

プログラマブルI/Oの機能と簡易ライブラリ

森岡 澄夫

ラズベリー・パイ Pico (以降、Pico) には、普通のマイコン・ボードにはない、とてもユニークな機能があります。プログラマブルI/O、略してPIOと呼ばれるものです。ここでは、PIOはパラレルI/Oの略ではありません。

PIOは、GPIO端子のアクセスを極めて正確なタイミングで高速に(125MHzクロック精度)行えるカスタムI/Oペリフェラルを最大8個作れます。

従来のラズベリー・パイやPCの弱点は、センサやアクチュエータなどI/Oデバイスの制御を行にくいことです。Picoは、それを代行する高速I/Oインターフェース・ボードになります。同じ目的でFPGAを使うよりも、簡単で安価です。本稿ではPIOの基本的な使い方を説明します。

プログラマブルI/O機能のメリット

● 正確なタイミングで信号入出力をするインターフェースを作る

ラズベリー・パイやPCを使うときにしばしば悩みの種となるのが、センサやアクチュエータなどのI/Oデバイスを接続しにくいことです。ラズベリー・パイにはGPIO端子があり、SPIやI²Cなどのペリフェラル回路も幾つかあるのですが、デバイス個数や自由度は大きく制限されます。特にソフトウェアからGPIO端子を直接読み書きしてデバイスを操作する方法^{注1}は、時間精度を気にしないLEDのようなデバイスを除き、とても実用に耐えません。

表1は、ラズベリー・パイ4のLinux (Raspberry Pi OS) からソフトウェアでGPIO出力をトグルさせた例です。ウェイト設定値は信号をトグルさせるための待ち時間を意味します。ウェイト設定値が x マイクロ秒(μ s)であるとき、 $1000000/(2x)$ Hzの周波数が得られれば、正確な時間でのコントロールができていますと結論できます。例えば、 $x=10000\mu$ sの場合、50Hzの周波数が出力されるならば、正確な時間でのコントロールができていますということ。しかし、表1によると、 x が小さいところでは、そのような周波数は得ら

表1 ラズベリー・パイ4においてC言語上で関数`usleep()`でウェイトし出力トグルさせた結果

従来のラズベリー・パイではソフトウェアから高速なGPIO操作を行うことは難しかった

ウェイト設定値 x [μ s]	最小周波数 [Hz]	最大周波数 [Hz]	平均周波数 [Hz]
10	4474.0	5335.0	4600.0
100	2000.0	2455.0	2390.0
1000	413.4	447.2	441.4
10000	48.0	49.4	49.2

れていません。例えば、 $x=100\mu$ sならば5000Hz、 $x=10\mu$ sならば50000Hzが出ていなければなりませんが、そのような周波数は出ていません。したがってラズベリー・パイ4では、正確な時間でのコントロールができていないことが分かります。表1では、出力される周波数を同一設定にしても大きな幅で揺らぐことを示すために、最小周波数、最大周波数、および平均周波数の3種類を示しています。

多くのデバイスを利用したり正確なタイミング・コントロールをしたりするためには、マイコンやFPGAをインターフェースとして使うのがこれまでの定石です(表2)。ただし、どれも一長一短あり、万能な方法はありません。マイコンは安価で手軽に使えますが、GPIOによるデバイス・アクセスはそれほど高速ではなく通信に広い帯域は取れません(例えばカメラを接続するなど難しい)。FPGAはピン数が多くて帯域を広く取れ、タイミング管理も数10nsの精度でできますが、設計は誰でもできるとは言えず、価格も高いです。

● 8つの並列動作するミニ・プロセッサとして使える

PicoのPIOは、FPGAほどではありませんが、正確にタイミング制御できる高速インターフェースを安価に作れる面白い機能です。図1がPIOのブロック構成

注1: ビット・バンギングと呼ばれる、自由なプロトコルやビット幅で通信できる。