

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

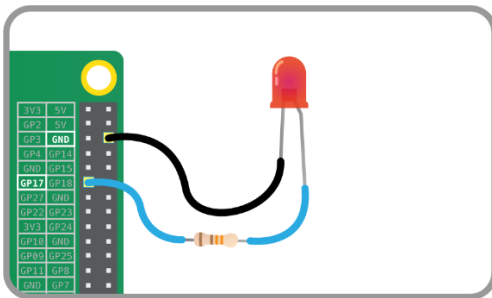


【操作步骤】

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
4. 在 HA 中配置 rpi_gpio_pwm
5. 在 HA 前端控制 LED

【参考】

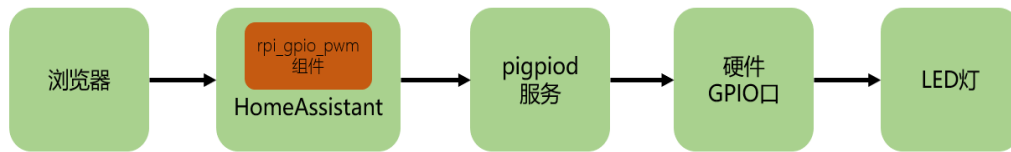
- 连接示意图



- 树莓派引脚

3.3V Power - 1	2 - 5V Power
SDA1 - GPIO02 - 3	4 - 5V Power
SCL1 - GPIO03 - 5	6 - Ground
GPIO04 - 7	8 - GPIO14 - TxD
Ground - 9	10 - GPIO15 - RxD
GPIO17 - 11	12 - GPIO18
GPIO27 - 13	14 - Ground
GPIO22 - 15	16 - GPIO23
3.3V Power - 17	18 - GPIO24
MOSI - GPIO10 - 19	20 - Ground
MISO - GPIO09 - 21	22 - GPIO25
SCLK - GPIO11 - 23	24 - GPIO8 - CE0
Ground - 25	26 - GPIO7 - CE1
SD - 27	28 - SC
GPIO05 - 29	30 - Ground
GPIO06 - 31	32 - GPIO12
GPIO13 - 33	34 - Ground
GPIO19 - 35	36 - GPIO16
GPIO26 - 37	38 - GPIO20
Ground - 39	40 - GPIO21

- 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

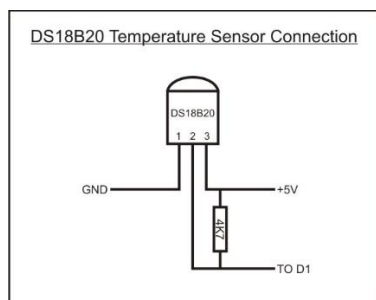


【操作步骤】

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