第4章

モダンなマイコンOSいろいろ使える

ESP32で使えるOS百科

宮田 賢一

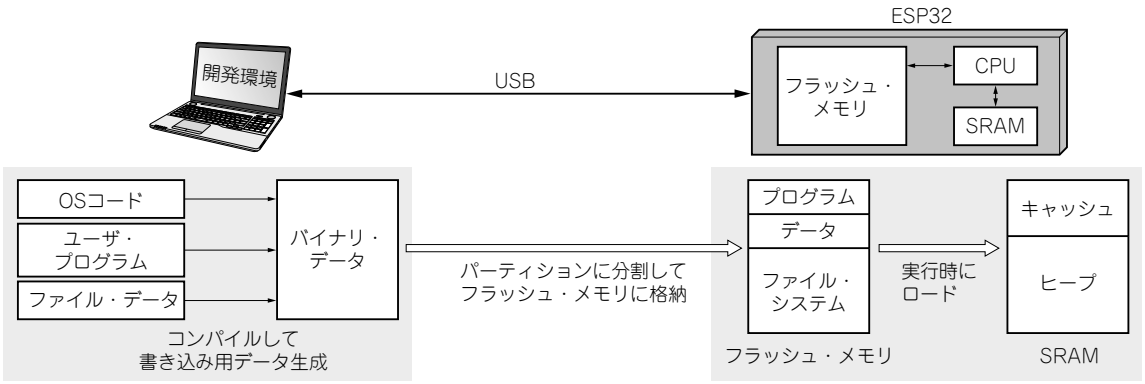

図1　OSはプログラムとひとまとめにしてESP32のフラッシュ・メモリに書き込まれる

マルチコアや大容量メモリをうまく利用するために

マイコンにおけるプログラミング・モデルは，大きく2種類に分けられます．

- ベアメタル・モデル
- オペレーティング・システム（OS）ベース・モデル

ベアメタル・モデルとは，電源を投入してCPUが動作を開始する直後からの全ての動作をプログラミングするモデルです．マイコン・ベンダが用意しているハードウェア・ライブラリやCPUが持つ，割り込み機構などを直接呼び出して，デバイス制御や複数処理の並列実行を実装します．ベアメタル・モデルは余分な処理を作り込まない分，性能やメモリ消費をチューニングできますが，ハード依存性が高いため移植性が低下し，またマルチタスクのような複雑な処理を実装するとバグを作り込むリスクも大きくなります．

一方OSベース・モデルでは，OSが提供する機能を活用してプログラミングします．

一般的にオペレーティング・システム（OS）が持つ役割は，ハードウェア資源（CPU，メモリ，記憶装置，各種デバイスなど）を管理することと，ユーザに対する標準的なインターフェースを提供することです．OSを使うことで個々のハードウェアを意識せずにプログラミングできたり，マルチタスクの処理を簡単に書けたりするようになります．

つまりOSベース・モデルはベアメタル・モデルの裏返しで，移植性が高くシンプルなプログラムにできる代わりに，性能が犠牲になったり，メモリ消費量が比較的に多くなることを意味します．

しかしESP32はマイコン向けOSを動かすのに十分なCPU性能とメモリを搭載しているため，性能低下やメモリ消費量増大というデメリットよりも，プログラム開発のしやすさというメリットの方が大きくなります．特にIoT向けアプリケーションでは，ネットワーク処理やデータ処理，センサ処理，ディスプレイやボタンといったユーザ・インターフェース処理など，多くの処理を組み合わせる必要があるので，OSによる開発支援は欠かせません．

ところで，多くのマイコン応用機器にはキーボードやディスプレイのようなユーザ・インターフェースがありません．つまり，WindowsやLinuxのようなOSをインストールしておいて，OS実行中にマウスやキーボードから実行ファイルをダブルクリックで選ぶというようなことができません．

そこで，ユーザが作成したプログラムと必要なデー