

教科書には
載っていない!

現場で役立つ プログラミングのちょい技

第12回(最終回) I/Oアクセスでハマらないためのちょい技

邑中 雅樹

組み込みシステムは、CPUからI/Oアクセスを行うことにより、周辺デバイスを制御したり、情報を読み取ったりする処理が欠かせません。最終回では、CPUからI/Oアクセスをするプログラムを書く際に陥る罠とそれを回避する方法について解説します。
(編集部)

12回に渡った本連載も最終回です。最終回なのに…いや、最終回なのであえて立ち返ります。

組み込みシステムって何でしょうか？組み込みソフトウェアと、一般的なソフトウェアの違いって、何でしょうか？

アカデミックな定義は難しそうなので、エンジニア以外の方も参照する、知恵蔵2012の説明を引用します。

「家電製品、携帯電話、デジタルカメラ、医療機器、自動車、ロボット、人工衛星など、特定の目的をもったシステムの内部で、それらを制御するために組み込まれている専用のコンピューターシステムである。」

(組み込みシステム：知恵蔵2012より引用)

上記説明に「制御」という言葉が出てきます。物理的に存在するものを制御するには、周辺の情報を取得し、また周辺に対して影響を与える必要があります。組み込みソフトウェアの世界の言葉でいうならば、CPUの周辺に存在する入出力デバイス(I/Oデバイス)に対するアクセス(I/Oアクセス)が必要ということになります。

そんなわけで本稿では、I/Oアクセスにまつわるちょい技をいくつか紹介します。

おさらい：CPUからI/Oデバイスにアクセスする方法

● CPUとデバイスをつなぐ「バス」

ソフトウェアはCPU上で動作します。つまり、ソフトウェアで操作したいすべてのデバイスは、何らかの形でCPUと物理的に接続されている必要があります。この接続経路のことをバス(bus)と呼びます。

最も基本的なコンピュータでは、バスはCPUとデバイスを直結します。CPUの命令セットには、バスを経由してデ

バイスを操作するための命令が含まれており、プログラムは、その命令を使ってデバイスを操作します。

現在のコンピュータは複雑なバスを持っており、制御対象のデバイスにたどり着くまで、ブリッジを介して複数のバスを経由することも珍しくありません。例えば、組み込みソフトウェアの開発現場ではお馴染みのデバイスに、USB-シリアル変換デバイスがあります。このデバイスは、USBバスをブリッジにして、シリアル・ポートというバスを提供しています。

● CPUは必ずバスを介してデバイスにアクセスする

ソフトウェアは、CPUが用意している命令を用いて直結されたバス・コントローラへアクセスすることで、間接的に複数のバスを経由して接続されたデバイスを操作しています。USB-シリアル変換デバイスの例であれば、ソフトウェアは主にUSBホスト・コントローラを操作することでシリアル・ポートの向こうにあるデバイスを操作します。

このような複雑なバス操作は、通常はデバイス・ドライバに隠蔽されています。デバイス・ドライバそのものを書く必要に迫られもしない限り、意識することはないかもしれせん。なので、あえて書きました。

大事なことなので繰り返します。ソフトウェアは、CPUが用意した命令を用いて、CPU直結のデバイスへアクセスすることで、目的のデバイスを操作します。複雑なバス構成の場合は、バス・コントローラを介した間接的なアクセスになりますが、原則は変わりません。

● バスの特定のアドレスにデバイスが割り当てられている

CPUとは何をやる機械でしょうか。抽象的に表現すると次のようにいえます。

CPUは、外部から与えられた命令を逐次実行する機械。