

```
/*-----
/
/ 実験2-2, 2-3, 2-4を実現する汎用ソース
/ 用いたい信号に応じて、適宜読み込み先のファイル名を変える。
/ スペクトル(ゲイン)を算出する前の段階で標準出力を行えば2-2が、
/ ゲインを出力すれば2-3, 2-4が実行できる。
/
/ 作成:木村優太郎
/ 日時:2017/04/24
/
/-----*/

#include<stdio.h>
#include<math.h>

#define DIGI_ROUND 10

const int M = 512;          //[-]

FILE *fp;

double blkHrsWindow(int n){

    return (0.423 - 0.498*(cos( (2*M_PI*n/M) )) + 0.0792*(cos( (4*M_PI*n/M) )));
}

int main(void){

    int i = 0;
    int n = 0;
    int k = 0;              //フーリエ変換後横軸(周波数[Hz])

    double ans = 0;
    double r = 0;

    double signANA[512] = {0};
    double valFilted[512] = {0};

    char *fname = "exp1-5.csv";
    double time = 0;
    double signTmp = 0;
```

```
double valIm[512] = {0};
double valRe[512] = {0};

double gain[512] = {0};

// 離散フーリエ変換対象のファイル読み込み
fp = fopen( fname, "r" );

if( fp == NULL ){
    printf( "%sError Can't Open Fiel \n", fname );
    return -1;
}
while( (fscanf( fp, "%lf,%lf", &time, &signTmp ) ) != EOF){

    //printf( "%lf,%lf \n", time, signTmp);
    signANA[i] = signTmp;
    i++;
}
fclose(fp);

//窓関数をかける
for(i=0; i<=M; i++){

    valFilted[i] = signANA[i]*blkHrsWindow(i);
}

//以下離散フーリエ変換
for(k=0; k<=M; k++){

    for(n=0; n<M; n++){

        valRe[k] += valFilted[n] * cos( -2.0 * M_PI * n * k /M);
        valIm[k] += valFilted[n] * sin( -2.0 * M_PI * n * k /M);

    }

    //ゲイン[dB]の算出
    r = (valRe[k]*valRe[k]) + (valIm[k]*valIm[k]);
    gain[k] = 10* log10(r);

    //標準出力へ表示
    printf("%d,%lf\n", k, gain[k] );
}

return 0;
}
```