

迁移至M5Unified

本文档简要概述如何从其他的M5Stack主控驱动库迁移到M5Unified。

与现有M5Stack库的主要区别

- 1.在之前的M5Stack库中, M5.begin()在不同的设备库中的参数内容和数量都不同。为了统一它们, M5Unified使用结构体作为参数。用于传入不同的配置信息。
- 2.M5Unified不集成RGB LED驱动, 需要单独安装库文件[FastLED](#)。
- 3.SDCard将不会集成默认初始化, 为了支持ESP-IDF将默认不使用SD.h, 内部不执行SD.begin()。
- 4.图形库从TFT-eSPI更改为M5GFX, 绘图API将不兼容。关于M5GFX库的更多信息, 请参考M5GFX库的教程[Getting Started with M5GFX](#)
- 5.扬声器功能相对之前的库有所改进。此外, 集成蜂鸣器的型号同样支持播放音频文件等。
- 6.Button名称的区别, 为了通用起见, 即使设备只有一个按钮, 名称也是BtnA而不是Btn。

M5Unified使用注意事项

注意事项

M5Unified不能现有的M5Stack主控驱动库结合使用。因为包含同名实例M5, 编译时会出现如下错误。

cpp



```
1 error: reference to 'jpeg_div_t' is ambiguous jpeg_div_t scale = JPEG_DIV_NONE);
2 error: conflicting declaration 'M5Stack M5' extern M5Stack M5;
```

M5.begin变更

早期的驱动库 M5.begin()对于每个设备具有不同的内容和参数数量, 如下所示。

cpp



```
1 // M5Stack
2 M5.begin(bool LCDEnable = true, bool SDEnable = true, bool SerialEnable = true, bool I2CE
3
4 // M5Core2
5 M5.begin(bool LCDEnable = true, bool SDEnable = true, bool SerialEnable = true, bool I2CE
6
7 // M5StickC
8 M5.begin(bool LCDEnable = true, bool PowerEnable = true, bool SerialEnable = true);
9
10 // M5StickCPlus
11 M5.begin(bool LCDEnable = true, bool PowerEnable = true, bool SerialEnable = true);
12
13 // M5Atom
14 M5.begin(bool SerialEnable = true, bool I2CEnable = true, bool DisplayEnable = false);
```

更改为Unified后, 一些初始化的配置信息将在cfg结构体中实现。

cpp



```
1 auto cfg = M5.config();
2
3 // Set the items you want to configure. Omit the following two lines if you use the default
4 cfg.serial_baudrate = 115200;
5 cfg.output_power = true;
6
7 M5.begin(cfg);
```

```
- M5.begin(true, true, true, true);
+ auto cfg = M5.config();
+ cfg.internal_imu = true;
+
+ M5.begin(cfg);
```

| M5.Btn变更

可用的按钮数量因不同的M5Stack产品有所不同, 主要使用以下命名。

- 一般按钮组:
 - BtnA, BtnB, BtnC
- 电源按钮组:
 - BtnPWR
- 拓展按钮组:
 - BtnEXT

变更案例：按下按钮A时，显示消息"BtnA isPressed"; 释放时，显示消息"BtnA isReleased"。

cpp



```
1  #include <M5Unified.h>
2
3  void setup() {
4      auto cfg = M5.config();
5      cfg.clear_display = true;
6      M5.begin(cfg);
7      M5.Lcd.setTextSize(2);
8      M5.Lcd.println("Press BtnA");
9  }
10
11 void loop() {
12     // update the button state.
13     M5.update();
14
15     if (M5.BtnA.isPressed()) {
16         M5.Display.setCursor(0, 16);
17         M5.Display.println("BtnA isPressed ");
18     } else {
19         M5.Display.setCursor(0, 16);
20         M5.Display.println("BtnA isReleased");
21     }
22 }
```

需要通过执行M5.update()来更新按钮状态。



```
1  #include <M5Unified.h>
2
3  uint32_t loop_count = 0;
4
5  void setup() {
6      auto cfg = M5.config();
7      cfg.clear_display = true;
8      M5.begin(cfg);
9      M5.Lcd.setTextSize(2);
10     M5.Lcd.println("Press BtnA");
11 }
12
13 void loop() {
14
15     M5.update(); // update the button state.
16
17     if (M5.BtnA.wasPressed()) {
18         M5.Display.fillScreen(TFT_BLACK);
19         M5.Display.setCursor(0, 0);
20         M5.Display.println("BtnA wasPressed");
21         M5.Display.println("Press BtnA Stop Count");
22
23         while (true) {
24
25             M5.update(); // update the button state within the subloop.
26
27             M5.Display.setCursor(0, 32);
28             M5.Display.printf("count: %10d\n", loop_count);
29             if (M5.BtnA.wasPressed()) {
30                 M5.Display.println("Count Stop");
31                 loop_count = 0;
32                 break;
33             }
34             loop_count++;
35         }
36     }
37 }
```

更多信息，请参考[M5Unified Button Class](#)。

| M5.Lcd变更

M5Unified依赖于[M5GFX library](#)，在Arduino IDE中在安装M5Unified时会弹窗提示安装该依赖，请点击安装全部。

变更案例：该案例程序兼容M5Stack使用LCD或e-Ink显示屏的主控，不同的屏幕类型将使用同一套绘图API。

cpp



```
1  #include <M5Unified.h>
2
3  void setup() {
4
5      auto cfg = M5.config();
6      M5.begin(cfg);
7      M5.Display.setTextSize(3);
8      M5.Display.println("HelloWorld!");
9
10 }
11
12 void loop() {
13 }
```

| SDCard变更

更改为M5Unified.h后将不会自带包含SD.h, 使用时需要手动添加 `#include <SD.h>`

```
- #include <M5Stack.h>
+ #include <SD.h>
+ #include <M5Unified.h>
```

添加SD.begin()

```
+ while (false == SD.begin(GPIO_NUM_4, SPI, 25000000))
+ {
+     delay(500);
+ }
```

| 内部I2C释放

M5Unified将默认初始化内部I2C, 对于PORT.A接口复用为内部I2C的设备如Basic/Gray/Fire。若需要将PORT.A映射为其他功能, 则需要调用 `M5.In_I2C.release()` 。

cpp



```
1  auto cfg = M5.config();
2  .
3  .
4  .
5  M5.begin(cfg);
6  M5.In_I2C.release();
```

| M5Unified迁移案例

如果将M5Stack库中的Example FactoryTest.ino改成M5Unified，就会变成下面这样。

- 更改头文件
- 更改IMU流程
- 添加SD.begin
- 添加SPEAKER_PIN的定义
- 更改了M5.Btn.pressedFor的参数

```

diff --git a/src/main.cpp b/src/main.cpp
index b735e30..dae9b99 100644
--- a/src/main.cpp
+++ b/src/main.cpp
@@ -1,10 +1,10 @@
#include <Arduino.h>
-#include <M5Stack.h>
+#include <SD.h>
+#include <M5Unified.h>
#include <stdlib.h>
// #include "FastLED.h"

#include "WiFi.h"
-#include "utility/MPU9250.h"

extern const unsigned char gImage_logoM5[];
extern const unsigned char m5stack_startup_music[];
@@ -13,7 +13,8 @@ extern const unsigned char m5stack_startup_music[];
#define min(a, b) (((a) < (b)) ? (a) : (b))
#endif

-MPU9250 IMU;
+
+#define SPEAKER_PIN 25

// #define LEDS_PIN 15
// #define LEDS_NUM 10
@@ -127,15 +128,15 @@ void writeFile(fs::FS &fs, const char *path, const char *message) {
}

void buttons_test() {
-   if (M5.BtnA.wasReleased() || M5.BtnA.pressedFor(1000, 200)) {
+   if (M5.BtnA.wasReleased() || M5.BtnA.pressedFor(1000)) {
       M5.Lcd.printf("A");
       Serial.printf("A");
   }
-   if (M5.BtnB.wasReleased() || M5.BtnB.pressedFor(1000, 200)) {
+   if (M5.BtnB.wasReleased() || M5.BtnB.pressedFor(1000)) {
       M5.Lcd.printf("B");
       Serial.printf("B");
   }
-   if (M5.BtnC.wasReleased() || M5.BtnC.pressedFor(1000, 200)) {
+   if (M5.BtnC.wasReleased() || M5.BtnC.pressedFor(1000)) {
       M5.Lcd.printf("C");
       Serial.printf("C");
   }
}
@@ -475,8 +476,11 @@ void setup() {
    //   GPIO_test();
    // }

```

```

+   auto cfg = M5.config();
+   cfg.internal_imu = true;
+
    // initialize the M5Stack object
-   M5.begin();
+   M5.begin(cfg);

    /*
        Power chip connected to gpio21, gpio22, I2C device
@@ -484,6 +488,10 @@ void setup() {
    If used battery, please call this function in your project
    */
    M5.Power.begin();
+   while (false == SD.begin(GPIO_NUM_4, SPI, 25000000))
+   {
+       delay(500);
+   }

    // dac test
    // if (gpio_test_flg)
@@ -492,7 +500,6 @@ void setup() {
    // }
    startupLogo();
    // ledBar();
-   Wire.begin();

    // Lcd display
    M5.Lcd.setBrightness(100);
@@ -609,54 +616,33 @@ void setup() {
    M5.Lcd.setBrightness(i);
    delay(2);
}

-
-   byte c = IMU.readByte(MPU9250_ADDRESS, WHO_AM_I_MPU9250);
-   Serial.print("MPU9250 ");
-   Serial.print("I AM ");
-   Serial.print(c, HEX);
-   Serial.print(" I should be ");
-   Serial.println(0x71, HEX);
-   Serial.println("");
-   M5.Lcd.setCursor(20, 0);
-   M5.Lcd.print("MPU9250");
-   M5.Lcd.setCursor(0, 10);
-   M5.Lcd.print("I AM");
-   M5.Lcd.setCursor(0, 20);
-   M5.Lcd.print(c, HEX);
-   M5.Lcd.setCursor(0, 30);
-   M5.Lcd.print("I Should Be");

```

```

- M5.Lcd.setCursor(0, 40);
- M5.Lcd.println(0x71, HEX);
- M5.Lcd.println();
- delay(100);
-
- IMU.initMPU9250();
- // Initialize device for active mode read of acclerometer, gyroscope, and
- // temperature
- Serial.println("MPU9250 initialized for active data mode...");
-
- // Read the WHO_AM_I register of the magnetometer, this is a good test of
- // communication
- byte d = IMU.readByte(AK8963_ADDRESS, WHO_AM_I_AK8963);
- Serial.print("AK8963 ");
- Serial.print("I AM ");
- Serial.print(d, HEX);
- Serial.print(" I should be ");
- Serial.println(0x48, HEX);
-
- // M5.Lcd.fillScreen(BLACK);
- M5.Lcd.setCursor(20, 100);
- M5.Lcd.print("AK8963");
- M5.Lcd.setCursor(0, 110);
- M5.Lcd.print("I AM");
- M5.Lcd.setCursor(0, 120);
- M5.Lcd.print(d, HEX);
- M5.Lcd.setCursor(0, 130);
- M5.Lcd.print("I Should Be");
- M5.Lcd.setCursor(0, 140);
- M5.Lcd.print(0x48, HEX);
+ const char* name;
+ // run-time branch : imu model check
+ switch (M5.Imu.getType())
+ {
+     case m5::imu_t::imu_mpu6050:
+         name = "MPU6050";
+         break;
+     case m5::imu_t::imu_mpu6886:
+         name = "MPU6886";
+         break;
+     case m5::imu_t::imu_mpu9250:
+         name = "MPU9250";
+         break;
+     case m5::imu_t::imu_sh200q:
+         name = "SH200Q";
+         break;
+     default:
+         name = "none";
+         break;

```

```
+ }  
+ M5.Display.print("IMU:");  
+ M5.Display.println(name);  
+ M5.Display.endWrite();  
+ ESP_LOGI("setup", "imu:%s", name);  
  delay(1000);  
  
+  
  M5.Lcd.setCursor(0, 0);  
  M5.Lcd.println("wifi test:");  
  M5.Lcd.fillScreen(BLACK);
```