

Unit MIC 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](#)
- 使用到的硬件产品：
 - [Core2 v1.1](#)
 - [Unit MIC](#)



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 Core2 v1.1，搭配 Unit MIC。请根据实际的电路连接修改程序中的引脚定义，设备连接后对应的引脚为 **G33 (AIN)**，**G32 (DIN)**。

```
1  #include <M5Unified.h>
2  #include <M5GFX.h>
3
4  // Input pins
5  #define AIN_PIN 33      // Analog input pin
6  #define DIN_PIN 32      // Digital input pin (ADC)
7  #define LCD_W 320
8  #define LCD_H 240
9  #define MAX_LEN LCD_W  // Buffer length (matches screen width)
10 #define X_OFFSET 0
11
12 #define Y_AIN_OFFSET 60  // Vertical offset for AIN waveform
13 #define Y_DIN_OFFSET 150 // Vertical offset for DIN waveform
14 #define X_SCALE 1        // Amplitude scaling factor
15
16 // Draw real-time waveforms and values for AIN and DIN channels
17 static void draw_waveform_dual() {
18     static int16_t ain_buf[MAX_LEN] = {0}; // Buffer for AIN samples
19     static int16_t din_buf[MAX_LEN] = {0}; // Buffer for DIN samples
20     static int16_t pt = MAX_LEN - 1;      // Circular buffer pointer
21
22     int ainValue = analogRead(AIN_PIN);    // Read AIN ADC value
23     int dinValue = analogRead(DIN_PIN);    // Read DIN ADC value
24
25     // Map AIN: higher ADC values produce higher positions
26     ain_buf[pt] = map((int16_t)(ainValue * X_SCALE), 1800, 4095, 0, 80);
27
28     // Map DIN: ADC=0 at waveform bottom, ADC=4095 at waveform top
29     din_buf[pt] = map((int16_t)(dinValue * X_SCALE), 0, 4095, 80, 0);
30
31     if (--pt < 0) pt = MAX_LEN - 1;        // Circular buffer wrap-around
32
33     for (int i = 1; i < MAX_LEN; i++) {
34         uint16_t now_pt = (pt + i) % MAX_LEN;
35
36         // Erase previous AIN waveform points
37         M5.Lcd.drawLine(
38             i + X_OFFSET,
39             ain_buf[(now_pt + 1) % MAX_LEN] + Y_AIN_OFFSET,
40             i + 1 + X_OFFSET,
41             ain_buf[(now_pt + 2) % MAX_LEN] + Y_AIN_OFFSET,
42             TFT_BLACK);
43
44         // Erase previous DIN waveform points
45         M5.Lcd.drawLine(
46             i + X_OFFSET,
47             din_buf[(now_pt + 1) % MAX_LEN] + Y_DIN_OFFSET,
48             i + 1 + X_OFFSET,
49             din_buf[(now_pt + 2) % MAX_LEN] + Y_DIN_OFFSET,
50             TFT_BLACK);
```

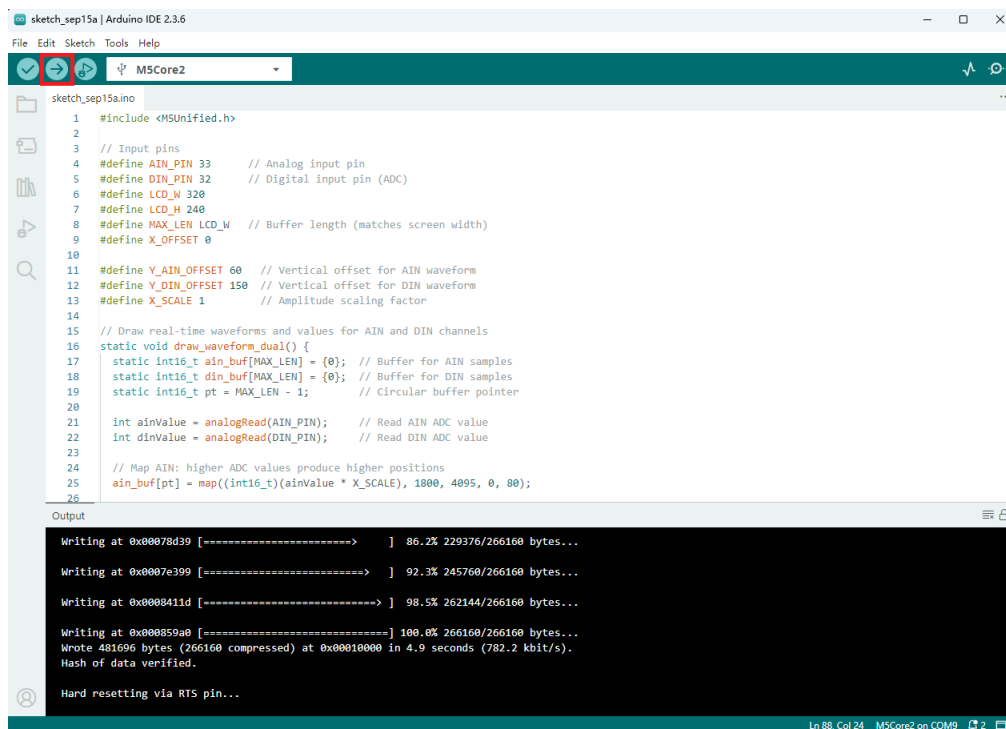
```

51
52     if (i < MAX_LEN - 1) {
53         // Draw current AIN waveform in green
54         M5.Lcd.drawLine(
55             i + X_OFFSET,
56             ain_buf[now_pt] + Y_AIN_OFFSET,
57             i + 1 + X_OFFSET,
58             ain_buf[(now_pt + 1) % MAX_LEN] + Y_AIN_OFFSET,
59             TFT_GREEN);
60
61         // Draw current DIN waveform in cyan
62         M5.Lcd.drawLine(
63             i + X_OFFSET,
64             din_buf[now_pt] + Y_DIN_OFFSET,
65             i + 1 + X_OFFSET,
66             din_buf[(now_pt + 1) % MAX_LEN] + Y_DIN_OFFSET,
67             TFT_CYAN);
68     }
69 }
70
71 // Display current AIN ADC value (upper area)
72 M5.Lcd.fillRect(10, Y_AIN_OFFSET - 25, 100, 20, TFT_BLACK);
73 M5.Lcd.setTextColor(TFT_GREEN, TFT_BLACK);
74 M5.Lcd.setFont(&fonts::FreeMonoBold9pt7b);
75 M5.Lcd.setTextDatum(TL_DATUM);
76 M5.Lcd.drawString("AIN: " + String(ainValue), 10, Y_AIN_OFFSET - 24);
77
78 // Display current DIN ADC value (lower area)
79 M5.Lcd.fillRect(10, Y_DIN_OFFSET + 80 - 25, 100, 20, TFT_BLACK);
80 M5.Lcd.setTextColor(TFT_CYAN, TFT_BLACK);
81 M5.Lcd.drawString("DIN: " + String(dinValue), 10, Y_DIN_OFFSET + 80 - 24);
82 }
83
84 void setup() {
85     M5.begin();
86     M5.Lcd.setFont(&fonts::FreeMonoBold9pt7b);
87     M5.Lcd.setTextDatum(top_center);
88
89     // AIN waveform label
90     M5.Lcd.setTextColor(TFT_GREEN, TFT_BLACK);
91     M5.Lcd.setTextDatum(top_center);
92     M5.Lcd.drawString("AIN Wave", 220, Y_AIN_OFFSET - 24);
93
94     // DIN waveform label (near bottom)
95     M5.Lcd.setTextColor(TFT_CYAN, TFT_BLACK);
96     M5.Lcd.drawString("DIN Wave", 220, Y_DIN_OFFSET + 80 - 24);
97 }
98
99 void loop() {
100     draw_waveform_dual();
101 }

```

4. 编译上传

- 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。



```
sketch_sep15a.ino
1 #include <M5Unified.h>
2
3 // Input pins
4 #define AIN_PIN 33 // Analog input pin
5 #define DIN_PIN 32 // Digital input pin (ADC)
6 #define LCD_W 320
7 #define LCD_H 240
8 #define MAX_LEN LCD_W // Buffer length (matches screen width)
9 #define X_OFFSET 0
10
11 #define V_AIN_OFFSET 60 // Vertical offset for AIN waveform
12 #define V_DIN_OFFSET 150 // Vertical offset for DIN waveform
13 #define X_SCALE 1 // Amplitude scaling factor
14
15 // Draw real-time waveforms and values for AIN and DIN channels
16 static void draw_waveform_dual() {
17     static int16_t ain_buf[MAX_LEN] = {0}; // Buffer for AIN samples
18     static int16_t din_buf[MAX_LEN] = {0}; // Buffer for DIN samples
19     static int16_t pt = MAX_LEN - 1; // Circular buffer pointer
20
21     int ainValue = analogRead(AIN_PIN); // Read AIN ADC value
22     int dinValue = analogRead(DIN_PIN); // Read DIN ADC value
23
24     // Map AIN: higher ADC values produce higher positions
25     ain_buf[pt] = map((int16_t)ainValue * X_SCALE, 1800, 4095, 0, 80);
26
27     // Draw waveforms and values (omitted for brevity)
28 }
```

Output

```
Writing at 0x00078d39 [=====>] 86.2% 229376/266160 bytes...
Writing at 0x0007e399 [=====>] 92.3% 245760/266160 bytes...
Writing at 0x0008411d [=====>] 98.5% 262144/266160 bytes...
Writing at 0x000859a0 [=====] 100.0% 266160/266160 bytes...
Wrote 481696 bytes (266160 compressed) at 0x00010000 in 4.9 seconds (782.2 kbit/s).
Hash of data verified.
Hard resetting via RTS pin...
```

5. 麦克风功能效果展示

- 该例程效果为实时绘制音频波形和采集到的 ADC 值，主控设备显示如下图所示。

