

Raspberry Pi 3 Model B による IoT の実践 (第八回)

三重大大学 機械工 松井博和

<http://www.robot.mach.mie-u.ac.jp/~hmatsui/RasPi3B/>

Keywords: シリアル通信 (SPI) , A/D 変換器 (MCP3002)

1. A/D 変換のための SPI 通信

SPI 通信は、i2c 通信とともによく使われるシリアル通信の規格の一つである。送信するバイト数と同バイト数受信する。Raspberry Pi Model 3B では、40 ピンの GPIO (General-Purpose Input/Output) の一部で SPI 通信ができる。RasPi で SPI 通信をするには、本報告前の第七回の「ssh サーバ設定」と同様に spi を設定し、第五回の「GPIO ライブラリのインストール」をする。RasPi3B の GPIO と A/D 変換チップの MCP3002 との接続図を Fig. 1、写真を Fig. 2 に載せる。

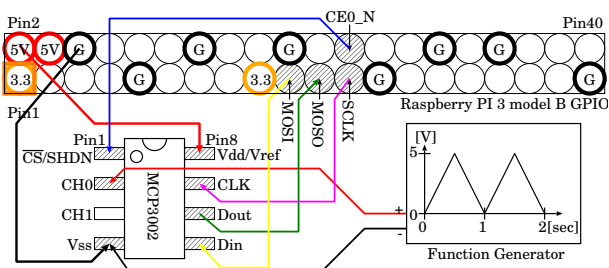


Fig. 1 RasPi3B と MCP3002 との接続図



Fig. 2 RasPi3B と MCP3002 との接続写真

ここでは、A/D 変換チップの CH0 に与える入力電圧として、ファンクションジェネレータから最低 0V から最高 5V で 0.5 秒上昇、0.5 秒下降の 1 秒周期のランプ入力を用いる。

2. SPI 通信する C プログラム

本プログラムは、SPI 通信で 2bytes データを RasPi3B から送り、2bytes データをガジェットから受け取るプログラムである。A/D 変換チップの MCP3002 は、0x68 と 0x00 の 2 バイトを送ると、CH0 の電圧を 10bits で量子化し、それを 2bits と 8bits の 2 バイトデータで返す。

```
8>-- 8>-- 8>-- 8>-- 8>-- SPI 通信 8>-- 8>-- 8>-- 8>-- 8>--
#include <wiringPiSPI.h>
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#define SsPort 8 //GPIO8
#define SpiChannel 0 //SPI チャンネル
#define BuffSize 2
#define SpiSpeed 1000000 //通信速度 (100kHz)

void main()
{
    int i;
    unsigned char spiBuff[BuffSize]; //送受信バッファ

    //SPI チャンネル初期化
    if((wiringPiSPISetup (SpiChannel, SpiSpeed)) < 0){
        fprintf(stderr, "SPISetup error\n");exit(1);
    }
    if(wiringPiSetupGpio() == -1){ //GPIO 初期化
        fprintf(stderr, "SetupGpio error\n");exit(1);
    }
    pinMode(SsPort, OUTPUT); //SSpin を出力に設定
    digitalWrite(SsPort, 1); //SS 信号初期化
    fprintf(stderr, "Start SPI\n");
    for(i=0; i<200; i++){ //2 秒間計 200 点のサンプリング
        spiBuff[0] = 0x68; spiBuff[1] = 0x00; //CH0 用
        //SPI 通信実行
        digitalWrite(SsPort, 0); //通信開始
        //BuffSize のデータを送受信する
        wiringPiSPIDataRW(SpiChannel, spiBuff, BuffSize);
        digitalWrite(SsPort, 1); //通信終了
        printf("%4.2f %d\n", 0.01*i, spiBuff[0]*256+spiBuff[1] );
        usleep(10*1000); // 10msec インターバル
    }
    fprintf(stderr, "End SPI...\n");
}
8>-- 8>-- 8>-- 8>-- 8>-- SPI 通信 8>-- 8>-- 8>-- 8>-- 8>--

上記プログラムをターミナル上の下記コマンドにてコンパイルし、実行ファイル spi を作り、実行する。

$ cc -o spi spi.c -lwiringPi
$ ./spi
Start SPI
0.00 234
0.01 213
197 行の省略
1.99 211
End SPI...
```

上記のデータをプロットするとファンジェネの出力波形とほぼ同型の Fig. 3 のグラフを得、A/D 変換がうまく動作していることが分かる。

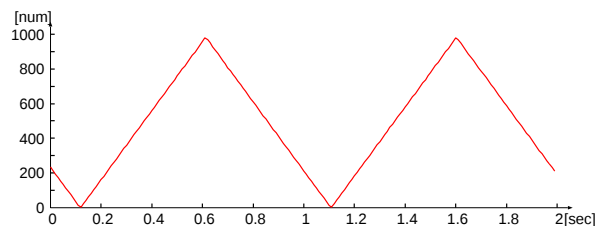


Fig. 3 実行出力のグラフ化