

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

第5回(最終回) FFTを利用するFIRフィルタ—DSP/BIOSを使う場合

三上 直樹

今回は前回(2009年6月号, pp.146-156)作成したマルチスレッド・システムをDSP/BIOSを使って実現する。前回よりプログラムを簡単に作成できることが理解できるだろう。(筆者)

関連データ

1 DSP/BIOSの コンフィグレーション

DSP/BIOSを使って作成するシステムの概要を図1に示します。

DSP/BIOSのコンフィグレーションを行うため、コンフィグレーション・ファイルを新規に作成します^{注1}。このファイルの名前をConf_OLS.cdbとします。

このシステムでは、HWI(Hardware Interrupt)やSWI

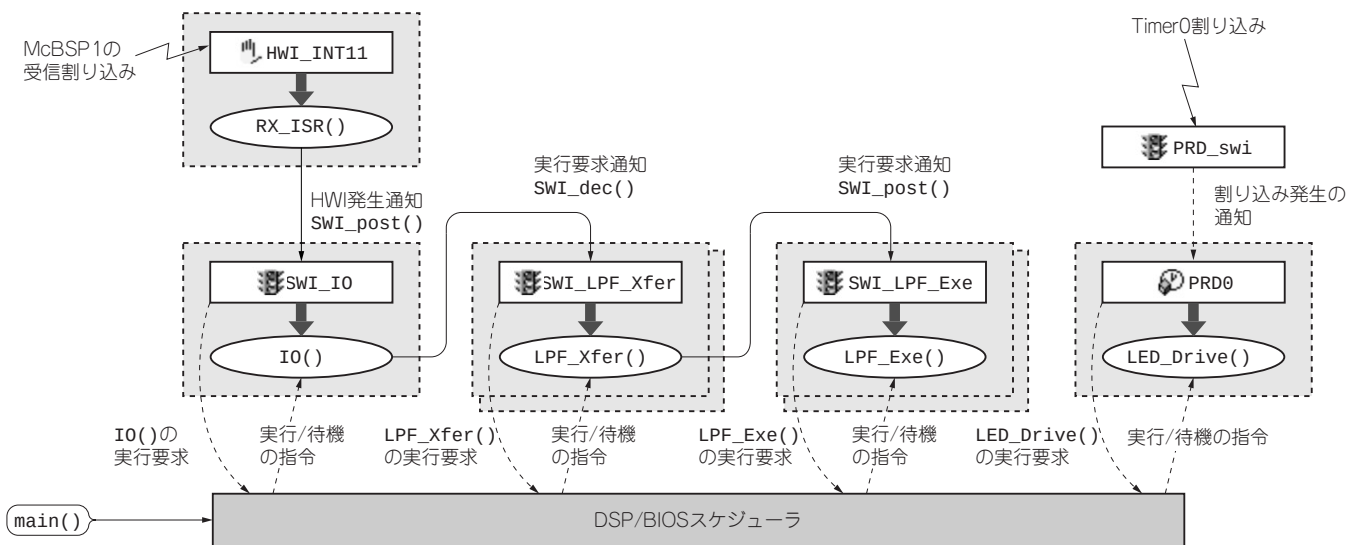
(Software Interrupt)、PRDオブジェクトとメモリを設定する必要があります。図2は、システムに必要なPRDオブジェクトとSWIオブジェクトをConf_OLS.cdbに追加したようすです。SWIオブジェクトの中のPRD_swiは、PRDオブジェクトを追加したときに自動的に追加されます。

● HWI オブジェクトの設定

A-Dコンバータからのデータ受け取りに使用するMcBSP1の受信割り込みを、HWIオブジェクトのHWI_INT11に対応させます。その設定方法は連載第1回^{注2}と同じなので省略します。

● SWI オブジェクトの設定

手動で追加したSWIオブジェクトは五つです。それらの名前をSWI_IO, SWI_LPF_Xfer, SWI_HPF_Xfer,



main()関数終了後に起動

IO(): アナログ信号入力処理, LPF_Xfer(): バッファ間のデータの転送, LPF_Exe(): FFTを利用するFIRフィルタの計算, LED_Drive(): LEDの点滅制御

図1 DSP/BIOSを利用して作成するシステム(右チャンネルの高域通過フィルタは省略)