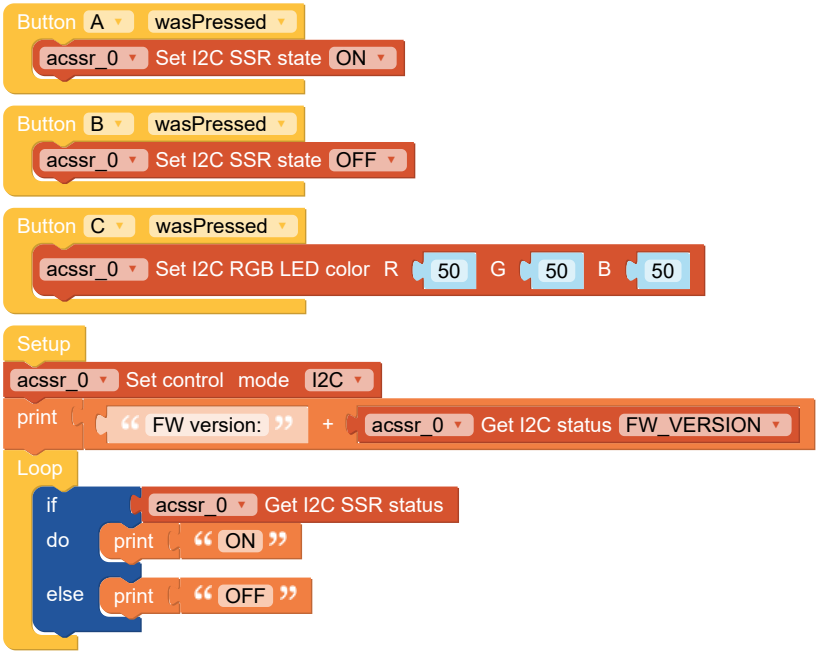
Unit ACSSR

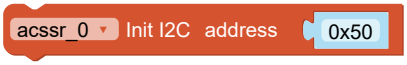
Example

> Click the button to control the input output state of the relay and the RGB light on and off



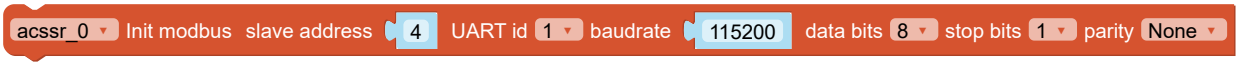
```
from m5stack import *
from m5ui import *
from uiflow import *
import unit

setScreenColor(0x222222)
acssr_0 = unit.get(unit.AC_SSR, unit.PORTA)

def buttonA_wasPressed():
  # global params
  acssr_0.set_i2c_ssr_state(1)
  pass
btnA.wasPressed(buttonA_wasPressed)

def buttonB_wasPressed():
  # global params
  acssr_0.set_i2c_ssr_state(0)
  pass
btnB.wasPressed(buttonB_wasPressed)

def buttonC_wasPressed():
  # global params
  acssr_0.set_i2c_rgb_led(50, 50, 50)
  pass
btnC.wasPressed(buttonC_wasPressed)

acssr_0.init_mode(1)
print((str('FW version:') + str((acssr_0.get_i2c_status(0xFE)))))
while True:
  if acssr_0.get_i2c_ssr_status():
    print('ON')
  else:
    print('OFF')
  wait_ms(2)
```

API




```
acssr_0.init_i2c_address(0x50)
```

- Initializes the I2C communication address of the ACSSR Unit




```
acssr_0.init_modbus(4, 1, 115200, 8, 1, None)
```

- Initializes the Modbus communication format


```
print(acssr_0.get_i2c_rgb_led())
```

- Get the RGB value through i2c


```
print(acssr_0.get_i2c_ssr_status())
```

- Obtain the switching status of the SSR through I2C


```
print(acssr_0.get_i2c_status(0xFE))
```

- Obtain the firmware version in I2C mode


```
print(acssr_0.get_modbus_rgb_led())
```

- Obtain the RGB value through Modbus communication


```
print(acssr_0.get_modbus_ssr_status())
```

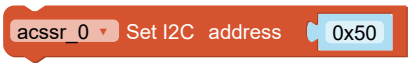
- Obtain the switching status of the SSR over Modbus


```
print(acssr_0.get_modbus_status(0x0001))
```

- Obtain the firmware version over Modbus

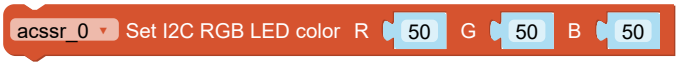

```
acssr_0.init_mode(1)
```

- Set the communication mode of the ACSSR Unit, I2C or Modbus




```
acssr_0.set_i2c_address(0x50)
```

- Set the i2c address of the ACSSR

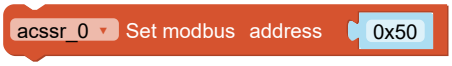



```
acssr_0.set_i2c_rgb_led(50, 50, 50)
```

- Set the RGB value using i2c

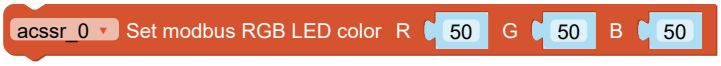

```
acssr_0.set_i2c_ssr_state(1)
```

- Set the switch status of the SSR device



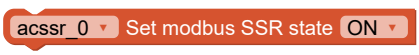

```
acssr_0.set_modbus_address(0x50)
```

- Set the i2c address of the ACSSR




```
acssr_0.set_modbus_rgb_led(50, 50, 50)
```

- Modbus Sets the RGB value of the LED



```
acssr_0.set_modbus_ssr_state(1)
```

- Modbus Sets the status switch of the SSR