第5章

ステレオ入力のA-D/D-A変換のひな型

C6713DSKのサンプル・プログラム

付属CD-ROMに収録されたC6713 DSK用サンプル・プログラムについて解説します．

DSK上のステレオA-Dコンバータから入力したデータをそのままステレオD-A変換するだけですが，読者自身でプログラムを作成する場合のひな型として利用してください

本章ではライン入力のプログラムについて解説しますが，マイク入力のプログラムも付属CD-ROMに収録してあります(マイク入力プログラムはモノラル入力になる)．

A-D/D-A(CODEC)と，DSP間のデータ転送方式の異なる下記の3種類のプログラムがあります．

- through_poll：ポーリングで1サンプルずつデータ転送
- through_intr：シリアル・ポート受信割り込みを使って1サンプルずつデータ転送
- through_edma：DMAを使ってブロックI/O

through_pollはポーリングを用いたプログラムです．A-D変換が終了して，CODECからデータが転送されるのをプログラム内でつねに監視していて，データが送られてきたら次の処理に移ります．A-D変換が終わるまでの監視中は他の処理ができず非効率なので，通常はこの方法は用いられません．

through_intrは，DSPのシリアル・ポートにA-Dコンバータからデータが1サンプル転送されるたびに割り込みを発生させて，割り込みルーチン内でデータを読み取り，D-A出力を行います．これがDSPプログラムのもっとも基本的な構成です．

through_edmaは，A-D/D-A変換とDSP間のデータ転送にDMAを用いています．A-D/D-A変換ともにそれぞれ二つのバッファを交互に切り替えながらブロック転送を行います〔第2章の図2-6(b)参照〕．

1サンプルずつのデータ入出力と比較すると，ブロック転送はI/Oのオーバー・ヘッドが少なくなります．当然ながらブロック転送を用いたときにはDSPでの演算処理もブロック処理となりますから，C6713内蔵の複数の演算器を効率的に用いるプログラムを作成して処理の高速化を図ることが可能です．その反面，バッファ長ぶんの遅延がつくために，低処理遅延が求められるアプリケーションには向きません．

■ ひな型を利用したプログラム作成について

本章以降で紹介しているサンプル・プログラムの大部分は，シリアル・ポート受信の割り込みを使って1サンプル単位でのA-D，D-Aを行っています．一部のプログラムのみDMAを用いたブロック転送をしています．

これらのプログラムをひな型として読者自身でプログラムを作成する場合は，新たに作成したディレクトリにファイルをすべてコピーしてそのまま使ってください．

プログラム名を変更して使用する場合は，以下の手順で作業を行ってください．以降ではサンプル・プログラムthrough_intrを元にmyprogという名前のプログラムを作成するものとして説明しています．

(**1**) サンプル・プログラムthrough_intrのプロジェクトに含まれるファイルのうち，以下の五つのファイルを新たに作成したディレクトリにコピーします〔**図5-1**(**a**)，(**b**)〕．

```
c6713dsk.h
lnk.cmd
through_intr.c
through_intr.pjt
vectors.asm
```

各ファイルの読み取り専用の属性はすべてはずしてください．

(**2**) through_intr.c，through_intr.pjtを，それぞれmyprog.c，myprog.pjtにリネームします〔**図5-1**(**c**)〕．

(**3**) CCSを起動して，Project→Openでプロジェクトmyprog.pjtを開きます．すると，CCSはリネーム前のthrough_intr.cを読み込もうとして，**図5-1**(**d**)のようなエラー・メッセージを出します．ここでRemoveのボタンを押します(プロジェクトからthrough_intr.cを削除)．

(**4**) Project→Add Files to Projectでmyprog.cをプロジェクトに追加します．

(**5**) Project→Build Optionsでリンカのオプションの出力ファイル名(Output Filename)の指定をthrough_intr.outからmyprog.outに変更します〔**図5-1**(**e**)〕．

(**6**) これでプロジェクトの修正は終了しました．Project→Saveで修正したプロジェクトをいったんセーブします．

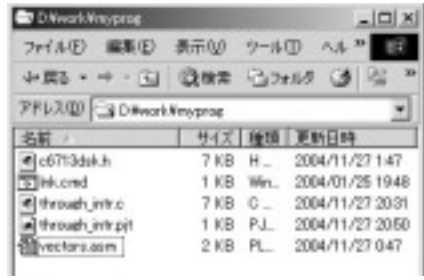
(**7**) Project→Rebuild Allでコンパイル，リンクが正常に終了して実行プログラムmyprog.outが生成されることを確認してください．

(**8**) Debug→ConnectでCCSとDSKを接続します．

(**9**) 次に，File→Load Programでmyprog.outをDSKにダウンロードして，Debug→Runで実行します．プログラムの中身は修正していないので，元のサンプル・プログラムと同様にA-Dコンバータに入力した信号がそのままD-Aから出力されるはずです．

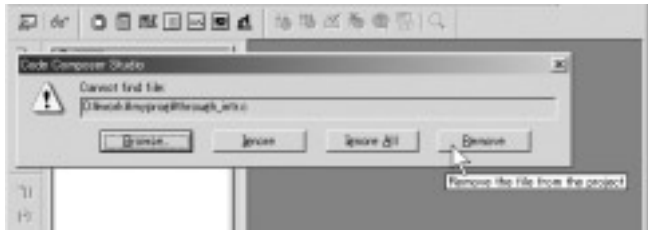
ここまでの確認が終わったら，次はCソース・プログラムmyprog.cに手を加えて自分のプログラムの作成に取りかかってください．

（a）サンプル・プログラム through_intr に含まれるファイル

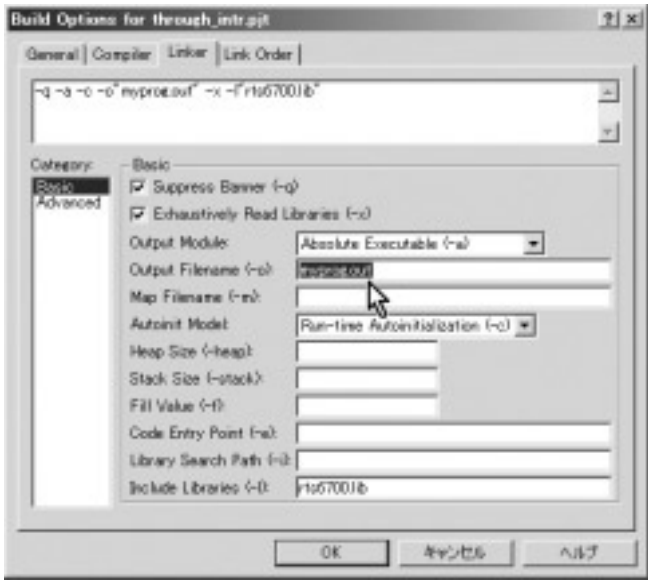


（b）新たに作成したディレクトリに五つのファイルをコピーする

（c）through_intr.c，through_intr.pjt をそれぞれ myprog.c，myprog.pjt にリネームする



（d）Project→Open でプロジェクト myprog.pjt を開くとエラー・メッセージが出るので，Remove のボタンを押す



（e）Project→Build Options で出力ファイル名を through_intr.out から myprog.out に変更する

〈図5-1〉付属CD-ROMの収録サンプル・プログラムをひな型にプログラムを作成するときの手順
プロジェクト through_intr を元に myprog を作成する場合

5-1 サンプル・プログラムの解説

サンプル・プログラムの処理の内容は，**図5-2**のとおりです．

A-D変換した信号をD-A変換するだけですが，左チャネル出力にのみ2秒のディレイがつきます（サンプリング周波数48kHz，ディレイの値は96000サンプル）．ディレイはリング・バッファを使って実現しています．

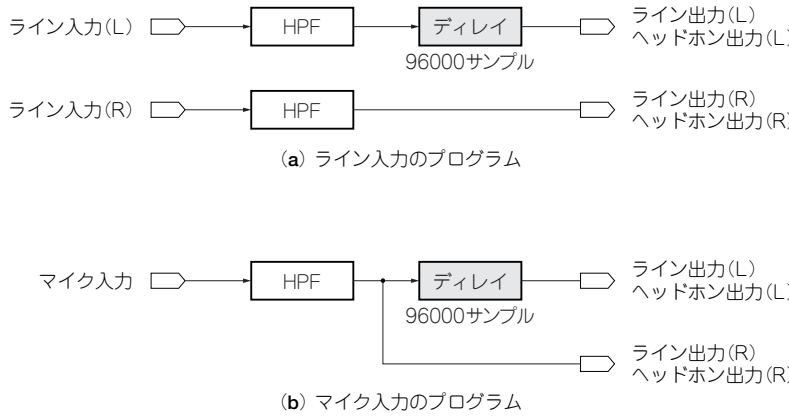
また，入力には直流成分をカットするためのハイパス・フィルタ（以降HPF）が付いています．音声・オーディオ用のΣΔ型A-Dコンバータは直流信号を扱うことができませんが，出力にDCオフセットが付くことがあるので，それを除去するためのHPFを使います．DSK搭載のTLC320AIC23はもともとDCカットのためD-A出力側にディジタルHPFがついていますが，説明のためにプログラム側にもHPFの処理を入れました．

■ through_intr.c

まず，ライン入力，シリアル・ポート受信割り込みを使ったプログラムthrough_intr.cの内容について説明します（**リスト5-1**）

11行目で直流成分のカット用のHPFの係数を定義しています．12行目で定義しているBUF_LENがディレイを実現するためのリング・バッファ長です．

ここでサンプリング周波数は48 kHzなので，BUF_LEN = 96000としたときの遅延時間は，



〈図5-2〉サンプル・プログラムthrough_poll, through_intr, through_edmaの処理
同じファイル名でライン入力とマイク入力のプログラムがそれぞれ付属CD-ROMの別々のディレクトリに収録されている．左チャネルの出力にのみ2秒のディレイがかかる（サンプリング周波数48 kHz）．HPFは直流成分をカットするためのもの

注5-1：英単語pointerの意味のとおりのポインタ．C言語のポインタ変数ではない．本書のプログラムではポインタ変数は一切使用していない．ポインタ変数を用いたコーディングも可能だが，その必要性はない．算術演算主体のディジタル信号処理で，明示的にポインタ変数を使わなければうまく記述できないような処理はないはず．