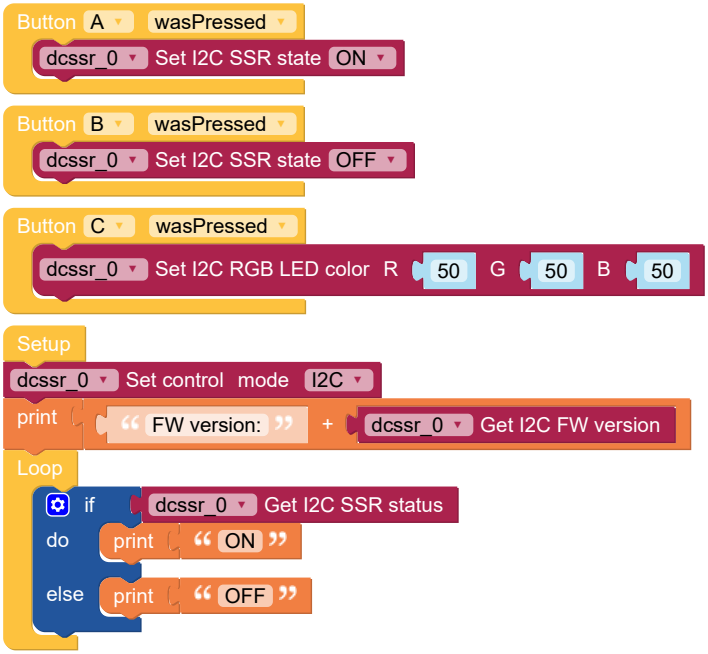
Unit DCSSR

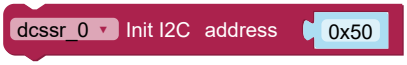
案例程序

> 点击按键，控制继电器的输人输出状态以及 RGB 的灯亮灭



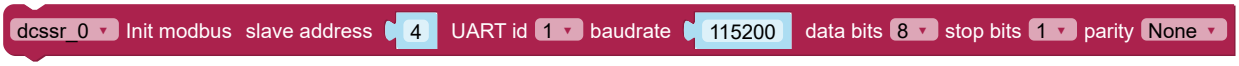
```
from m5stack import *
from m5ui import *
from uiflow import *
import unit

setScreenColor(0x444444)
dcssr_0 = unit.get(unit.DC_SSR, unit.PORTA)

def buttonA_wasPressed():
  # global params
  dcssr_0.set_i2c_ssr_state(1)
  pass
btnA.wasPressed(buttonA_wasPressed)

def buttonB_wasPressed():
  # global params
  dcssr_0.set_i2c_ssr_state(0)
  pass
btnB.wasPressed(buttonB_wasPressed)

def buttonC_wasPressed():
  # global params
  dcssr_0.set_i2c_rgb_led(50, 50, 50)
  pass
btnC.wasPressed(buttonC_wasPressed)

dcssr_0.init_mode(1)
print((str('FW version:') + str((dcssr_0.get_i2c_status(0xFE)))))
while True:
  if dcssr_0.get_i2c_ssr_status():
    print('ON')
  else:
    print('OFF')
  wait_ms(2)
```

功能说明




```
dcssr_0.init_i2c_address(0x50)
```

- 初始化设置 DCSSR Unit 的 I2C 通讯地址




```
dcssr_0.init_modbus(4, 1, 115200, 8, 1, None)
```

- 初始化 Modbus 通讯格式


```
print(dcssr_0.get_i2c_rgb_led())
```

- 通过 i2c 获取 RGB 的值


```
print(dcssr_0.get_i2c_ssr_status())
```

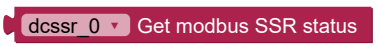
- 通过 I2C 获取 SSR 的开关状态


```
print(dcssr_0.get_i2c_status(0xFE))
```

- 通过 I2C 模式获取固件版本

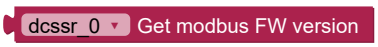

```
print(dcssr_0.get_modbus_rgb_led())
```

- 通过 Modbus 通讯方式获取 RGB 的值




```
print(dcssr_0.get_i2c_ssr_status())
```

- 通过 Modbus 获取 SSR 的开关状态

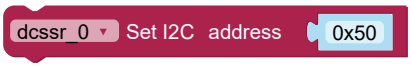



```
print(dcssr_0.get_modbus_ssr_status())
```

- 通过 Modbus 模式获取固件版本

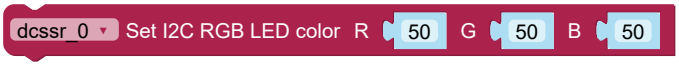

```
dcssr_0.init_mode(1)
```

- 设置 DCSSR Unit 的通讯模式，I2C 或者是 Modbus




```
dcssr_0.set_i2c_address(0x50)
```

- 设置 DCSSR 的 i2c 地址

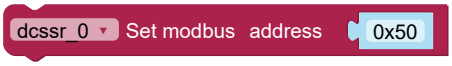



```
dcssr_0.set_i2c_rgb_led(50, 50, 50)
```

- 通过 i2c 设置 RGB 值


```
dcssr_0.set_i2c_ssr_state(1)
```

- 设置 SSR 设备的开关状态

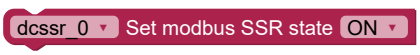



```
dcssr_0.set_modbus_address(0x50)
```

- 设置 DCSSR 的 i2c 地址


```
dcssr_0.set_modbus_rgb_led(50, 50, 50)
```

- Modbus 设置 LED 的 RGB 值



```
dcssr_0.set_modbus_ssr_state(1)
```

- Modbus 设置 SSR 的状态开关