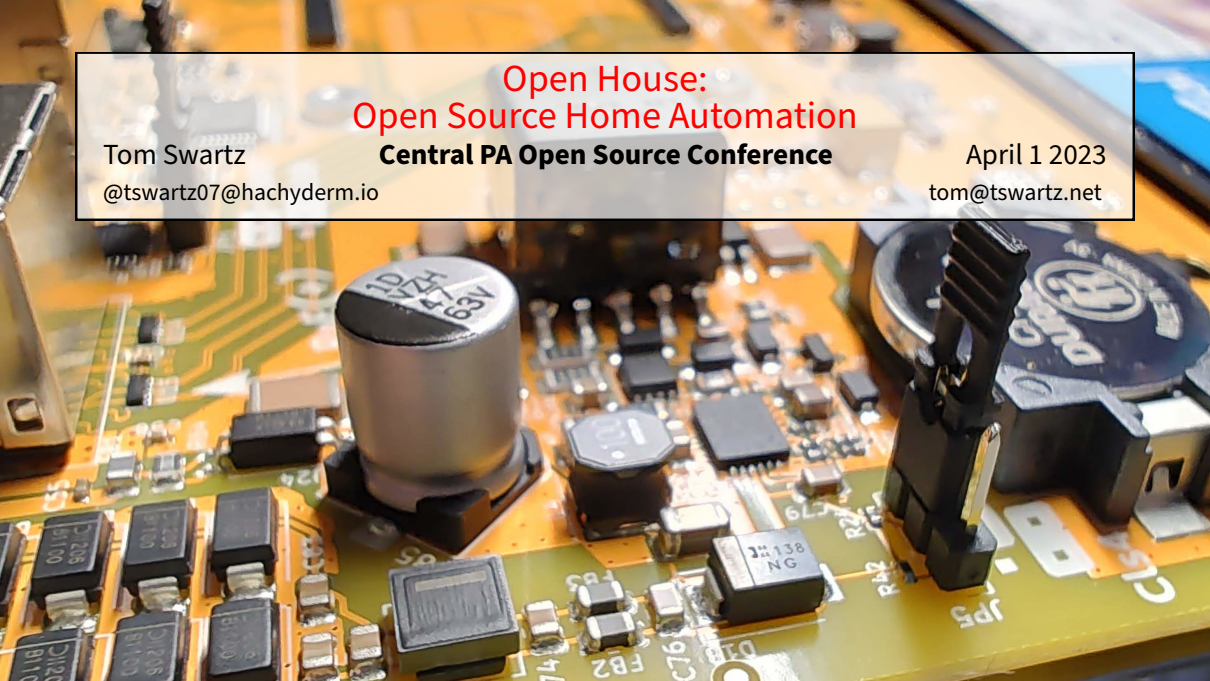


Open House:
Open Source Home Automation

Tom Swartz
@tswartz07@hachyderm.io

Central PA Open Source Conference

April 1 2023

tom@tswartz.net

Who is this guy?

And what is he doing here?

Tom Swartz

- Crunchy Data
  Associate Director of Support

Who is this guy?

And what is he doing here?

Tom Swartz

- Crunchy Data
  Associate Director of Support
- Hobbyist interests in ‘Big Data’, IoT, RF Communications

Who is this guy?

And what is he doing here?

Tom Swartz

- Crunchy Data
  Associate Director of Support
- Hobbyist interests in ‘Big Data’, IoT, RF Communications
- Prior CPOSC talks on RF comms/SDR & homemade PCBs

Who is this guy?

And what is he doing here?

Tom Swartz

- Crunchy Data
  Associate Director of Support
- Hobbyist interests in ‘Big Data’, IoT, RF Communications
- Prior CPOSC talks on RF comms/SDR & homemade PCBs
- I’ve got some pretty specific hyperfixations

Summary

We'll be reviewing the following topics, and how they relate to Home Automation

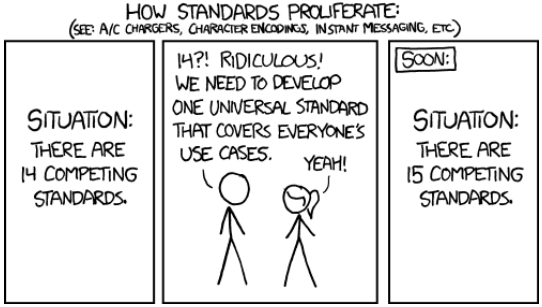
1. Home Assistant
2. ESPHome
3. Building your own Sensors

Subsection 1

Home Assistant

Why Home Assistant?

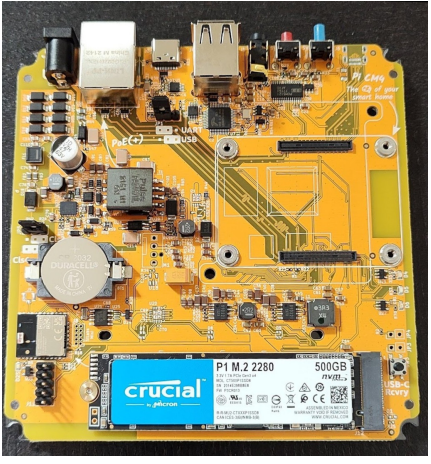
- There are a million apps for every ‘Smart’ thing (sometimes multiple for the same item!)
- A practical application for consolidating software, apps, and tool-sets that many people already use
- Needs for better usability, security, and maintenance
- A solid use case for HomeLab equipment, for nerd cred
- It’s Open Source!



Installation Methods

1. Home Assistant Yellow
2. Docker Container
3. Dedicated Raspberry Pi (Good luck finding one!)
4. Running as a background app on other hosts/servers

Worth noting that certain install methods lack certain features (such as third party add-ons, Zigbee support, etc)



Components and Overview

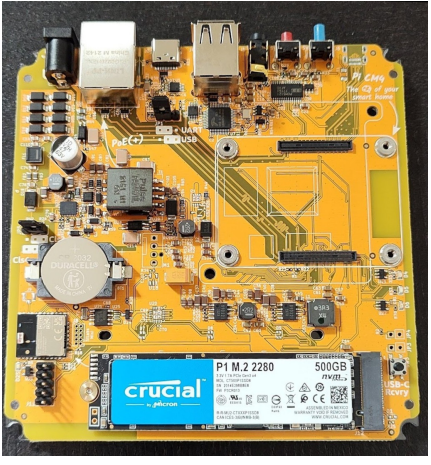
Home Assistant is made of several parts:

1. Dashboards
2. Integrations
3. Devices and Entities
4. Automations

Dashboards

Home Assistant's way of showing what's happening

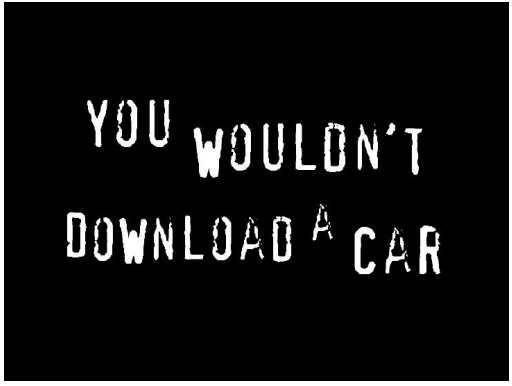
1. Per-User Dashboards
2. Show live and historical data
3. Completely customizable, and totally optional



Integrations

Arguably, the thing that makes Home Assistant so lucrative

1. All devices are community-driven
2. 2424+ total integrations supported
3. Everything from 3D Printers to Vacuum cleaners

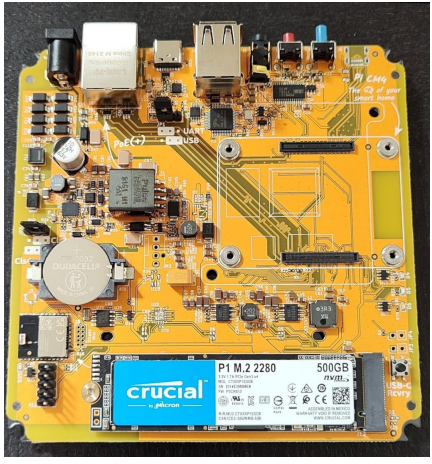


Devices and Entities

The nuts and bolts of Home Assistant

Devices A physical (or logical) group of sensors and inputs. For example, a motion sensor is a device

Entities Any type of ‘data point’ for a Device. For example, a motion sensor’s *Entities* can be: ‘Motion Status’, ‘Battery Level’, ‘Light Level’



Automations

Home Assistant's key features are Automations

Trigger	Actions which will start the Automation action
Conditions	Optional conditions which limit when/where/how the automation will run
Actions	Items to perform if conditions are met

It's kind of like *cron*, but for Internet of Things!

Example Automations

Home Assistant performs Automations which allow for endless possibilities, but here are some examples:

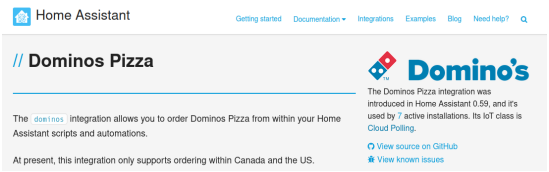
Christmas Lights Turn on all of the Christmas lights 30 minutes before the sun sets

Extra doorbell coverage Home Audio speakers announce if the doorbell is ringing

Never forget to close the garage Automatically close the garage doors if no one is home

Fix a dark office Turn on office lamps if it gets cloudy/dark

So that means…



Section 2

Security

Subsection 1

The Elephant in the Room

Subsection 2

How does it compare?

How does it compare to others?

General IoT Apps

- Almost all depend on external, third-party servers
- Services can be discontinued, making certain devices useless
- Has the potential for information gathering

Home Assistant

- The majority of the service is hosted locally, no cloud connection
- Support for services can continue after the Manufacturer ends support
- Can be completely firewalled off, in certain cases

Integration Classifications

Classifying the Internet of Things

Local Push	Gold Standard, entirely local control directly from Home Assistant
Local Poll	Home Assistant direct read/write, but on-device updates may be delayed
Cloud Push	Integration of this device happens via the cloud and requires an active internet connection
Cloud Poll	Delayed integration of this device happens via the cloud and requires an active internet connection

Subsection 1

Demo of Home-Assistant

Section 4

Custom Sensors

Subsection 1

ESPHome

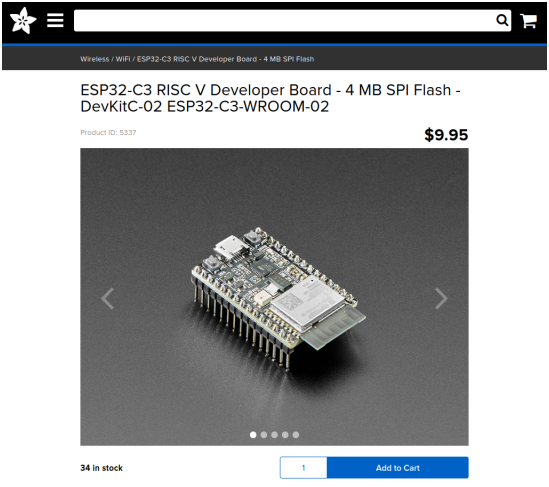
What is ESPHome?

- Sponsored and directed by same group as HomeAssistant
- Tight integrations with HomeAssistant
- Easy to set up and use
- Allow for specific, inexpensive, widespread sensors

Building Sensor Devices: The Controller

ESPHome runs on easily obtained microcontroller hardware.
Typical cost for an entire USB-powered sensor can be as low as $5

1. ESP32
2. ESP8266
3. Raspberry Pi RP2040 (*Slightly* easier finding one!)



Building Sensor Devices: The Sensors

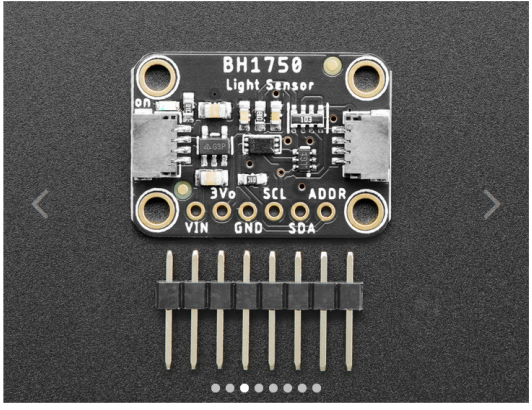
Types of common ‘sensor components’:

- Air Quality
- Digital Signals
- Distance
- Electricity/Power
- Environmental
- Light
- Magnetic
- Motion
- Weight

Adafruit BH1750 Light Sensor - STEMMA QT / Qwiic

Product ID: 4681

$4.50



Building Sensor Devices: Putting It All Together

A typical ESPHome config contains the following sections:

1. Device name
2. Board/Controller Definition
3. WiFi Info
4. Sensor Connection Setup
5. Sensor Definition

Example code configuring a custom light level sensor:

```
esphome:
  name: luminance
esp32:
  board: esp32doit-devkit-v1
  framework:
    type: arduino
api: # Enables connection to Home Assistant API
  encryption:
    key: !secret haas_encryption_key
ota: # Allows over-the-air updates
  password: !secret ota_password
wifi:
  networks:
  - ssid: !secret wifi_ssid_main
    password: !secret wifi_password_main
i2c: # I2C setup for HB1750 device
  sda: GPIO21
  scl: GPIO22
sensor: # https://esphome.io/components/sensor/bh1750.html
- platform: bh1750
  name: "BH1750 Illuminance"
  address: 0x23
```

Subsection 1

Demo of ESPHome

Luminance Driven Light Switch

Home Assistant performs an Automation which does the following:

1. Trigger
   1. 1.5 hours before the sun sets
   2. The time is 8am Eastern
   3. The BH1750 light sensor triggers an event
2. Conditions
   1. Time between 8am and 5pm Eastern, on a weekday
   2. The BH1750 sensor has been ‘dim’ for 15 minutes
   3. My lamps are already off
3. Actions
   1. Turn on Office Table Lamp
   2. Turn on Desk Lamp

Section 6

Wrapping Up

Key Takeaways

- Home Assistant is a pretty neat central, locally controlled, home automation system

Key Takeaways

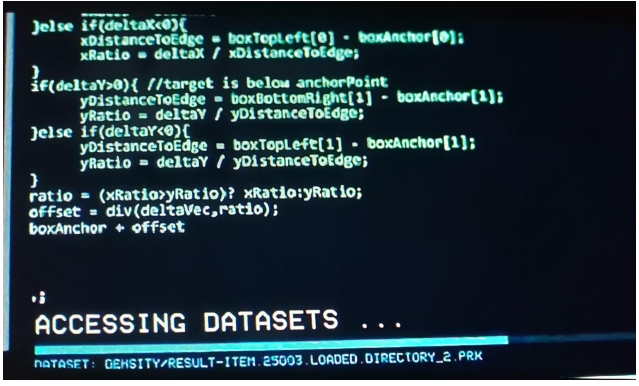
- Home Assistant is a pretty neat central, locally controlled, home automation system
- Creating your own sensors and doing useful things with them is very easy and inexpensive

Key Takeaways

- Home Assistant is a pretty neat central, locally controlled, home automation system
- Creating your own sensors and doing useful things with them is very easy and inexpensive
- Excellent way to ensure your devices ‘just work’ and continue to stay that way

Thank You

Questions?



References

Code for this presentation, as well as some sample device manifests and other information can be found on GitHub and GitLab.

- https://github.com/tomswartz07/CPOSC2023
- https://gitlab.com/tom.swartz07/CPOSC2023

Further Reading:

1. https://www.home-assistant.io/
2. https://esphome.io/
3. https://www.troyhunt.com/ IoT Unravelled Series