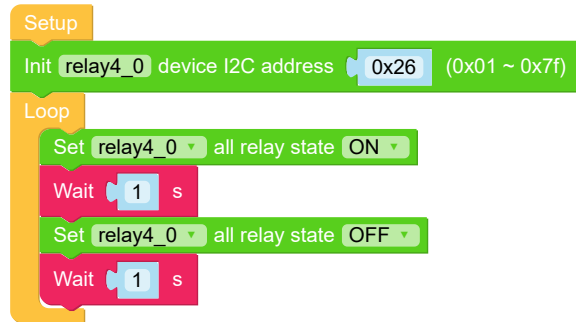


Module13.2 4Relay

| 案例程序

间隔一秒闭合和断开继电器



```
from m5stack import *
from m5ui import *
from uiflow import *
import module

import time

setScreenColor(0x222222)

relay4 = module.get(module.RELAY4)

relay4_0 = relay4.init_i2c_address(0x26)
while True:
    relay4_0.set_all_relay_state(1)
    wait(1)
    relay4_0.set_all_relay_state(0)
    wait(1)
    wait_ms(2)
```

| 功能说明

Get Please init a 4Relay Module ADC 12bit raw value (return 0 ~ 4095)

```
relay4_0.get_adc_12bit_value(0)
```

- 获取12位 ADC 原始值，返回范围为0到4095的整数。这是从指定模块的 ADC 读取的未处理电压数据

Get Please init a 4Relay Module ADC 12bit voltage value (V, return 0 ~ 26)

```
relay4_0.get_adc_12bit_value(1)
```

- 获取12位 ADC 的电压值，返回范围为0到26的电压值(单位：伏特)

Get Please init a 4Relay Module ADC 8bit raw value (return 0 ~ 255)

```
relay4_0.get_adc_8bit_value(0)
```

- 获取8位 ADC 原始值，返回范围为0到255的整数。这是从指定模块的 ADC 读取的未处理电压数据

Get Please init a 4Relay Module ADC 8bit voltage value (V, return 0 ~ 26)

```
relay4_0.get_adc_8bit_value(1)
```

- 获取8位 ADC 的电压值，返回范围为0到26的电压值(单位：伏特)

Get Please init a 4Relay Module relay 1 (1 ~ 4) status (return 0 or 1)

```
relay4_0.get_relay_status(1)
```

- 获取继电器模块的状态，返回值为0或1

Init relay4_0 device I2C address 0x26 (0x01 ~ 0x7f)

```
relay4.init_i2c_address(0x26)
```

- 初始化设备的 I2C 地址

Set Please init a 4Relay Module all relay state ON

```
relay4_0.set_all_relay_state(1)
```

- 设置所有继电器的状态

Set Please init a 4Relay Module device I2C slave address 0x26 (0x01 ~ 0x7f)

```
relay4_0.set_i2c_address(0x26)
```

- 设置设备的 I2C 从属地址

Set Please init a 4Relay Module ▾ relay 1 (1 ~ 4) state ON ▾

```
relay4_0.set_relay_state(1, 1)
```

- 设置继电器1的状态
 - 参数1：第1-4个继电器
 - 参数2： "ON":闭合继电器， "OFF":断开继电器