

ラズパイを使ってシェルのプロが1行ずつ解説!

# IoTのための Linux コマンド超入門

ご購入はこちら

## 第6回 awkを利用したSPI コマンドの高速化

中村 和敬, 大野 浩之

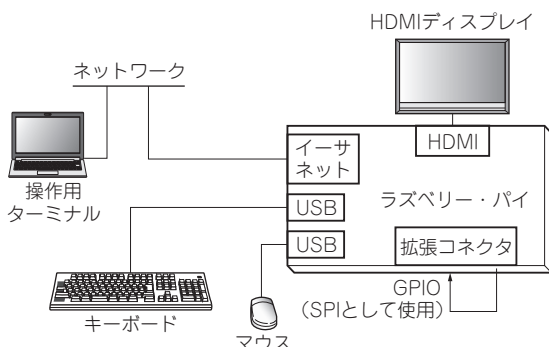


図1 今回やること…GPIO操作によるSPI通信コマンドを作成して速度を計測する

今回はSPIデバイスは使用しない、速度の計測はループバックで行う

本連載では、UART、SPI、1-Wireなどのインターフェースやイーサネット通信、データ処理などをシェルから対話型で行う方法について解説していきます。(編集部)

本連載ではラズベリー・パイの上で、シェルを通じてさまざまなデバイスからデータを取得して処理できることを説明してきました。

書き捨てのシェル・スクリプトでもさまざまなことができますが、ある程度まとまった処理はモジュールとして扱いたくなります。

そこで今回はシェル・スクリプトを、コマンドとして作成する際の考え方を説明します。

### ハードウェア

#### ● その1: ホスト・コンピュータ「ラズベリー・パイ3」

本稿では、I/O制御で使いやすいコンピュータとして代表的なラズベリー・パイ3を使用します(図1)。OSとして、Raspbian Jessie LiteのVersion April 2017を使用していることを前提としています。シェルは、ネットワークからsshを使用して操作します。

誌面の都合で、これらの設定方法などは割愛します。

ラズベリー・パイ3  
(40ピン拡張コネクタ)

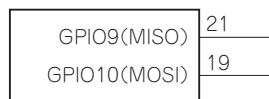


図2 GPIO操作によるSPI通信速度テストのための接続  
純粋に速度を計測するため、SPI出力をそのまま入力にループバックさせる

#### ● その2: SPI接続用のワイヤ

例題として、SPIを使います。

連載第2回(2017年12月号)<sup>(1)</sup>では、SPI接続できるリアルタイム・クロック・モジュール<sup>(3)</sup>をラズベリー・パイに接続して実験を行いました。

今回は純粋な処理速度を測定したいので、SPIデバイスは使用せず、SPIの出力になるGPIO9(MISO)とSPIの入力になるGPIO10(MOSI)を短絡して実験を行います(図2)。これにより、入力内容をそのまま返すSPIデバイスのように動作します。

また、今回はGPIOを直接操作してSPIを操作します。RaspbianはSPIのドライバを備えていますが使用しません。デフォルトでSPIドライバは無効化されていますが、もし有効化されている場合は無効化しておきます。

### 実験1…シェルによる操作

SPIはシンプルな規格であり、動作クロックに下限がありません。そのため、シェル・スクリプトからGPIOを直接操作することで、SPI通信ができます<sup>(2)</sup>。

このとき、シェルのwhile文を利用した実装では、非常に低速な通信しか行えません。しかしデータ・フローによる処理に書き換えることで、性能を向上できます<sup>(2)</sup>。

ここではwhile文による単純な実装を行い、処理時間を確認します。次にデータ・フローによる実装を行い、処理性能の向上を確認します。

第1回 UARTインターフェースのGPSモジュールからデータを取得(2017年11月号)

第2回 GPIOコマンドでSPIデバイスを操作(2017年12月号)

第3回 少ピン低速1-Wire接続の温度センサからデータを取得(2018年1月号)