

CoreS3 RTC 实时时钟

CoreS3 RTC 时钟相关 API 与案例程序。

| 案例程序

| 编译要求

- M5Stack 板管理版本 $\geq 3.2.2$
- 开发板选项 = M5CoreS3
- M5Unified 库版本 $\geq 0.2.11$

```
1  #if defined(ARDUINO)
2
3  #define WIFI_SSID      "*****"
4  #define WIFI_PASSWORD "*****"
5  #define NTP_TIMEZONE   "UTC-8"
6  #define NTP_SERVER1    "0.pool.ntp.org"
7  #define NTP_SERVER2    "1.pool.ntp.org"
8  #define NTP_SERVER3    "2.pool.ntp.org"
9
10 #include <WiFi.h>
11
12 // Different versions of the framework have different SNTP header file names and
13 // availability.
14 #if __has_include(<esp_sntp.h>)
15 #include <esp_sntp.h>
16 #define SNTP_ENABLED 1
17 #elif __has_include(<sntp.h>)
18 #include <sntp.h>
19 #define SNTP_ENABLED 1
20 #endif
21
22 #endif
23
24 #ifndef SNTP_ENABLED
25 #define SNTP_ENABLED 0
26 #endif
27
28 #include <M5Unified.h>
29
30 void setup(void) {
31     M5.begin();
32     M5.Display.setFont(&fonts::FreeMonoBold12pt7b);
33
34     if (!M5.Rtc.isEnabled()) {
35         Serial.println("RTC not found.");
36         for (;;) {
37             vTaskDelay(500);
38         }
39     }
40
41     Serial.println("RTC found.");
42
43     // It is recommended to set UTC for the RTC and ESP32 internal clocks.
44     /* /// setup RTC ( direct setting )
45         //
46         //          YYYY MM DD      hh mm ss
47         M5.Rtc.setDateTime( { { 2021, 12, 31 }, { 12, 34, 56 } } );
48
49     ///*/
50
51     /// setup RTC ( NTP auto setting )
```

```

51
52     M5.Display.println("WiFi Connecting...");
53
54     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
55     while (WiFi.status() != WL_CONNECTED) {
56         Serial.print('.');
57         delay(500);
58     }
59     M5.Display.println("WiFi Connected.");
60     Serial.println("\r\n WiFi Connected.");
61
62     configTzTime(NTP_TIMEZONE, NTP_SERVER1, NTP_SERVER2, NTP_SERVER3);
63
64     #if SNTP_ENABLED
65         while (sntp_get_sync_status() != SNTP_SYNC_STATUS_COMPLETED) {
66             Serial.print('.');
67             delay(1000);
68         }
69         Serial.println("\r\n NTP Connected.");
70         M5.Display.println("NTP Connected");
71     #else
72         delay(1600);
73         struct tm timeInfo;
74         while (!getLocalTime(&timeInfo, 1000)) {
75             Serial.print('.');
76         };
77     #endif
78
79     time_t t = time(nullptr) + 1; // Advance one second.
80     while (t > time(nullptr))
81         ; // Synchronization in seconds
82     M5.Rtc.setDateTime(gmtime(&t));
83     M5.Display.clear();
84 }
85
86 void loop(void) {
87     static constexpr const char* const wd[7] = {"Sun", "Mon", "Tue", "Wed",
88                                                  "Thr", "Fri", "Sat"};
89
90     delay(500);
91
92     auto dt = M5.Rtc.getDateTime();
93     Serial.printf("RTC   UTC   :%04d/%02d/%02d (%s)  %02d:%02d:%02d\r\n",
94                  dt.date.year, dt.date.month, dt.date.date,
95                  wd[dt.date.weekDay], dt.time.hours, dt.time.minutes,
96                  dt.time.seconds);
97     M5.Display.setCursor(0, 0);
98     M5.Display.printf("RTC   UTC   :%04d/%02d/%02d (%s)  %02d:%02d:%02d\n",
99                      dt.date.year, dt.date.month, dt.date.date,
100                      wd[dt.date.weekDay], dt.time.hours, dt.time.minutes,
101                      dt.time.seconds);
102
103     /// ESP32 internal timer

```

```

104     auto t = time(nullptr);
105     {
106         auto tm = gmtime(&t); // for UTC.
107         Serial.printf("ESP32 UTC   :%04d/%02d/%02d (%s)  %02d:%02d:%02d\r\n",
108             tm->tm_year + 1900, tm->tm_mon + 1, tm->tm_mday,
109             wd[tm->tm_wday], tm->tm_hour, tm->tm_min, tm->tm_sec);
110         M5.Display.printf("ESP32 UTC   :%04d/%02d/%02d (%s)  %02d:%02d:%02d\n",
111             tm->tm_year + 1900, tm->tm_mon + 1, tm->tm_mday,
112             wd[tm->tm_wday], tm->tm_hour, tm->tm_min,
113             tm->tm_sec);
114     }
115
116     {
117         auto tm = localtime(&t); // for local timezone.
118         Serial.printf("ESP32 %s:%04d/%02d/%02d (%s)  %02d:%02d:%02d\r\n",
119             NTP_TIMEZONE, tm->tm_year + 1900, tm->tm_mon + 1,
120             tm->tm_mday, wd[tm->tm_wday], tm->tm_hour, tm->tm_min,
121             tm->tm_sec);
122         M5.Display.printf("ESP32 %s:%04d/%02d/%02d (%s)  %02d:%02d:%02d\n",
123             NTP_TIMEZONE, tm->tm_year + 1900, tm->tm_mon + 1,
124             tm->tm_mday, wd[tm->tm_wday], tm->tm_hour,
125             tm->tm_min, tm->tm_sec);
126     }
127 }

```

烧录成功后，CoreS3 屏幕上会显示当前的 RTC 时间以及 ESP32 内部时钟的时间信息。RTC 时间是通过 NTP 同步设置的，而 ESP32 内部时钟则是通过系统时间函数获取的，它们应该是同步的。



API

CoreS3 RTC 时钟部分使用了 M5Unified 库中的 RTC8563_Class, 更多相关的 API 可以参考下方文档:

- [M5Unified - RTC8563 Class](#)