付録G うまく動かない場合の対処

1 よくある「うまくいかない場合」

本や雑誌に書かれているプログラムをそのまま[注1]使ったのに，ビルドがうまくできない，あるいはビルドはできたけれどもうまく動作しないということを聞くことがある．本書のプログラムで，多くの場合はCode Composer Studio(以下，CCS)のバージョンが違うか，フォルダ(ディレクトリ)の設定が，書籍などに書かれているものと異なっているということが原因である．**図1**には，筆者のパソコンで，CCSに関するフォルダの構成をツリー形式で示す．

筆者が以前に執筆した「C言語によるディジタル信号処理入門」(CQ出版社)でも，その書籍の中のプログラムがうまく動かないということを言われて，調べたところ，フォルダの設定が異なっているためにうまくビルドできないというケースが多かった．

そこで，ここではうまくいかないケースとその対処について説明する．なお，筆者が相談を受けた中で，次の二つのケースで，どちらも「うまく動かない」と表現している場合もよく見受けられた．一つはビルドがうまくいかない場合[注2]である．もう一つは，ビルドはうまくいったが，実行形式のファイル(*.outというファイル)をロードしてDSKボードのうえで実行したところうまく動作しない場合である．したがって，ここでは，両方のケースについてその対処法を説明する．

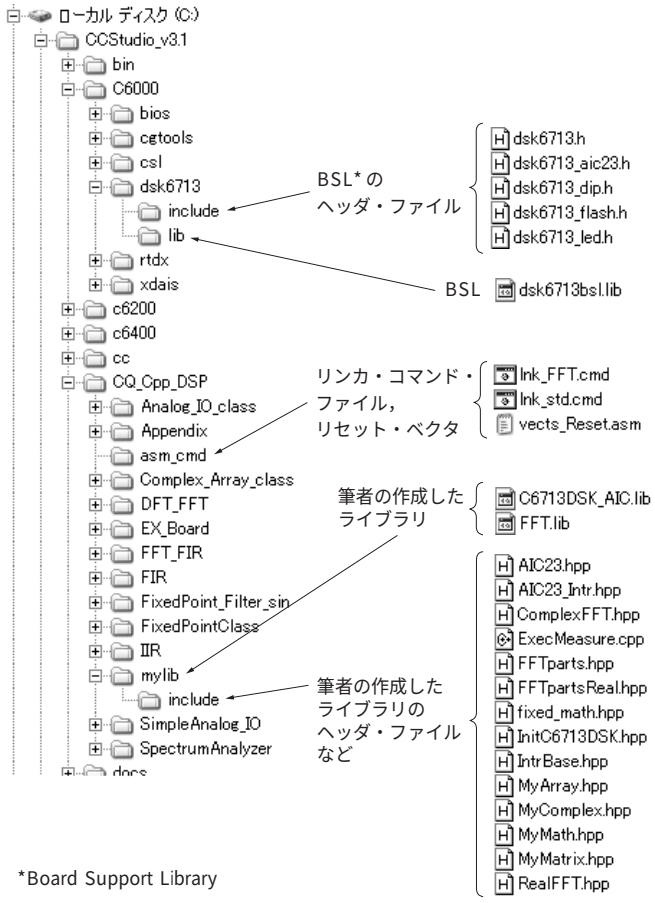

図1　筆者のパソコンのCCSに関連するフォルダの構成

なお，本書付属のCD-ROMからコピーしたファイルは，基本的に読み取り専用になっている．その関係で，再ビルドを行った場合に，ファイルを上書きできないためにビルドが正しく行われずエラーになることもある．そこで，本書付属のCD-ROMからコピーしたファイルは，読み取り専用の属性を最初に解除しておいたほうがよい．以下に読み取り専用の解除の方法を二つ示す．

▶ 一つのフォルダ内のファイルの場合

ファイルを選択し，右クリックして，「プロパティ(R)」を開く．そこで，「読み取り専用(R)」のチェック・マークを外す．

▶ 複数のフォルダのファイルの場合

読み取り専用の属性を複数のフォルダにわたって一度に解除するには，コマンド・プロンプトのattribコマンドを使用すると便利である．

例：フォルダ(ディレクトリ)CQ_Cpp_DSP以下にあるすべてのファイルの「読み取り専用」属性を解除する．

```
attrib -r CQ_Cpp_DSP\*.* /s
```

また，CCSのバージョンにより，CCSに関連するファイルが置かれるデフォルトのフォルダ名

注1：本書付属のCD-ROMに収録したプログラムをそのまま利用した場合も同じ．
注2：これも細かく分けると，コンパイルでエラーが出るケースと，コンパイラは通るがリンクでエラーが出るケースがある．

見本

が以下のように異なっている．

- Ver. 3.0より前：ti
- Ver. 3.0　　：CCStudio
- Ver. 3.1　　：CCStudio_v3.1

したがって，以下で説明する対処では，このことを踏まえてほしい．

うまくいかない場合は，まずどの段階でうまくいかないのかを見きわめる必要がある．**図2**にプログラムをビルドしてからDSKボードのうえで実行形式のプログラムを実行するまでの手順の概略を示した．ビルド(build)とは，この図でわかるように，コンパイルとリンクの両方が含まれる．ロードとは，ビルドの結果生成された実行形式のファイル(*.out)の内容をDSKボードのメモリへ転送することである．

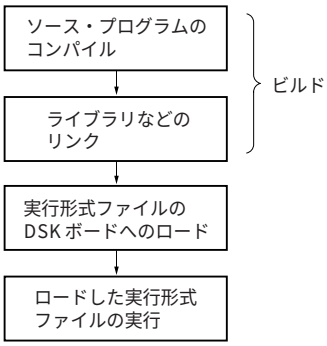

図2 プログラムをビルドしてDSKボードの上で実行するまでの過程

2 途中で発生する問題への対処

以下では，コンパイル，リンク，ロード，実行の各段階で発生した問題への対処について説明する．

● コンパイル

ここでは，本書付属のCD-ROMに収録されているファイルを使った場合について説明するので，プログラムに文法的な誤りはないものとする．

▶ ヘッダ・ファイルのインクルード

よくあるケースは，必要なファイルをインクルードできないというものである．本書に出ている大部分のプログラムは，ボード・サポート・ライブラリ(以下，BSL)のヘッダ・ファイル(dsk6713.hなど)と筆者の作成したライブラリのヘッダ・ファイル(AIC23.hppなど)をインクルードするようになっている．

ところで，CCS ver.3.1に関連するファイルは，デフォルトでフォルダCCStudio_v3.1に置かれるので，CCS ver.3.1以外のCCSを使っている場合には，ファイルをインクルードできないため，エラーを発生する．また，CCS ver.3.1を使っていてもCCSに関連するファイルが，デフォルトのフォルダであるCCStudio_v3.1に置かれていない場合や，インクルード・ファイルを**図1**に示す，筆者の例と異なるフォルダにおいている場合も，同様にエラーが発生する．

そのような場合は，BSLおよび筆者の作成したライブラリのヘッダ・ファイルにアクセスするため，パス(path)を適切に設定する必要がある．これらのパスの設定の例は，第3章の第1節の中の「実行ファイルの作成と実行」の項目，および第3章の図2を参照することとする[注3]．ここで設定するパス以外のところにヘッダ・ファイルが置かれている場合は，これを参考に適切に設定する必要がある．

▶ DSPファミリの指定のまちがい

プロジェクトの新規作成を新たに行った際，DSPファミリの指定が正しく行われていない場合にコンパイル・エラーが発生することがある．それは，TMS320C6713のintrinsics関数の中で，浮動小数点演算を行うものを使うようなintrinsics関数を使っているプログラムを，デフォルト状態あるいはC62xやC64xを指定した状態でコンパイルを行った場合に発生する．その場合は，付録Eの第3節の中の「ビルド・オプションの設定」の項目の中の「ターゲットとなるDSPファミリの設定」と付録Eの図17を参考に，C671xを選択すればよい．

▶ ファイルが読み取り専用になっている

本書付属のCD-ROMからコピーしたファイルは，読み取り専用になっている．したがって，コンパイルの結果生成される*.objファイルなどを上書きすることができず，エラーが発生する．本書付属のCD-ROMからコピーしたファイルなどをビルドする場合には，事前に付録Gの冒頭に説明した方法で，読み取り専用を解除しておかなければならない．

▶ *.asmファイルが見つからない

リセット・ベクタが記述されたファイル(vects_Reset.asm)が筆者の環境と同じフォルダに存在しない場合にエラーが発生する．その場合は，いったんvects_Reset.asmを取り除き，再度プロジェクトに追加すればよい．追加の方法については，第3章の第1節の「実行ファイルの作成と実行」を参照のこと．

● リンク

▶ リンカ・コマンド・ファイルが見つからない

リンカ・コマンド・ファイルが見つからない場合には，先に述べた「*.asmファイルが見つからない」と同様に対処する．

▶ ライブラリが見つからない

本書に出ている大部分のプログラムは，BSL(dsk6713bsl.lib)と筆者の作成したライブラリ(C6713DSK_AIC.lib)[注4]をリンクする．これらのライブラリをロードする

注3：この設定の例は，CCSのVer.3.1の場合になっている．なお，CCSのVer.3.0より前のバージョンでは，BSLのヘッダ・ファイルのフォルダはパスを指定する必要はなかった．
注4：FFTを使っているプログラムでは，FFT.libも必要．

見本

ためのオプションの記述が，リンカ・コマンド・ファイル（lnk_std.cmd，lnk_FFT.cmd）にある．このとき，ファイルはフルパス名で指定されているので，これらのファイルが実際に存在するフォルダ名を使って，フルパス名で指定する[注5]．

その書き方については，第3章の第1節にある「リンカ・コマンド・ファイル」の項目と第3章のリスト3，または第7章の第5節の説明と，第7章のリスト7を参照のこと．

▶ ライブラリ作成時に異なるバージョンのコンパイラを使用

本書付属のCD-ROMに収録されている筆者の作成したライブラリは，CCSのver.3.1のコンパイラで作成したものである．その関係で，ほかのバージョンのCCSを使った場合，うまくリンクできないこともある．その場合は，筆者が作成したライブラリをコンパイルし直す必要がある．

▶ ファイルが読み取り専用になっている

本書付属のCD-ROMからコピーしたファイルは読み取り専用になっている．したがって，リンクの結果生成される*.outファイルなどを上書きすることができないため，エラーが発生する．本書付属のCD-ROMからコピーしたファイルなどをビルドする場合には，事前に冒頭に説明した方法で，読み取り専用を解除しておかなければならない．

● ロード

▶ メモリ・サイズが小さい

最近のC6713 DSKボードに搭載されている外部メモリは16Mバイトであるが，古いボードでは8Mバイトのものがある．本書付属のCD-ROMに収録されている実行形式のファイルを古いDSKのボードにそのままロードした場合，エラーが発生することがある．

その場合は，リンカ・コマンド・ファイルのMEMORY指示命令で指定する外部メモリのサイズを，ボードに搭載されている外部メモリのサイズ以下に設定してから，再度ビルド[注6]を行う必要がある．リンカ・コマンド・ファイルでの外部メモリのサイズを設定する方法については，第3章の第1節にある「リンカ・コマンド・ファイル」の項目，および第3章のリスト3を参照のこと．

▶ CCSのバージョンの違い

CCS ver.3.1以外の環境で使っている場合に，上記の「メモリのサイズが小さい」の項目で説明した問題以外の原因でうまくロードできない可能性もある．その場合は，再度ビルドを行った後，ロードする．

● 実行

▶ 実行形式のファイルをロードしていない

プログラムをビルドしただけでは，DSKボードのうえで実行することはできない．そこで実行する前に，生成された実行形式のファイルの内容をかならずDSKボードのメモリへ転送する必要がある．その方法については，付録Eの第3節にある「プロジェクトのビルドと実行」の項目を参照のこと．なお，ビルドするたびに実行形式のファイルをロードするのはやっかいなので，付録Eの第2節にある「自動ロード・オプション」の項目，および付録Eの図8を参考にして，ビルドが終わったら実行形式のファイルが自動的にロードされるように設定しておくとよい．

▶ Debugモードでビルドした

Debugモードでビルドを行い，エラーを発生せずにビルドが終わり，実行形式のファイルをロードして実行した場合に，おかしな動作をする場合がある．Debugモードでビルドされた実行形式のファイルは，デバッグのためのいろいろな情報が組み込まれ，最適化も行われていないので，実行スピードが遅くなる．その関係で，とくに処理量の多いプログラムを実行させた場合に，処理が入力のスピードに追いつかなくなり，入力データの取りこぼしを起こし，正常に動作しない場合がある．そのときは，付録Eの第3節にある「ビルド前の準備」の項目，および付録Eの図15を参考にしてReleaseモードに切り替えてから，ビルドすればよい．

▶ CCSのバージョンに違い

CCS ver.3.1以外のものを使っている場合に，うまく実行できない場合があるかもしれない．その場合は，再度ビルドを行った後，ロードして実行する．

▶ DSPファミリの指定のまちがい

プロジェクトの新規作成を新たに行ったとき，DSPファミリの指定が正しく行われていない場合におかしな動作をする場合がある．プロジェクトの新規作成の際は，C62xxファミリのようにハードウェアの浮動小数点演算器を内蔵しないDSPがデフォルトで選択される．そのため実行速度がデータの入力速度に追いつかなくなり，おかしな動作になる．その場合は，付録Eの第3節にある「ビルド・オプションの設定」の項目の中の「ターゲットとなるDSPファミリの設定」と，付録Eの図17を参考に，C671xを選択すればよい．

▶ そのほか

DSKボードのRAM内のCCSに関係する部分が，何らかの原因で破壊されている場合があり，その場合もうまく実行することはできない．その場合は，いったんCCSを終了してから，DSKボードのリセット・ボタンを押すか，DSKボードの電源を落として再度電源を加える．そのとき，ボードのLEDが点滅するので，それが終わるまで待ってから，次の操作を始める．

なお，場合によっては不幸にもDSKボードのハードウェアが壊れている場合もある．そこで，付録Dの第2節に従って，DSKボードの動作の確認を行う．そのとき問題がなければ，その後CCSを立ち上げる．

注5：CCSのVer.3.0より前では，BSLをフルパス名で指定する必要はなかった．

注6：もちろん，コンパイルの必要はないので，リンクを行うだけでもかまわない．

見本