

Unit ACSSR/DCSSR Arduino 使用チュートリアル

1. 準備作業

- 1. 環境構築: [Arduino IDE クイックスタート](#)を参考に IDE をインストールし、使用する開発ボードに対応するボードマネージャーと必要なドライバライブラリをインストールしてください。
- 2. 使用するドライバライブラリ:
 - [M5Unified](#)
 - [M5GFX](#)
 - [M5Unit-ACSSR](#)
- 3. 使用するハードウェア製品:
 - [CoreS3](#)
 - [Unit ACSSR / Unit DCSSR](#)
 - [Unit RS485-ISO](#)



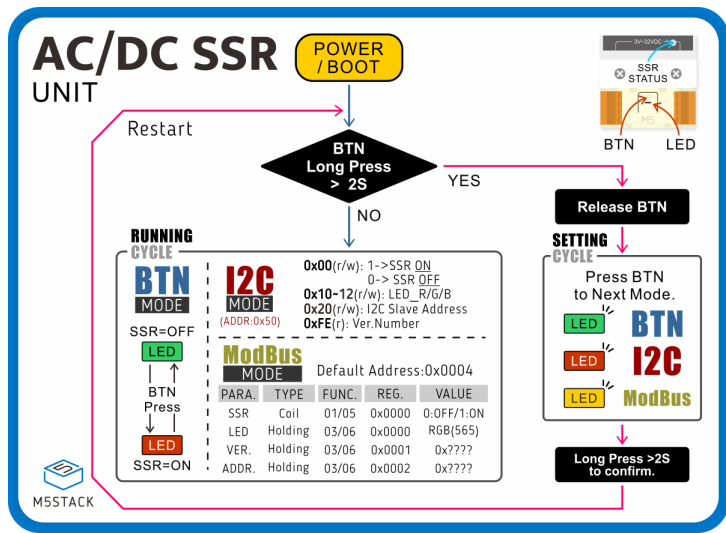
2. 動作モード

動作モード設定

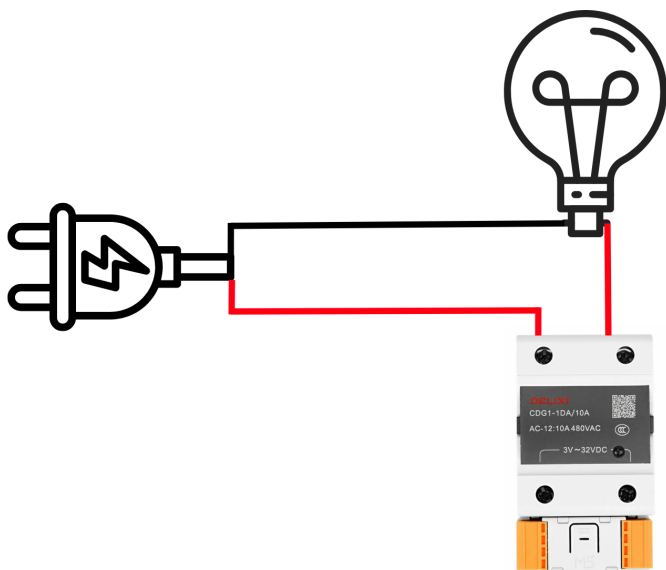
使用前に以下の説明フローチャートに従って動作モードを設定してください

- 1. デバイスの電源を切った状態で、中央のボタンを長押しします。
- 2. デバイスに電源を入れると、RGB LED が高速点滅し、モード設定状態に入ります。
- 3. ボタンを 1 回押してモードを切り替えます:
 - 緑: ボタン手動モード。動作中は中央ボタンでスイッチを制御。
 - 赤: I2C 通信モード。動作中は HY2.0-4P Grove 端子から I2C バス接続で制御。
 - 黄: Modbus 通信モード。動作中は両端の HT3.96-4P 端子から Modbus バス接続で制御。
- 4. 長押しで現在のモードを選択保存します。

[unit_dc SSR mode_change_guide_01.mp4](#)



○ 負荷接続例



3. I2C モード制御

デバイスを I2C 動作モードに設定後、HY2.0-4P Grove 端子から I2C バスに接続して制御します。本例では CoreS3 の PORT.A 端子で通信を行っています。他のコントローラを使用する場合は、接続ピンに応じてプログラムを修正してください。

```
1  #include "M5Unified.h"
2  #include "M5_ACSSR.h"
3
4  M5_ACSSR SSR;
5  void printStatus();
6
7  void setup()
8  {
9      M5.begin();
10     Serial.begin(115200);
11     while (!SSR.begin(&Wire, 2, 1, ACSSR_DEFAULT_ADDR)) {
12         M5.Display.println("ACSSR I2C INIT ERROR");
13         Serial.println("ACSSR I2C INIT ERROR");
14         delay(1000);
15     }
16     M5.Display.setFont(&fonts::FreeMonoBold12pt7b);
17 }
18
19 void loop()
20 {
21     SSR.on();
22     SSR.setLEDColour(0xff0000);
23     printStatus();
24
25     delay(1000);
26
27     SSR.off();
28     SSR.setLEDColour(0x00ff00);
29     printStatus();
30
31     delay(1000);
32 }
33
34 void printStatus()
35 {
36     int relay    = SSR.status();
37     int color    = SSR.getLEDColour();
38     int version  = SSR.getVersion();
39
40     M5.Display.clear();
41     M5.Display.setCursor(0, 0);
42     M5.Display.println("I2C Mode Control");
43     if (relay != -1) {
44         Serial.printf("Relay: %s\n", relay ? "ON" : "OFF");
45         M5.Display.printf("Relay: %s\n", relay ? "ON" : "OFF");
46     }
47
48     if (color != -1) {
49         Serial.printf("LED Color: 0x%04X\n", color);
50         M5.Display.printf("LED Color: 0x%04X\n", color);
```

```
51     }  
52  
53     if (version != -1) {  
54         Serial.printf("FW Version: 0x%04X\n", version);  
55         M5.Display.printf("FW Version: 0x%04X\n", version);  
56     }  
57 }
```



4. Modbus モード制御

デバイスを Modbus 動作モードに設定後、両端の HT3.96-4P 端子から Modbus バスに接続して制御します。本例では CoreS3 PORT.A と Unit RS485-ISO を使用し、RS485 通信を行っています。

```
1  #include "M5Unified.h"
2  #include "M5_ACSSR.h"
3  #include <ArduinoModbus.h>
4  #include <ArduinoRS485.h>
5
6  RS485Class RS485(Serial2, 1, 2, -1, -1);
7  uint8_t slave_id = 0;
8
9  void relayOn();
10 void relayOff();
11 void setLedColor(uint16_t color);
12 int getRelayStatus();
13 int getLedColor();
14 int getFirmwareVersion();
15 void printStatus();
16
17 void setup()
18 {
19     M5.begin();
20     Serial.begin(115200);
21     delay(2000);
22
23     Wire.end();
24     ModbusRTUClient.begin(115200, SERIAL_8N1);
25
26     slave_id = ACSSR_DEFAULT_SLAVE_ID;
27     M5.Display.setFont(&font::FreeMonoBold12pt7b);
28 }
29
30 void loop()
31 {
32     relayOn();
33     setLedColor(0xF800); // RED
34     printStatus();
35
36     delay(1000);
37
38     relayOff();
39     setLedColor(0x07E0); // GREEN
40     printStatus();
41
42     delay(1000);
43 }
44
45 void relayOn()
46 {
47     ModbusRTUClient.coilWrite(slave_id, ACSSR_RELAY_COIL_ADDR, 0xFF);
48 }
49
50 void relayOff()
```

```

51 {
52     ModbusRTUClient.coilWrite(slave_id, ACSSR_RELAY_COIL_ADDR, 0x00);
53 }
54
55 void setLedColor(uint16_t color)
56 {
57     ModbusRTUClient.holdingRegisterWrite(slave_id, ACSSR_LED_HOLDING_ADDR, color);
58 }
59
60 int getRelayStatus()
61 {
62     if (ModbusRTUClient.requestFrom(slave_id, COILS, ACSSR_RELAY_COIL_ADDR, 1)) {
63         while (ModbusRTUClient.available()) {
64             return ModbusRTUClient.read();
65         }
66     }
67     return -1;
68 }
69
70 int getLedColor()
71 {
72     if (ModbusRTUClient.requestFrom(slave_id, HOLDING_REGISTERS, ACSSR_LED_HOLDING_ADDR, 1)) {
73         while (ModbusRTUClient.available()) {
74             return ModbusRTUClient.read();
75         }
76     }
77     return -1;
78 }
79
80 int getFirmwareVersion()
81 {
82     if (ModbusRTUClient.requestFrom(slave_id, HOLDING_REGISTERS, ACSSR_VERSION_HOLDING_ADDR, 1)) {
83         while (ModbusRTUClient.available()) {
84             return ModbusRTUClient.read();
85         }
86     }
87     return -1;
88 }
89
90 void printStatus()
91 {
92     int relay    = getRelayStatus();
93     int color    = getLedColor();
94     int version  = getFirmwareVersion();
95
96     M5.Display.clear();
97     M5.Display.setCursor(0, 0);
98     M5.Display.println("Modbus Mode Control");
99     if (relay != -1) {
100         Serial.printf("Relay: %s\n", relay ? "ON" : "OFF");
101         M5.Display.printf("Relay: %s\n", relay ? "ON" : "OFF");
102     }
103 }

```

```

104     if (color != -1) {
105         Serial.printf("LED Color: 0x%04X\n", color);
106         M5.Display.printf("LED Color: 0x%04X\n", color);
107     }
108
109     if (version != -1) {
110         Serial.printf("FW Version: 0x%04X\n", version);
111         M5.Display.printf("FW Version: 0x%04X\n", version);
112     }
113 }

```

