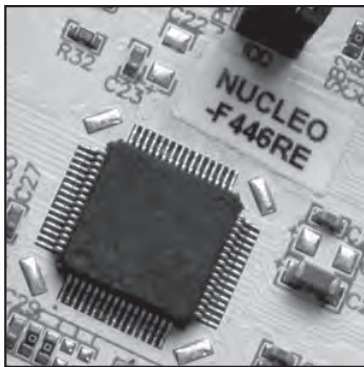


# 連載



FPGAを使わない! Cortex-M4のプログラムで  
誰でも簡単にソフトウェア・ラジオの実体験



## Armマイコンでつくる ダイレクト・サンプリングSDR

### 第3回 SDRで使う信号処理

三上 直樹 Naoki Mikami

今回は、SDRに限らず、アナログ信号をデジタル的に処理する場合に必要な基礎的知識について説明しました。今回は、SDRを作る際に必要となる特有のデジタル的な処理について解説します。

SDRには高速なsin/cos信号を生成する直交信号発生器が必要です。本稿ではデジタル・フィルタを応用して計算する方法を紹介します。

また、放送信号を標準化する際には、標準化周波数をより低い周波数に変換するダウン・サンプリングを行います。本稿ではCICフィルタを使うダウン・サンプリングの方法や、CICとFIRフィルタを組み合わせる方法を紹介し、**〈編集部〉**

—本連載の筆者によるセミナー開催のご案内—  
実習・Armマイコンでつくる  
ダイレクト・サンプリング方式のSDR

- 日時：2021年2月19日(金)10:00 ~ 17:00
- 講師：三上 直樹 ● 受講料：26,000円(税込み)
- 会場：東京・巣鴨 CQ出版社セミナー・ルーム  
参加希望者は、タイトル部(下記Webページ)にアクセスして登録を行ってください。

<https://seminar.cqpub.co.jp/ccm/ES20-0124>

### 直交信号発生器

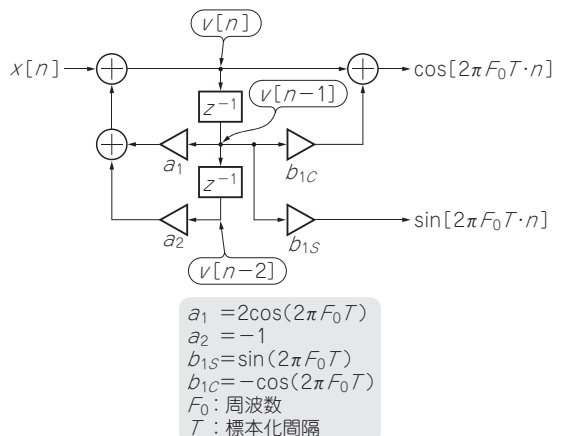
#### ● SDRには高速なsin/cos信号を生成する信号発生器が必要になる

第1回目で説明したように、通常SDRでは放送波の信号をI/Q信号に変換します。このとき必要となるのが、数値制御発振器NCO(Numerically-Controlled Oscillator)です。NCOはsin/cosの信号を同時に発生する必要があります。cos信号はsin信号の位相を90°ずらしたものであるため、このような信号発生器を直交信号発生器と言います。

C/C++言語ではsin/cosの値を求めるための関数が提供されているので、簡単に計算できます。しかし、これらの関数の実行スピードはそれほど速くないため、リアルタイム処理が難しくなります。

sin/cosの値を高速に求める方法としては、CORDICアルゴリズムがよく知られています<sup>(1)</sup>。この方法はマイコンの乗算器のスピードが遅い場合や、FPGAを使って固定小数点演算でsin/cos信号を求める場合には有利です。しかし、今回のように浮動小数点演算ユニットを有するマイコンで実行する場合は、CORDICアルゴリズムは適切な方法とは言えません。

本稿では、sin/cos信号を高速に生成する方法として、デジタル・フィルタを応用する方法を使います。



入力信号  $x[n]$  を次のようにすれば、sin/cosが出力される

$$x[n] = \delta[n] = \begin{cases} 1, & n=0 \\ 0, & n \neq 0 \end{cases}$$

プログラムでは、 $v[n]$  を次のように初期設定し、 $n=2$  から順に計算するようにしている

$$\text{初期設定} \quad \begin{cases} v[0]=0 \\ v[1]=1 \end{cases}$$

$$n=2 \text{ から順に計算} \quad v[n] = a_1 v[n-1] + a_2 v[n-2]$$

図1 直交信号発生器によるNCOの構成

NCOとして使う場合は、フィルタの係数  $a_1$ ,  $b_{1s}$ ,  $b_{1c}$  を  $F_0$  に応じて変更し、初期条件を設定して  $n=2$  から計算を始めるようにする

#### ● デジタル・フィルタを応用した直交信号発生器

IIRフィルタを使うと、高速なsin/cos信号を簡単に発生できます。図1に直交信号発生器によるNCOの