

リアルタイム os「DSP/BIOS」を利用する DSP アプリケーションの開発

三上 直樹

第3回 簡単なマルチスレッド・システム構築の基礎とリアルタイム解析

本連載の第1回と第2回では、DSP/BIOSを使って非常に簡単なシステムを作成した。今回は、簡単なマルチスレッド・システムをDSP/BIOSを使って作成する。さらに、作成したDSP/BIOSのシステムを使って、リアルタイム解析機能を適用した例を紹介する。これにより、DSP/BIOSを使うメリットを実感できるだろう。(筆者)

関連データ

今回作成するマルチスレッド・システムは、A-Dコンバータで変換された信号をそのままD-Aコンバータへ送る処理と、入力信号の大きさに応じてDSPボードのLEDの明るさをコントロールする処理を並行して行います。LEDの明るさはPWM(Pulse Width Modulation)制御でコントロールし、その際にタイマ割り込みを使います。

このシステムを、初めにDSP/BIOSを使わずに作成し、次にDSP/BIOSを使って作成します。そして、DSP/BIOSのリアルタイム解析機能を使って、システムの動作時に各スレッドの実行状況を表示する機能について紹介します。

1

DSP/BIOSを使わない マルチスレッド・システム

DSP/BIOSを使わずに、アナログ信号の入出力とLEDの明るさのコントロールを並行して行うマルチスレッド・システムを作成してみましょう。

● main()関数を含むファイルで行う処理

リスト1(LED_PWM_NoRTOS.cpp)にmain()関数を含むファイルの内容を示します。この中では、main()関数のほかに、二つの割り込みサービス・ルーチンAIC_RX_ISR()とTIMER0_ISR()が定義されています。

<グローバル宣言>

定数TmCLKやPrdTri, CmpNum, PrdSmpは、PWM制御用のノコギリ波を発生するために使うDSP内蔵のタイマの周期を決めます。これらの定数により、ノコギリ波の基本周期は15.36ms(約65Hz)に設定されます。この基本

周期を128に分割して、ノコギリ波を発生させます。

グローバル変数powerは入力信号のレベルの推定値を格納します。

オブジェクトの宣言の中で、AIC23_6713_DSP_BIOSクラスのオブジェクトcodecはA-D/D-A変換に使うものです。もう一つはタイマ用のTimerクラス(リスト2を参照)のオブジェクトtimer0で、タイマ0を使い、CPU割り込み番号として14を割り当てます。また、タイマ周期レジスタに対応する値TmCLK*PrdSmpで、タイマ割り込みの周期が0.12msになるように設定しています。

<main()関数>

最初に、A-Dコンバータのデータを読み出すためのMcBSP1^{注1}の受信割り込み関連の設定をします。関数IRQ_map()は、McBSP1のCPU割り込み番号11を割り当てます。

次に、変数powerの初期化と、タイマ0のカウント開始を指令します。その次の二つの関数IRQ_nmiEnable()とIRQ_globalEnable()により、割り込みが受け付けられる状態になります。

最後に、while文による無限ループに入り、割り込み発生を待つ状態に入ります。

<割り込みサービス・ルーチン>

割り込みサービス・ルーチンの宣言や定義を行う場合は、以下の2点が必須です。

- ① 関数名の先頭にキーワードinterruptを付ける
- ② 戻り値はvoidにする

● AIC_RX_ISR()

この割り込みサービス・ルーチンはMcBSP1の受信割り込み発生時に呼び出され、A-Dコンバータのデータを読み込み、それをD-Aコンバータへ送ります。

そのほかに、両チャンネルの入力信号の2乗和を求め、こ

注1: McBSPとは、使用するDSPボードに搭載されているTLV320AIC23(A-D/D-Aコンバータなどを内蔵するCODEC用IC)と、DSPとのインターフェースのために使用する、多チャンネル対応のシリアル・ポートのこと。