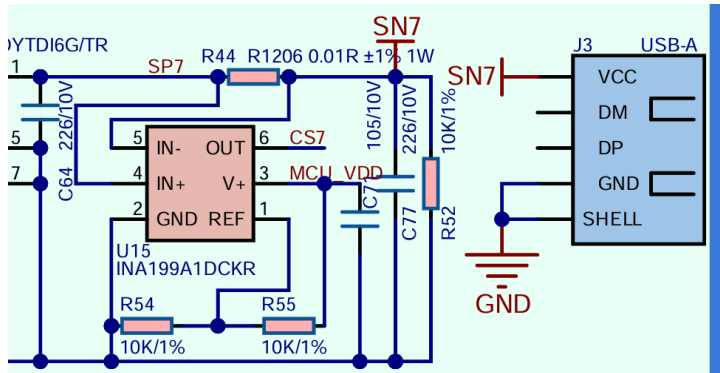


Station Grove Power 電源管理システムです

Station インターフェース電源管理関連 API とケースプログラムです。

USB-A ポート出力電流 の 計算原理

Station の USB-A ポートは出力電流情報の取得のみをサポートしており、GPIO34 を通じて INA199Ax1 の出力電圧値を取得して換算することで USB-A の出力電流値を求める必要があります。この電圧値は USB-A の出力電流値とほぼ比例しています。



上記の原理図から分かるように、INA199x1 の基準電圧 (REF) の理想値は: $(R54 \setminus (R54 + R55)) \text{MCU_VDD}$, すなわち $1 \setminus 2 \text{MCU_VDD}$ です。しかし、ハードウェア部品の許容誤差と本設計構造が精密構造ではないため、この基準電圧 (REF) の実際の値は実測が必要です。INA199 の [データシート](#) を参照すると、分流抵抗 (すなわち上図の R_{44}) の両端の電圧差を検出して電流を測定し、内部で設定されたゲイン (INA199x1 の場合 50V/V) , に基づいて、実時間で検出した電流信号を OUT ピンから対応するアナログ電圧信号 (すなわち以下のコードの `usb_vol`) 。に変換して出力することがわかります。USB-A ポートが無負荷の場合、INA199x1 の出力電圧値は基準電圧 (REF) の実際の値に近づき、すなわち以下のコードの `usb_vref` です。OUT ピンの出力電圧の計算式は $\text{usb_vol} = \text{usb_vref} + I_o * (R44 * 50)$ で、換算すると $I_o = (\text{usb_vol} - \text{usb_vref}) \setminus (R44 * 50)$ となります。具体的な実装は以下のコードを参照してください。

サンプルプログラム

```

1  #include <Arduino.h>
2  #include "M5Unified.h"
3
4  #define USB_PIN 34
5  const float V_REF = 3.3;    // Analog reference voltage (e.g., 5V or 3.3V) due to hardware
6  const float Res_BITS = 12.0; // ADC resolution (bits)
7  const float ADC_STEPS = (1 << int(Res_BITS)) - 1; // Number of steps (2^Res_BITS - 1)
8  float usb_vref; // INA199x1 REF
9
10 float get_USB_Volt(float num){
11     float volt;
12     for (size_t i = 0; i < num; i++) {
13         volt = volt + analogRead(USB_PIN);
14         delay(10);
15     }
16
17     volt = volt / num / ADC_STEPS * V_REF;
18     return volt; //unit: V
19 }
20
21 void setup()
22 {
23     auto cfg = M5.config();
24     M5.begin(cfg);
25     M5.Display.setTextDatum(middle_center);
26     M5.Display.setTextColor(TFT_BLACK);
27     M5.Display.setFont(&fonts::FreeSansOblique9pt7b);
28     M5.Display.setTextSize(1);
29     M5.Power.setExtOutput(false, (m5::ext_port_mask_t)(m5::ext_USB|m5::ext_PA|m5::ext_PB1|m5::ext_PB2));
30     analogReadResolution(12); //Set ADC Resolution
31     analogSetAttenuation(ADC_11db); //Set ADC Attenuation
32     M5.Lcd.clear(TFT_WHITE);
33     M5.Power.setExtOutput(true, (m5::ext_port_mask_t)(m5::ext_USB|m5::ext_PA|m5::ext_PB1|m5::ext_PB2));
34     delay(200);
35     usb_vref = get_USB_Volt(5); //Obtain the output voltage of INA199 when the USB is in a no-load state
36 }
37
38 void loop()
39 {
40     M5.Display.clear(TFT_WHITE);
41
42     bool isSupplying = M5.Power.getExtOutput();
43     if(isSupplying){
44         M5.Lcd.setCursor(0, 10);
45         float usb_vol = get_USB_Volt(5); //Obtain the real-time output voltage of INA199
46         float usb_current = ((usb_vol - usb_vref) / 50.0f / 0.01f * 1000.0f); //unit: mA  usb_vol = us
47         M5.Lcd.printf("USB-A %3.0f mA", usb_current);
48         M5.Lcd.setCursor(0, 30);
49         M5.Lcd.printf("A1%3.0f mA %1.1fV  A2%3.0f mA %1.1fV\n",
50                     M5.Power.Ina3221[0].getCurrent(0) * 1000,

```

```

51         M5.Power.Ina3221[0].getBusVoltage(0),
52         M5.Power.Ina3221[0].getCurrent(1) * 1000,
53         M5.Power.Ina3221[0].getBusVoltage(1));
54     M5.Lcd.setCursor(0, 50);
55     M5.Lcd.printf("B1%3.0fmA %1.1fV  B2%3.0fmA %1.1fV\n",
56         M5.Power.Ina3221[0].getCurrent(2) * 1000,
57         M5.Power.Ina3221[0].getBusVoltage(2),
58         M5.Power.Ina3221[1].getCurrent(0) * 1000,
59         M5.Power.Ina3221[1].getBusVoltage(0));
60     M5.Lcd.setCursor(0, 70);
61     M5.Lcd.printf("C1%3.0fmA %1.1fV  C2%3.0fmA %1.1fV\n",
62         M5.Power.Ina3221[1].getCurrent(1) * 1000,
63         M5.Power.Ina3221[1].getBusVoltage(1),
64         M5.Power.Ina3221[1].getCurrent(2) * 1000,
65         M5.Power.Ina3221[1].getBusVoltage(2));
66 }
67 delay(1000);
68 }

```

このプログラムは画面上に各 Grove のリアルタイムの出力電圧と電流情報を表示します。



API

Station の電源管理部分では M5Unified ライブラリ内の `Power_Class` を使用しています。詳細なAPIは以下のドキュメントを参照してください:

- [M5Unified - Power Class](#)

