

Station RTC

Station RTC に関連するAPIとサンプルプログラムです。

| サンプルプログラム

| コンパイル要件

- M5Stack ボードマネージャ バージョン $\geq 2.1.4$
- 開発ボードの選択 = M5Fire
- M5Unified ライブラリ バージョン $\geq 0.2.5$
- M5GFX ライブラリ バージョン $\geq 0.2.7$

```
1  #if defined(ARDUINO)
2  #define WIFI_SSID      "YOUR WIFI SSID NAME"
3  #define WIFI_PASSWORD "YOUR WIFI PASSWORD"
4  #define NTP_TIMEZONE   "UTC-8"
5  #define NTP_SERVER1    "0.pool.ntp.org"
6  #define NTP_SERVER2    "1.pool.ntp.org"
7  #define NTP_SERVER3    "2.pool.ntp.org"
8  #include <WiFi.h>
9
10 // Different versions of the framework have different SNTP header file names and availability.
11 #if __has_include(<esp_sntp.h>)
12 #include <esp_sntp.h>
13 #define SNTP_ENABLED 1
14 #elif __has_include(<sntp.h>)
15 #include <sntp.h>
16 #define SNTP_ENABLED 1
17 #endif
18
19 #endif
20
21 #ifndef SNTP_ENABLED
22 #define SNTP_ENABLED 0
23 #endif
24
25 #include <M5Unified.h>
26
27 void setup(void) {
28     M5.begin();
29     Serial.begin(115200);
30
31     M5.Display.setRotation(1);
32     M5.Display.setFont(&fonts::FreeMonoBold9pt7b);
33     if (!M5.Rtc.isEnabled()) {
34         Serial.println("RTC not found.");
35         M5.Display.println("RTC not found.");
36         for (;;) {
37             vTaskDelay(500);
38         }
39     }
40
41     Serial.println("RTC found.");
42     M5.Display.println("RTC found.\n");
43
44     // It is recommended to set UTC for the RTC and ESP32 internal clocks.
45     // setup RTC ( direct setting )
46     ////          YYYY  MM  DD      hh  mm  ss
47     // M5.Rtc.setDateTime( { { 2021, 12, 31 }, { 12, 34, 56 } } );
48     // setup RTC ( NTP auto setting )
49
50     M5.Display.print("WiFi:");
```

```

51  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
52  while (WiFi.status() != WL_CONNECTED) {
53      Serial.print('.');
54      delay(500);
55  }
56  Serial.println("\r\n WiFi Connected.");
57  M5.Display.print("Connected.");
58
59  configTzTime(NTP_TIMEZONE, NTP_SERVER1, NTP_SERVER2, NTP_SERVER3);
60
61  #if SNTP_ENABLED
62      while (sntp_get_sync_status() != SNTP_SYNC_STATUS_COMPLETED) {
63          Serial.print('.');
64          delay(1000);
65      }
66  #else
67      delay(1600);
68      struct tm timeInfo;
69      while (!getLocalTime(&timeInfo, 1000)) {
70          Serial.print('.');
71      };
72  #endif
73
74      Serial.println("\r\n NTP Connected.");
75
76      time_t t = time(nullptr) + 1; // Advance one second.
77      while (t > time(nullptr)); // Synchronization in seconds
78      M5.Rtc.setDateTime(gmtime(&t));
79      M5.Display.clear();
80  }
81
82  void loop(void) {
83      static constexpr const char* const wd[7] = { "Sun", "Mon", "Tue", "Wed", "Thr", "Fri", "Sat" };
84
85      delay(500);
86
87      //RTC chip
88      auto dt = M5.Rtc.getDateTime();
89      Serial.printf("RTC   UTC   :%04d/%02d/%02d (%s)  %02d:%02d:%02d\r\n", dt.date.year, dt.date.month,
90                    wd[dt.date.weekDay], dt.time.hours, dt.time.minutes, dt.time.seconds);
91      M5.Display.setCursor(0, 0);
92      M5.Display.printf("RTC   UTC   :%04d/%02d/%02d (%s)  %02d:%02d:%02d", dt.date.year, dt.date.month,
93                      wd[dt.date.weekDay], dt.time.hours, dt.time.minutes, dt.time.seconds);
94
95      /// ESP32 internal timer
96      auto t = time(nullptr);
97      {
98          auto tm = gmtime(&t); // for UTC.
99          Serial.printf("ESP32 UTC   :%04d/%02d/%02d (%s)  %02d:%02d:%02d\r\n", tm->tm_year + 1900, tm->tm_
100                        tm->tm_mday, wd[tm->tm_wday], tm->tm_hour, tm->tm_min, tm->tm_sec);
101          M5.Display.setCursor(0, 40);
102          M5.Display.printf("ESP32 UTC   :%04d/%02d/%02d (%s)  %02d:%02d:%02d", tm->tm_year + 1900, tm->tm_
103                        tm->tm_mday, wd[tm->tm_wday], tm->tm_hour, tm->tm_min, tm->tm_sec);

```

```

104 }
105
106 {
107     auto tm = localtime(&t); // for local timezone.
108     Serial.printf("ESP32 %s:%04d/%02d/%02d (%s) %02d:%02d:%02d\r\n", NTP_TIMEZONE, tm->tm_year + 1970,
109                   tm->tm_mon + 1, tm->tm_mday, wd[tm->tm_wday], tm->tm_hour, tm->tm_min, tm->tm_sec);
110     M5.Display.setCursor(0, 80);
111     M5.Display.printf("ESP32 %s:%04d/%02d/%02d (%s) %02d:%02d:%02d", NTP_TIMEZONE, tm->tm_year + 1970,
112                      tm->tm_mon + 1, tm->tm_mday, wd[tm->tm_wday], tm->tm_hour, tm->tm_min, tm->tm_sec);
113 }
114 }

```

プログラム内に接続するWi-FiのSSIDとパスワードを入力し、アップロードボタンをクリックすると、リアルタイムクロックが表示されます。



API

Station RTC に関連するAPIは、 **M5Unified** ライブラリの **RTC8563_Class** を使用しています。詳細なAPIについては、以下のドキュメントを参照してください。

- [M5Unified - RTC8563 Class](#)