Google Smart Home Ceiling Fan Control

VJ Sananda (vj.sananda@gmail.com)

Inspired by this:

A Wifi ON/OFF switch that can be controlled by Google Home or Alexa

“Turn light on”

Surprisingly reliable

$7 in US

$5 direct from China

Smart Home Devices

- Attractive use model when controlled using voice commands from Alexa or Google Home
- But many devices have a high price point. Used to be close to $100, but dropping now.
- SONOFF is the lowest priced on/off smart switch.
- Piqued my interest to look for ceiling fan controllers that could recognize voice commands.
  - Like “Set Fan speed high”

Market Survey for Smart Ceiling Fan controllers (Alexa and Google Home support)

1. Found Wifi Ceiling Fans, 2X markup $300+ cost for a fan that would otherwise be $120-$150. Most have poor reviews.
2. BOND : A $99 device that needs a fan to have an existing wireless remote. Good reviews.
3. Z wave controllers for $45 that require a Z wave hub.

So...

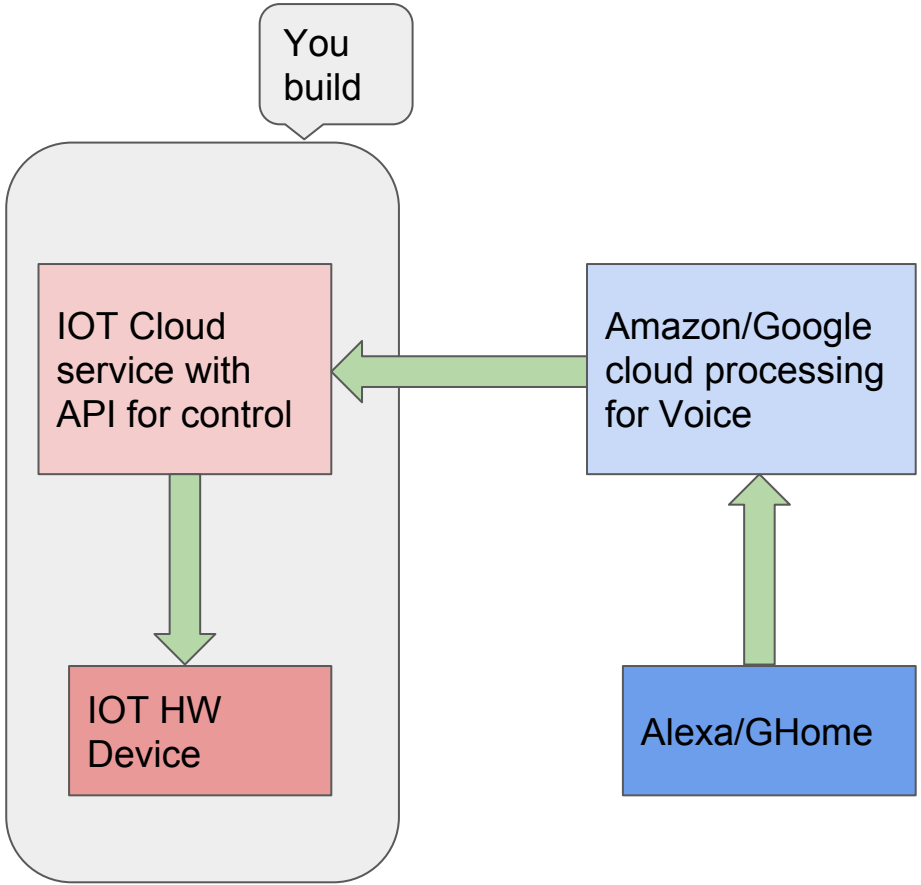
Wondered what it would take to build a low cost but reliable wifi smart fan controller, that would work with Google home and Alexa. (without any additional device, hub or remote)

That started my journey on figuring out the entire flow in getting Google home talking to the IOT device.

More than just hardware, requires a dedicated endpoint in the cloud for the Google home/Alexa servers to talk to.

Transaction Flow

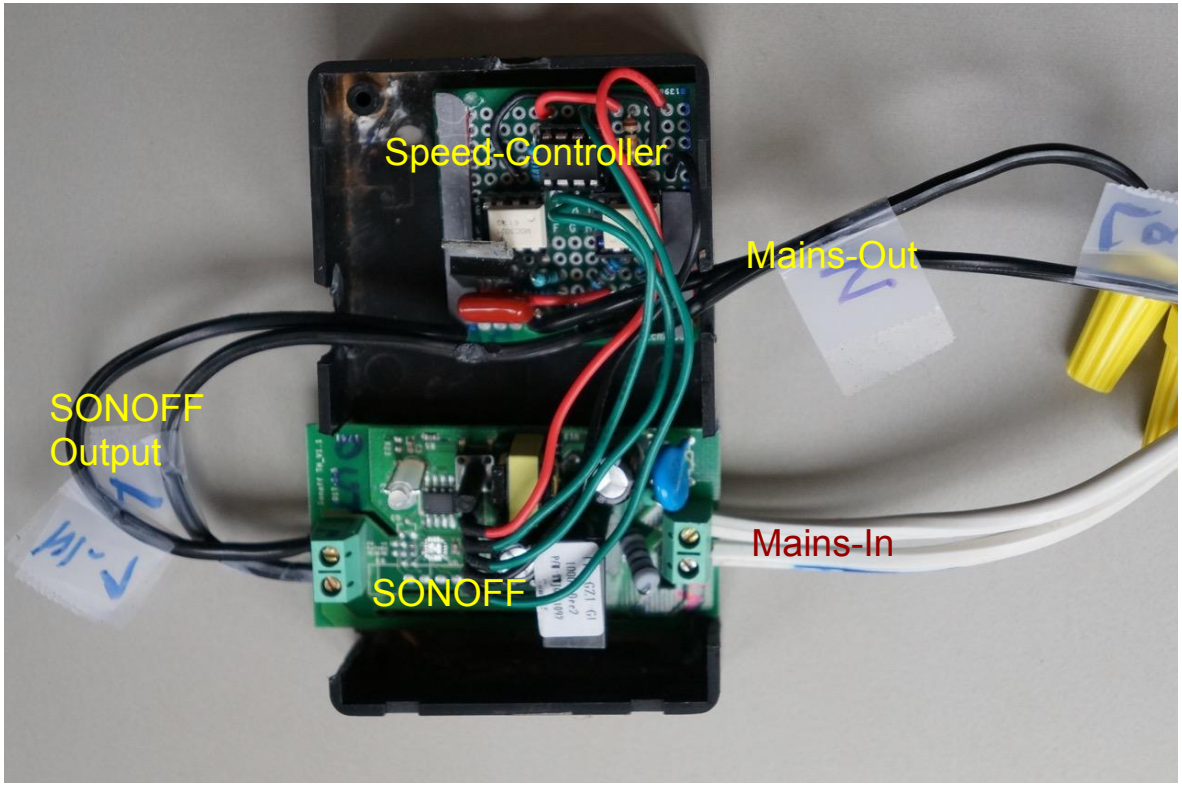
- The IOT device provider has to provide an authenticated cloud service to control their device.
- Google/Amazon will process the voice command, find the intent and call your control API.
- Have to factor your cloud service in the cost equation.
- Summary: Your device is a combination of HW along with dedicated cloud service.



Hardware: Wifi Smart Fan Controller

The SONOFF module has the ESP8266 Wifi chip along with a relay to switch AC on and off.

The output from the SONOFF is connected to the Fan speed controller in daisy chain fashion.



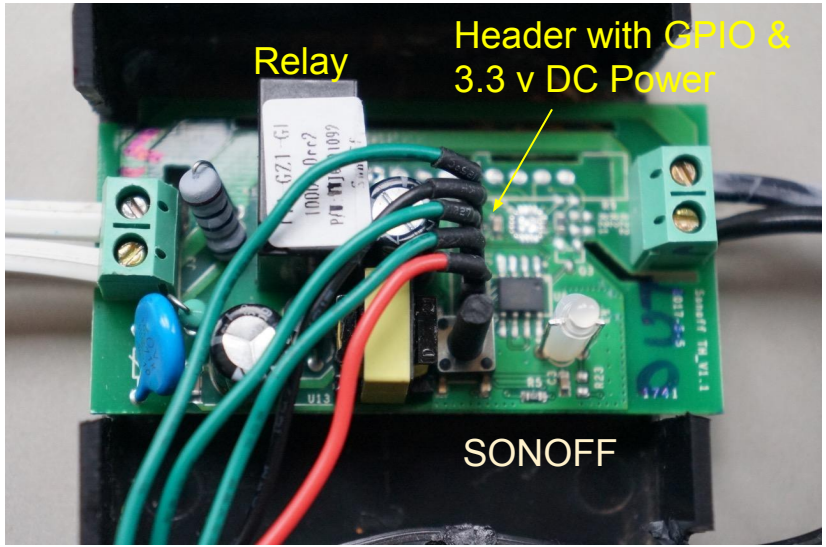
Hardware: SONOFF Detail

The SONOFF has an onboard DC 3.3v power source (from the 120v AC line in), to power the Wifi chip (ESP8266).

This power along with a few GPIO pins from the ESP8266 are available on a header.

We can load custom firmware on the ESP8266 to establish Wifi connectivity and then receive commands to drive GPIO appropriately.

3 GPIO lines can encode 8 speeds to the Fan speed controller.



Hardware: Speed Controller detail

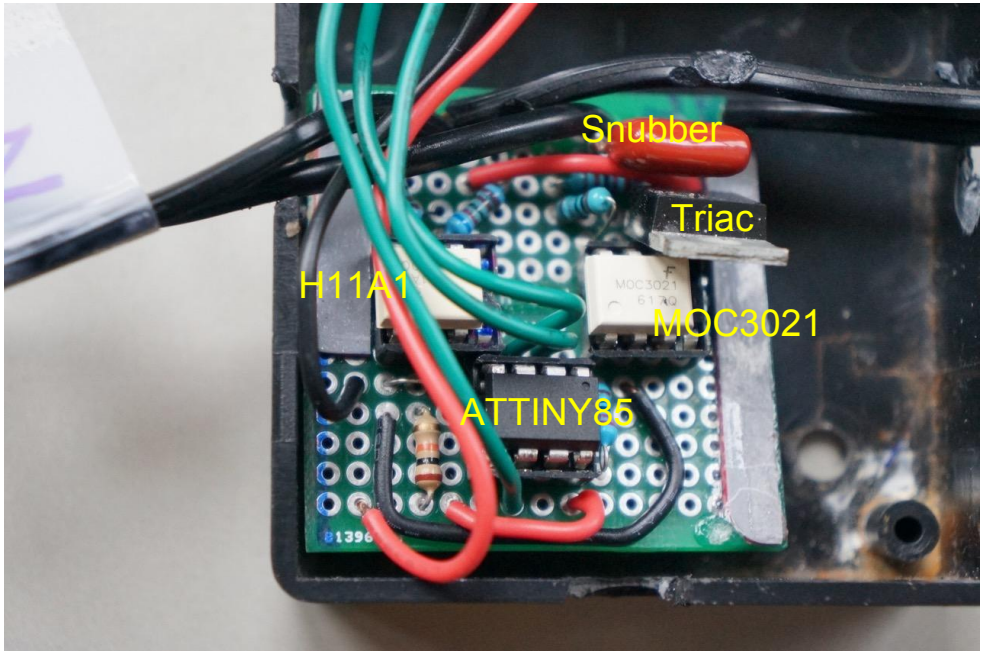
The GPIO output from the SONOFF are inputs to an ATTINY85 microcontroller.

The ATTINY samples the AC for the 0-crossings using an H11A1 optocoupler.

By delaying driving the Triac using a MOC3021 Triac driver (with opto isolator), the duty cycle of the output AC waveform can be changed.

Snubber circuit , resistor + bypass capacitor to drive inductive loads, like a fan.

Hardware fairly straightforward.



Hardware : Cost of Active components

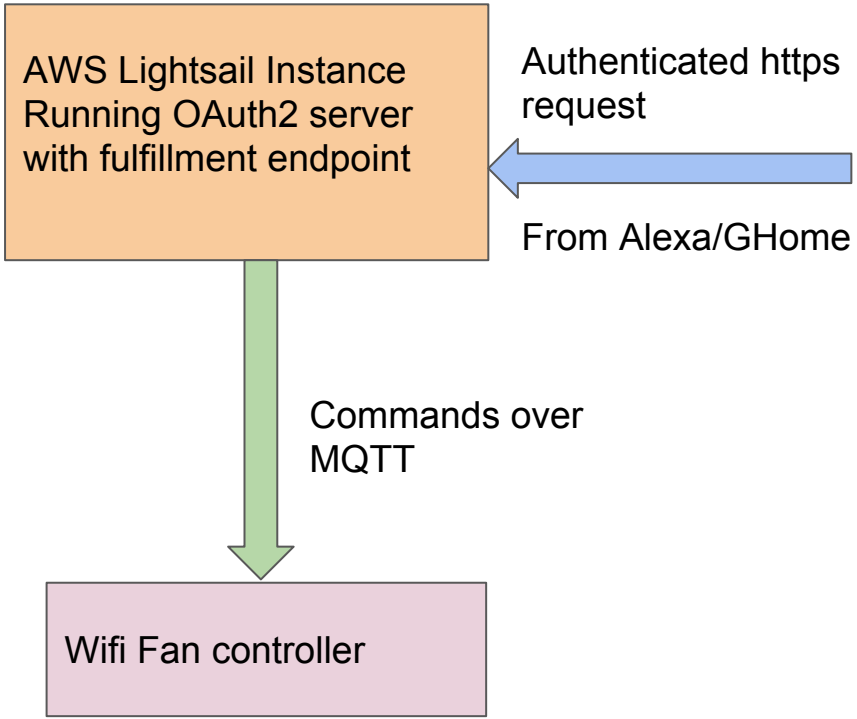
From Aliexpress

- SONOFF (wifi module with relay) : $5.00
- H11A1 (optoisolator) : $0.10
- MOC3021 : $0.15
- ATTINY85 : $1.50
- BT136 : $0.10
- Total : $6.85

Doesn’t break the bank !

Cloud Service Setup (from scratch)

1. Virtual Linux machine in cloud using AWS lightsail.
2. SSL certificates from Let's Encrypt. Google Home requires a secure cloud service using https
3. Cloud Linux machine runs OAuth2 server and fulfillment API. Written in nodejs.
4. Cloud Linux machine also runs an MQTT broker.
5. Nodejs fulfillment API sends commands to Fan controller via MQTT.



Cloud Service Setup

6. Actions on Google console (https://console.actions.google.com) allows you to add sample project to test.

7. Once added under your google account, the device shows up in the Google Home app on your phone.

8. Authenticate via login and password on your phone to link device to Google home.

9. Now your device should respond to Google home voice commands. Command vocabulary based on device type (in this case a fan, “on, off, set speed high, low”)

Challenges

Needless to say nothing as smooth.

Much time spent building OAuth2 server, ensuring token exchange and authentication.

Essentially the username, password combination maps to a unique token, which is then passed along in the header of the https request to authenticate the fulfillment API call.

First prototype built using low voltage AC from a transformer, before wiring up 120v AC.

2 sets of firmware, one for the ESP8266 and the other for the ATTINY85 to complete the control loop. Written using the Arduino IDE.

Google Home Mini for testing

Demo (youtube video)

ADDITIONAL RESEARCH

Wall switch vs. Module in Ceiling

A wall switch convenient as a drop in replacement for existing fan speed controllers.

However, Wall switch requires neutral wire so that wifi can be powered up.

Neutral Wire Issue

https://theiotpad.com/tips/smart-switches-no-neutral-wire

There have been attempts by Lutron to allow some leakage to earth (UL allows a small amount), but we would likely exceed that.

Looked at 3 switch outlets in my home (2009 construction), 2 of 3 had a neutral wire

Output voltage of existing controllers

Measured output voltage for different settings

High setting : 120 v (looks like this just bypasses controller)

Medium : Varies from 100v to 65v , Fan load is inductive and non-linear, so output voltage varies with each fan.

Low : Varies from 36v to 50v.

This inconsistent output voltage, opens up the possibility of an improved controller. With more fine grained control, can give user the ability to choose and tune the speeds for speed settings other than High.