Build Your Own Smart Home with Home Assistant

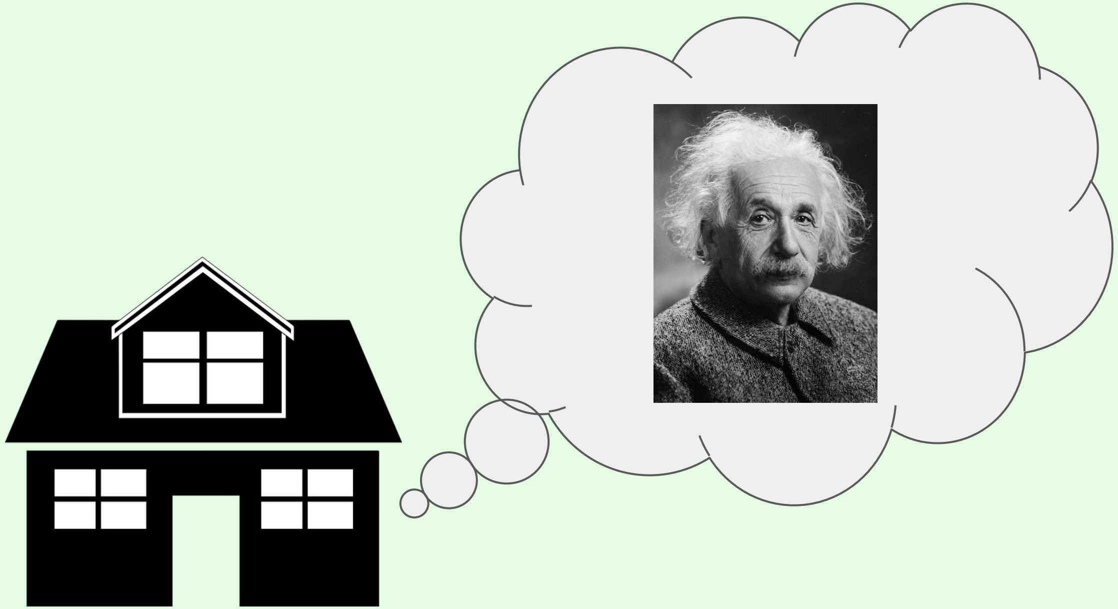
Corey Edwards

OpenWest Conference

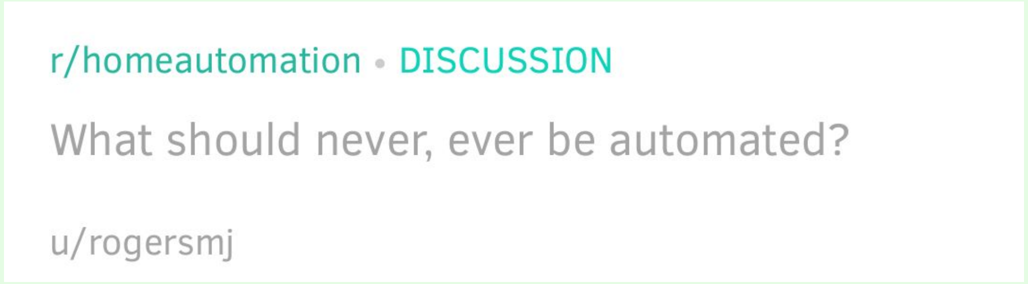
June 8, 2018

v1.0 Updated: 2018-06-03

Why A Smart Home?



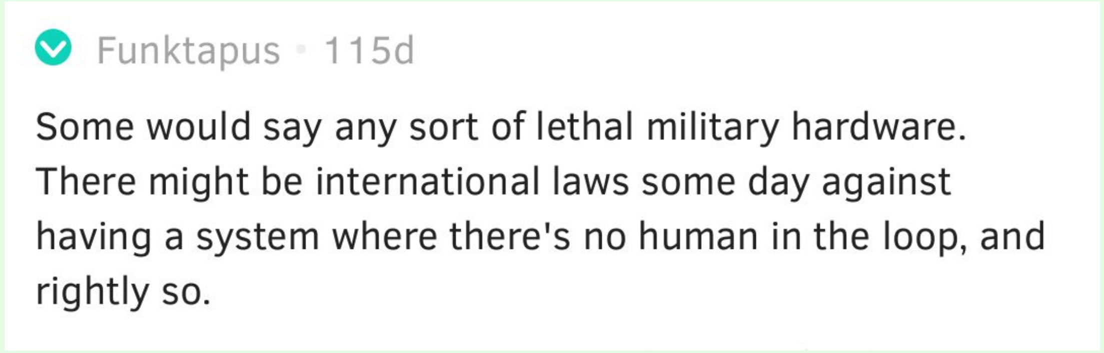
First, Why Not A Smart Home?



r/homeautomation • DISCUSSION

What should never, ever be automated?

u/rogersmj



Funktapus • 115d

Some would say any sort of lethal military hardware. There might be international laws some day against having a system where there's no human in the loop, and rightly so.

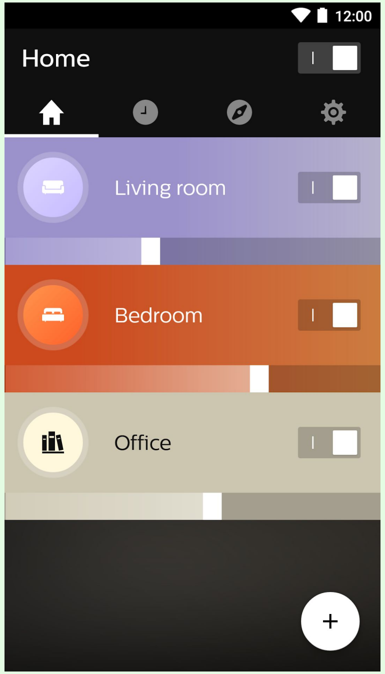
No Seriously, Why Not A Smart Home?

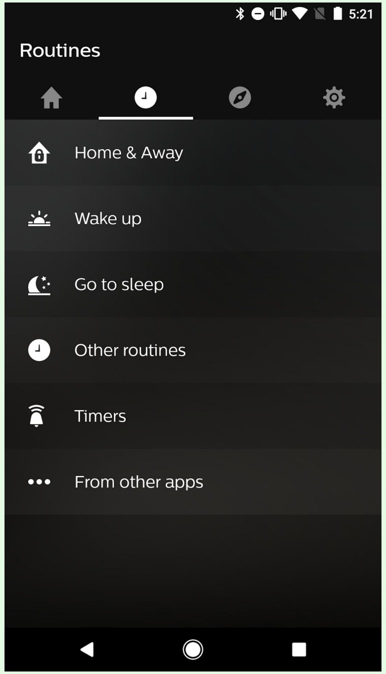
Expensive

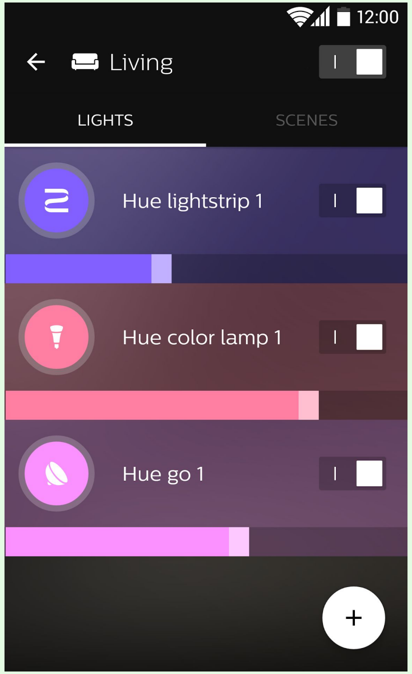
Very early tech (read: buggy)

Why A Smart Home?

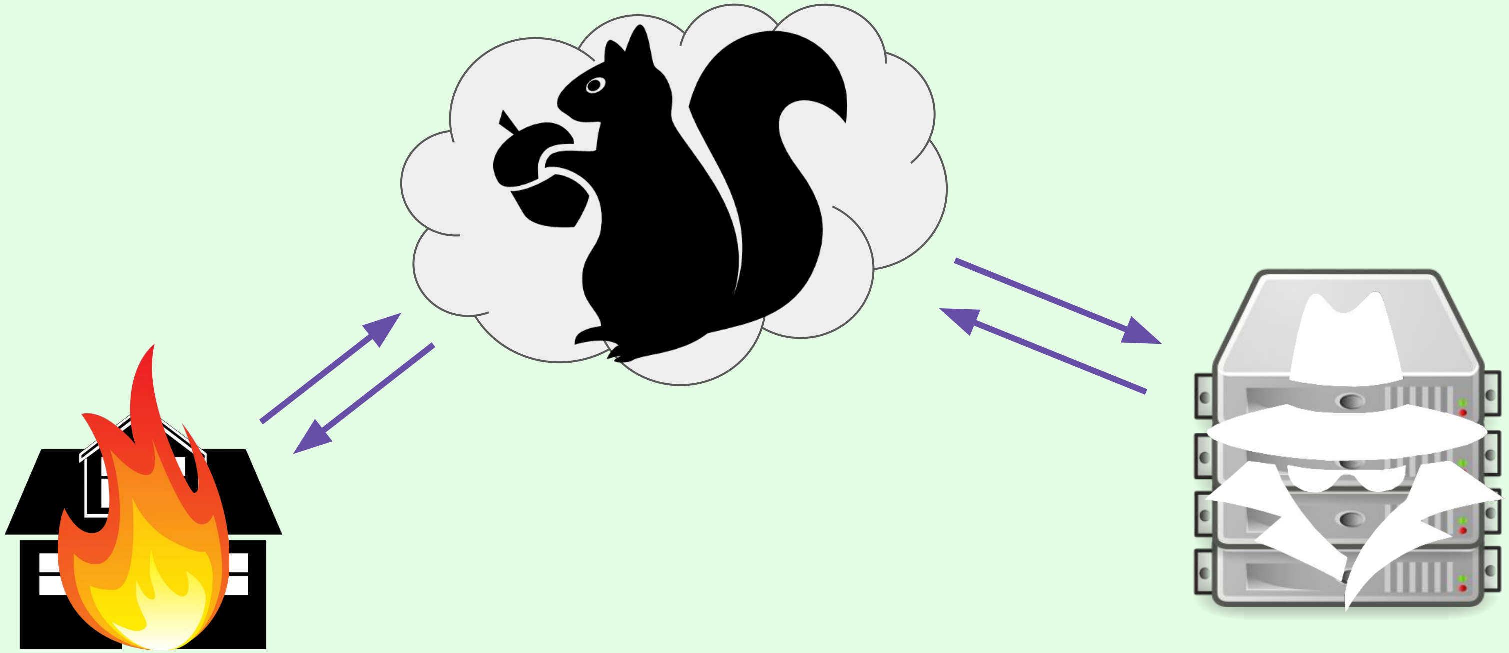
An App For Everything

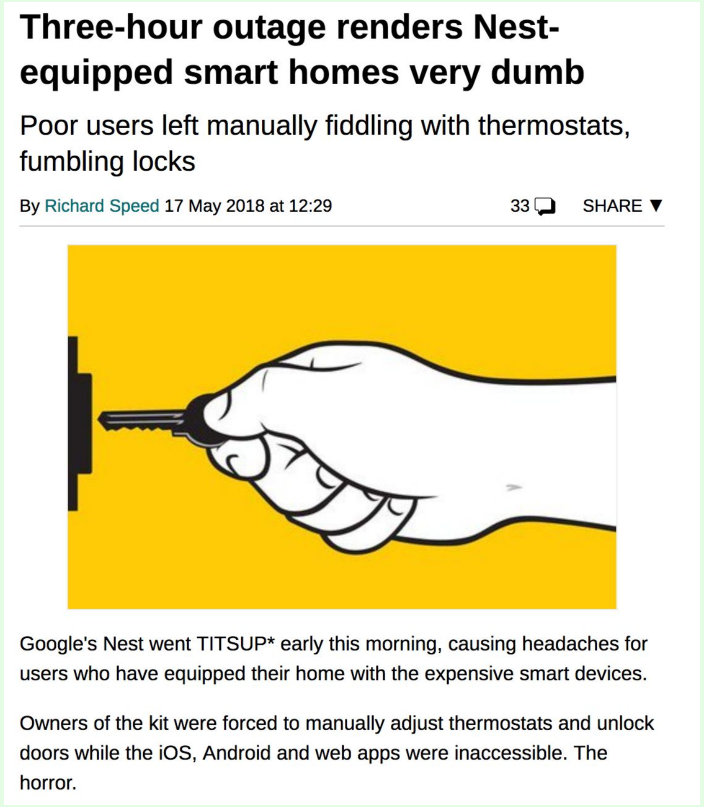






An Internet App For Everything

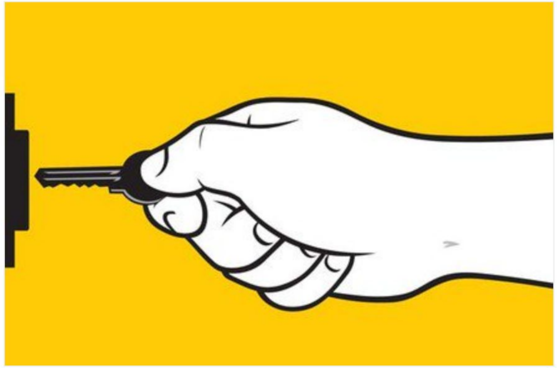




Three-hour outage renders Nest-equipped smart homes very dumb

Poor users left manually fiddling with thermostats, fumbling locks

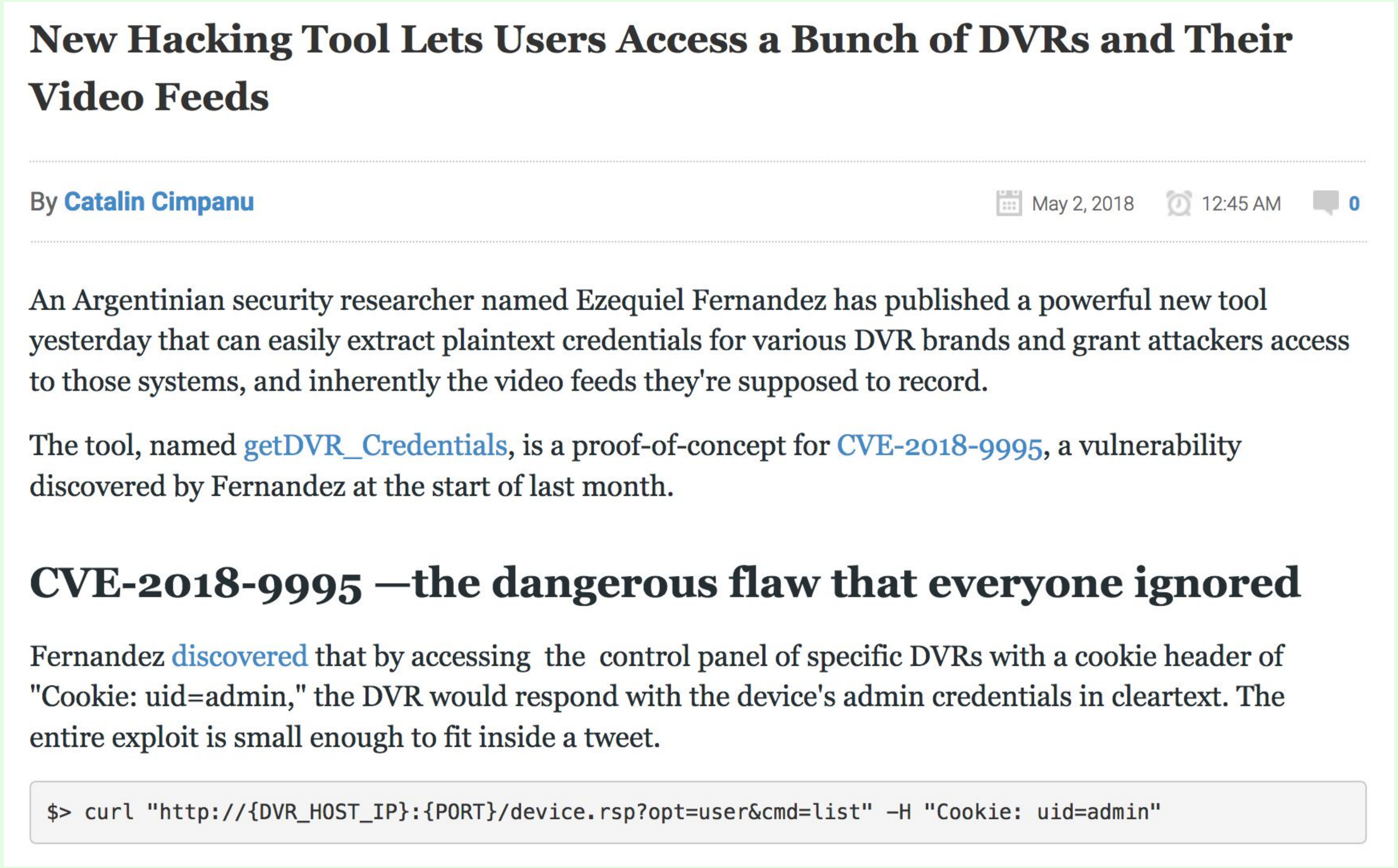
By Richard Speed 17 May 2018 at 12:29 33 SHARE ▼



Google's Nest went TITSUP* early this morning, causing headaches for users who have equipped their home with the expensive smart devices.

Owners of the kit were forced to manually adjust thermostats and unlock doors while the iOS, Android and web apps were inaccessible. The horror.

* Toasted Infrastructure Totally Stops Unlocking Properties



New Hacking Tool Lets Users Access a Bunch of DVRs and Their Video Feeds

By Catalin Cimpanu May 2, 2018 12:45 AM 0

An Argentinian security researcher named Ezequiel Fernandez has published a powerful new tool yesterday that can easily extract plaintext credentials for various DVR brands and grant attackers access to those systems, and inherently the video feeds they're supposed to record.

The tool, named getDVR_Credentials, is a proof-of-concept for CVE-2018-9995, a vulnerability discovered by Fernandez at the start of last month.

CVE-2018-9995 —the dangerous flaw that everyone ignored

Fernandez discovered that by accessing the control panel of specific DVRs with a cookie header of "Cookie: uid=admin," the DVR would respond with the device's admin credentials in cleartext. The entire exploit is small enough to fit inside a tweet.

```
$> curl "http://{DVR_HOST_IP}:{PORT}/device.rsp?opt=user&cmd=list" -H "Cookie: uid=admin"
```

No More Silos

Home Assistant is

- Home automation hub
- Written in Python
- Depends on PyPI modules
- Web based
- 1084 modules (as of 0.70)
- Releases every 2-3 weeks
- 100% Open Source

Installation Options

1. Python virtual env
2. Virtual Machine
3. Docker
4. Hassbian
5. Hass.io (HassOS)

Virtualenv Install

```
$ python3 -m venv homeassistant
$ cd homeassistant
$ source bin/activate
$ python3 -m pip install wheel
$ python3 -m pip install homeassistant
$ hass --open-ui
```

Do You Love YAML?

```yaml
homeassistant:
  # Name of the location where Home Assistant is running
  name: Home
  # Location required to calculate the time the sun rises and sets
  latitude: !secret home_lat
  longitude: !secret home_long
  # Impacts weather/sunrise data (altitude above sea level in meters)
  elevation: 1428
  # metric for Metric, imperial for Imperial
  unit_system: imperial #sadface
  # Pick yours from here: http://en.wikipedia.org/wiki/List_of_tz_database_time_zones
  time_zone: America/Boise

recorder:
  purge_keep_days: 10
  exclude:
        domains:
        - automation
        - weblink
        - updater
        - group
        - zwave
        - media_player
```

Secrets

```
configuration.yaml

mqtt:
  broker: !secret mqtt_hostname
  port: 1883
  username: !secret mqtt_username
  password: !secret mqtt_password
```

```
secrets.yaml

mqtt_username: joe_user
mqtt_password: mysecretpassword
mqtt_hostname: mqtt.example.com
```

Including Files

configuration.yaml

```yaml
sensor: !include sensor.yaml
```

sensor.yaml

```yaml
- platform: sun
- platform: moon
- platform: season
  type: meteorological
- platform: time_date
  display_options:
    - 'time'
    - 'date'
```

Remote Access

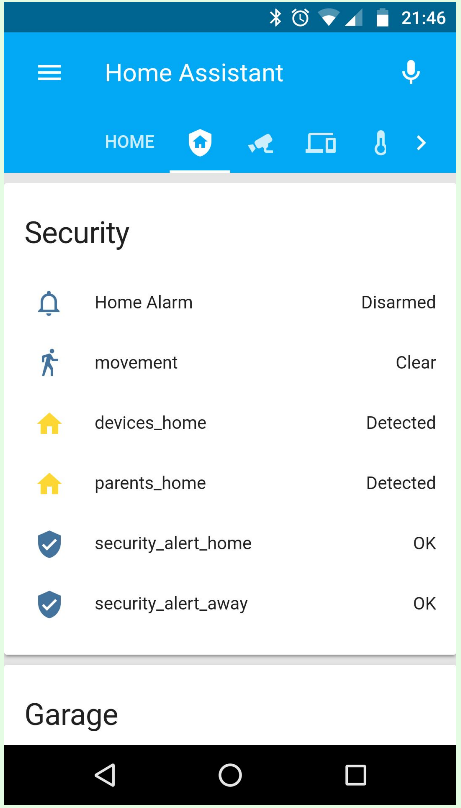
- Port forwarding
- VPN
- Tor
- MQTT
- Reverse Proxy (Apache, NGINX)
- TLS
- … or none

Apache Reverse Proxy

```
ProxyPreserveHost On
ProxyPass "/.well-known" !  # for Let’s Encrypt
ProxyPass "/" "http://192.0.2.1"
ProxyPassReverse "/" "http://192.0.2.1/"
ProxyPass "/api/websocket" "ws://192.0.2.1/api/websocket"
ProxyPassReverse "/api/websocket" "ws://192.0.2.1/api/websocket"

RewriteEngine on
RewriteCond %{HTTP:Upgrade} =websocket [NC]
RewriteRule /(.*)  ws://192.0.2.1/$1 [P,L]
RewriteCond %{REQUEST_URI} !^/.well-known  # again, for Let’s Encrypt
RewriteCond %{HTTP:Upgrade} !=websocket [NC]
RewriteRule /(.*)  http://192.0.2.1/$1 [P,L]
```

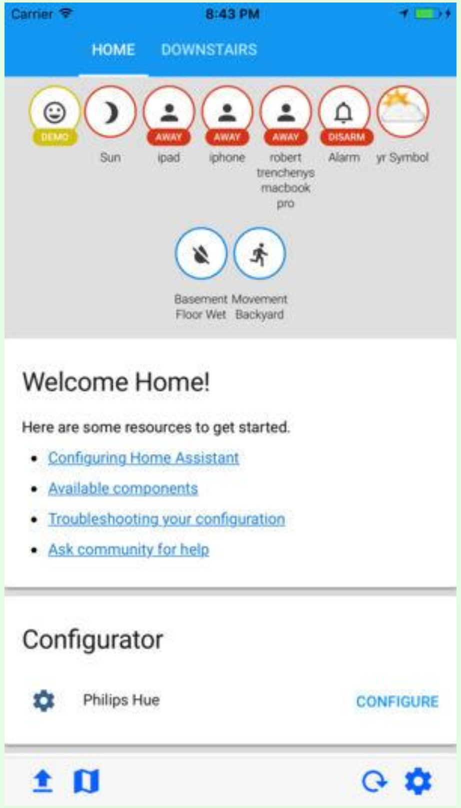
Mobile Apps



Security

Home Alarm	Disarmed
movement	Clear
devices_home	Detected
parents_home	Detected
security_alert_home	OK
security_alert_away	OK

Garage



Welcome Home!

Here are some resources to get started.

- Configuring Home Assistant
- Available components
- Troubleshooting your configuration
- Ask community for help

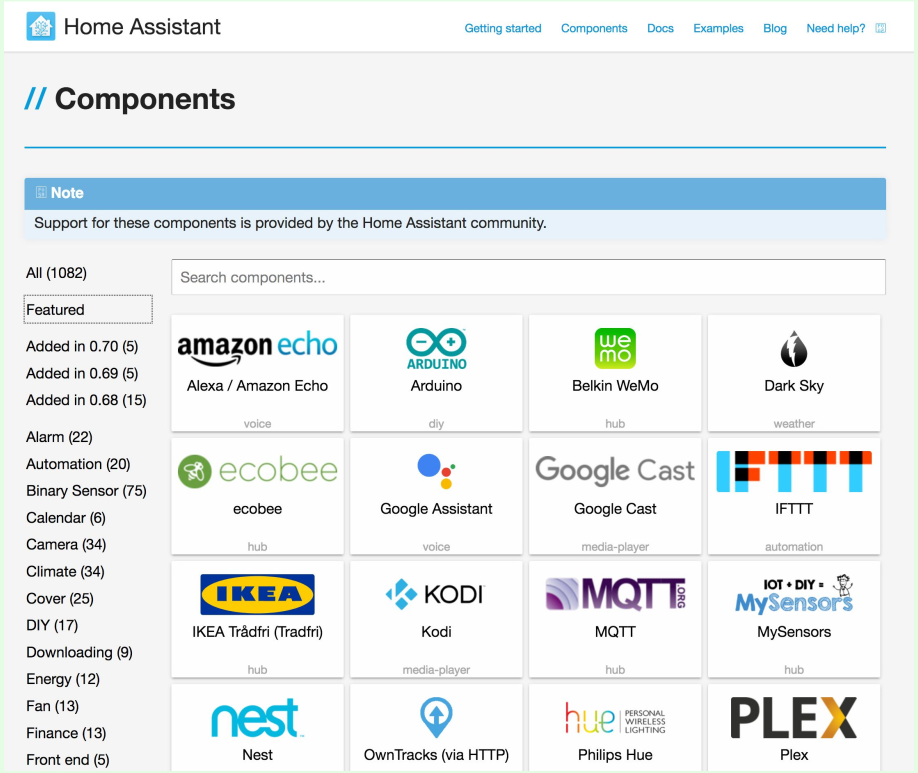
Configurator

Philips Hue CONFIGURE

Basic Sensor Types

- Sensor
  - Has a non-discrete value
  - Could be a string, int, float
  - Examples
    - Temperature
    - Weather forecast
- Binary Sensor
  - Either true or false, on or off
  - Examples
    - Door or window
    - Motion
- Cover
  - Window shades
- Switch
  - Either on or off
  - Can be controlled
  - Examples:
    - Light switch
    - Camera recording control
- Media Player
  - Play, pause, stop
  - Can report metadata about media
- Light
  - On or off
  - Can have a brightness
  - Can have a color

Home Assistant Components



Home Assistant

Getting started Components Docs Examples Blog Need help?

// Components

Note

Support for these components is provided by the Home Assistant community.

All (1082)

Featured

Added in 0.70 (5)
Added in 0.69 (5)
Added in 0.68 (15)

Alarm (22)
Automation (20)
Binary Sensor (75)
Calendar (6)
Camera (34)
Climate (34)
Cover (25)
DIY (17)
Downloading (9)
Energy (12)
Fan (13)
Finance (13)
Front end (5)

Search components...

Alexa / Amazon Echo voice	Arduino diy	Belkin WeMo hub	Dark Sky weather
ecobee hub	Google Assistant voice	Google Cast media-player	IFTTT automation
IKEA Trådfri (Tradfri) hub	Kodi media-player	MQTT hub	MySensors hub
Nest	OwnTracks (via HTTP)	Philips Hue	Plex

https://www.home-assistant.io/components/

- Full list of all supported modules
- Grouped by type and category
- Searchable
- Links directly to full documentation

Templates

- Build your own sensors using data from other sensors
- Available for many types
  - Binary sensor
  - Cover
  - Fan
  - Light
  - Switch
  - Generic sensor
- Uses the Jinja2 template engine

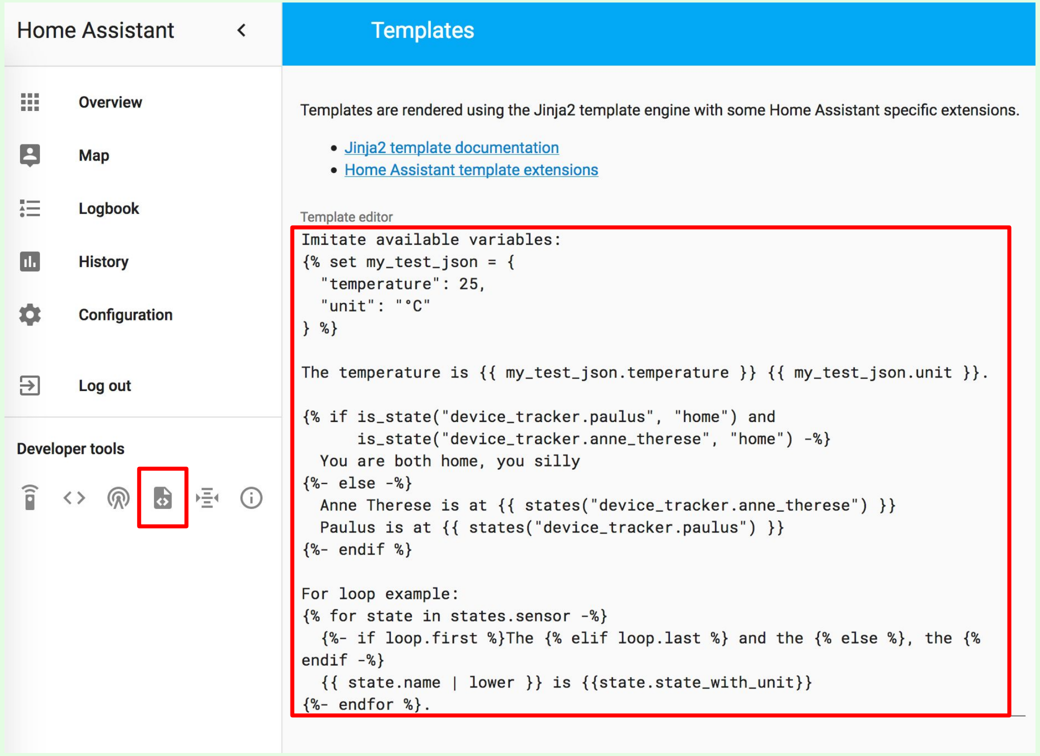
Binary Sensor Template Example

```
- platform: template
  sensors:
    sprinkler_is_rain_bypass:
      value_template: >-
        {{ states.sensor.rainfall_24h_mean.attributes["max_value"] > 0.5 }}
    door_open:
      device_class: door
      value_template: >-
        {{ is_state('binary_sensor.door_garage_rear', 'on')
          or is_state('binary_sensor.door_garage_front', 'on')
          or is_state('binary_sensor.front_door', 'on')
          or is_state('binary_sensor.door_back_deck', 'on')
        }}
    work_time:
      device_class: presence
      value_template: >-
        {% set timey = now().strftime("%H%M") %}
        {% if timey < "0800" or timey > "1700" %} False {% else %} True {% endif %}
```

Sensor Template Example

```
- platform: template
  sensors:
    battery_status:
      friendly_name: "Battery Status"
      value_template: >
        {% set value = 0 %}
        {% for zwave in states.zwave if zwave.attributes.battery_level %}
        {% if zwave.attributes.battery_level < 40 %}
        {% set value = 1 %}
        {% endif %}
        {% endfor %}
        {{ value }}
    dining_temp:
      friendly_name: Dining Room Temp
      unit_of_measurement: '°F'
      value_template: "{{ states('sensor.vision_zp3102_pir_motion_sensor_temperature') }}"
```

Template Testing



Templates are rendered using the Jinja2 template engine with some Home Assistant specific extensions.

- Jinja2 template documentation
- Home Assistant template extensions

```
Imitate available variables:
{% set my_test_json = {
  "temperature": 25,
  "unit": "°C"
} %}

The temperature is {{ my_test_json.temperature }} {{ my_test_json.unit }}.

{% if is_state("device_tracker.paulus", "home") and
      is_state("device_tracker.anne_therese", "home") -%}
  You are both home, you silly
{%- else -%}
  Anne Therese is at {{ states("device_tracker.anne_therese") }}
  Paulus is at {{ states("device_tracker.paulus") }}
{%- endif %}

For loop example:
{% for state in states.sensor -%}
  {%- if loop.first %}The {% elif loop.last %} and the {% else %}, the {%
endif -%}
  {{ state.name | lower }} is {{state.state_with_unit}}
{%- endfor %}.
```

DIY

Lots of options:

- Command Line Scripts
- REST API
- Python Module
- MySensors
- GPIO

Command Line

```
configuration.yaml

sensor:
  platform: command_line
  name: uptime
  command: /home/homeassistant/bin/uptime 1
```

```
bin/uptime

#!/bin/bash

case "${1}" in
  1)
    output=$(uptime | perl -lane 'print $F[9]')
  ;;
  5)
    output=$(uptime | perl -lane 'print $F[10]')
  ;;
  15)
    output=$(uptime | perl -lane 'print $F[11]')
  ;;
esac

echo "${output}"
```

REST

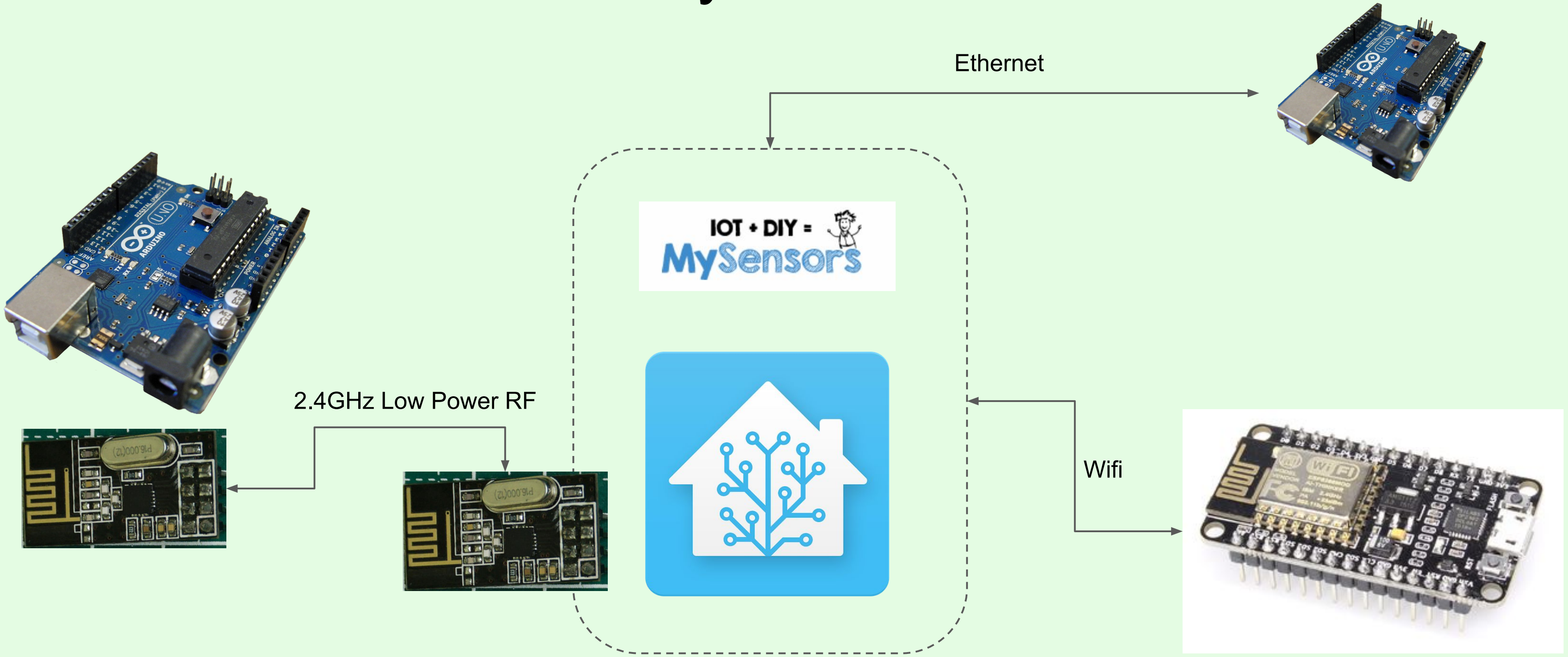
configuration.yaml

```yaml
sensor:
  platform: rest
  name: uptime
  resource: https://192.0.2.10/api/status
  value_template: ‘{ value_json.five }’
```

sample json

```json
{
  “one”: 1.2,
  “five”: 1.5,
  “fifteen”: 1.6
}
```

MySensors



MySensors HA Configuration

```
gateways:
  - device: ‘192.0.2.128’
    tcp_port: 5003
    persistence_file: ‘/home/homeassistant/.homeassistant/garage_mysensors.json
persistence: true
version: ‘2.0’
```

MySensors Arduino Code

```
#define MY_GATEWAY_W5100
#define MY_PORT 5003
#define MY_MAC_ADDRESS 0x00, 0x01, 0x02, 0x03, 0x04, 0x05
#define MY_IP_ADDRESS 192,0,2,18
#define MY_IP_SUBNET_ADDRESS 255,255,255,0
#define MY_IP_GATEWAY_ADDRESS 192,0,2,1

int pin = 4;
int id = 4;
MyMessage switch(id, V_STATUS);
bool need_init = false;

void setup() {
  pinMode(pin, OUTPUT);
}

void presentation(){
  sendSketchInfo(“Garage”, “1.0”);
  present(id, S_BINARY);
  need_init = true;
}
```

```
void loop(){
  if (need_init){
    need_init = false;
    switch.set(0);
    send(switch);
  }
}

void receive(const MyMessage &msg){
  if (msg.isAck()){
    // ignore or print or something
    return;
  }
  if (msg.sensor == id){
    digitalWrite(pin, HIGH);
    wait(100);
    digitalWrite(pin, LOW);
  }
}
```

Automations

Make the computer do all the hard work

Automations

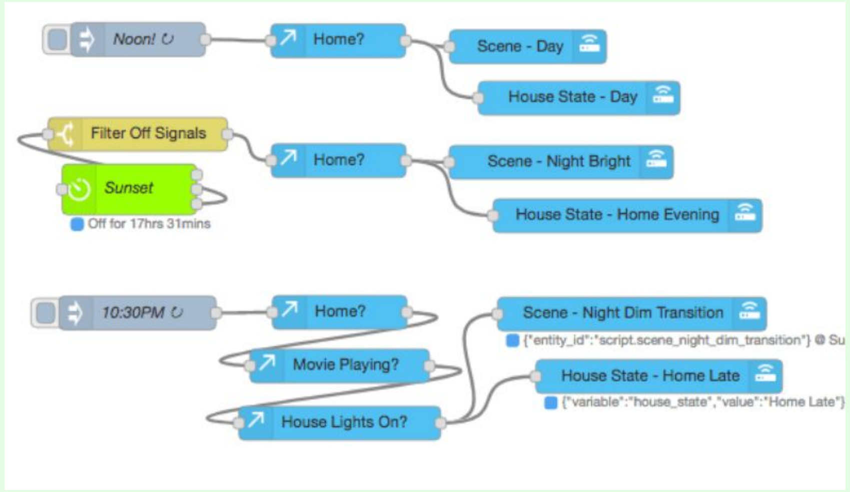
```
- alias: name of your automation
  initial_state: True
  trigger:
    <when this event happens>
  condition:
    - <only if these conditions are true>
  action:
    - <what to do>
```

Automations

```yaml
- alias: run sprinkler
  initial_state: True
  trigger:
    platform: sun
    event: sunrise
    offset: '-02:00:00'
  condition:
    - condition: state
      entity_id: binary_sensor.sprinkler_is_run_day
      state: 'on'
    - condition: state
      entity_id: binary_sensor.sprinkler_bypass
      state: 'off'
    - condition: state
      entity_id: binary_sensor.sprinkler_is_rain_bypass
      state: 'off'
  action:
    service: homeassistant.turn_on
    entity_id: script.run_sprinklers
```

Automations With Node Red

- Node Red is a browser based workflow dev tool built on Node.js
- Not specific to Home Assistant
- If you like visualizing workflows, or just hate YAML, this might be the tool for you



So Many Possibilities

- Zwave
- Zigbee
- Google Home
- Alexa
- Device Tracking
  - Router/Wifi
  - Owntracks
  - GPS Logger
- Cameras
- Bayes Statistics Sensor
- Weather
- Bluetooth
- Web Scraping
- Graphing
  - Influx
  - Graphite
- Inputs
- Cars
- Media
  - Sonos
  - Plex
  - MPD
  - Roku
- Notifications
  - Slack
  - Telegram
  - Twilio