

CoreS3 Power Management

CoreS3 power management related APIs and example programs.

Example Program

Compilation Requirements

- M5Stack Board Manager version $\geq 3.2.2$
- Board selection = M5CoreS3
- M5Unified library version $\geq 0.2.11$

cpp

```
1  #include "M5Unified.h"
2
3  void setup()
4  {
5      auto cfg = M5.config();
6      // if using ext power input(Grove Port or DC input power supply) needs to be set to false.
7      // cfg.output_power = false;
8      M5.begin(cfg);
9      M5.Display.setTextSize(2);
10     M5.Power.setChargeCurrent(200);
11 }
12
13 void loop()
14 {
15     M5.Display.clear();
16
17     bool bat_ischarging = M5.Power.isCharging();
18     M5.Display.setCursor(10, 30);
19     M5.Display.printf("Bat Charging: %d", bat_ischarging);
20
21     int bat_vol = M5.Power.getBatteryVoltage();
22     M5.Display.setCursor(10, 50);
23     M5.Display.printf("Bat Voltage: %dmv", bat_vol);
24
25     int bat_level = M5.Power.getBatteryLevel();
26     M5.Display.setCursor(10, 70);
27     M5.Display.printf("Bat Level: %d", bat_level);
28
29     int vbus_vol = M5.Power.getVBUSVoltage();
30     M5.Display.setCursor(10, 90);
31     M5.Display.printf("VBus Voltage: %dmv", vbus_vol);
32     delay(1000);
33 }
```

After successful upload, you can view battery charging status, battery voltage, battery level percentage, and VBUS voltage information on CoreS3.



API

The CoreS3 power section uses the **Power_Class** from the M5Unified library. For more related APIs, please refer to the documentation below:

- [M5Unified - Power Class](#)