サンプル・プログラム付き!

高速ワンチップ・マイコンではじめる ソフトウェア無線

第11回 信号処理結果のスペクトラム表示

高橋 知宏

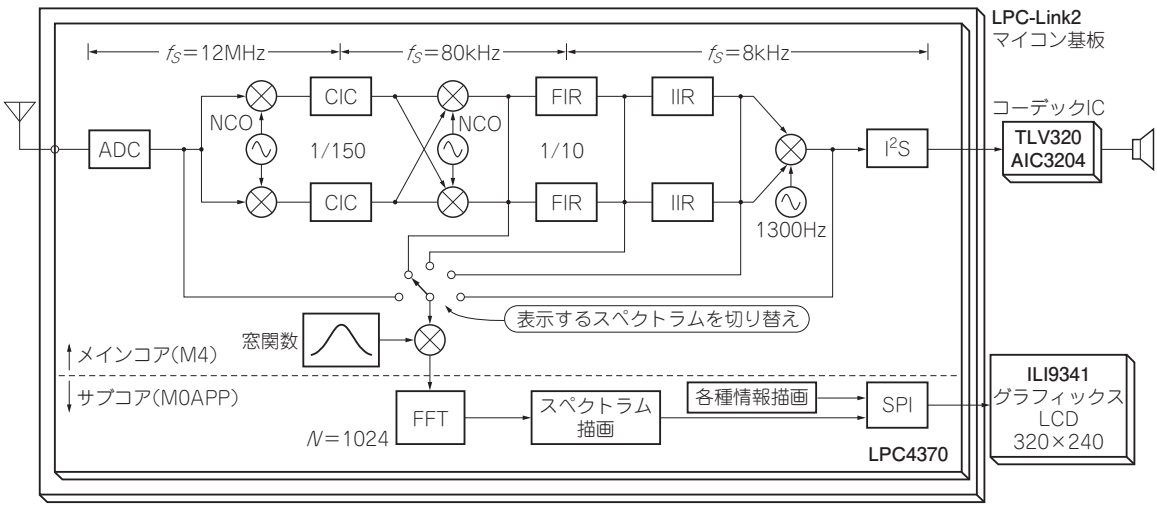

図1 ソフトウェア無線信号処理の各ステージの波形をグラフィックス液晶ディスプレイでリアルタイム・モニタできるようにした
FFT計算や画面表示は，メインCortex-M4Fコアじゃなくて，サブCortex-M0コアにやらせる

前回（第10回，2016年7月号）は，ソフトウェア無線信号処理の各ステージの波形をグラフィックスLCDでリアルタイム・モニタするために，追加するFFT処理とマルチコア・プログラミング方法を紹介しました（**図1**）．

今回は実際に動かしてみます．

ターゲットARMマイコンのマルチコア動作

● サブコアの動作

ターゲットLPC4370マイコンはメインCortex-M4Fコア（M4）に加えてCortex-M0コア（M0APP）を内蔵しています．FFTと描画はこのサブコアで行います．

リセット時にはサブコアは起動していません．メインコアのmain()関数の先頭部分で，明示的にサブコアを実行開始している箇所があります（**リスト1**）．

この呼び出しを行うことでサブコアの実行が開始されます．内部的には，サブコア用の渡されたイメージ・アドレスにより，PCやスタック・ポインタが格納されたベクタを設定した後，リセット状態を解除することで，サブコアが起動されるようになっています．イメージを参照するシンボル（__core_m0app_START__）は，リンカ・スクリプトで定義されるようになっています．

サブコアはCortex-M0ですが，メインコアと違っていくつか注意点があります．

リスト1 メインコアのmain関数でサブコアを起動している部分

```
int main(void) {

    // Start M0APP slave processor
#if defined (__MULTICORE_MASTER_SLAVE_M0APP)
    cr_start_m0(SLAVE_M0APP,&__core_m0app_START__);
#endif

    // Start M0SUB slave processor
#if defined (__MULTICORE_MASTER_SLAVE_M0SUB)
    cr_start_m0(SLAVE_M0SUB,&__core_m0sub_START__);
#endif
```

▶ (1) クロック設定が不要

クロックはメインコアと共用ですので，サブコアでは設定する必要がありません．これに関係する落とし穴として，設定不要であるが故にライブラリが必要とするクロック関係の呼び出しやソフトウェア的な設定が行われておらず，ペリフェラルを扱うライブラリが

第1回 AMラジオを作る（2015年10月号）
第2回 AMラジオで行う信号処理（2015年11月号）
第3回 短波ラジオを受信する（2015年12月号）