

StickS3 Wakeup

StickS3 sleep/wakeup related APIs and example program.

Example

Compilation Requirements

- M5Stack Board Manager version $\geq 3.2.5$
- Development board option = M5StickS3
- M5Unified library version $\geq 0.2.12$
- M5GFX library version $\geq 0.2.18$

cpp

```
1  #include "M5Unified.h"
2
3  void setup(void) {
4      M5.begin();
5
6      M5.Display.setTextDatum(middle_center);
7      M5.Display.setTextFont(&fonts::FreeMonoBoldOblique9pt7b);
8      M5.Display.setRotation(1);
9
10     M5.Display.drawString("BtnA to Sleep 5s", M5.Display.width() / 2, M5.Display.height() / 2);
11 }
12
13 void loop(void) {
14     M5.update();
15
16     if (M5.BtnA.wasPressed()) {
17         M5.Display.clear();
18         M5.Power.timerSleep(5); //sec
19     }
20 }
```

The example program runs as follows:

[StickS3_Arduino_wakeup.mp4](#)

API

The power management part of StickS3 uses the **Power_Class** from the **M5Unified** library. For more related APIs, you can refer to the following documentation:

- [M5Unified - Power Class](#)

PMIC Timer Wakeup

StickS3 can be driven using the [M5PM1 Arduino Library](#), enabling timed device wake-up through the PMIC. In this mode, power to peripherals such as the ESP32-S3 can be shut off, resulting in lower power consumption.

For more tutorials on using the StickS3 PMIC, please refer to [StickS3 PMIC Configuration](#).

| Compilation Requirements

- M5Stack Board Manager version $\geq 3.2.5$
- Development board option = M5StickS3
- M5Unified library version $\geq 0.2.12$
- M5GFX library version $\geq 0.2.18$
- M5PM1 library version $\geq 1.0.1$

Description: After the device is powered on, press button A to configure a 10-second timer to trigger a wake-up.

```
1  #include <M5Unified.h>
2  #include <M5PM1.h>
3  #include <Wire.h>
4
5  M5PM1 pm1;
6
7  void setup(void)
8  {
9      M5.begin();
10     M5.Display.setRotation(1);
11     Serial.begin(115200);
12     auto pin_num_sda = M5.getPin(m5::pin_name_t::in_i2c_sda);
13     auto pin_num_scl = M5.getPin(m5::pin_name_t::in_i2c_scl);
14     M5_LOGI("getPin: SDA:%u SCL:%u", pin_num_sda, pin_num_scl);
15     Wire.end();
16     Wire.begin(pin_num_sda, pin_num_scl, 1000000U);
17
18     // Initialize PM1
19     m5pm1_err_t err = pm1.begin(&Wire, M5PM1_DEFAULT_ADDR, pin_num_sda, pin_num_scl, M5PM1_I2C_FREQ_
20
21     if (err == M5PM1_OK) {
22         Serial.println("PM1 initialization successful");
23     } else {
24         Serial.printf("PM1 initialization failed, error code: %d\n", err);
25     }
26     M5.Display.fillScreen(BLACK);
27     M5.Display.setTextSize(2);
28     M5.Display.setTextColor(WHITE);
29     M5.Display.setCursor(0, 10);
30     M5.Display.println("Timer Power Test");
31     M5.Display.println("BtnA: After 10s Wakeup");
32 }
33
34 void loop(void)
35 {
36     M5.update();
37     if (M5.BtnA.wasPressed()) {
38         M5.Display.fillScreen(BLACK);
39         M5.Display.setCursor(0, 10);
40         M5.Display.println("Shutdown");
41         M5.Display.println("After 10s");
42         M5.Display.println("Wakeup");
43         delay(1000);
44         pm1.timerSet(10, M5PM1_TIM_ACTION_POWERON);
45         pm1.shutdown();
46     }
47 }
```