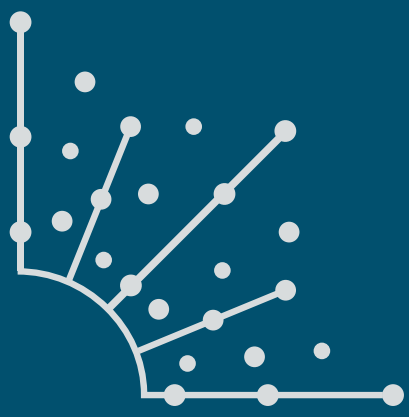
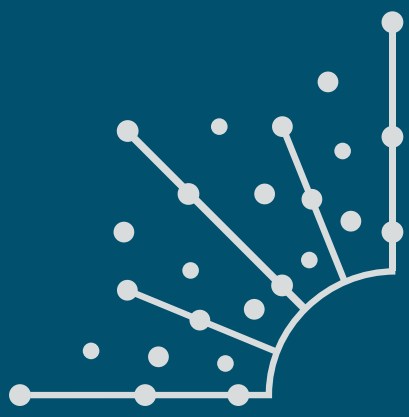
A practical guide to home automation using open source tools





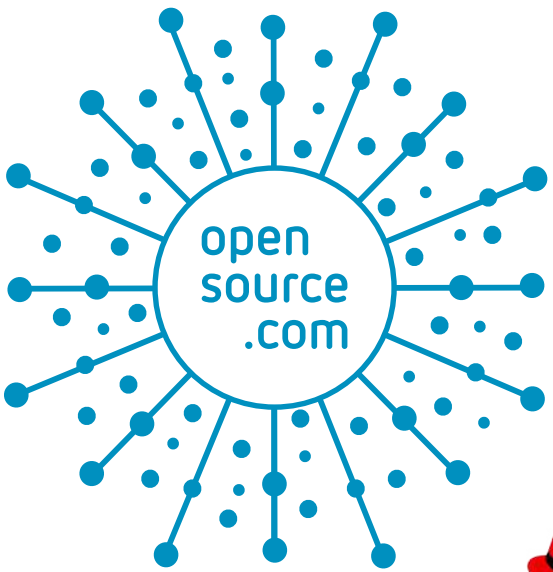
ABOUT OPENSOURCE.COM

What is Opensource.com?

OPENSOURCE.COM publishes stories about creating, adopting, and sharing open source solutions. Visit Opensource.com to learn more about how the open source way is improving technologies, education, business, government, health, law, entertainment, humanitarian efforts, and more.

Submit a story idea: opensource.com/story

Email us: open@opensource.com



Supported by
Red Hat

STEVE OVENS

STEVE OVENS is a dedicated IT professional and Linux advocate. Prior to joining Red Hat, he spent several years in financial, automotive, and movie industries. Steve currently works for Red Hat as an OpenShift consultant and has certifications ranging from the RHCA (in DevOps), to Ansible, to Containerized Applications and more. He spends a lot of time discussing technology and writing tutorials on various technical subjects with friends, family, and anyone who is interested in listening.

Follow me at @linuxovens

CHAPTERS

Why I use Home Assistant for open source home automation

Home automation can be a slippery slope. The right open source tools can get you on firmer footing.

HOME AUTOMATION is a slippery slope; you have been warned! In this multipart series, I will discuss home automation using the open source project Home Assistant. This introductory article will cover my journey to Home Assistant [1], what the application does, and why it's important.

How my journey began

Some time ago, when I set out on this journey, my goal was not lofty. I was solving a need. You see, I have a fairly sizable homelab [2]. Nothing on the scale of some notable YouTubers, but I have eight machines ranging from 16GB RAM all the way up to 96GB. I have a Netgear 10G Ethernet switch as the backbone of my networking infrastructure. However, I have a small problem. Every once in a while, this switch's state table fills up, and then it crashes, taking the network with it. This is a known issue with this model (although it was not known to me ahead of time). The only way to resolve the issue, without replacing the switch, is to power it off for a few seconds and then power it back on.

This would not seem like a big deal. Especially because I live in an apartment, so I don't even have to travel up or down a flight of stairs to do this. However, I work for Red Hat as a senior OpenShift consultant, which means I am often on the road (at least pre-COVID). I use my lab almost every day for work-related activities, and my family uses the network as well for things like playing games, watching our Blu-ray collection, etc. So, it's a giant pain to have the network go down when I don't have physical access. My solution was to get a smart plug and join it to a completely different network with nothing else on it, completely firewalled off from other equipment. If the Netgear switch needs to be rebooted, I should be able to access the smart plug remotely to reboot the switch.

My journey away from the cloud

Being privacy-conscious, perhaps bordering on the tinfoil hat breed, I am immediately uncomfortable with a "cloud"-connected anything. We run Plex [3], Kodi [4], Nextcloud [5], and a host of other services because, well, I am "that guy." But I'm not "anti-cloud." In fact, a large part of my day job is working with the big three: AWS, Google Compute, and Azure. But when it comes to standing up services that I rely on, I have an almost irrational need to host things locally.

After doing a significant amount of digging and speaking to my coworker Alex Kretzschmar (who also hosts the Self-Hosted [6] podcast), I discovered alternative firmware projects, such as Tasmota [7] and ESPHome [8], available for certain wireless chipsets. I'll cover the different chipsets and protocols (Zigbee, WiFi, Z-Wave, and the like) in a future article, but suffice it to say, you are not stuck buying products that are dependent on the cloud. There are online stores such as CloudFree [9] that sell devices pre-flashed with Tasmota. There are even companies like Shelly [10] that produce high-quality products with an optional cloud component, but the buyer maintains local control.

Anyway, back to my story. I bought a plug, flashed the firmware to run Tasmota, and remotely power-cycled my switch on occasion, and that was the end of it, right? Well, if it was, I imagine I would not be writing this article. I currently have 43 Internet of Things (IoT) devices—from sensors I built to smart bulbs, smart LED strips, infrared blasters, and more. Remember how I said home automation is a slippery slope?

Did I mention that I have had virtually no experience with soldering, electronic theory/repairs, or anything remotely related to home automation? In fact, outside of Linux and related technology, I am one of the least "handy" people I know. Sure, I can punch a hole in a wall or put some screws in a board, but up to the age of 30, the only tools that I ever owned were a Dremel and some screwdrivers for installing computer components.

Why mention this? So you understand exactly how far behind square one I was when I started. If I can take this on, so can you!

Centralized local control

Great, I have a ton of IoT devices, and I can toggle things on and off from a phone. For a while, I was satisfied with simply calling up the Tasmota web UI and using its sparse but functional controls.

Even my wife, God bless her willingness and patience

(Steve Ovens, CC BY-SA 4.0)