

Unit 8Angle Arduino User Guide

1. Preparations

- 1. Environment Setup: Refer to the [Arduino IDE Getting Started Tutorial](#) to complete the IDE installation, and install the corresponding board manager and required driver libraries according to your development board.
- 2. Driver Libraries Used:
 - [M5Unified](#)
 - [M5GFX](#)
 - [M5Unit-8Angle](#)
- 3. Hardware Products Used:
 - [CoreS3](#)
 - [Unit 8Angle](#)



2. Example

Example Explanation

Unit 8Angle is an input unit that integrates 8 adjustable potentiometers, along with 9 programmable RGB LEDs and a toggle switch. In this example, we use the CoreS3 main controller via its PORT.A expansion interface to read the input status of each potentiometer channel and the status of the toggle switch, while also controlling the RGB LEDs.

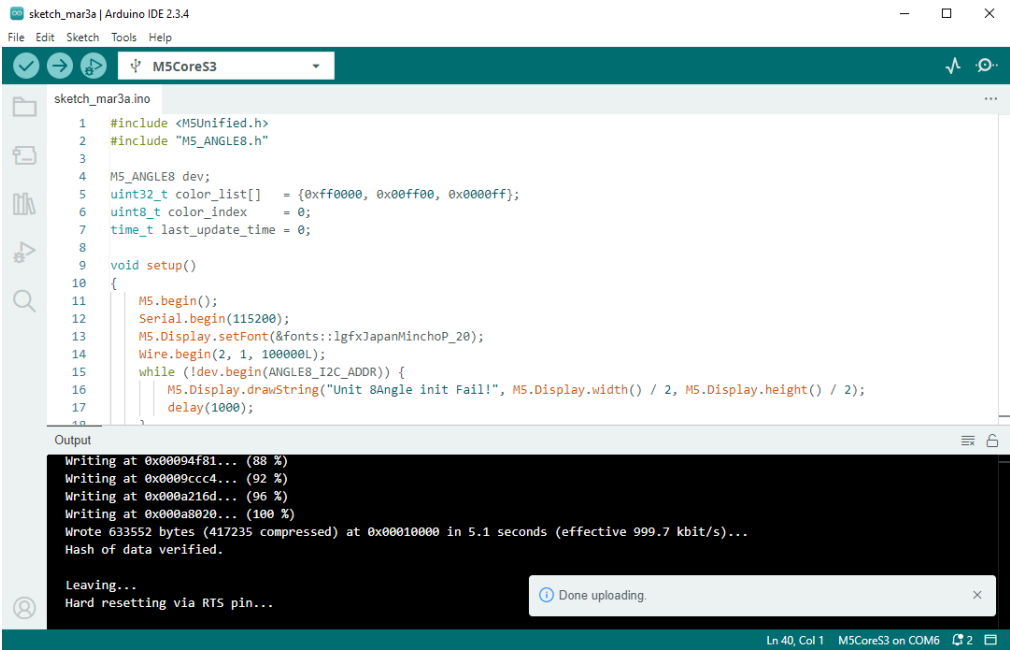
```
1  #include <M5Unified.h>
2  #include "M5_ANGLE8.h"
3
4  M5_ANGLE8 dev;
5  uint32_t color_list[] = {0xff0000, 0x00ff00, 0x0000ff};
6  uint8_t color_index = 0;
7  time_t last_update_time = 0;
8
9  void setup()
10 {
11     M5.begin();
12     Serial.begin(115200);
13     M5.Display.setFont(&fonts::lgfxJapanMinchoP_20);
14     Wire.begin(2, 1, 100000L);
15     while (!dev.begin(ANGLE8_I2C_ADDR)) {
16         M5.Display.drawString("Unit 8Angle init Fail!", M5.Display.width() / 2, M5.Display.height() ,
17             delay(1000);
18     }
19 }
20
21 void loop()
22 {
23     bool switch_status = dev.getDigitalInput();
24     M5.Display.clear();
25     M5.Display.setCursor(0, 0);
26     M5.Display.printf("Switch: %s\r\n", switch_status ? "ON" : "OFF");
27     for (uint8_t i = 0; i < ANGLE8_TOTAL_ADC; i++) {
28         uint16_t adc_v = dev.getAnalogInput(i, _12bit);
29         M5.Display.printf("CH[%d]: %d\r\n", i, adc_v);
30     }
31     if (millis() - last_update_time > 1000) {
32         color_index = (color_index + 1) % 3;
33         for (uint8_t i = 0; i < ANGLE8_TOTAL_LED; i++) {
34             dev.setLEDColour(i, color_list[color_index], 80);
35         }
36         last_update_time = millis();
37     }
38     delay(100);
39 }
```

3. Compile and Upload

- 1. Download Mode: Different devices require a download mode before flashing the program, and this step may vary depending on the main controller. For details, please refer to the device programming download tutorial list at the bottom of the [Arduino IDE Getting Started Tutorial](#) page for specific instructions.
- For the CoreS3, press and hold the reset button (approximately 2 seconds) until the internal green LED lights up, then release the button. At this point, the device will enter download mode and wait for flashing.



- 2. Select the device port, click the compile and upload button in the upper left corner of the Arduino IDE, and wait for the program to compile and upload to the device.



4. Reading Knob Status

Read and display the statuses of the potentiometers and the switch, while also controlling the RGB LED color to change every 1 second.

