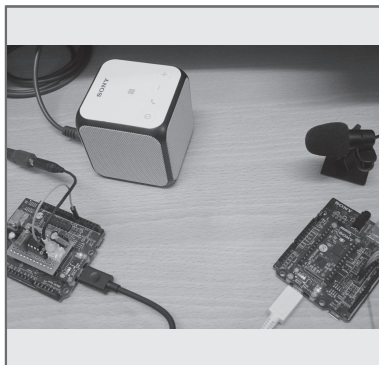




第9章

サイエンス実験！
通信はサブコアに任せるのがよし！

Spresense マルチコア入門！ デジタル・ワイヤレス大実験

よしの たろう

Taro Yoshino

本稿ではSpresenseのマルチコア・プログラミングに入門してみます。テーマは、マルチコアの使いどころ「通信」です。

Spresenseの弱点は、無線通信が標準でないことです。それでも、アナログの音声入出力があるなら音で通信できるはずということで、音や光でつなぐ通信に挑戦してみます。

デジタル無線通信といえば通常は電波を使うのが一般的ですが、波の振幅や周波数、位相を制御して情報を伝えるという原理は、どの媒介波でも共通しています。身近な音や光でワイヤレス通信ができれば、電波によるワイヤレス通信の理解も深まります。ここでは、まず一般的なデジタル無線通信の方式について紹介し、その応用として音や光を使った通信にチャレンジします。

実験の前に…デジタル通信の基本 「非同期シリアル通信」

デジタル通信にはさまざまな方式がありますが、多くはシリアル(直列)なビット列としてデータが送信されます。デジタル通信を理解するためには、ビット列を時間的に順番に送るシリアル伝送の仕組みを押さえておくことが重要です。シリアル伝送のなかでも、

もっとも基本的な方式がUARTで広く知られる非同期シリアル通信です。ここでは、非同期シリアル通信の仕組みについて解説します。

● ビット列を「意味のあるデータ」にするには

物理的にビットを伝送できても、それだけでは情報として意味をもちません。送られたビット列をどう区切り、どこからどこまでを1つのデータとするかというルールが必要です。そこで利用するのが、組み込み開発でよく使われている非同期シリアル通信です。これは、UART(Universal Asynchronous Receiver/Transmitter)と呼ばれる通信方式です。

● 非同期シリアル通信の基本構造

非同期シリアル通信は、1本の信号線で、1ビットずつ順番にデータを送る方式で、送信側と受信側に共通のクロックがありません。そのため、データの区切りを示すためにスタート・ビットとストップ・ビットを付け加える必要があります。

例えば、8ビットのデータ(1バイト)を送る場合は、図1のように8ビットのデータの前にスタート・ビットを付け、最後にストップ・ビットを付け加えます。また、データが正しく届いたかどうか確認するために、データの最後にパリティ・ビットを付ける場合もあります。このようにして、受信側はビットの並びを「ここからここまでが1バイト」と理解できるようになります。

● ボー・レート(通信速度)を合わせる重要性

非同期通信では送信側/受信側に共通のクロックがないため、あらかじめ通信速度(ボー・レート)を決めておく必要があります。ボー・レートとは、1秒間に何ビット送るかという指標です。ボー・レートを設定することで受信側は、ビットを受け取るタイミングを決めることができます。

例えば9600 bpsの場合、送信側も受信側も「1秒間に9600ビットを送る」前提でタイミングを合わせて

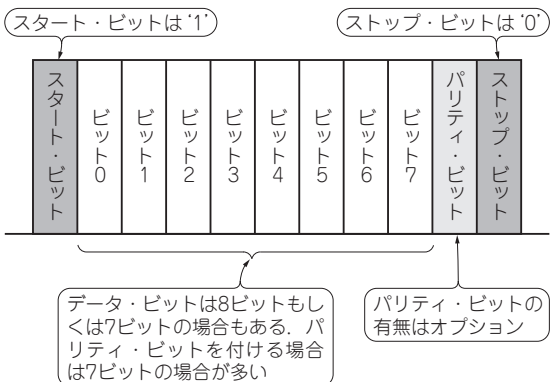


図1 非同期シリアル通信のデータ構造

データの最後にパリティ・ビットを付け加えることもある