

第3章

GPIOにはできない速度! リアルタイムがうれしい

簡易ライブラリで複数のRC
サーボや測距センサを動かす

森岡 澄夫

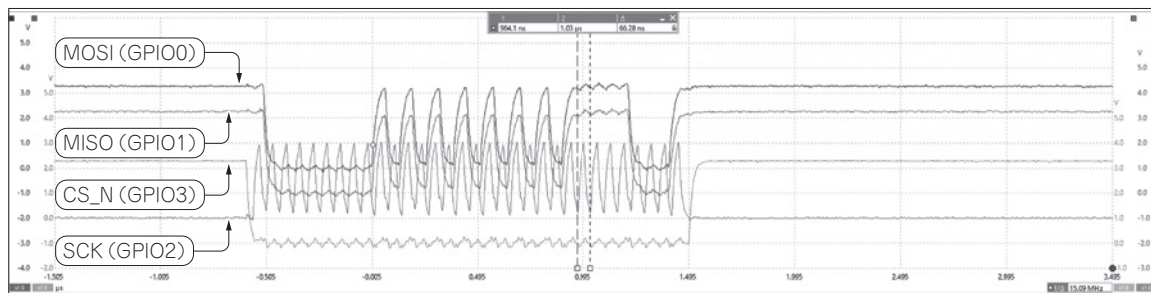


図1 PIOによるSPI入出力の実力

CPOL=0, CPHA=0, MSBファーストに相当する動作になっているが、容易に変更できる。通信周波数もステート・マシンの動作周波数を変えることで簡単に調整できる

前章に続き、プログラマブルI/O(以降、PIO)を実際に使ってみた4つの例と、それらの製作に当たってつまづいた点などを紹介します。ラズベリー・パイPico(以降、Pico)単独で完結するのではなく、PCやラズベリー・パイと接続し、拡張I/Oインターフェースとして使うことを想定しています。作例は前章で用いた頒布アーカイブに収録されています。

作例1: SPI入出力の処理

● 製作物の機能と特徴

信号入出力の実践的な例題として、センサとの通信に用いるSPIのマスタ・インターフェースを作ってみます。

4線式のSPIでは、マスタ出力MOSI、マスタ入力MISO、クロックSCK(またはSCLK)、チップ・セレクトCS_N(またはSS)があります。図1のようにクロックに同期して出力と入力が同時並列で行われます。送受信データ長は32ビットにしていますが、作り替えは難しくありません。Armプロセッサからステート・マシンへ送信データを送ると送受信が始まり、完了すると受信データがArmプロセッサへ送られます。

SPIにはクロックの極性やデータ入出力のタイミングによる幾つかのモードがありますが、作例では立ち上がりでサンプリング動作するようにしています(いわゆるモード0)。しかし、どのモードへも簡単に作

り替えることができます。

SPIクロックの周波数はステート・マシンの動作周波数の1/4であり、簡単に変えられます。筆者が実機で試したところ、SPIクロックは15.625MHzで安定した入出力ができました。一般的なマイコンに搭載されているSPIペリフェラルと同等かそれ以上の性能で、十分な実用性があります。通常、マイコンのGPIO制御ではこのような速度は出せませんので、PIOがとても強力であることがよく分かります。

● ソフトウェア構成とコード

ステート・マシンのコードがリスト1、Armプロセッサのコードがリスト2です。前章で例を示した通り、out命令やin命令を使ってGPIOの読み書きを行い、push命令やpull命令を使ってArmプロセッサとのデータ交換をします。OSRレジスタとISRレジスタを、それぞれ1ビットずつGPIO入出力するためのバッファとして利用しています。

全32ビットを送受信したかは、OSRレジスタが空になったかをjmp命令で見ることによって判定しています。

● サイドセットを使ったおかげで高速インターフェースができた

この例では、SPIのSCK出力やCS_N出力をout命