

Unit AIN4-20mA Arduino 使用ガイド

1. 準備作業

- 環境設定: [Arduino IDE 入門チュートリアル](#)を参照し、IDE のインストールを完了し、使用する開発ボードに対応するボード管理と必要なドライバライブラリをインストールしてください。
- 必要なドライバライブラリ:
 - [M5Unified](#)
 - [M5GFX](#)
 - [M5Module-4-20mA](#) (本モジュールは Module13.2 AIN4-20mA と同じドライバライブラリを共有しています)

注意

GitHub から最新版のライブラリをダウンロードしてください: [M5Module-4-20mA - M5Stack GitHub](#)。Arduino ライブラリからはダウンロードしないでください (質問がある場合は[このチュートリアル](#)を参照)。

- 必要なハードウェア製品:
 - [CoreS3](#)
 - [Unit AIN4-20mA](#)

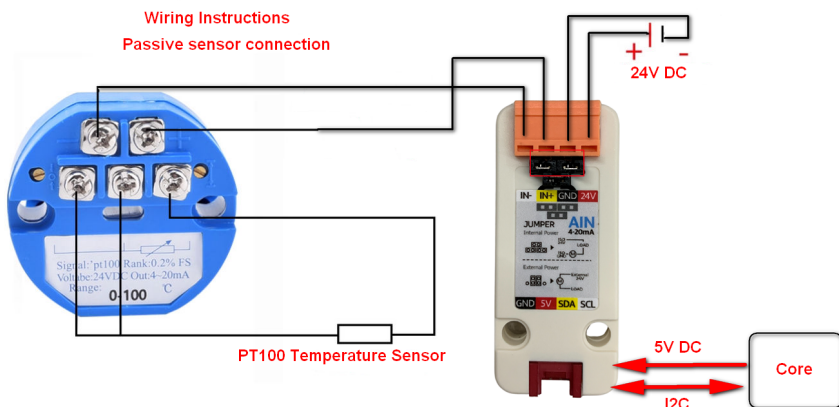


2. 注意事項

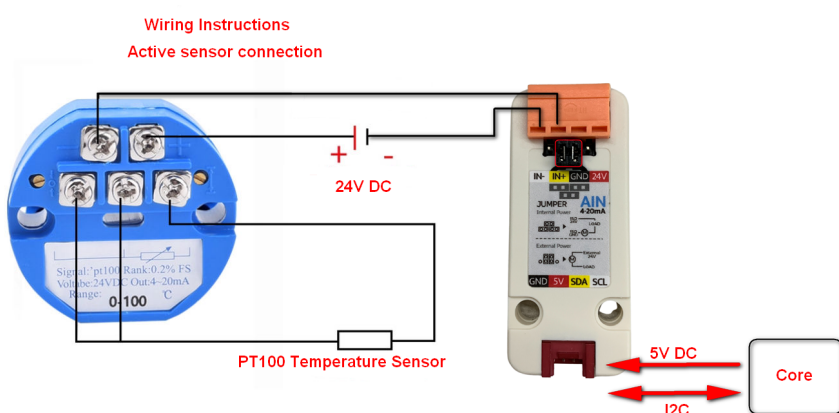
ジャンパーキャップ接続説明

製品パッケージには3つのジャンパーキャップが付属しており、電流型センサーの有源/無源状態に応じて切り替え可能です。

- 無源電流型センサー**を使用する場合: DC 24V 電源を接続し、センサー信号を IN+ / IN- に接続し、ジャンパーキャップを下図のように設定してください (極性に注意) :



- 有源電流型センサーを使用する場合: センサー信号を IN+ / IN- に接続し、ジャンパーキャップを下図のように設定してください（極性に注意）:



ピン互換性

各ホストデバイスのピン構成が異なるため、使用前に製品ドキュメントの[ピン互換性表](#)を参照し、実際のピン接続状況に応じてサンプルプログラムを修正してください。

3. サンプルプログラム

- 本チュートリアルで使用するメインコントロールデバイスは、Unit AIN4-20mA と組み合わせた CoreS3 です。この電流型アナログ取得モジュールは I2C 通信方式を採用しています。実際の回路接続に応じてプログラム内のピン定義を修正してください。デバイス接続後、対応するシリアル IO は G1（SCL）と G2（SDA）です。

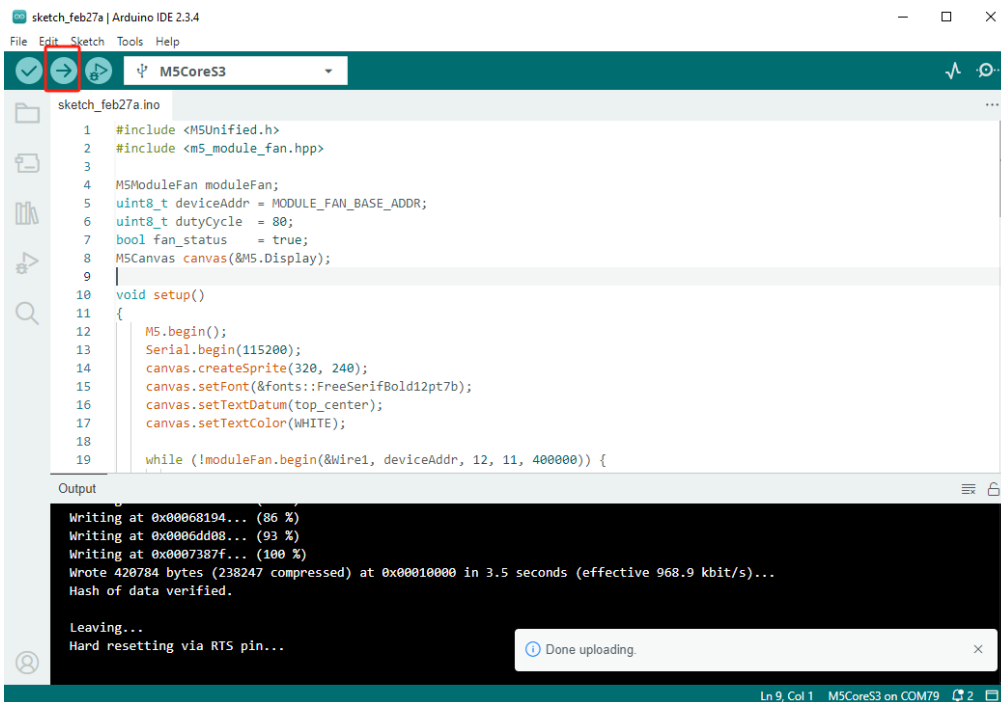
```
1  #include <M5Unified.h>
2  #include "MODULE_4_20MA.h"
3  #include <M5GFX.h>
4
5  MODULE_4_20MA meter;
6
7  void show_current_value(void) {
8      M5.Display.fillScreen(WHITE);
9      M5.Display.setCursor(0, 0);
10     M5.Display.println("Unit 4-20mA Demo");
11     M5.Display.setCursor(0, 50);
12     M5.Display.printf("Current: %.2f mA\r\n", (float)(meter.getCurrentValue(0)) / 100.0);
13     M5.Display.printf("ADC_Value: %.2f\r\n", (float)(meter.getADC12BitsValue(0)) / 100.0);
14     M5.Display.printf("Cal_Current: %.2f mA", (float)(meter.getCurrentValue(0)));
15 }
16
17 void setup() {
18     M5.begin();
19     Serial.begin(115200);
20     M5.Display.fillScreen(WHITE);
21     M5.Display.setTextColor(BLACK);
22     M5.Display.setFont(&fonts::FreeMonoBold12pt7b);
23     M5.Display.setCursor(0, 0);
24     while (!(meter.begin(&Wire, MODULE_4_20MA_ADDR, 2, 1, 1000000UL))) {
25         M5.Display.println("No Module!");
26     }
27 }
28
29 void loop() {
30     show_current_value();
31     delay(1000);
32 }
```

4. コンパイルとアップロード

- ダウンロードモード：プログラム書き込み前にデバイスごとにダウンロードモードに入る必要があります。この手順はメインコントロールデバイスによって異なる場合があります。詳細は[Arduino IDE入門チュートリアル](#)ページ下部のデバイスプログラムダウンロードチュートリアルリストを参照してください。
- CoreS3 の場合：リセットボタンを長押し（約 2 秒）、内部の緑色 LED が点灯するまで押し続け、離します。これでデバイスはダウンロードモードに入り、書き込みを待機します。

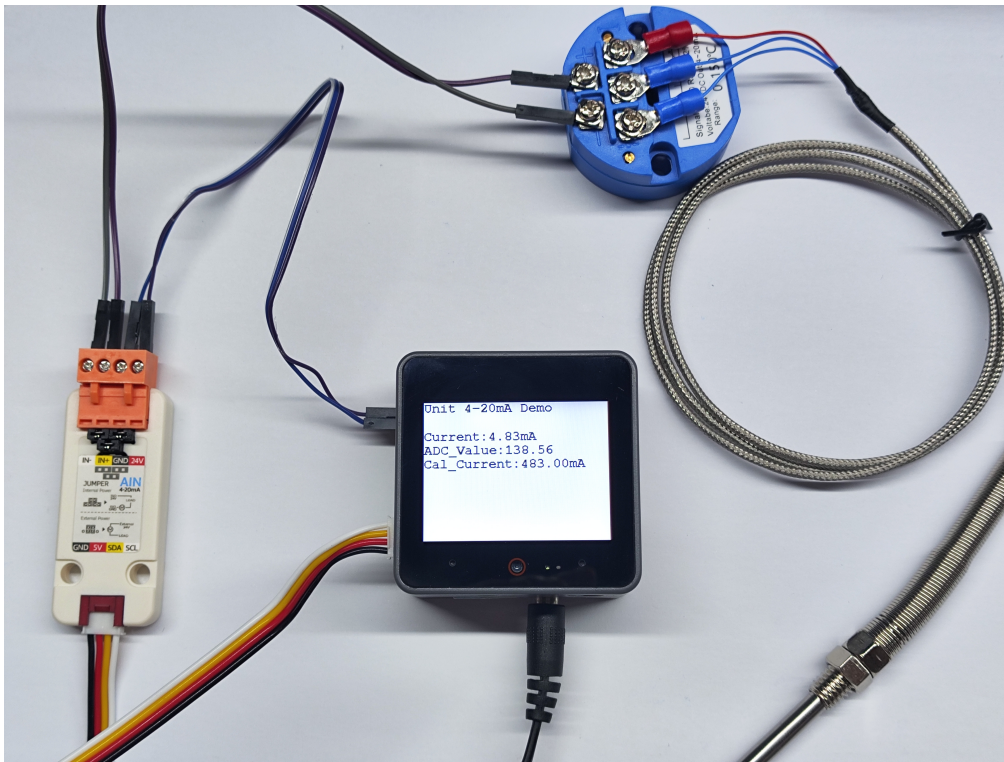


- デバイスポートを選択し、Arduino IDE 左上のコンパイル/アップロードボタンをクリックします。プログラムのコンパイルが完了し、デバイスにアップロードされるまで待機します。



5. 電流値取得

- 本実験では PT100 熱電対温度センサーと無源電流型センサーを使用します。ハードウェア配線完了後、コードを書き込むと取得した電流アナログデータを確認できます。



- 。 センサー温度が上昇すると、電流センサーが取得する電流値も上昇します。

