

AD9833 DDS は、0~20MHz の出力レンジ・28 ビット分解能・サイン波/三角波/方形波の出力・周波数/位相変調が可能な信号源です。制御は 3 線式 SPI です。その機能を有する PIC 16F1823 マイクロプロセッサ等で簡単に行うことができます。それ等のデバイスの詳細は、次の URL からダウンロードできます。

[www.analog.com/media/jp/technical\\_documentation/data-sheets/AD9833\\_JP.pdf](http://www.analog.com/media/jp/technical_documentation/data-sheets/AD9833_JP.pdf)  
[http://www1.microchip.com/downloads/jp/DeviceDoc/41413C\\_JP.pdf](http://www1.microchip.com/downloads/jp/DeviceDoc/41413C_JP.pdf)

ただし、AD9833 のレジスタは”16 ビット・ワード”として説明されていますが、PIC は”8 ビット・バイト”を取り扱いますので、インターフェースの設定に悩まされます。そこで、その疑問解明を目的にして実回路で確認しましたので、本稿ではその結果をメモとして残します。

図 1 に、インターフェースと書き込みタイミングを示します。

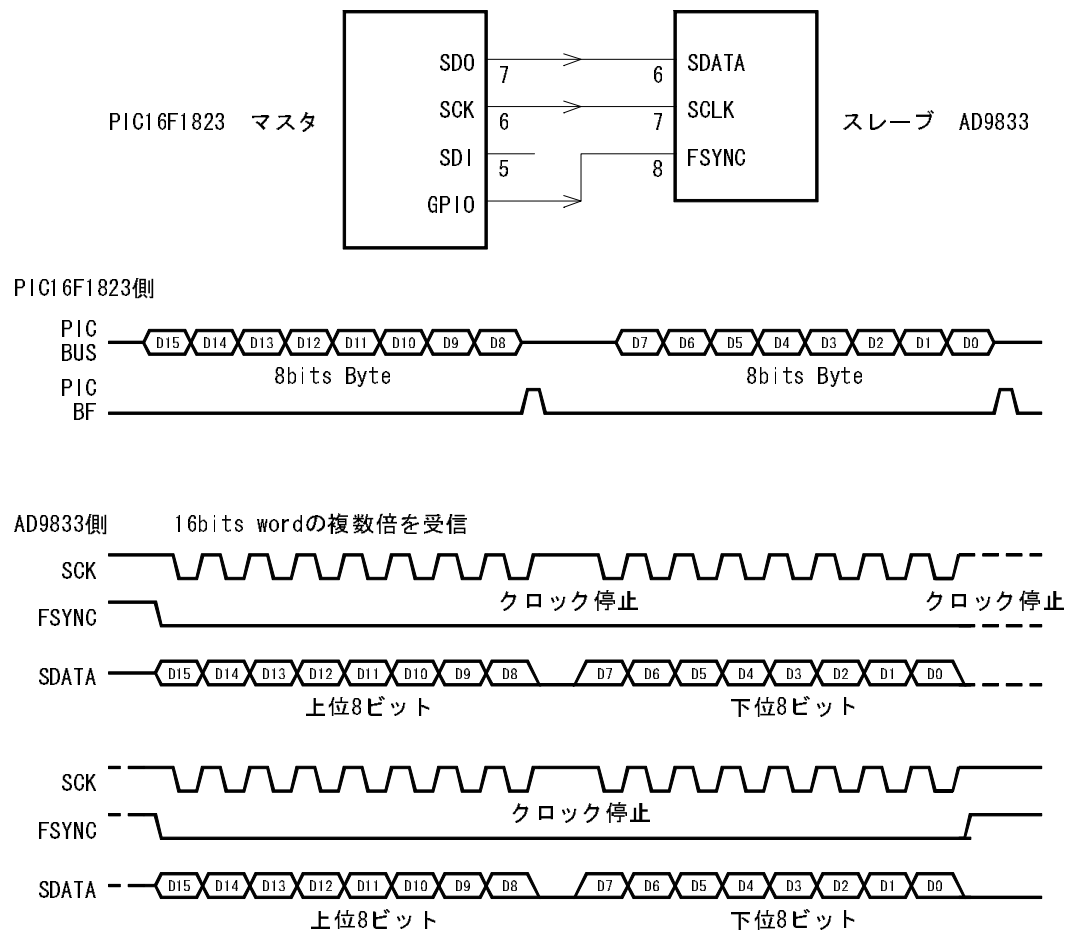


図 1: PIC SPI による AD9833 のレジスタ書き込み

もちろん、PIC 側は SPI マスタとして動作します。データが SSP1BUF にロードされると、SCK からクロックと SDO からデータの送付が行われ、8 ビット送り終わると BF が立ち上がります。これで PIC 内部では 1 バイトの送信が完了しますので、SCK をアイドル状態に保ちます。ソフトウェアが次のデータである 8 ビット・バイトを SSP1BUF にロードして、SCK と SDO が出力されることとなります。

いっぽう、AD9833 側では SCLK に同期してデータ  $D_{15}$  から  $D_0$  までをレジスタに収納します。ここで、FSYNC がローレベルの間をデータ長として、16 ビット・ワードの整数倍まで認識します。ですから、制御レジスタ、周波数/位相レジスタへの書き込みを 1 つの FSYNC 周期内で行うことができます。

次のプログラム例は、単純に 2.4576MHz 基準周波数から 137.5kHz の方形波クロックを出力するサンプルです。この場合の VOUT 出力は MSB/2 を使用しますので、MSB としては  $137.5kHz \times 2 = 275kHz$  を生成することになります。算式は次のとおり。

$$FREQ0 = 275 \cdot 10^3 \times 2^{28} / 2.4576 \cdot 10^6 = 30,037,333$$

この値は 16 進法では 0x1ca5555 となります。AD9833 の周波数レジスタは 28 ビット構成ですので、ここで 14 ビットの MSB と 14 ビットの LSB に分解します。すなわち、MSB が 0x0729 と LSB が 0x1555 です。それぞれの前に周波数レジスタへのアドレス (0b01) を付加して、LSB は 0x5555 と MSB は 0x4729 として SDATA を送ります。

可変周波数・周波数変調/位相変調・出力波形等の応用は各パラメータを変更すれば可能ですが、この段階では省略します。

```

// "gy9833.c" ver. 0r00 (c) 2019.01.05 JA5FP
// This program may be compiled with MPLAB XC8 C Compiler on MPLAB X IDE v2.20.
// Target hardware is PIC16F1823 and DDS unit GY9833 replaced XTAL 2.4576MHz.
// The functions are:
// (1) to learn how to configure 16F1823's SPI
// (2) to set simply specific frequency to AD9833 DDS

#include <xc.h>
#include <pic16f1823.h>
#pragma config FCMEN=OFF, IESO=OFF, CLKOUTEN=OFF, BOREN=ON, CPD=OFF, CP=OFF
#pragma config MCLRE=OFF, PWRTE=OFF, WDTE=OFF, FOSC=ECM
#pragma config LVP=OFF, BORV=LO, STVREN=OFF, PLLEN=OFF, WRT=OFF

void device1823(void){
    // hardware setting
    APFCON=0b10000100; // RX=RA1
    ANSELA=0b00000000; // digital portA
    ANSELC=0b00000000; // digital portC
    TRISA=0b00101111; // RA5=CLK, RA4=/PTT, RA3=D1, RA2=1PPS,
    // RA1=NMEA, RA0=D0
    TRISC=0b00110010; // RC5=D3, RC4=D2, RC3=FSYNC, RC2=SDATA,
    // RC1=DDI, RCO=SCLK
    OPTION_REG=0b00000010; // WPU enable
    WPUA=0b00001001; // WPU portA
    WPUC=0b00110010; // WPU portC
    SSP1CON1=0b00110000; // SPI master mode, clock=Fin/4=614.4kHz
    SSP1CON3=0b00000000;
    SSP1STAT=0b00000000;
    INTCON=0b00100000;
} // device1823()

void spi(unsigned int variable){
    SSP1BUF=variable; while(!BF); // transmit 8bits Byte via SPI
} // spi()

void main(void){
    device1823(); // initialize 16F1823
    RC3=1; RC3=1; // SPI idle state
    RC3=0; RC3=0; // SPI start
    spi(0x21); // reset AD9833 register
    spi(0x40); // output from MSB/2

    spi(0x55); // frequency register with 14LSB
    spi(0x55); // remained 8bits
    spi(0x47); // frequency register with 14MSB
    spi(0x29); // remained 8bits

    spi(0x20); // unreset AD9833 register
    spi(0x40); // output from MSB/2
    RC3=1; RC3=1;

    while(1){}; // continue DDS
} // main()

```