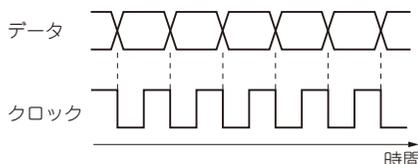


I²C & SPI をゼロから作る①

…全体像と制御プログラム

岡野 彰文



データがシリアルに、かつクロックに同期して転送される

図1 同期式シリアル・インターフェースの概念

I²CとSPIは、IC同士データのやり取りによく使われるシリアル・インターフェースです。

どちらのインターフェースもデータをシリアル化することで信号線を減らしています。順次送信されてくるデータを正確に受け取るために、もう1本のクロック信号で同期を取るようにした同期式シリアル・インターフェースです(図1)。

I²Cではデータとクロックの2本だけで、SPIは4本の信号線を用いて通信を行います。

I²CとSPIのコントローラ機能をビット・バンキングで作ってみる

I²CとSPIは多くの部分が共通したプログラムで作成します。そのため、第2章では共通部分を解説します。第3章と第4章ではI²C、第5章と第6章ではSPIを取り上げます。

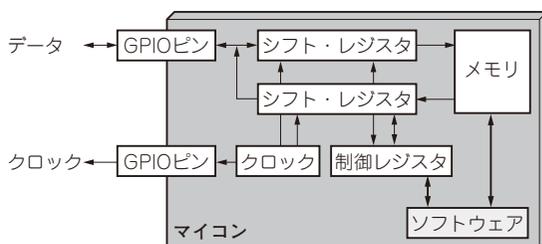
● ビット・バンキングとは

▶ ソフトウェアでGPIOを操作する

ビット・バンキングとは、ソフトウェアによってGPIOを直接操作し、信号の“H”/“L”を出力してビット・パターンを作り出し、信号の“H”/“L”を読み取ってデータを入力する方法です(図2)。

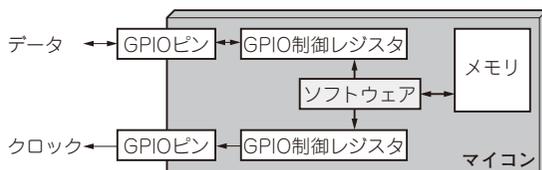
通常のI²CやSPIの通信はビット・バンキングではなく、マイコン内蔵のハードウェアを用いて行います。もしそのようなハードウェアがなかったり、数が足りなかったり、あるいは任意のピンが使えないなどの制限がある場合に、ビット・バンキングを使います。

I²CやSPIの通信をビット・バンキングで行うには、



(a) ハードウェアによる制御

クロックやデータのタイミングは全てハードウェアで制御される



(b) ビット・バンキングによる制御

ソフトウェアによるGPIOの直接制御

図2 ビット・バンキングとハードウェアによる制御

GPIOポートをビット・レベルで、適切なタイミングで操作して、信号の入出力を行うことになります。

このような制御を行うプログラムを読めば、プロトコルが理解できるようになります。

▶ ビット・バンキングの不利な点…速度と正確さ

ビット・バンキングには利点ばかりでなく不利な点ももちろん存在します。まず、この方法は通信をソフトウェアだけで実装するために、ソフトウェアの負荷が重くなってしまいます。

また、ソフトウェアで直接信号を扱うことになるので正確なタイミングの管理ができません。とは言え、I²CやSPIはデータの他にクロック信号を持った同期式通信です。このため、タイミングについてはあまり大きな制約は受けません。常に一定間隔で入出力を管理しなければならないUARTなどに比べると、データとクロックのタイミングさえ合えばよい、同期式通信の方が作りやすいというメリットがあります。