

Smart Home Unified Control System With Homekit and Raspberry Credits

Hong Liu*, Yupeng Tao and Mohan Liu

School of Electronics Information, Shanghai Dianji University, Shanghai, China, 200240,
** liumohan@sdju.edu.cn, *** taoyupeng@sdju.edu.cn

ABSTRACT

This paper introduces a way to emerge smart-home sensors and devices of different bands to one control system or a control app. In the case that apple homekit-based devices follow the security protocols of Apple's privacy policy. It is hard to gather data from these devices. The best way to unite all the data is by guiding other bands' data to the Apple Home. By transforming all the data to apple homekit, all the data will show in the Home App and we can control all the data much easier. For the security reason, all the data need to encrypt through the homebridge system. Use homebridge and gateway to route data to the apple system[1]. Experiments and data testing showing that this way is suitable to centralize all the data to Apple Home.

Keywords: Smart home; HomeKit; Raspberry; IoT; System architecture; Hardware infrastructure requirements.

1. INTRODUCTION

It is noticeable that a smart home is wildly used in lots of modern families. Many companies such as Apple, Inc, Tp-Link, Mi, Huawei, Ikea. . . developed their own smart devices and sensors, which means they have their control center and gateway hub.

If a family install several brands of smart devices, they need to use several controlling app or control hub to control all the devices. And it is even hard to make all the devices cooperated because of the separation of platforms. Cross-platform cooperation seems difficult if we only use software to transform data via a different platform. Beyond all of the hubs and gateways, a central hub to merge all the data is pretty necessary.

Homekit is an open-source IoT control platform developed by Apple, with safe API and Siri, this platform makes smart devices' control much easier than other platforms^[2]. By merging other platforms' data to Homekit, it is a safe and innovating way to unify all cross-platform devices.

2. SYSTEM ARCHITECTURE

The system architecture contains the application layer, the service layer, the Interference layer, and the hardware

layer, as the figure 1.

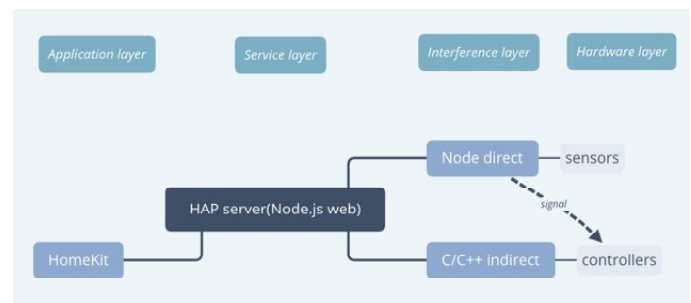


Figure 1 System layers

- (1) Application layer: This layer is directly based by Homekit API, for the safety reason, Apple secure its own safety protocol. An alternate way in the service layer will solve this problem.
- (2) Service layer: By modifying open-source Homebridge from GitHub with Nodejs, a web backstage server will generate in the Nodejs. Through this server, Homebridge hardware can get verification from Apple. It is a critical step to get this verification so the system can link the gateway to Homekit and get devices running well on Home App.

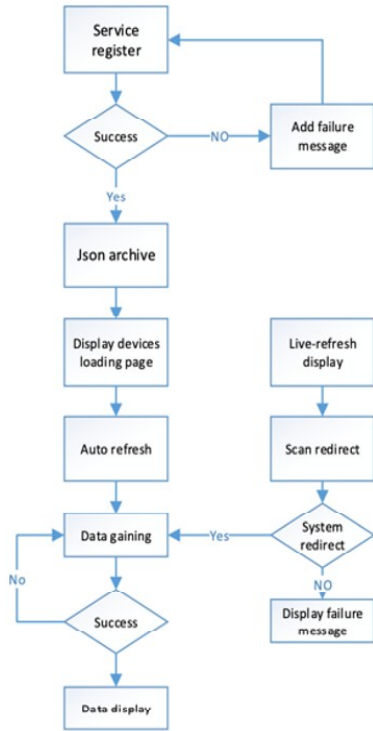


Figure 2 Sensors process

(3) Interference layer: This layer will be the bridge to the service layer. According to different types of chips, each company and the open-source community provide specifically modified C/C++ module. Through those modules, the system can access the hardware layer and control it by Python and Node.js.

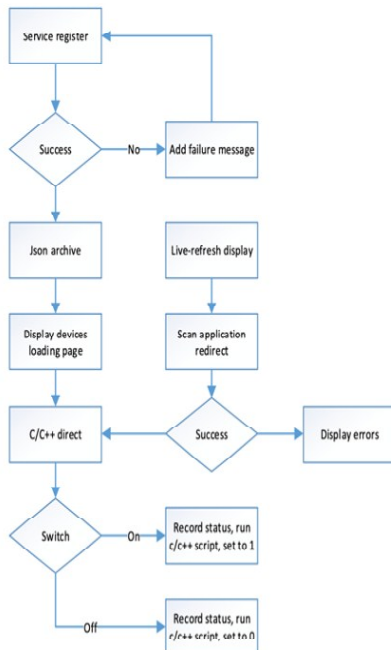


Figure 3 Controller process

(4) Hardware layer: Hardware is the actual working layer. It contains sensors and control modules. Whose working process as figure 2 and 3. The interference layer gathers data from sensors and transfers it to service data, then those data will be acquired by the application layer and display it in the Home App

3. HARDWARE INFRASTRUCTURE REQUIREMENTS

This system use plugins that require access to the Raspberry Pi’s GPIO or Bluetooth radio, or require hardware video decoding, beware using Docker to run Homebridge may add an additional layer of complexity to this setup. If this is the case may wish to consider Setting up Homebridge with Systemd instead^[3].

3.1 Initial Raspberry Pi Setup

Get the latest copy of Raspbian Stretch Lite from the official Raspberry Pi website and burn this to the SD card using Etcher. Raspbian Stretch Lite is preferred as there is no need to run a GUI desktop. By default SSH access and WiFi are disabled in Raspbian. Enable both these services before booting from the SD card for the first time - this will avoid the need to ever connect a screen to the Pi.

The following changes to the freshly imaged SD card should be made on the computer before plugging the card into the Raspberry Pi for the first time. Enable SSH: To enable remote SSH access on first boot create an empty file called ssh in the root of the SD card.

Enable WiFi (Optional): If the Raspberry Pi with built-in WiFi (Pi 3 or Zero W), configure WiFi on first boot. To do this create a file named wpa_supplicant.conf in the root of the SD card that contains the following (replace the YOUR_COUNTRY_CODE, YOUR_SSID and YOUR_PASSWORD values):

```

country=YOUR_COUNTRY_CODE
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="YOUR_SSID"
    scan_ssid=1
    psk="YOUR_PASSWORD"
    key_mgmt=WPA-PSK
}
  
```

3.2 Manual Install

Install Node: Starting with version 4.0.0, NodeJS now

supports ARM-based platforms like Raspberry Pi by default. Node.js package manager install (Easiest, Preferred) If the system is running a newish Raspberry Pi with an ARMv7 chip or better, install NodeJS using their apt-get repository^[4]. Follow the instructions here.

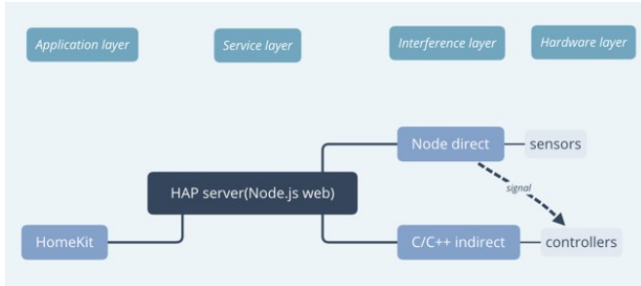


Figure 4 Install process

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E
bash -
sudo apt-get install -y nodejs
```

Install Avahi and other Dependencies: This is required by the mdns package in HAP-NodeJS library.

```
sudo apt-get install libavahi-compat-libdnssd-dev
```

Install homebridge and homebridge-mi-aqara:

```
sudo npm install -g --unsafe-perm homebridgesudo npm
install -g --unsafe-perm homebridge-mi-aqara
```

Start homebridge Docker container by running:

```
docker-compose up -d
```

It might take some time to download the initial image which is about 125 MB compressed. Docker will now download the latest oznu/homebridge docker image. Run the container as a background process. After processing, the ios pin code and the QR code will show on the terminals.



Figure 5 QR set

The raspberry is all set. It means a transfer station is built successfully. To connect the smart home gateway (this paper use Mi Gateway), this system acquires mac address and SID. All of these configurations can be detected in Apps (Mi Home) or routers. SID is a critical configure that the system needs it to control devices and send signals.

3.3 Config File Setup

There are 3 components to the config.json file. The “Bridge” at the top, this information is a default and the same for everyone, the system can technically change it. Format sample is down below:

```
“bridge”: {
  “name”:”Homebridge”,
  “username”:”FC:35:HD:HF:KL:26”,
  “port”:51826,
  “pin”:”233-78-123”
},
“platforms”: [{
  “platform”: “MiAqaraPlatform”,
  “sid”: [“”],
  “password”: [“”]
```

After setting configurations, use iOS devices to scan the code see if this system work properly:

If this pops up on your devices, Homebridge is ready to go. A new gateway icon will show on the Home app, all the devices connect to it will show data and control options, as Fig. 6 shows.

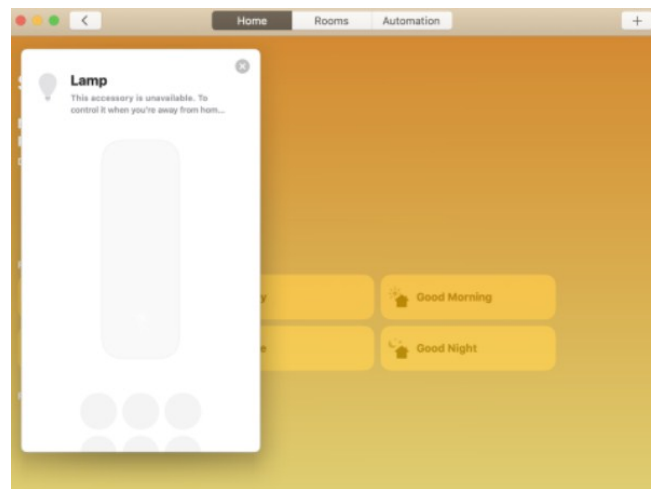


Figure 6 Control panel

Experiments take a mi lamp that does not support Homekit. With the modified raspberry, this device now shows on the home app and ready to be controlled. Sensors follow the same protocols, the system will automatically add configurations if a new device adds to the gateway. This unifies seemly all the devices do not

support homekit, and homebridge community is still improving system stability and supporting more devices^[5].

CONCLUSION

Homebridge is a lightweight NodeJS server you can run on your home network that emulates the iOS HomeKit API. It supports Plugins, which are community-contributed modules that provide a basic bridge from HomeKit to various 3rd-party APIs provided by manufacturers of “smart home” devices.

Since Siri supports devices added through HomeKit, this means that with Homebridge you can ask Siri to control devices that don’t have any support for HomeKit at all.

Biographical Notes

Hong Liu is an associate professor of Electronics and Information in Shanghai Dianji University. She received her Ph.D. degree from East China Normal University in 2012. Her research interest is in the areas of Internet of things technology and the latest communication

technology.

Yupeng Tao is a student in the Department of Electronics and Information in Shanghai Dianji University. His research interests include micro-nano electronic detectors and sensors.

Mohan Liu is a student in the Department of Electronics and Information in Shanghai Dianji University. He current research interests include image registration, medical image processing and pattern recognition.

References

- [1] Anonymous. Crestron Announces Integration with Apple TV, Siri, and HomeKit[J]. Residential Systems,2018,19(7).
- [2] Honeywell Home and Building Technologies; Honeywell Lyric Controller Home Security System Compatible With Apple® HomeKit[J]. Computers, Networks & Communications,2017.
- [3] Anonymous. Yales Adds Apple HomeKit Support[J]. Residential Systems,2017,18(11).
- [4] Silver, Stephen. The Future of Apple’s HomeKit[J]. Dealerscope,2015,57(10).



This document was created with the Win2PDF "print to PDF" printer available at <http://www.win2pdf.com>

This version of Win2PDF 10 is for evaluation and non-commercial use only.

This page will not be added after purchasing Win2PDF.

<http://www.win2pdf.com/purchase/>