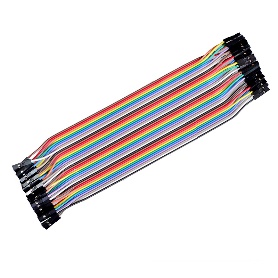
直连树莓派的LED(1)

——NodeRED接入/HA中rpi\_gpio\_pwm组件

【硬件准备】

LED小灯 / 1kΩ左右保护电阻 / 杜邦线

【操作步骤】

1. 连接LED硬件
2. 在NodeRED中控制LED
3. 启动pigpiod服务

重新配置jupyter notebook端口号

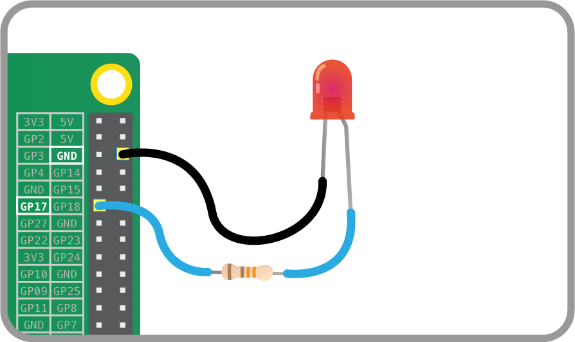
配置pigpiod服务配置

启动pigpiod服务

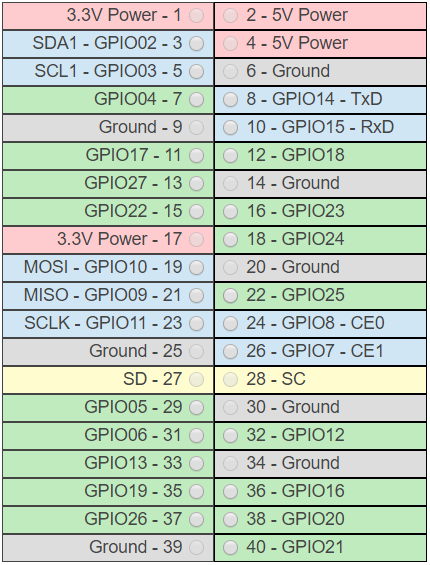
1. 在HA中配置rpi\_gpio\_pwm
2. 在HA前端控制LED

【参考】

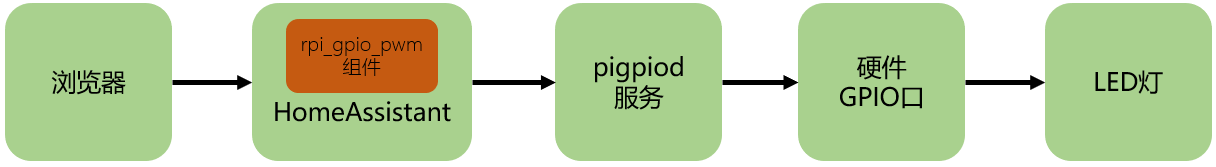
* 连接示意图



* 树莓派引脚



* rpi\_gpio\_pwm组件逻辑示意图



* pigpiod服务配置文件（/lib/systemd/system/pigpiod.service）

[Unit]

Description=Daemon required to control GPIO pins via pigpio

[Service]

ExecStart=/usr/bin/pigpiod -l -n 127.0.0.1

ExecStop=/bin/systemctl kill pigpiod

Type=forking

[Install]

WantedBy=multi-user.target

* pigpiod服务控制

重载服务配置： sudo systemctl --system daemon-reload

设置为自启动： sudo systemctl enable pigpiod

启动服务： sudo systemctl start pigpiod

* HomeAssistant中GPIO口LED配置

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

light:

- platform: rpi\_gpio\_pwm

leds:

- name: my\_led

driver: gpio

pins: [17]

type: simple

直连树莓派的LED(2)

——HA 中的shell\_command/binary\_sensor.command\_line/light.template

【操作步骤】

1. 尝试使用shell命令控制LED
2. 使用shell\_command和light.template构建设备
3. 使用binary\_sensor.command\_line组件反馈灯的状态

【参考】

* sysfs控制命令

创建GPIO控制文件: echo 17 > /sys/class/gpio/export

删除GPIO控制文件： echo 17 > /sys/class/gpio/unexport

配置输入/输出方向： echo out > /sys/class/gpio/gpio17/direction

输出电压： echo 1 > /sys/class/gpio/gpio17/value

输出零： echo 0 > /sys/class/gpio/gpio17/value

* 配置（example\_15\_2\_1.yaml）

shell\_command:

gpio17\_init: (echo 17 > /sys/class/gpio/export) && (sleep 1) && (echo out > /sys/class/gpio/gpio17/direction)

gpio17\_deinit: (echo 17 > /sys/class/gpio/unexport)

gpio17\_turn\_on: (echo 1 > /sys/class/gpio/gpio17/value)

gpio17\_turn\_off: (echo 0 > /sys/class/gpio/gpio17/value)

automation:

- alias: create GPIO17 sysfs

initial\_state: True

trigger:

- platform: homeassistant

event: start

action:

service: shell\_command.gpio17\_init

- alias: delete GPIO17 sysfs

initial\_state: True

trigger:

- platform: homeassistant

event: shutdown

action:

service: shell\_command.gpio17\_deinit

light:

- platform: template

lights:

gpio17\_led\_light:

friendly\_name: GPIO17\_LED

turn\_on:

service: shell\_command.gpio17\_turn\_on

turn\_off:

service: shell\_command.gpio17\_turn\_off

# value\_template: "{{ states.binary\_sensor.gpio17\_value.state }}"

#binary\_sensor:

# - platform: command\_line

# name: gpio17\_value

# command: cat /sys/class/gpio/gpio17/value

# payload\_on: 1

# payload\_off: 0

# scan\_interval: 1

* shell\_command组件配置

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

* light.template组件配置

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

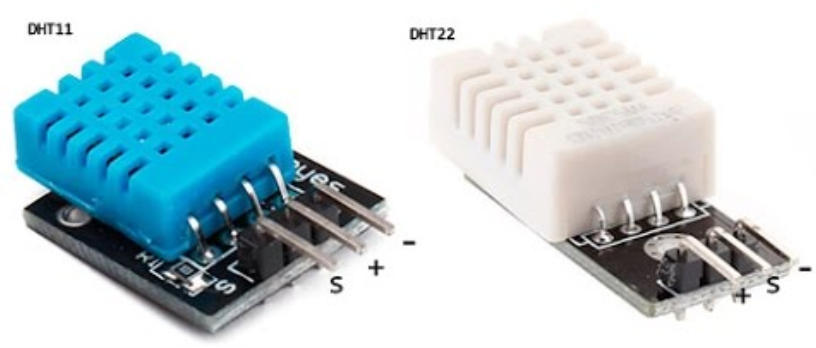
* binary\_sensor.command\_line组件配置

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

直连树莓派的温湿度传感器

【硬件准备】

温湿度传感器：dht11/dht22/18b20

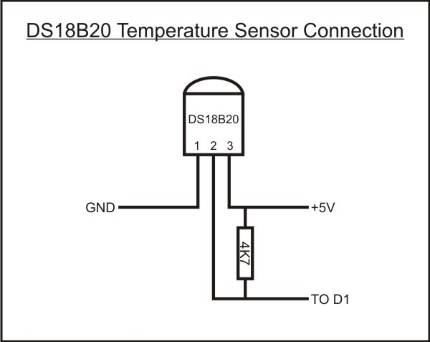
  

【操作步骤】

1. 在HomeAssistant中配置DHT传感器
2. 在Node-RED中配置DHT传感器
3. 在树莓派上打开1-wire通讯
4. 在HomeAssistant中配置18b20
5. 在Node-RED中配置18b20

【参考】

* 连接示意图



* DHT温湿度传感器在HA中的配置

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

sensor:

- platform: dht

sensor: DHT11

pin: 18

monitored\_conditions:

- temperature

- humidity

* bcm2835开发包

<http://www.airspayce.com/mikem/bcm2835/>

* 18b20在HA中的配置

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

sensor:

- platform: onewire