

std環境だから2行のプログラムでサッと表示!
プロジェクトの中身もチェック

ステップ2…コンソールに「Hello World」を出力

中林 智之

組み込み界のHello World, いわゆるLチカではなくて、コンソールにHello Worldを出力します(図1)。というのも、M5Stamp C3U Mateに搭載されているLEDはフルカラーRGB LEDのSK6812(いわゆるNeoPixel)なので、stdが使える今回の環境ではLEDを光らせるより、文字をコンソールに出力する方がよっぽど簡単なのです。

SK6812はRGB LEDと制御用のマイコンとを搭載しており、GPIO 1本だけで任意の色で光らせることができる優れたものです。その上、数珠つなぎにすることができて、使うGPIOは1本のまま、好きな位置のLEDを好きな色に光らせることができます。

SK6812もESP32-C3のRMT (Remote Control) ペリフェラルを使ったws2812-esp32-rmt-driver-crate⁽¹⁾を使えばわりと簡単に制御できます。クレート名はws2812ですが、制御方法が同じSK6812でも使えます。Hello Worldが終わった後に各自で試してみてください。

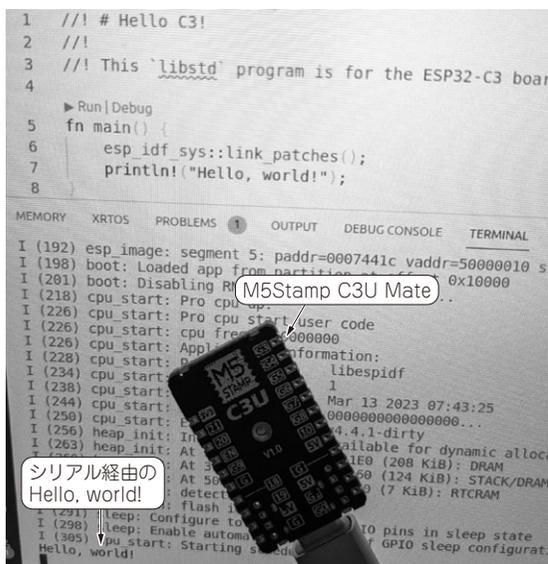


図1 M5Stamp C3U MateでHello World

本章のプログラムはM5Stamp C3U Mateに限らず、ESP32-C3が搭載されていて、USBシリアル通信が可能なボード、例えばESP32-C3-DevKitM-1⁽²⁾などであれば動かすことができます。

まずは動かしてみる

● プログラムの確認

ビルド用のソースコードはGitHubのplay-stdディレクトリ下にあります。また、本誌に載せきれなかった図A～図Cは本誌ウェブ・ページに掲載しています。

```
https://github.com/tomoyuki-nakabayashi/interface202305-c3-std-rust
https://interface.cqpub.co.jp/2305rust2/
```

Hello WorldのプログラムはGitHubのhello_c3(リスト1)⁽³⁾です。

あらかじめコードを確認しておきましょう。

M5Stamp C3U MateをホストPCとUSB Type-Cケーブルで接続し、hello_c3ディレクトリに移動したら次のコマンドを実行します。

```
$ cargo espflash --release --monitor
```

ここで、--releaseは最適化のかかったリリース・ビルドをするためのオプションで、--monitorはフラッシュ・メモリへの書き込み完了後にespsmonitorでESP32-C3の動作&ログ出力を開始するためのオプションです。

以降、ビルド・コマンド例では--releaseは省略します。必要に応じてリリース・ビルドは別途試してください。

リスト1 Hello, world!を表示するhello_c3/src/main.rs

```
fn main() {
    esp_idf_sys::link_patches();
    println!("Hello, world!");
}
```