the full manifesto is too large for a forum post, preserve at least the following block with the links:

Security warning

Security / provenance note. This executable is an old archived copy that I believe I originally downloaded from Eaglemoss while the official Vector 3 software was still available. The original name was `setup.exe`; my preserved copy is called `setup (2).exe` because I downloaded it more than once. Do not trust the file solely because I say it is legitimate. Verify the SHA-256 hash shown here, scan it, use an isolated VM/sandbox if appropriate, and please compare against any independently preserved historical copy you possess. A matching independent hash is much stronger evidence than my provenance story alone.

C. Colophon

This document was intentionally overengineered in LaTeX because a plain forum post was apparently insufficient for the second coming of a file named `setup (2).exe`.

It contains actual photographs from the Vector 3 repair attempt, deliberately redundant artifact slots, diagrams of the repair dependency failure, a distinction between availability and provenance, and far more typographic infrastructure than any abandoned 3D-printer installer reasonably deserves.

I’m gay.

NYEH HEH HEH.

Field Index

Arduino, 11

EEPROM, 11

mirrors, 25
moderation, 5

necroposting, 5

provenance, 23

Repetier, 11

setup (2).exe, 20
SHA-256, 23
shipping box, 7

ten screws, 7

USBasp, 11