NERO PROJECT

Alarm Revival - AI That Sees, Describes and Confronts Intruders

It sees you. It describes you. It tells you to leave.

Paul Schindler | Built with Claude | November 2026

Executive summary

Our house came with a dead alarm. Its control panel was locked to an installer that had gone out of business, yet every wireless door and window sensor still worked. Instead of a new system and a monthly contract, I mapped every sensor by hand and set up a $52 radio receiver to hear them directly, feeding an open-source alarm system under my control. A live-fire test in June 2026 (arm, breach, siren, phone alerts) passed and caught two real bugs, which I had fixed and re-tested before relying on it. In September 2026 I gave the alarm a voice. When it trips, a local AI vision model reads the camera facing that door, and a deep processed voice tells the intruder over the siren what they are wearing and to leave, looping until someone disarms it.

$52
in new hardware to revive the alarm

$0
monthly fee, no dealer, no subscription

2
real bugs caught by the live-fire test, both fixed

0 s
from siren to voice (13 s before the pre-build fix)

What is in this report

Page	Contents
2	The problem, why it mattered and what I ruled out
3	Act 1, the revival - mapping the sensors, the signal path and the live-fire test that caught two bugs
4	Act 2, the talking alarm - how a camera frame becomes a spoken warning over the siren
5	Safeguards - who may arm and disarm, and what keeps the AI in its lane
6	Results, status, next steps and what this project shows about how I work

My role

I decided to keep the old sensors, designed the alarm's rules, approved every change and ran the tests myself. AI coding agents (Claude) wrote the code and the configuration under plans I approved. Sensor details, camera details and timings are left out of this report on purpose.

1. The problem and why it mattered

The alarm was already in the walls when we moved in. Wireless sensors sat on the doors and windows, and all of them still transmitted. The control panel was the problem. The installer had locked it to their own account and then gone bankrupt, so nobody could unlock it. The house had a full set of working sensors reporting to a box that would never listen again.

The standard answer is rip and replace. A new panel, new sensors, a new installation and a new monthly monitoring contract. That path throws away hardware that works and adds a recurring bill for something the house already owned. It also leaves the owner dependent on a vendor's servers to decide whether the alarm is allowed to work on a given day.

I started planning the revival in my notes in October 2025. I set three goals. Keep every working sensor. Run the alarm on software I control, inside the house, with no subscription. Prove it works by setting it off, not by reading its settings.

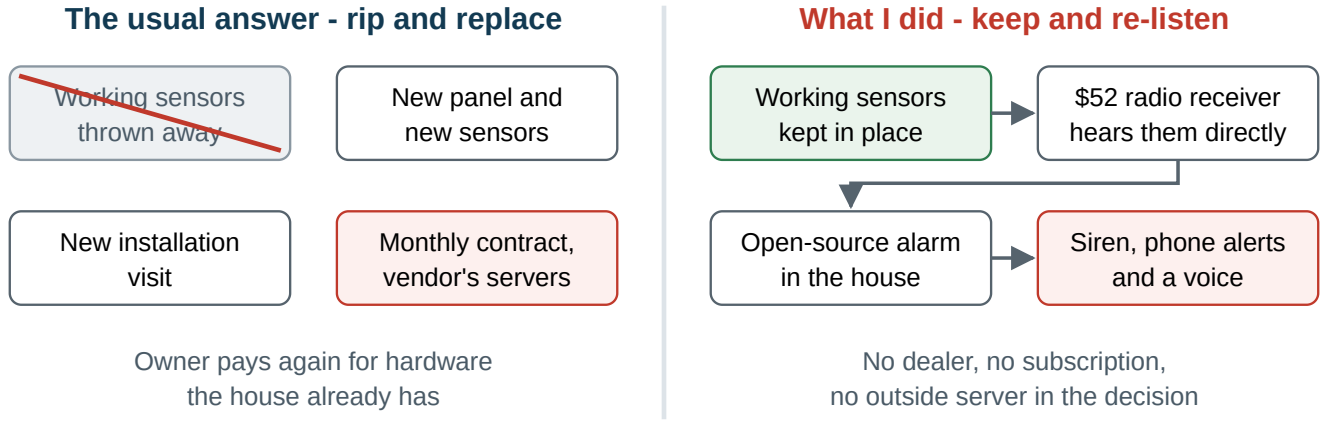


Figure 1. Two ways to fix a dealer-locked alarm. The right-hand path cost $52 in new hardware.

2. What I ruled out, and the design choices

Option considered	Decision and reason
Replace the system on a monitoring contract	Ruled out. It discards working sensors and adds a recurring fee for hardware the house already owned.
Get the old panel unlocked	Not possible. The dealer that locked it had gone bankrupt, so nobody held the key.
Move the receiver to the home-automation server	Ruled out. Reception was already proven where the receiver sat, so moving it gained nothing. Leaving it there kept the server single-purpose.
Let the software auto-discover sensors	Ruled out. Auto-discovery creates generic numbered entries. I had each sensor written in by hand, named for its door or window, so the alarm was ready to arm from day one.
A canned, pre-recorded warning	Rejected in favor of a live description. A warning that names what the intruder is wearing tells them they have been seen, not just heard.
Speak through small voice speakers in each room	Tried first and dropped the same evening. They were too quiet and the voice was too polite. The warning moved to the main speaker with the siren mixed underneath.
A larger vision model	Replaced after a side-by-side test. A smaller model answered in 0.55 seconds instead of 0.94, used 3.3 GB of memory instead of 5.9 and made fewer false calls.

ACT 1 - THE REVIVAL

3. Mapping every sensor and listening to it directly

The sensors were still transmitting. The missing piece was something to receive them, so I bought a $52 radio receiver and plugged it into my home workstation. Open-source decoder software turns each radio burst into a plain event, such as "this window opened".

Mapping came next, and I did it by hand. I walked the house and opened each door and window three times while an AI agent watched the radio log and matched each burst to the spot I named. One sensor moved at a time, so no two signals could be confused. The test also exposed a trap. The field that looked like the door's status stuck on "open" after the first opening. Repeated open-and-close cycles showed which field tracked the door, and the alarm reads that one.

1. Receive
A $52 radio receiver hears the old sensors directly.

2. Decode
Software turns each burst into a readable event.

3. Publish
Events land on the house's message bus.

4. Decide
The alarm software judges normal or breach.

5. Act
Siren, critical phone alert and a Telegram message.

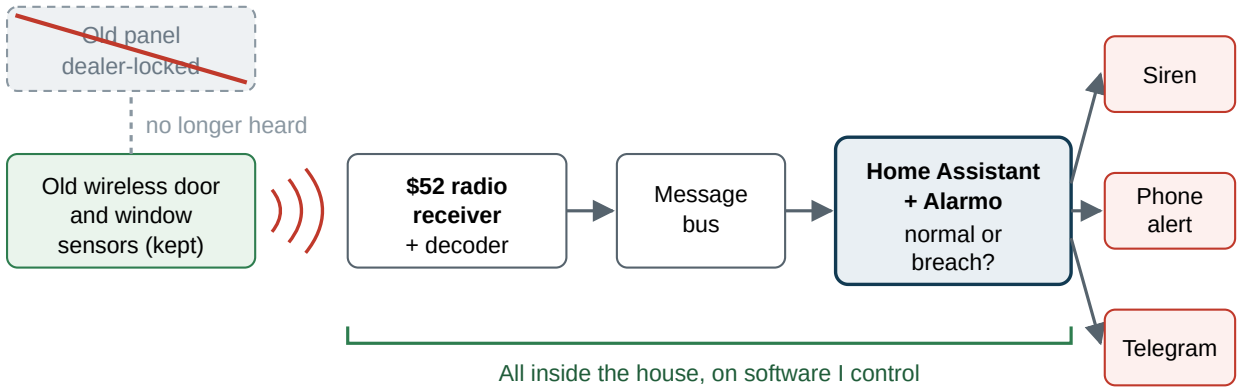


Figure 2. The revived signal path. Generic boxes on purpose, since the pattern is the story and the specifics stay home.

The live-fire test, June 2026

I consider a change finished when I have seen the result, so I tested the alarm by setting it off. I armed the house, opened a real door, waited out the entry countdown and listened for the siren while both phones received the alert. That first run found two faults that a review of the configuration had missed.

What the test showed	Cause, in plain terms	Fix and proof
The armed alarm ignored an opened door. Nothing happened at all.	A configuration gap left every sensor outside the alarm's zone, so the armed system was watching no sensors.	Every sensor assigned to the zone. Checked on disk, then armed and opened again. The countdown started.
The first seconds of the siren never played.	The speaker sleeps when idle, and its wake-up swallowed the start of the siren.	A quiet wake-up signal and a longer pause before full volume. Re-tested by ear with a clean start.

After both fixes the full chain passed. Arm, open, countdown, siren, phone alert, Telegram message, then disarm and silence.

Simple first, then build on trust

Version one did one job. Door and window sensors formed the armed layer, a person armed it by hand and a breach sounded the siren. Automation came only after that version had been heard firing and trusted.

ACT 2 - THE TALKING ALARM

4. When it trips, it talks back

A siren tells an intruder that an alarm went off. I wanted the house to tell them they had been seen. In September 2026 the alarm gained a voice. It looks at the camera facing the opened door, describes what the person is wearing and says it to them over the siren.

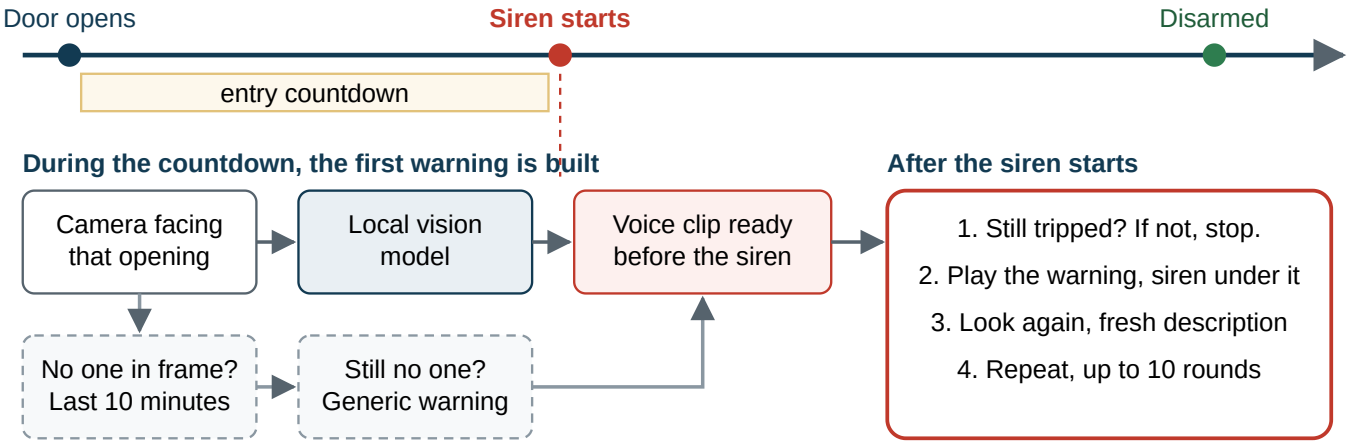
1. Door opens
An armed door or window opens and the entry countdown starts.

2. AI looks
The service picks the camera facing that opening and grabs a frame.

3. Describes
A local vision model completes "You are wearing ...".

4. Builds
A deep voice speaks the line, with the siren mixed underneath.

5. Talks back
It plays on the main speaker and loops until disarmed.



Pre-building during the countdown cut the gap from siren to voice from 13 seconds to 0.

Disarming stops the loop and puts every speaker back at its previous volume.

Figure 3. How a camera frame becomes a spoken warning. The camera image never leaves the house.

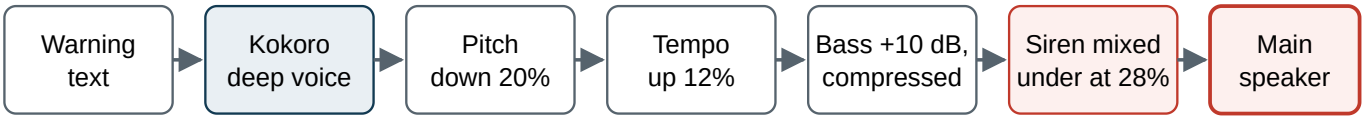
What the intruder hears

> *"I see you. You are wearing a gray t-shirt and black pants, holding a phone in your right hand. I saw you walk in and you are on camera right now. ... I know exactly what you look like. You are wearing [the same description]. Get out!"*
>
> Clean excerpt from the final rehearsal, September 2026, when I walked in on the armed house myself. The full script is blunter.

Why a description, not a recording

A canned warning sounds like every other alarm. A warning that names the intruder's clothes proves a camera saw them and that a record exists. It also refreshes each round, so a person who moves deeper into the house hears a new look at them. The full script is blunt, in language no burglar will mistake for a doorbell.

Making the voice sound like a threat



Open-source text-to-speech, then audio processing, so the voice carries over the siren instead of fighting it.

Figure 4. The voice chain. Every processing step runs on my home workstation, as does the vision model (served by Ollama).

5. Safeguards - who may arm, who may disarm and what keeps the AI in its lane

An alarm that talks and takes orders by text and voice needs hard limits on authority. I set the rules. Code enforces them, not the AI's judgment.

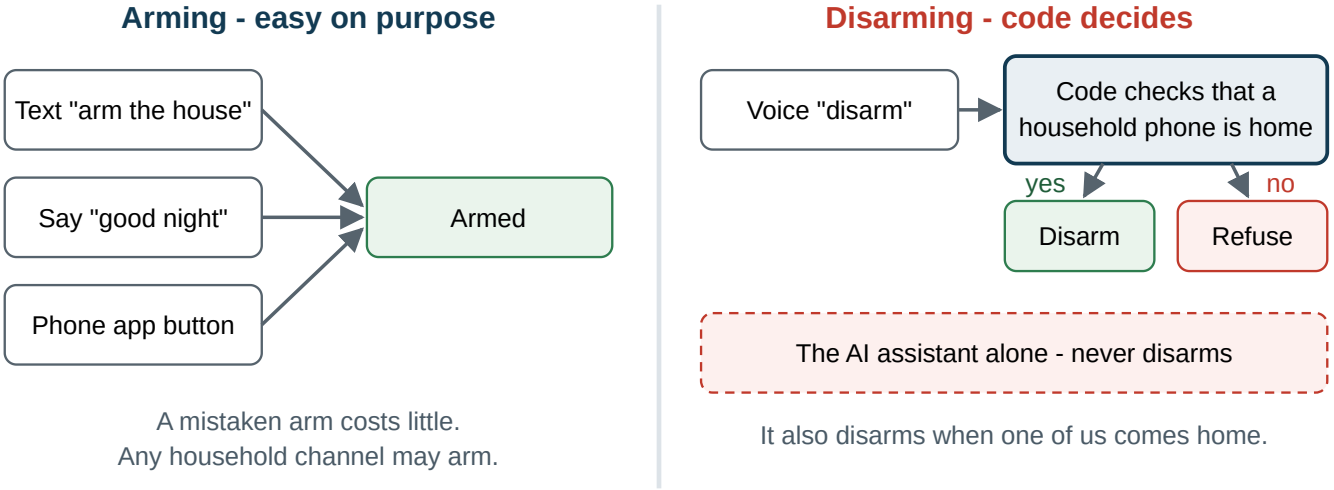


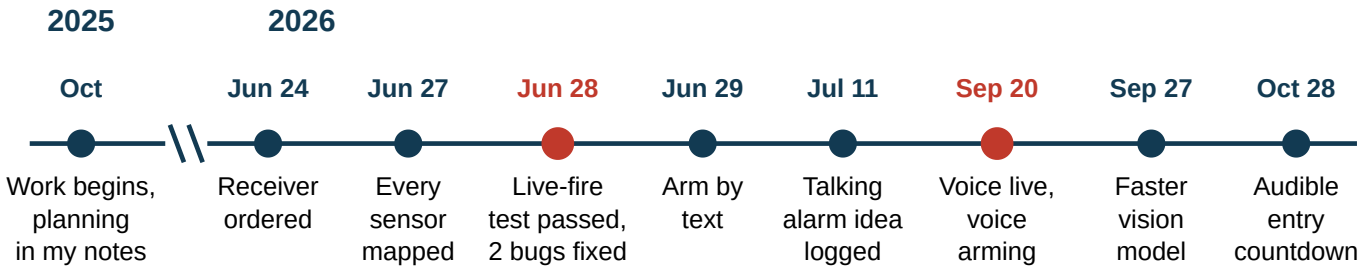
Figure 5. Arming is easy because a mistaken arm is cheap. Disarming runs through code the AI cannot override.

1. **The AI never disarms on its own judgment.** It disarms only when one of us asks, and separate code decides whether the request is allowed.
2. **Voice disarm needs proof.** A voice request disarms only when the code confirms a household phone is home. Otherwise the script refuses, whatever the AI thinks of the request.
3. **The siren never depends on the AI.** The voice service takes the speaker only once its first clip is ready. If it is down, the home-automation system plays the plain siren on its own.
4. **No invented descriptions.** The vision model must answer "NO PERSON" when nobody is visible, and the voice then uses a generic warning instead of guessing.
5. **It stops when told.** The loop checks the alarm state before every round and again just before playing. It is capped at 10 rounds, and disarming restores every speaker to the volume it had before.
6. **Camera images stay home.** The vision model runs on my own workstation. No frame goes to a cloud service to be described.
7. **Secrets stay out of the code.** Passwords and access keys live in protected files on the workstation, and the code reads them by location at run time. The house rule is that a secret is never written into code, notes or a chat.
8. **Every change followed one discipline.** A written brief (what, why, risk, undo and blast radius) approved before the change, a backup first, one change at a time, then verification by outcome.

Escape hatches before more automation

Before the alarm gained any automation beyond a manual arm, I had every way to silence it confirmed working, and confirmed that the siren stops on its own if nobody responds. An alarm that cannot be stopped is a hazard to the people it protects.

6. Results, status and what is next



Four days from receiver order to a passed live-fire test. The voice followed once the alarm had earned trust.

Figure 6. Project timeline, October 2025 to October 2026.

Item	Status
Revived alarm	In service since June 28, 2026, after the live-fire test passed and both bugs were fixed and re-tested.
Arming by text	Live since June 29, 2026. A texted "arm the house" arms it.
Talking alarm and voice arming	Live since September 20, 2026. In the final rehearsal I armed the house by text, walked back in and heard a description of my own clothes the moment the siren started.
Vision model	Replaced September 27, 2026, by a smaller and faster model after a side-by-side test on saved camera frames.
Entry countdown	Since October 28, 2026 the countdown beeps like a standard panel, so anyone walking in hears the clock. I added it after a false alarm on our own return.
Next (designed, not built)	On a trigger, turn on a television inside the house and loop the intruder's own camera clip, so they see that they were seen.

7. What this project shows about how I work

Reuse the installed base before buying new. The house already owned a working sensor network. The vendor's answer was to replace it. A $52 receiver and open-source software returned it to service. Agencies face the same choice with legacy systems every budget cycle, and the cheapest working option is often the one already in the walls.

Accept work by outcome, not by paperwork. The configuration looked correct on review. Only setting the alarm off showed that it was watching no sensors at all. I treat a live test as the acceptance step for any program I run.

Put AI to work, and put limits on its authority. AI agents wrote the code and an AI model describes the intruder, but the rules about who may disarm the house live in code the AI cannot override. That is the posture I bring to AI adoption in any organization. Let it do the work, and decide in advance where its authority stops.

How it was built

I designed it, set the rules, approved every change and ran the tests. AI coding agents (Claude) wrote the code and configuration. Hardware, sensor and camera details are left out on purpose.