

Unit LCD Arduino Tutorial

1. Preparation

- Environment setup: Refer to the [Arduino IDE Getting Started Guide](#) to complete the IDE installation, then install the board package and driver libraries required for the development board you are using.
- Libraries used:
 - [M5Unified](#)
 - [M5GFX](#)
- `M5UnitLCD` is included in M5GFX and does not need to be installed separately.
- Hardware products used:
 - [CoreS3](#)
 - [Unit LCD](#)



2. Notes

Pin Compatibility

Since the pin configuration differs between host devices, M5Stack provides a pin compatibility table for reference. Check the actual pin connections and modify the example program as needed.

3. Example Program

- This tutorial uses CoreS3 as the host device with Unit LCD. The example follows the general graphics drawing method used by M5Unified and uses `draw_function(LovyanGFX* gfx)` to draw random circles and squares on Unit LCD. Unit LCD communicates over I2C with a default address of `0x3E` and a communication frequency of 400 kHz. When connected to CoreS3's Port A, the corresponding pins are `G2 (SDA)` and `G1 (SCL)`.

```
cpp
```

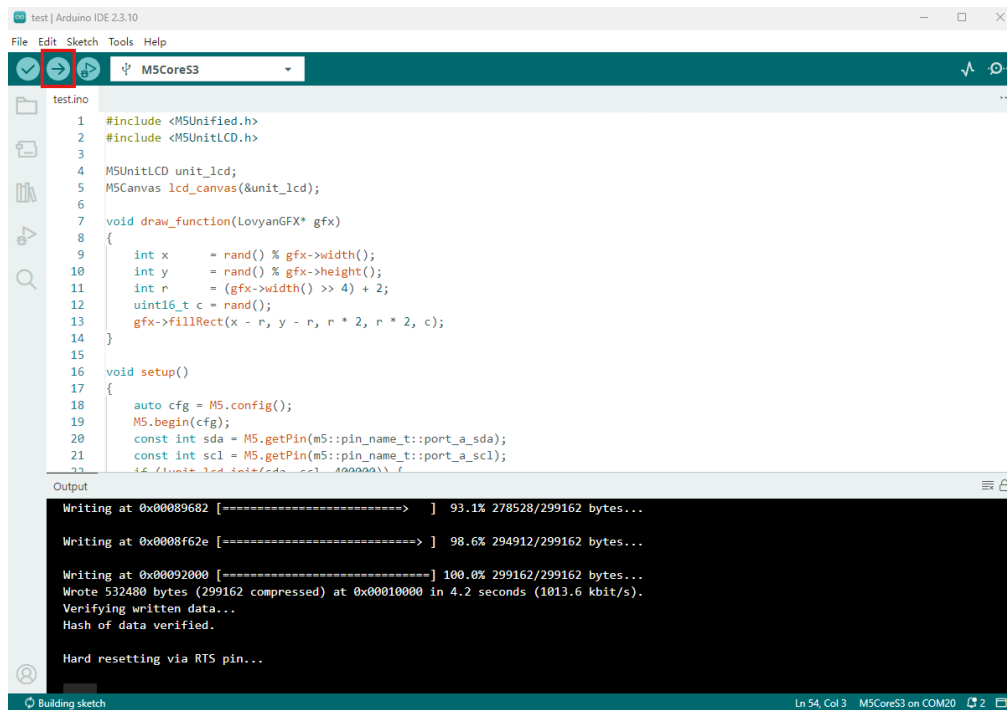
```

1  #include <M5Unified.h>
2  #include <M5UnitLCD.h>
3
4  M5UnitLCD unit_lcd;
5  M5Canvas lcd_canvas(&unit_lcd);
6
7  void draw_function(LovyanGFX* gfx)
8  {
9      int x      = rand() % gfx->width();
10     int y      = rand() % gfx->height();
11     int r      = (gfx->width() >> 4) + 2;
12     uint16_t c = rand();
13     gfx->fillRect(x - r, y - r, r * 2, r * 2, c);
14 }
15
16 void setup()
17 {
18     auto cfg = M5.config();
19     M5.begin(cfg);
20     const int sda = M5.getPin(m5::pin_name_t::port_a_sda);
21     const int scl = M5.getPin(m5::pin_name_t::port_a_scl);
22     if (!unit_lcd.init(sda, scl, 400000)) {
23         M5.Display.fillScreen(TFT_RED);
24         M5.Display.setTextSize(2);
25         M5.Display.setCursor(12, 32);
26         M5.Display.println("Unit LCD init failed");
27         while (true) {
28             delay(1000);
29         }
30     }
31
32     unit_lcd.setRotation(3);
33     lcd_canvas.setColorDepth(16);
34     lcd_canvas.createSprite(unit_lcd.width(), unit_lcd.height());
35     lcd_canvas.fillScreen(TFT_BLACK);
36     int textsize = lcd_canvas.height() / 60;
37     if (textsize == 0) {
38         textsize = 1;
39     }
40     lcd_canvas.setTextSize(textsize);
41 }
42
43 void loop()
44 {
45     int x      = rand() % lcd_canvas.width();
46     int y      = rand() % lcd_canvas.height();
47     int r      = (lcd_canvas.width() >> 4) + 2;
48     uint16_t c = rand();
49     lcd_canvas.fillCircle(x, y, r, c);
50     draw_function(&lcd_canvas);
51     lcd_canvas.pushSprite(0, 0);
52     unit_lcd.waitDisplay();
53     delay(50);
54 }

```

4. Compile and Upload

- Copy and paste the example program above into the project code editor, select the device port (see [Program Compile & Upload](#) for details), then click the Compile and Upload button in the upper-left corner of Arduino IDE and wait for the program to compile and upload to the device.



The screenshot shows the Arduino IDE 2.3.10 interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for opening files, saving, and uploading. The 'Upload' button, represented by a right-pointing arrow, is highlighted with a red box. Below the toolbar, the 'Sketch' dropdown menu is set to 'M5CoreS3'. The main code editor displays the following C++ code:

```
testino
1 #include <M5Unified.h>
2 #include <M5UnitLCD.h>
3
4 M5UnitLCD unit_lcd;
5 M5Canvas lcd_canvas(&unit_lcd);
6
7 void draw_function(LovyanGFX* gfx)
8 {
9     int x = rand() % gfx->width();
10    int y = rand() % gfx->height();
11    int r = (gfx->width() >> 4) + 2;
12    uint16_t c = rand();
13    gfx->fillRect(x - r, y - r, r * 2, r * 2, c);
14 }
15
16 void setup()
17 {
18     auto cfg = M5.config();
19     M5.begin(cfg);
20     const int sda = M5.getPin(m5::pin_name_t::port_a_sda);
21     const int scl = M5.getPin(m5::pin_name_t::port_a_scl);
22     if (!unit_lcd.init(sda, scl, 400000)) {
23 
```

The 'Output' window at the bottom shows the upload progress and status:

```
Writing at 0x00089682 [=====> ] 93.1% 278528/299162 bytes...
Writing at 0x0008f62e [=====> ] 98.6% 294912/299162 bytes...
Writing at 0x00092000 [=====] 100.0% 299162/299162 bytes...
Wrote 532480 bytes (299162 compressed) at 0x00010000 in 4.2 seconds (1013.6 kbit/s).
Verifying written data...
Hash of data verified.
Hard resetting via RTS pin...
```

The status bar at the bottom indicates 'Building sketch' and 'Ln 54, Col 3 M5CoreS3 on COM20'.

5. Random Graphics Display

- After the program starts, CoreS3 continuously overlays random circles and squares on the Unit LCD canvas.

