

# YeeLight 智能灯

## 【硬件准备】



## 【操作步骤】

1. 在 Yeelight App 中添加智能灯设备，打开外部访问模式
2. 在 HA 中配置 yeelight
3. 解决两个问题：重复自动发现与刷新速度

## 【参考】

- HA 中配置 yeelight 说明文档  
<https://www.home-assistant.io/components/light.yeelight/>
- yeelight 配置样例

```
discovery:
```

```
  ignore:
```

```
    - yeelight
```

```
yeelight:
```

```
  devices:
```

```
    192.168.3.162:
```

```
      name: my_yeelight
```

```
      transition: 1000
```

# 云端的自动化——IFTTT(1)

## 【操作步骤】

1. IFTTT 基本使用  
    样例：如果 gmail 收到邮件，就切换 YeeLight 智能灯开关状态
2. 在 HA 中执行脚本，触发 IFTTT 切换 YeeLight 智能灯开关状态
  - a) 配置 IFTTT 规则 (THIS=webhook, THAT=YeeLight)
  - b) 配置 HA 中脚本
  - c) 执行演示

## 【参考】

- IFTTT 网站  
<https://ifttt.com/>
- IFTTT 与 HomeAssistant 的集成



- HA 中集成 IFTTT 配置说明  
<https://www.home-assistant.io/components/ifttt/>
- 样例：在 HA 中执行脚本，触发 IFTTT 切换 YeeLight 智能灯开关状态  
HA 中配置(example\_10\_2\_1.yaml)

```
# example_10_2_1.yaml
ifttt:
  key: deZzryxvEslHpFoL_gsUsm
script:
  button_press:
    alias: 通过 IFTTT 开关灯
    sequence:
      - service: ifttt.trigger
        data:
          event: ButtonPressed
```

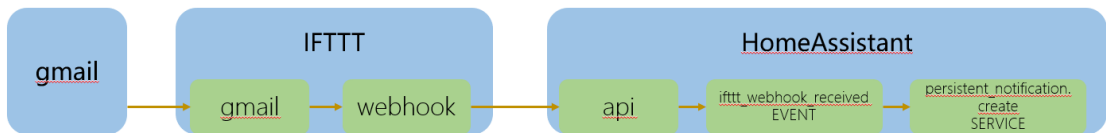
## 云端的自动化——IFTTT(2)

### 【操作步骤】

1. 配置 HA 中由 IFTTT 触发的自动化规则
2. 配置 HA 中 IFTTT 事件的接收 API
3. 配置 IFTTT 规则：收到邮件，触发 webhook
4. 测试演示

### 【参考】

- 样例逻辑驱动示意



- IFTTT 网站

<https://ifttt.com/>

- HA 中集成 IFTTT 配置说明

<https://www.home-assistant.io/components/ifttt/>

- HA 中自动化配置

- trigger:

```
platform: event
event_type: ifttt_webhook_received
event_data:
  action: email_arrived
```

action:

```
service: persistent_notification.create
data_template:
  title: 收到新邮件
  message: >
```

一封来自{{ trigger.event.data.from }}的邮件， 邮件主题: {{ trigger.event.data.subject }}

- IFTTT 中 webhook 发送的 json 数据

```
{ "action": "email_arrived", "from": "{{FromName}}", "subject": "{{Subject}} }
```

# 系统性能监控——SystemMonitor

## 【操作步骤】

1. 完整配置 SystemMonitor 组件
2. 解释每个监视项的含义， 以及对应 linux 命令

## 【参考】

- HA 中 SystemMonitor 组件配置说明

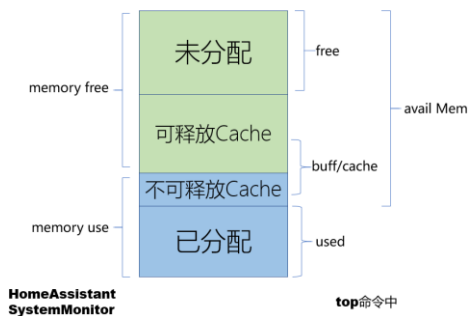
<https://www.home-assistant.io/components/sensor.systemmonitor/>

- systemmonitor 配置样例

sensor sm:

```
- platform: systemmonitor
  scan_interval: 5
  resources:
    - type: disk_use_percent
      arg: /
    - type: disk_use
      arg: /
    - type: disk_free
      arg: /
    - type: memory_use_percent
    - type: memory_use
    - type: memory_free
    - type: swap_use_percent
    - type: swap_use
    - type: swap_free
    - type: load_1m
    - type: load_5m
    - type: load_15m
    - type: network_in
      arg: wlan0
    - type: network_out
      arg: wlan0
    - type: packets_in
      arg: wlan0
    - type: packets_out
      arg: wlan0
    - type: ipv4_address
      arg: wlan0
    - type: ipv6_address
      arg: wlan0
    - type: processor_use
    - type: process
      arg: smbd
    - type: last_boot
```

- 内存概念



# 以不同的音色播报文字——百度 tts

## 【操作步骤】

1. 申请百度云免费开发者账号
2. 配置百度 tts
3. 调用服务，以不同音色播报文字

## 【参考】

- 百度云-语音合成网站  
<https://cloud.baidu.com/product/speech/tts>
- HA 中百度 tts 配置介绍  
<https://www.home-assistant.io/components/tts.baidu/>
- 百度 tts 的配置

```
tts:  
  - platform: baidu  
    app_id: 9931448  
    api_key: YaEF9KGD6WvoXovGMZxtX3Q  
    secret_key: 70e71c2425dwccb67439dafdcf9b999f  
    speed: 5  
    pitch: 5  
    volume: 5  
    person: 4
```

- 对话脚本

```
dialog1:  
  alias: 对话一  
  sequence:  
    - service: tts.baidu_say  
      entity_id: "all"  
      data:  
        message: "作战时，踩到地雷咋办？"  
        options:  
          person: 1  
    - delay: 00:00:03  
    - service: tts.baidu_say  
      entity_id: "all"  
      data:  
        message: "靠，能咋办？踩坏了照价赔偿。"  
        options:  
          person: 3  
          pitch: 9  
  
dialog2:  
  alias: 对话二  
  sequence:  
    - service: tts.baidu_say  
      entity_id: "all"  
      data:  
        message: "泽哥，可能是睹物思人，在外面玩了一天，看见什么都像你"  
        options:  
          person: 4  
    - delay: 00:00:07  
    - service: tts.baidu_say  
      entity_id: "all"  
      data:  
        message: "哎哦，良心发现啊？看你这么想我的份上，我去接你吧。你在哪儿呢？"  
        options:  
          person: 3  
          speed: 3  
          pitch: 1  
    - delay: 00:00:08  
    - service: tts.baidu_say
```

```
entity_id: "all"
data:
  message: "动物园"
  options:
    person: 4
    volume: 15
- delay: 00:00:03
- service: tts.baidu_say
entity_id: "all"
data:
  message: "你, 该吃药了"
  options:
    person: 3
    speed: 3
    pitch: 1
```