

ステップ7…クライアント&サーバでセンサ・データ送信

中林 智之

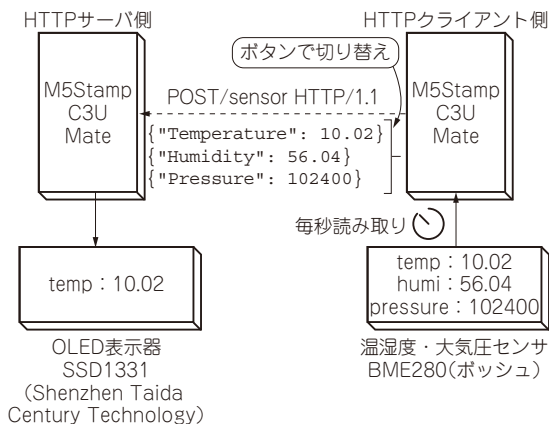


図1 今回作るシステム

M5Stamp C3U Mateを使った組み込みstd Rustへの入門もとうとう大詰めです。ここまで積み上げてきた基礎知識と，ちょっと新しい知識を使ってアプリケーションを作っていきます。

作るアプリケーションの概要は第1章で少しふれましたが，改めて説明すると，図1のような組み込みシステムです。

右のM5Stamp C3U Mateは毎秒センサ値を読み取り，そのうちの1つをJSON形式でHTTP POSTします。このとき，温度，湿度，大気圧のうち，どのセンサ値をPOSTするか，はボタンで切り替えできるようにします。

左のM5Stamp C3U MateはPOSTされたセンサ値のJSONをパースし，接続されているディスプレイに出力します。

何となくこれまでの組み合わせでできそうな予感がありますね。

M5Stamp C3U Mateを2台使って，サーバとクライアントとに分ける構成でも動きますが，localhost宛にHTTPリクエストを送ることで1台でサーバとクライアントを兼ねることもできます。

ビルド用のソースコードはGitHubのhttp-appディレクトリおよびhttp-app-separated下に

リスト1 HTTPサーバ

```
use embedded_svc::{http::server::*, io::Write};
use esp_idf_svc::http::server::*;

let server_config = Configuration::default();
let mut server = EspHttpServer::new(
    &server_config);

server
    .fn_handler("/", Method::Get, |req| {
        let html = index_html();
        req.into_ok_response()?.write_all(
            &html.as_bytes());

        Ok(())
    })?
```

にあります。また，本誌に載せきれなかったリストA～リストR，図A～図Eは本誌ウェブ・ページに掲載しています。

<https://interface.cqpub.co.jp/2305rust2/>
プログラムはGitHubから提供します。

<https://github.com/tomoyukinakabayashi/interface202305-c3-std-rust>

1台のM5Stamp C3U Mateを使ってアプリケーションを作成しているソースコードはhttp-appにあります。2台のM5Stamp C3U Mateを使うソースコードはhttp-app-separatedになります。両者のソースコードはほとんど同じです。http-app-separatedはCargo workspaceを使って，クライアントとサーバ側とでビルド中間生成物を共有している点が異なります。

以下，http-appのコードを使って，アプリケーションの説明を行っていきます。

HTTPサーバが必要なので追加する

アプリケーションを作成する上で欠けている要素はHTTPサーバを実現する部分です。これはesp-idf-svc::http::serverを使います。使い方

