

Unit 4Relay Arduino 使用教程

1. 准备工作

- 环境配置：参考 [Arduino IDE 上手教程](#) 完成 IDE 安装，并根据实际使用的开发板安装对应的板管理，与需要的驱动库。
- 使用到的驱动库：
 - [M5Unified](#)
 - [M5GFX](#)
 - [M5Unit_RELAY](#)
- 使用到的硬件产品：
 - [Basic v2.7](#)
 - [Unit 4Relay](#)



2. 注意事项

引脚兼容性

由于每款主机的引脚配置不同，为了让用户更方便地使用，M5Stack 官方提供了引脚兼容性表，方便用户查看，请根据实际引脚连接情况修改案例程序。

3. 案例程序

- 本教程中使用的主控设备为 Basic V2.7，搭配 Unit 4Relay。本 Relay 模块采用 I2C 的方式通讯，根据实际的电路链接修改程序中的引脚定义，设备连接后对应的引脚为 `G21(SDA)`，`G22(SCL)`。

注意

本模块只能在同步模式下控制继电器，如果在异步模式下对继电器进行控制将无效，只能控制 LED 指示灯。

cpp

```

1  #include <M5Unified.h>
2  #include "Unit_4RELAY.h"
3
4  UNIT_4RELAY relay;
5
6  namespace {
7  constexpr uint16_t COLOR_BG      = 0x1082;
8  constexpr uint16_t COLOR_SURFACE = 0x2104;
9  constexpr uint16_t COLOR_HEADER  = 0x0349;
10 constexpr uint16_t COLOR_BORDER  = 0x52AA;
11 constexpr uint16_t COLOR_MUTED   = 0xBDF7;
12 constexpr uint16_t COLOR_GREEN   = 0x2E8B;
13 constexpr uint16_t COLOR_RED     = 0xD249;
14 constexpr uint16_t COLOR_AMBER   = 0xFD20;
15
16 constexpr int16_t CHANNEL_Y      = 148;
17 constexpr int16_t CHANNEL_WIDTH  = 68;
18 constexpr int16_t CHANNEL_HEIGHT = 43;
19 constexpr int16_t CHANNEL_GAP    = 8;
20
21 uint8_t stepIndex = 0;
22 bool syncMode     = false;
23 bool allOn        = false;
24 bool relayState[4] = {false, false, false, false};
25 bool ledState[4]   = {false, false, false, false};
26
27 void drawModeStatus()
28 {
29     M5.Display.fillRect(0, 38, 320, 36, COLOR_BG);
30     M5.Display.setFont(&fonts::Font2);
31     M5.Display.setTextDatum(textdatum_t::middle_left);
32     M5.Display.setTextColor(COLOR_MUTED, COLOR_BG);
33     M5.Display.drawString("MODE", 12, 56);
34     M5.Display.drawString("CONTROL", 161, 56);
35
36     const uint16_t modeColor = syncMode ? COLOR_GREEN : COLOR_AMBER;
37     M5.Display.fillRoundRect(57, 44, 82, 25, 4, modeColor);
38     M5.Display.fillRoundRect(231, 44, 77, 25, 4, COLOR_SURFACE);
39     M5.Display.drawRoundRect(231, 44, 77, 25, 4, COLOR_BORDER);
40
41     M5.Display.setTextDatum(textdatum_t::middle_center);
42     M5.Display.setTextColor(TFT_BLACK, modeColor);
43     M5.Display.drawString(syncMode ? "SYNC" : "ASYN", 98, 56);
44     M5.Display.setTextColor(TFT_WHITE, COLOR_SURFACE);
45     M5.Display.drawString(syncMode ? "RELAY" : "LED", 269, 56);
46 }
47
48 void drawAction(const char *message)
49 {
50     M5.Display.fillRoundRect(10, 81, 300, 38, 4, COLOR_SURFACE);
51     M5.Display.drawRoundRect(10, 81, 300, 38, 4, COLOR_BORDER);
52     M5.Display.setFont(&fonts::Font2);
53     M5.Display.setTextDatum(textdatum_t::middle_left);
54     M5.Display.setTextColor(COLOR_MUTED, COLOR_SURFACE);
55     M5.Display.drawString("LAST ACTION", 20, 100);

```

```

56     M5.Display.setTextColor(TFT_WHITE, COLOR_SURFACE);
57     M5.Display.drawString(message, 119, 100);
58 }
59
60 void drawChannel(uint8_t channel)
61 {
62     const bool isOn      = syncMode ? relayState[channel] : ledState[channel];
63     const uint16_t color = isOn ? COLOR_GREEN : COLOR_BORDER;
64     const int16_t x      = 10 + channel * (CHANNEL_WIDTH + CHANNEL_GAP);
65     char channelLabel[5] = {};
66     snprintf(channelLabel, sizeof(channelLabel), "CH%u",
67              static_cast<unsigned>(channel + 1));
68
69     M5.Display.fillRoundRect(x, CHANNEL_Y, CHANNEL_WIDTH, CHANNEL_HEIGHT, 4,
70                             COLOR_SURFACE);
71     M5.Display.drawRoundRect(x, CHANNEL_Y, CHANNEL_WIDTH, CHANNEL_HEIGHT, 4,
72                             color);
73     M5.Display.setTextDatum(textdatum_t::middle_center);
74     M5.Display.setFont(&fonts::Font2);
75     M5.Display.setTextColor(COLOR_MUTED, COLOR_SURFACE);
76     M5.Display.drawString(channelLabel, x + CHANNEL_WIDTH / 2, CHANNEL_Y + 12);
77     M5.Display.setTextColor(isOn ? COLOR_GREEN : TFT_WHITE, COLOR_SURFACE);
78     M5.Display.drawString(isOn ? "ON" : "OFF", x + CHANNEL_WIDTH / 2,
79                          CHANNEL_Y + 31);
80 }
81
82 void drawAllChannels()
83 {
84     M5.Display.fillRect(0, 125, 320, 22, COLOR_BG);
85     M5.Display.setFont(&fonts::Font2);
86     M5.Display.setTextDatum(textdatum_t::middle_left);
87     M5.Display.setTextColor(COLOR_MUTED, COLOR_BG);
88     M5.Display.drawString(syncMode ? "RELAY STATUS" : "LED STATUS", 10, 136);
89
90     for (uint8_t channel = 0; channel < 4; ++channel) {
91         drawChannel(channel);
92     }
93 }
94
95 void drawButtonHint(int16_t x, char button, const char *label, uint16_t color)
96 {
97     char buttonLabel[2] = {button, '\0'};
98
99     M5.Display.fillCircle(x, 220, 11, color);
100    M5.Display.setFont(&fonts::Font2);
101    M5.Display.setTextDatum(textdatum_t::middle_center);
102    M5.Display.setTextColor(TFT_BLACK, color);
103    M5.Display.drawString(buttonLabel, x, 220);
104    M5.Display.setTextDatum(textdatum_t::middle_left);
105    M5.Display.setTextColor(TFT_WHITE, COLOR_SURFACE);
106    M5.Display.drawString(label, x + 16, 220);
107 }
108
109 void drawMainUi()
110 {

```

```

111 M5.Display.fillScreen(COLOR_BG);
112 M5.Display.fillRect(0, 0, 320, 34, COLOR_HEADER);
113 M5.Display.setFont(&font4::Font4);
114 M5.Display.setTextDatum(textdatum_t::middle_center);
115 M5.Display.setTextColor(TFT_WHITE, COLOR_HEADER);
116 M5.Display.drawString("UNIT 4RELAY", 160, 17);
117
118 drawModeStatus();
119 drawAction("Ready");
120 drawAllChannels();
121
122 M5.Display.fillRect(0, 202, 320, 38, COLOR_SURFACE);
123 M5.Display.drawFastHLine(0, 202, 320, COLOR_BORDER);
124 drawButtonHint(15, 'A', "STEP", COLOR_GREEN);
125 drawButtonHint(121, 'B', "MODE", COLOR_AMBER);
126 drawButtonHint(229, 'C', "ALL", COLOR_RED);
127 }
128
129 void updateStateAfterAllWrite(bool state)
130 {
131     // relayAll() and ledAll() overwrite the complete output register.
132     for (uint8_t channel = 0; channel < 4; ++channel) {
133         relayState[channel] = syncMode ? state : false;
134         ledState[channel] = syncMode ? false : state;
135     }
136 }
137 } // namespace
138
139 void setup()
140 {
141     M5.begin();
142     Serial.begin(115200);
143
144     M5.Display.setRotation(1);
145     relay.begin(&Wire, 21, 22);
146     relay.Init(false); // Start in asynchronous mode with all outputs off.
147     drawMainUi();
148 }
149
150 void loop()
151 {
152     M5.update();
153
154     if (M5.BtnA.wasPressed()) {
155         const uint8_t channel = stepIndex % 4;
156         const bool state = stepIndex < 4;
157         char message[20] = {};
158
159         if (syncMode) {
160             relay.relayWrite(channel, state);
161             relayState[channel] = state;
162         } else {
163             relay.ledWrite(channel, state);
164             ledState[channel] = state;
165         }

```

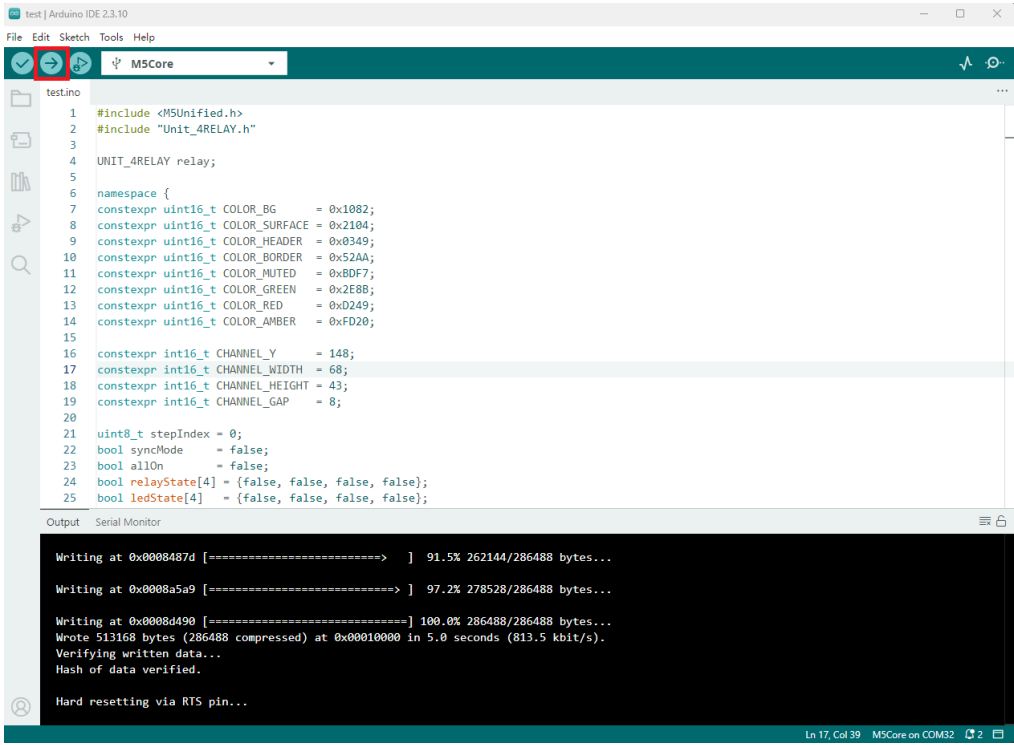
```

166
167     snprintf(message, sizeof(message), "%s %u %s",
168             syncMode ? "Relay" : "LED",
169             static_cast<unsigned>(channel + 1),
170             state ? "ON" : "OFF");
171     drawAction(message);
172     drawChannel(channel);
173     stepIndex = (stepIndex + 1) % 8;
174 }
175
176 if (M5.BtnB.wasPressed()) {
177     syncMode = !syncMode;
178     relay.switchMode(syncMode);
179     drawModeStatus();
180     drawAction(syncMode ? "Sync mode enabled" : "Async mode enabled");
181     drawAllChannels();
182 }
183
184 if (M5.BtnC.wasPressed()) {
185     allOn = !allOn;
186     if (syncMode) {
187         relay.relayAll(allOn);
188     } else {
189         relay.ledAll(allOn);
190     }
191
192     updateStateAfterAllWrite(allOn);
193     drawAction(allOn ? "All channels ON" : "All channels OFF");
194     drawAllChannels();
195 }
196 }

```

4. 编译上传

- 复制粘贴上述例程代码到项目代码区，选中设备端口（详情请参考 [程序编译与烧录](#)），点击 Arduino IDE 左上角编译上传按钮，等待程序完成编译并上传至设备。



5. 继电器控制

- 按键 A 可切换单个继电器控制模式，按键 B 可切换指示灯和继电器的同步 / 异步模式，按键 C 可全开 / 关继电器及 LED 指示灯。

