

DSP によるデジタル信号処理実験

盧 存偉

第一章 概論

1.1 DSP とは

DSP は Digital Signal Processing の頭文字で、デジタル信号処理装置の意味である。

本実験に使われる DSP はアメリカ テキサス・インスツルメンツ社の DSP Starter Kit (DSK) TMS320VC5416 である。C5416 DSK Board は、CPU、メモリ、レジスタ、I/O ポートなどが含まれ、高速なデジタル信号処理ができる

DSP はアセンブリ言語で動くが、開発のソフトを介して、C++言語を使う開発は可能になった

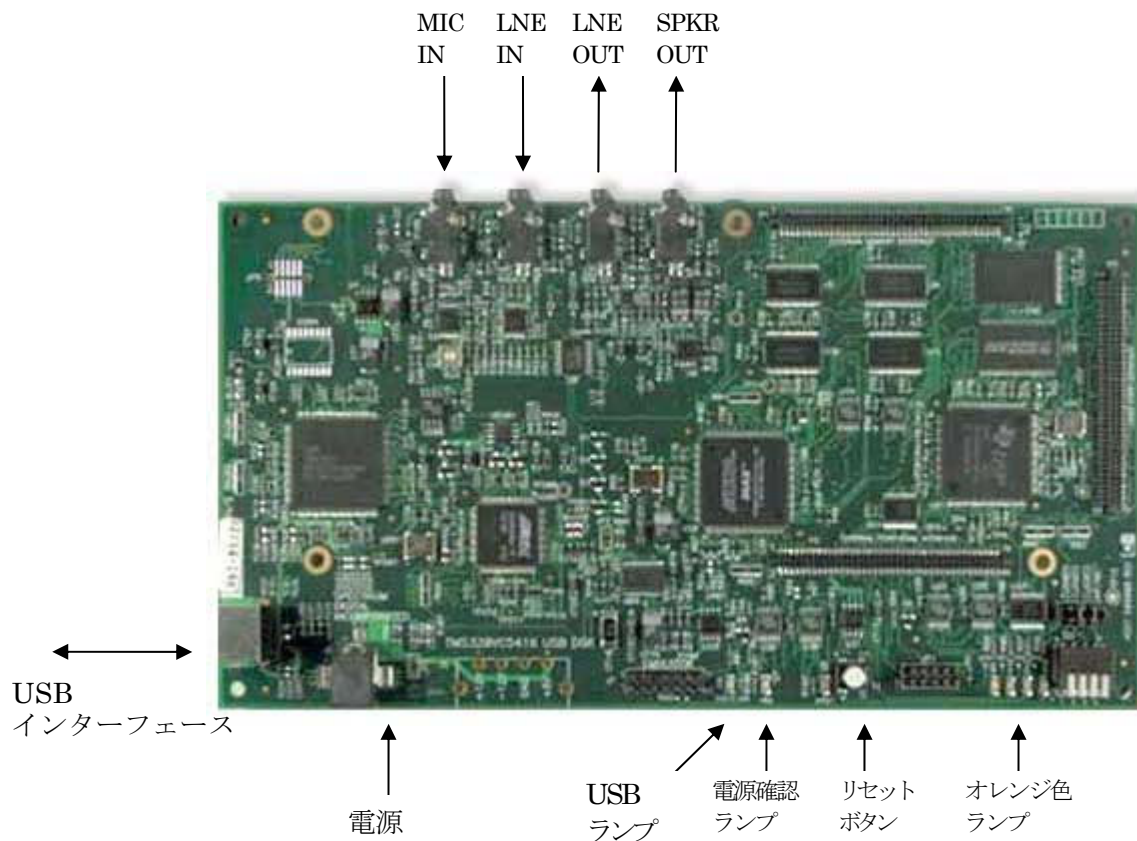
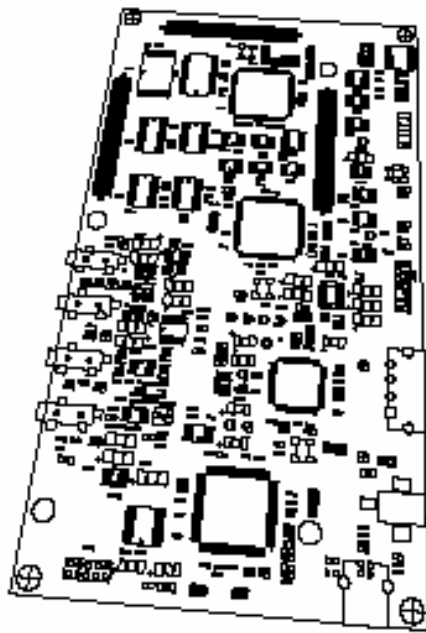


図 1 C5416 DSK Board

1.2 DSP 実験装置の構成

本実験で使う DSK はハードウェアとソフトウェアより構成される。

- ・ ハードウェアの基本構成は下記の四点がある。



C5416 DSK Board



+5V Universal
Power Supply



AC Power
Cord



USB Cable

図 2 TMS320VC5416 のハードウェア



- ・ ソフトウェアはデバッガ C5416 DSK CCS を使う.

- 主な特長：

- ハードウェア

- VC5416 160MHz 搭載

- 20bit/48kHz PCM3002 Stereo CODEC

- 64K ワード SRAM、256K ワード・フラッシュ・メモリ搭載

- USB 接続による快適な JTAG エミュレーション

- デバッグ作業の効率を高めるユーザ・スイッチ/ユーザ・LED

- マイクロフォン/スピーカ・インターフェース

- +5V ユニバーサル電源

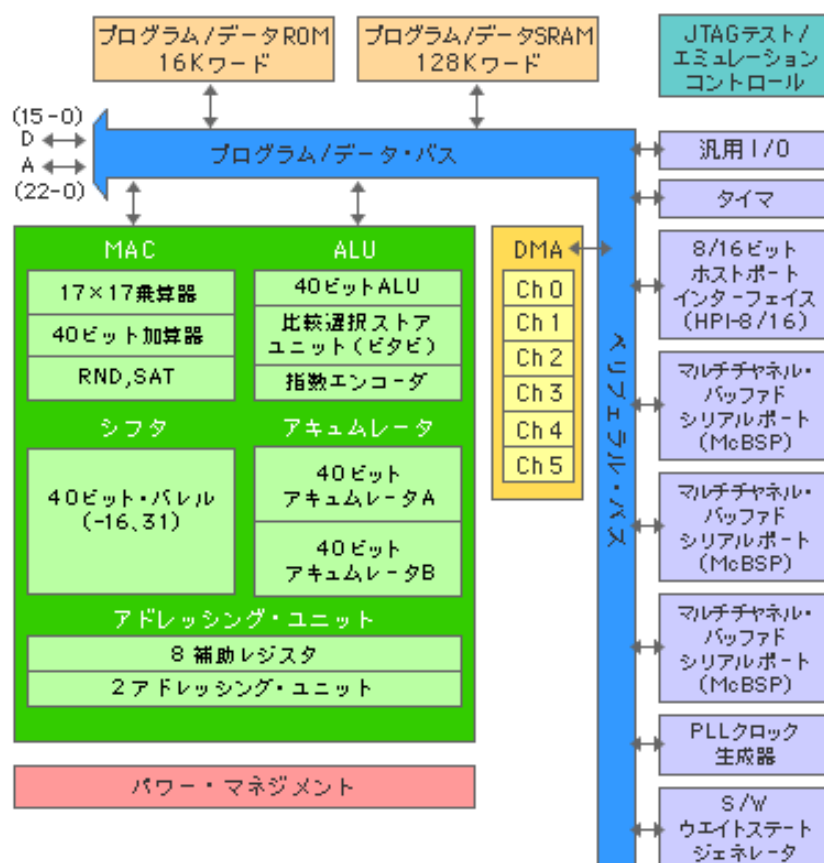
- ソフトウェア

- CC5416—DSK 用 Code Composer Studio

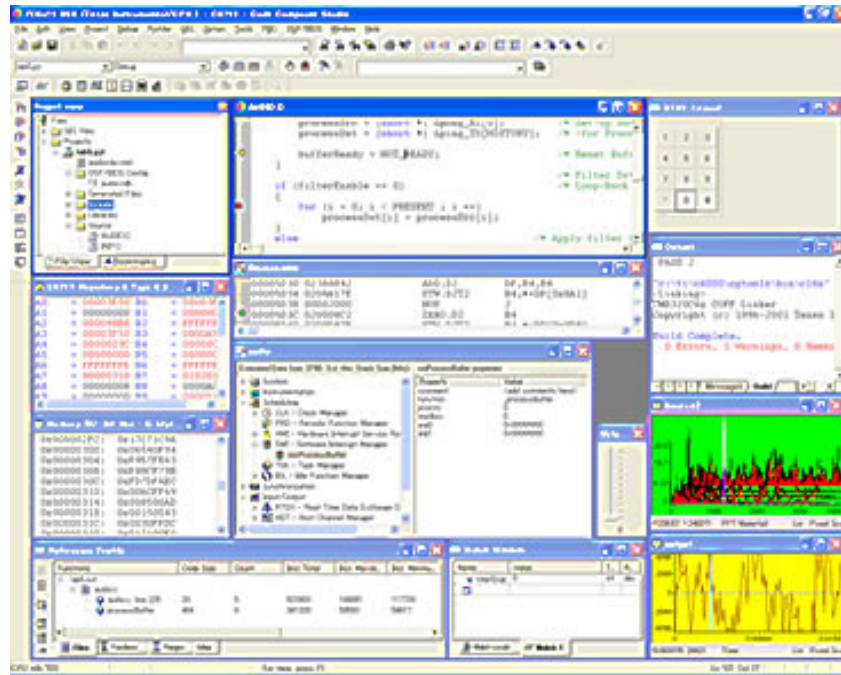
- C コンパイラ、アセンブラ、リンカ (オブジェクト・サイズ制限無し)

- サンプル・プログラム、チュートリアル

- C5416 DSK Board のブロック図：



- Code Composer Studio™ (CCS)の画面



1.3 実験システムの構築及び実験の進め方

静電気防止のため、チップや配線に絶対触れないこと！

- ① コンピュータを起動
- ② 指定された場所で、実験用フォルダを作成。フォルダ名は自分の苗字のローマ字表示を使用。
- ③ Windows が完全に起動されたから、USB ケーブルを用いて、DSK ボードをコンピュータに接続
- ④ 電源ケーブル、電源アダプタを用いて、DSK ボードを電源に接続
注意：アース付きコンセントに接続すること！
- ⑤ 開発ソフト CCS を起動
- ⑥ CCS を用いて、実験用ソースプログラムや実行プログラムを作成
- ⑦ 作成された実行プログラム*.out を DSK に送り、実行させ
- ⑧ 実行結果を確認

実験の方針：サンプルプログラムをベースにして、実験課題を完成すること。

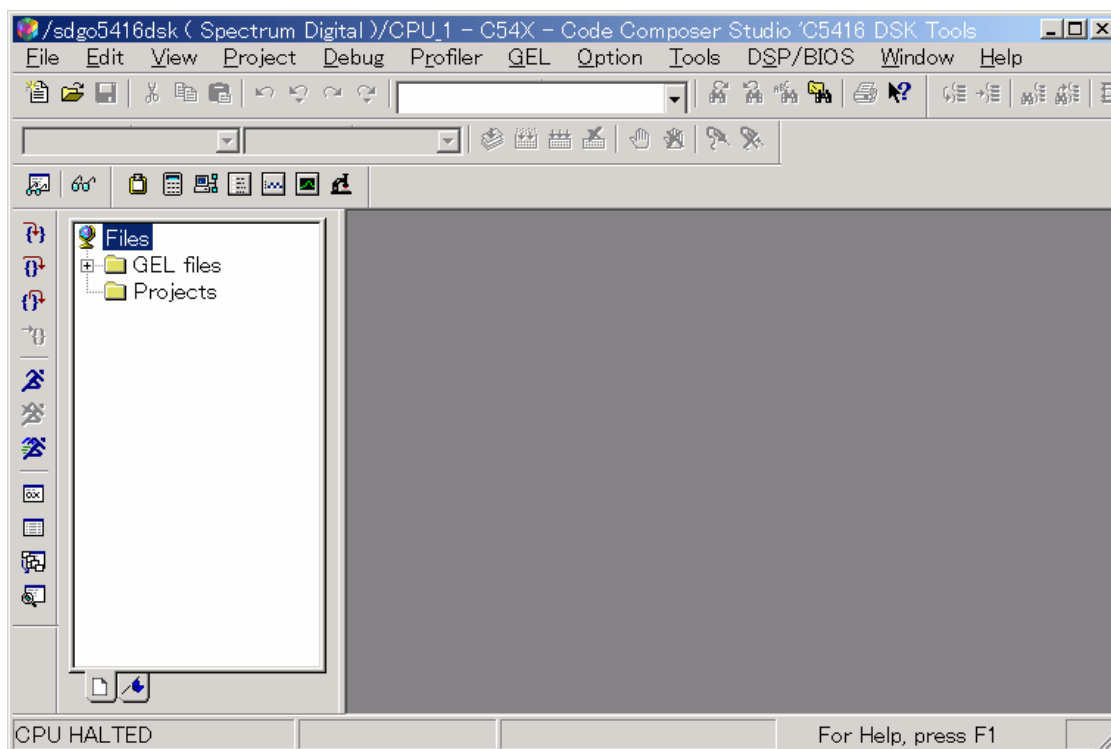
第二章 ランプ点滅制御

2-1 実験の概要

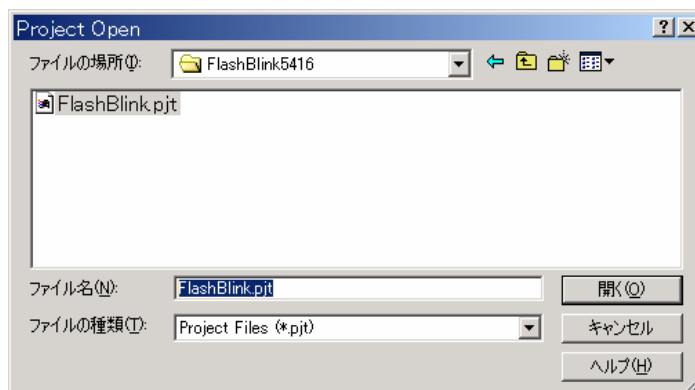
DSK ボードにあるオレンジ色のランプの点灯，消灯をコントロールするための実験

所要機器：DSK ボード

- ① 自分の実験用フォルダに「jiken1」フォルダを作り
- ② サンプルファイルを「jiken1」フォルダにコピー
- ③ 1.3 節の方法で実験の準備を完成，USB の接続状況を確認
- ④ CCS を起動、下記のようなデバッグ画面が表示され



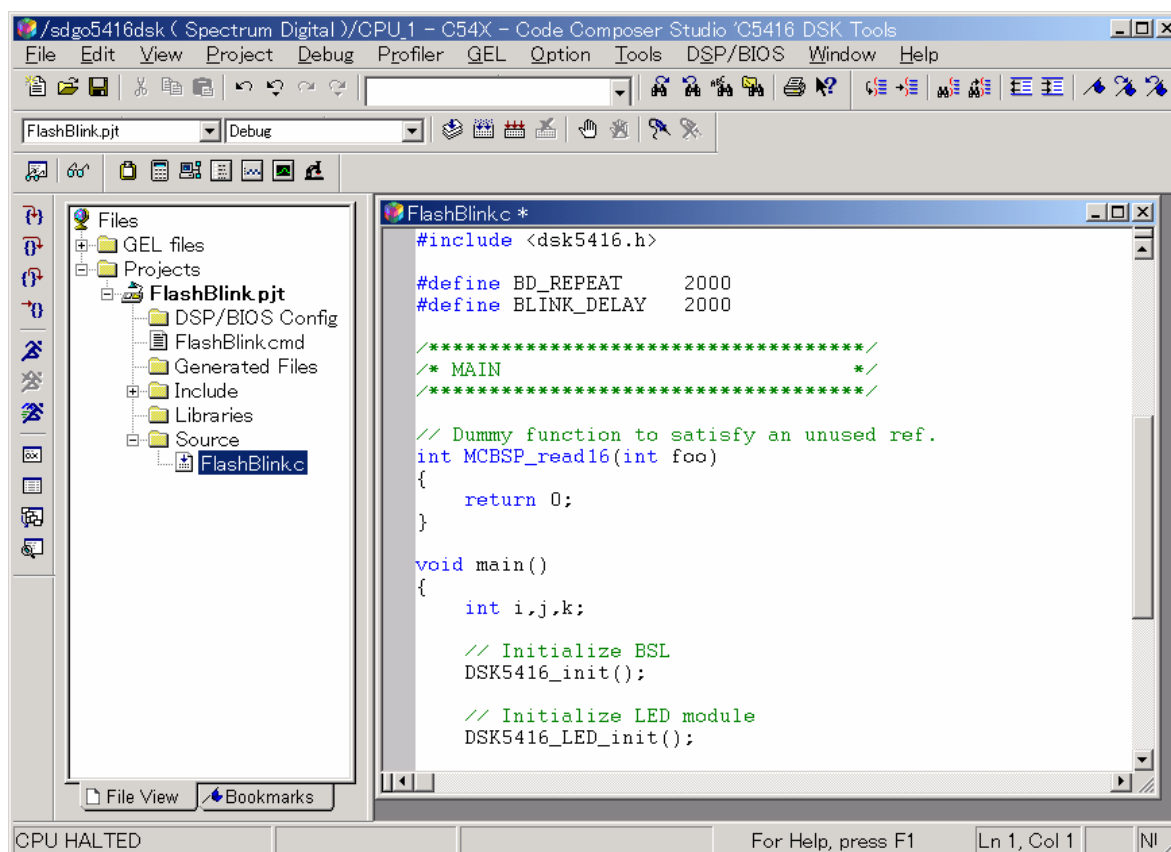
- ⑤ 下記の手順で、実験用プロジェクト(FlashBlink.pjt)を開く
メニューバーの **Project** => **Open** => [Project Open] ウィンドウから FlashBlink.pjt を開く



(囲い英文字はメニューバーの項目である。これから説明を省略)

⑥ Project の内容を確認

特に、ソースの C++プログラム FlashBlink.c の内容をきちんと確認すること。



⑦ 下記の手順で、プロジェクトファイルをビルドして、実行ファイルを生成

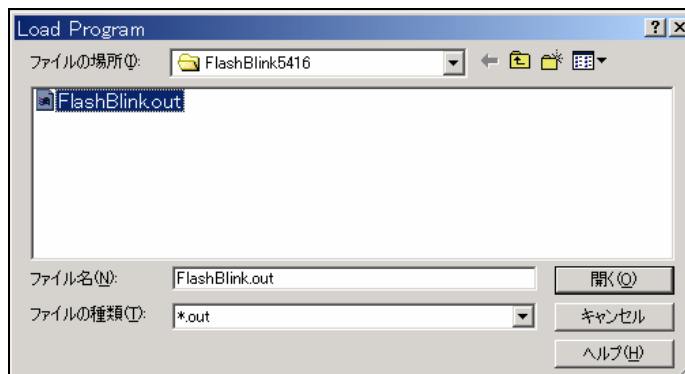
Project => **Rebuild**

もし問題が無ければ、下記のような表示が出るはず

Build Complete,
0 Errors, 0 Warnings, 0 Remarks.

⑧ バイナリ化された実行ファイル FlashBlink.out を DSK に送り

File => **Load Program**



⑨ DSK で実行プログラムを実行し、オレンジ色のランプの状態変化を観察

Debug => **Run**

- ⑩ 実験が終わったら、まずプロジェクトを閉じ

Project=> **Close**

- ⑪ 最後に、CCS を終了

File=> **Exit**

ソースファイルの関数説明：

DSK5416_LED_toggle(a) : a 番目の LED の状態(点灯・消灯)を切り替える。

2-2 課題：

1. Project にある C++ソースファイルの各部分に例のような説明文をつけ、意味を説明せよ。
(入れた説明は、プログラムの実行に対しては影響が無いはず)

```
#define BD_REPEAT    2000    // LED 状態転換時間を決める係数    (例)
#define BLINK_DELAY  1000
/*****
/* MAIN                                */
*****/

// Dummy function to satisfy an unused ref.
int MCBSP_read16(int foo)
{
    return 0;
}

void main()
{
    int i,j,k;

    // Initialize BSL
    DSK5416_init();

    // Initialize LED module
    DSK5416_LED_init();
    for(k=0; k<10; k++)
    {
        DSK5416_LED_toggle(0);
        for(i=0; i<BD_REPEAT; i+=1)
            for(j=0; j<BLINK_DELAY; j+=1);
            DSK5416_LED_toggle(1);
        for(i=0; i<BD_REPEAT; i+=1)
            for(j=0; j<BLINK_DELAY; j+=1);
            DSK5416_LED_toggle(2);
        for(i=0; i<BD_REPEAT; i+=1)
```

```

        for(j=0; j<BLINK_DELAY; j+=1);
            DSK5416_LED_toggle(3);
        for(i=0; i<BD_REPEAT; i+=1)
            for(j=0; j<BLINK_DELAY; j+=1);
    }
}

```

2. 各ランプの状態転換時間を現状の 2 倍に変換し、実行せよ。

ヒント: LED 状態転換時間を決める係数の値を変換してから、Build => Load Program => Run の順で操作を行う。

3. 実行プログラムの実行時間 (Run から停止までの所要時間) は大体どのくらいであるか、この時間内で、各 LED の状態は何回転換したか? この転換回数を現状の 2 倍にし、実行せよ。

4. 下記の機能を実現せよ。

まず、4 つの LED は全部点灯し、3 秒間を続ける。その後、左側の二つ LED を消灯し、4 秒間を続ける。続いて、右側の二つ LED を点灯し、左側の二つを消灯する。この状態は 5 秒間を続ける。最後に、全部の LED を消灯する。

状態一: ○○○○ 続く時間 3 秒
 状態二: ●●○○ 続く時間 4 秒
 状態三: ○○●● 続く時間 5 秒
 状態四: ●●●● 終了状態である。

5. 下記の関数を使って、実験をなさい。

DSK5416_LED_on(a) : a 番目の LED の状態を ON にする。

DSK5416_LED_off(a) : a 番目の LED の状態を OFF にする。

(1 - 3 は必須, 4 - 5 は選択課題である)

第三章 音波出力実験

3-1 実験の概要

DSK ボードでサイン状の波形を生成し、LNE OUT 端子より出力する。出力信号はヘッドホンやオシロスコープで確認できる。

所要機器：DSK ボード，ヘッドフォン

接続：ヘッドホンの「MIC」端子（ピンク色）⇔DSK ボードの「MIC IN」

ヘッドホンの「ヘッドホン」端子（緑色）⇔DSK ボードの「LNE OUT」

- ① 自分の実験用フォルダに「jiken2」フォルダを作り
- ② サンプルファイルを「jiken2」フォルダにコピー
- ③ 1.3 節の方法で実験の準備を完成，USB の接続状況を確認
- ④ CCS を起動
- ⑤ 実験用プロジェクト(Tone.pjt)を開き
- ⑥ Project の内容を確認
特に、ソースの C++プログラム Tone.c の内容をきちんと確認すること。
- ⑦ プロジェクトファイルをビルドして、実行ファイルを生成
- ⑧ バイナリ化された実行ファイルを DSK に送り
- ⑨ DSK で実行プログラムを実行し、ヘッドホンの右の耳には 1kHz の音波信号が聞こえるはず
- ⑩ 実験が終わったら，CCS を終了

3-2 課題：

1. Project にある C++ソースファイルの各部分に説明文をつけ、意味を説明せよ。
2. ヘッドホンの左の耳にも音波信号が聞こえるように，ソースプログラムを修正せよ。
3. 出力音波周波数を 2kHz にするように，ソースプログラムを修正せよ。
4. 出力周波数が 1kHz と 2kHz 交代で出るように，ソースプログラムを修せよ。
5. ド，レ，ミのような音を作ってみよう。

（1－3 は必須，4－5 は選択課題である）

第四章 チャレンジ実験：音声入出力

4-1 実験の概要：

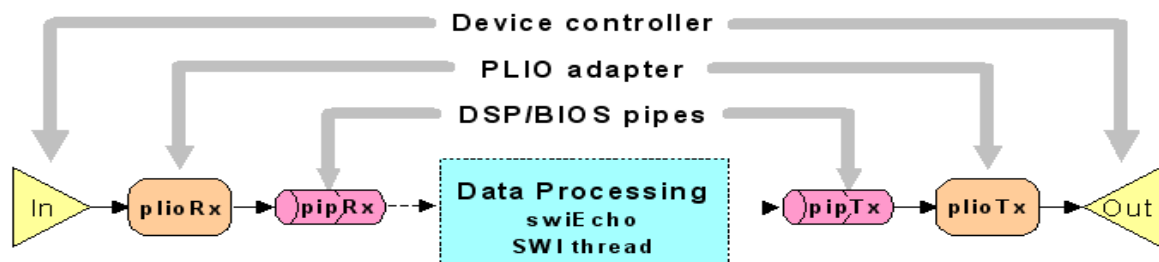
本実験はチャレンジ用で、余裕がない学生はしなくてもかまわない。

信号の流れは下記のように、マイクからの信号は DSK ボードに入力され、DSK において処理された（エコーを加える）後、出力される。

所要機器：DSK ボード、ヘッドホン

接続：ヘッドホンの「MIC」端子（ピンク色）⇔DSK ボードの「MIC IN」

ヘッドホンの「ヘッドホン」端子（緑色）⇔DSK ボードの「LNE OUT」



サンプル（audio.pjt）を参考して、進めてください。

4-2 課題：自由設定