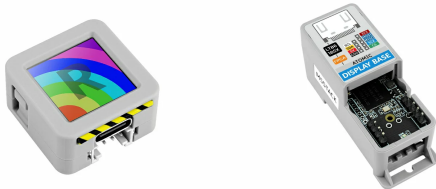


Atomic Display Base Arduino 使用チュートリアル

1. 準備作業

- 1. 開発環境の設定: [Arduino IDE 入門チュートリアル](#)を参考に IDE をインストールし、使用する開発ボードに応じてボードマネージャーと必要なライブラリをインストールしてください。
- 2. 使用ライブラリ:
 - [M5Unified](#)
 - [M5GFX](#)
- 3. 使用するハードウェア製品:
 - [AtomS3R](#)
 - [Atomic Display Base](#)



2. サンプルプログラム

[M5Unifiedライブラリ](#)に含まれる[Displaysサンプル](#)を参考に、本チュートリアルで使用するデバイスに合わせて設定を一部変更しました。使用する外部ディスプレイの解像度に応じて、13 行目と 14 行目のコードを変更してください。

ディスプレイ互換性 | Atomic Display Base は、解像度の自動スケーリングに対応したモニターと一緒に使用する必要があります。一部の非対応ディスプレイでは、表示に異常が発生する可能性があります。

```
1  #include <Arduino.h>
2  #include <M5AtomDisplay.h>
3  #include <M5Unified.h>
4
5  void setup() {
6      auto cfg = M5.config();
7
8      // external display setting. (Pre-include required)
9      cfg.external_display.atom_display = true; // default=true. use AtomDisplay
10
11     // Set individual parameters for external displays.
12     // (※ Use only the items you wish to change. Basically, it can be omitted.)
13     cfg.atom_display.logical_width = 1920;
14     cfg.atom_display.logical_height = 1200;
15     // cfg.atom_display.output_width   = 1920;
16     // cfg.atom_display.output_height  = 1080;
17     // cfg.atom_display.refresh_rate   = 60;
18     // cfg.atom_display.scale_w        = 2;
19     // cfg.atom_display.scale_h        = 2;
20     // cfg.atom_display.pixel_clock    = 74250000;
21
22     // begin M5Unified.
23     M5.begin(cfg);
24
25     // Get the number of available displays
26     int display_count = M5.getDisplayCount();
27
28     for (int i = 0; i < display_count; ++i) {
29         // All displays are available in M5.Displays.
30         // ※ Note that the order of which displays are numbered is the order in which they are detected,
31
32         M5.Displays(i).clear();
33         int textsize = M5.Displays(i).height() / 120;
34         if (textsize < 5) { textsize = 3; }
35         M5.Displays(i).setTextSize(textsize);
36         M5.Displays(i).printf("\n\nNo.%d\n\n", i);
37     }
38
39     // If an external display is to be used as the main display, it can be listed in order of priority
40     M5.setPrimaryDisplayType({
41         m5::board_t::board_M5AtomDisplay,
42         // m5::board_t::board_M5ModuleDisplay,
43     });
44
45     // The primary display can be used with M5.Display.
46     M5.Display.print("primary display\n\n");
47
48     // Examine the indexes of a given type of display
49     int index_atom_display = M5.getDisplayIndex(m5::board_t::board_M5AtomDisplay);
50
```

```

51     if (index_atom_display >= 0) {
52         M5.Displays(index_atom_display).print("This is Atom Display\n");
53     }
54
55     M5.delay(2500);
56 }
57
58 void loop() {
59     M5.delay(100);
60
61     int x = rand() % M5.Displays(0).width();
62     int y = rand() % M5.Displays(0).height();
63     int r = (M5.Displays(0).width() >> 2) + 2;
64     uint16_t c = rand();
65     M5.Displays(0).fillCircle(x, y, r, c);
66
67     draw_function(&M5.Displays(1));
68 }
69
70 // When creating a function for drawing, it can be used universally by accepting a LovyanGFX type as
71 void draw_function(LovyanGFX* gfx) {
72     int x = rand() % gfx->width();
73     int y = rand() % gfx->height();
74     int r = (gfx->width() >> 6) + 2;
75     uint16_t c = rand();
76     gfx->fillRect(x - r, y - r, r * 2, r * 2, c);
77 }

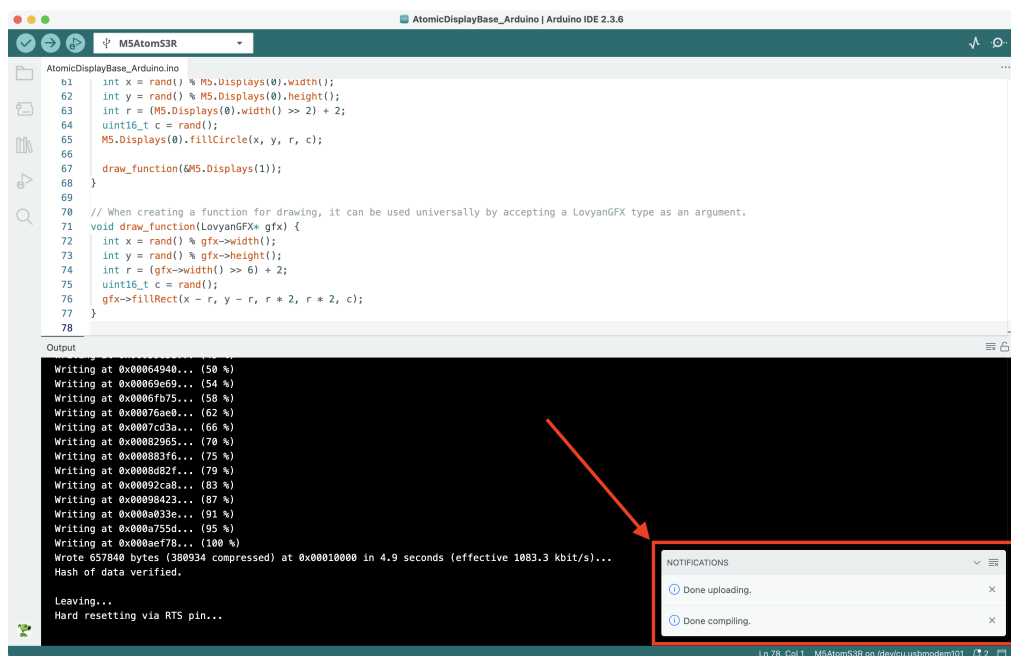
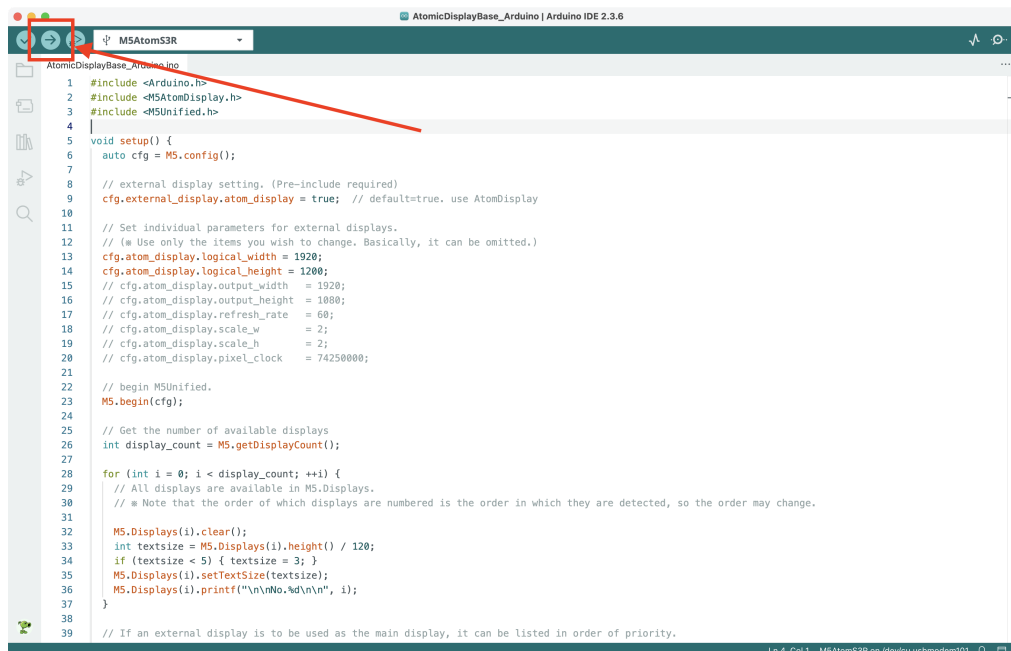
```

3. コンパイルと書き込み

- 1. ダウンロードモード：プログラムを書き込む前に、メインコントローラーをダウンロードモードに切り替える必要があります。デバイスによって手順が異なるため、詳細は[Arduino IDE 入門チュートリアル](#)のページ下部にあるデバイス別の書き込み手順を参照してください。
- AtomS3R の場合、リセットボタンを約 2 秒間長押しして、内部の緑色 LED が点灯したらボタンを離します。この状態でダウンロードモードに入り、書き込み待機状態になります。



- 2.Arduino IDE の左上にあるコンパイル / アップロードボタンをクリックし、使用するポートを選択します。プログラムがコンパイルされ、デバイスに書き込まれるのを待ちます。



4. 実行開始

AtomS3R と Atomic Display Base を組み合わせ、高精細ビデオ拡張インターフェースを介してディスプレイに接続した後、リセットボタンを短く 1 回押してください。AtomS3R 本体の画面には多数のカラフルな円が表示され、外部ディスプレイには多数のカラフルな四角形が表示されます。表示効果は以下のとおりです。

