

TR - H - 172

0001

磁気脳波計測装置を用いた実験のための
聴覚刺激呈示系の試作

小幡 靖

正木 信夫

(奈良先端科学技術大学院大学)

1995. 11.10

ATR人間情報通信研究所

〒619-02 京都府相楽郡精華町光台2-2 ☎ 0774-95-1011

ATR Human Information Processing Research Laboratories

2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan

Telephone: +81-774-95-1011

Facsimile: +81-774-95-1008

磁気脳波計測装置を用いた実験のための聴覚刺激呈示系の試作

小幡 靖 (奈良先端科学技術大学院大学)

正木 信夫

1 目的

95年1月、ATR人間情報通信研究所は、住友金属工業株式会社ハイクオリティライフ研究所(京都府相楽郡精華町光台3-5)の所有する磁気脳波計測装置(Magneto-Encephalo-Graphy; MEG)を用いて、音声の聴取と生成に関する研究を行うための研究協力を開始した。本報では、このMEG装置を用いて、音声の聴取に対する脳内活動を観測する際に必要な、聴覚刺激の呈示装置について報告する。

2 聴覚刺激に対する磁気脳波計測の実験装置

聴覚刺激に対する磁気脳波計測のための装置の全体図を図1に示す。実験装置は3つの部分からなる。

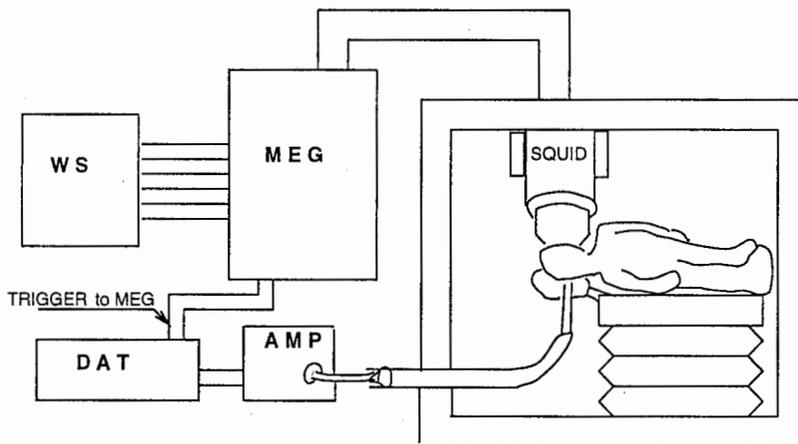


図 1: 実験系の全体図

1. 聴覚刺激生成系: 聴覚刺激生成系は、予め録音された聴覚刺激を再生する。データ収集のタイミングを知らせるためのトリガー・パルス、を、脳磁界情報収集系である MEG 装置に与える。

2. 聴覚刺激伝達系：聴覚刺激伝達系は、生成系で再生された聴覚刺激をアンプで増幅し、音響管を用いて磁気シールド室内の人間の聴覚に伝達する。
3. 脳磁界情報収集系：脳磁界情報収集系は、脳内で生成される磁界を測定し、ワークステーションに伝え、記録する。

なお、聴覚刺激生成系と聴覚刺激伝達系を合わせて「聴覚刺激呈示系」とする。以下、聴覚刺激伝達系、聴覚刺激生成系および聴覚刺激用テープの作成方法について説明する。

3 聴覚刺激伝達系の作成

MEG 装置を用いた実験においては、通常聴覚刺激呈示装置に用いられるスピーカーやヘッドフォン等の電気音響変換装置を刺激呈示装置として用いることはできない。それは、スピーカーから発生する磁界が脳内で発生する磁界よりも強いために、MEG 装置に用いられている高精度の磁気センサ、超伝導量子干渉装置 (Superconducting QUantum Interference Device, SQUID) では、脳内活動を正確に観測できない可能性があるからである。[1, 2, 3, 4, 5]。

そこで、聴覚刺激を磁気センサである SQUID の格納されている磁気シールド室の外で音響信号に変換し、これを音響管により被験者に伝えることにした。

3.1 音響管の作成

聴覚刺激伝達系に使用した機器は以下のとおり。

1. 増幅器：オーディオ・テクニカ、AT-MA500。
2. イヤホン：ソニー、MDR-E837(L)。アンプのイヤホン端子に接続。音源としては1チャンネルを使用。
3. 音響管：以下の説明参照。

音響管は、図2に示す構造にした。ホースの一方の端にイヤホンを挿入し、音源とした。音を伝達する管の壁面による損失を小さくするために、ホースの内径をイヤホンの内径と同じものにし、耳への挿入部の近くで、しだいに細くなる構造とした。表1に、ホースとビニールチューブの寸法を示す。各部分の接合は、細い方の接合部の外周にビニールテープを巻き、太い方の内周にちょうど合う太さにしてから、細い方を太い方の内側に挿入する方法を採った。表1中の長さの括弧内の数字は、重なり部分の長さ2cmを除いた長さである。

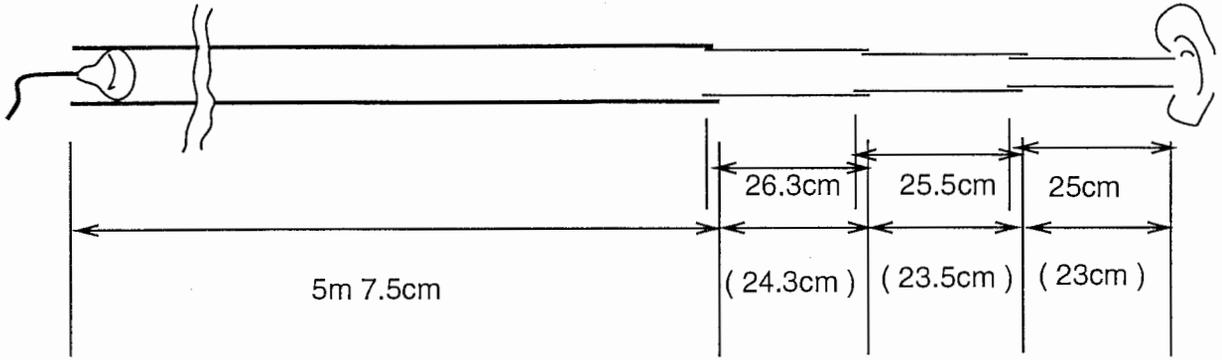


図 2: 聴覚刺激伝達系

表 1: ホースとビニールチューブの寸法

	外側の直径 [mm]	内側の直径 [mm]	長さ [mm]
ホース	20	15	5075
チューブ太	14	12	26.3 (24.3)
チューブ中	10.5	8.5	25.5 (23.5)
チューブ細	7.5	5.5	25 (23)

3.2 聴覚刺激伝達系の伝達特性

聴覚刺激再生系から、上記のように作成した音響管を経由して耳に呈示されるまでの系全体としての、音響的な伝達特性を測定した。

3.2.1 方法

測定に使用した機器は以下のとおり。

1. 刺激音再生装置：ソニー・マグネスケール製データレコーダ、PC-216A
2. 増幅器：オーディオ・テクニカ、AT-MA500
3. イヤホン：ソニー、MDR-E838
4. マイクロホン：B & K のダミーヘッドとそれに内装された同社のマイクロホン

測定は ATR 内の無響室で行った。

音源としては、TSP(Time Stretched Pulse, 時間引き伸ばしパルス)を用いた [6]。ここでは 8192 点の TSP 信号を、16384 点 (8192 × 2) の間隔で生成し、ダミーヘッド内のマイクで採取し、16384 点の FFT から伝達特性を求めた。なお、信号音の再生に用いた標本化周波数は 48kHz である。

3.2.2 結果

図 3に、各周波数の音圧の伝達特性を示す。図 3をみると明らかに、200Hz 以上では約 10dB/oct の減衰がみられる。

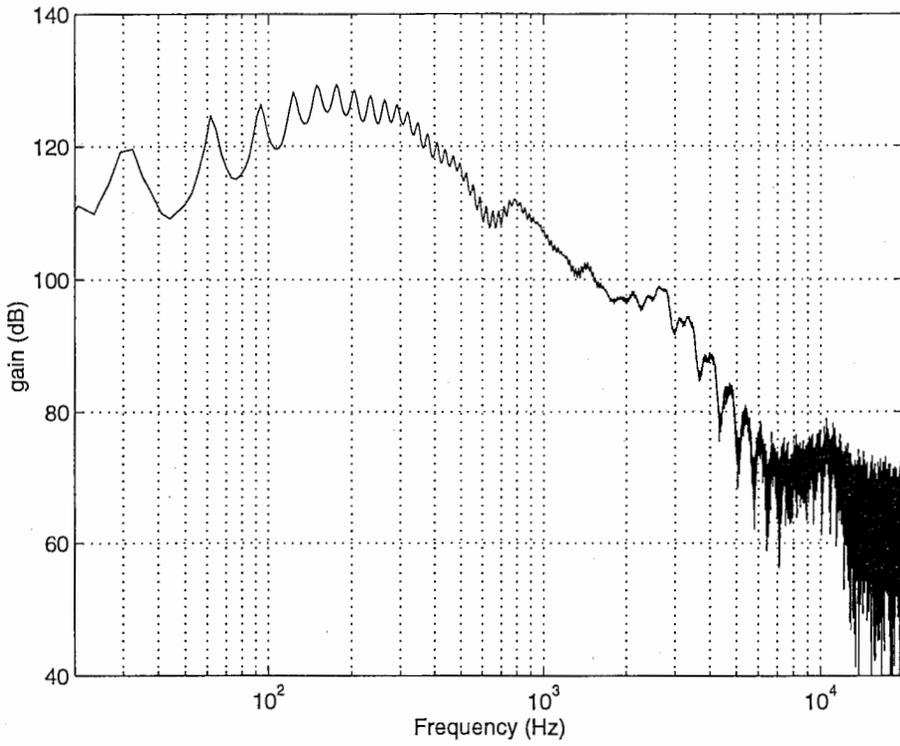


図 3: 音圧の伝達特性

また、音響管内の音の伝搬にかかる時間が無視できないので、遅れの特性として測定した。図 4に、遅れの特性を示す。図 4より、周波数 100Hz から 10000Hz の間では、5ms から 32ms の範囲でばらつきがあるが、800Hz から 10000Hz に限定すれば、約 17ms の遅れであることがわかる。そこで、伝達系の遅れとして、17ms を用いることにする。

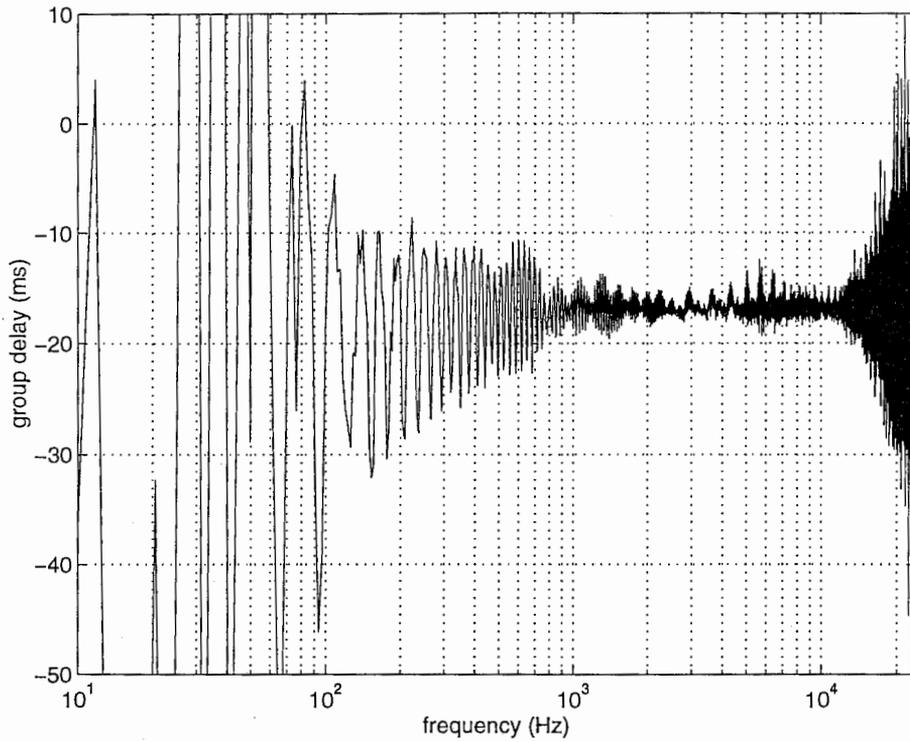


図 4: 遅れの特徴

3.3 伝達特性の補正

伝達特性を測定した結果（図 3）、高域で落ち込みがあることがわかったので、入力聴覚刺激の高域をあらかじめ強調しておく必要がある。ここでは高域強調のために、微分（48kHz 標本化信号の差分）を用い高域強調を実現することにした。

図 5 に、TSP 音源の微分波形を入力としたときの、原音からみた伝達特性を示す。前節と同じ測定方法を用いて測定した結果を、図 3 と比較すると、高域での減衰は約 4dB / oct に改善されていることがわかる。

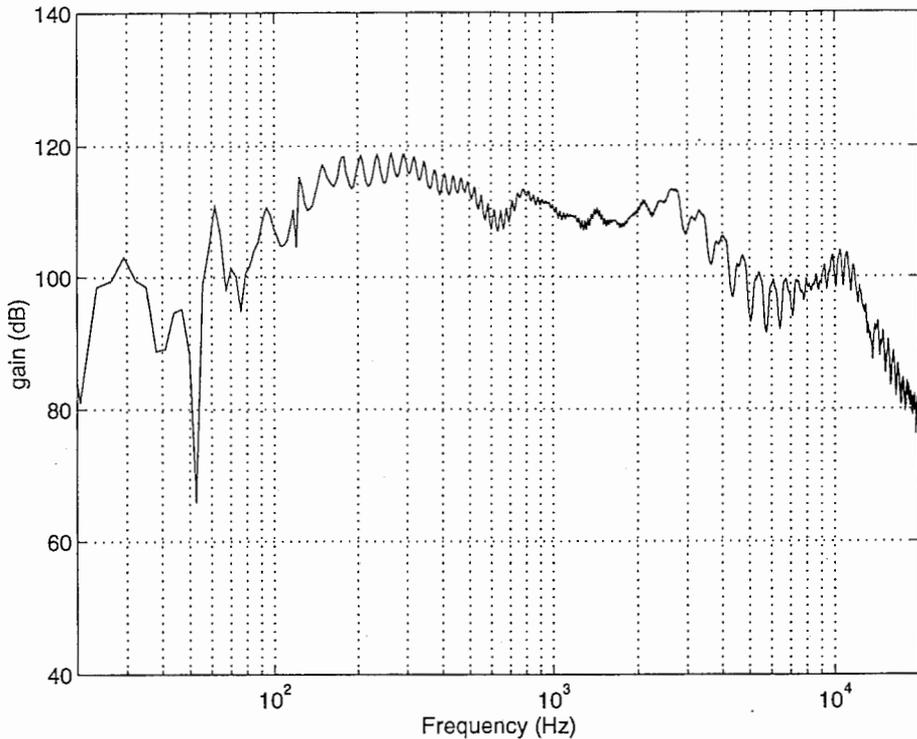


図 5: 音源を微分した波形を入力としたときの、原音からみた伝達特性

4 聴覚刺激生成系

刺激音の生成には、ソニー・マグネスケールのデータレコーダ (PC-216A) を用いた。このデータレコーダは 2 チャンネルで使用した場合、標本化周波数 48kHz、16bit の精度で信号を記録・再生できる。ただし、16bit のうち LSB の 1bit は、A/D, D/A 変換の対象とならず、アナログ系とは別の TTL レベルの 1bit 信号の入出力に使用される。したがって、聴覚刺激等の入出力には残りの 15bit が使用される。詳細はセクション 5.2 を参照。

5 聴覚刺激用テープの作成

5.1 音響管の特性を補正するための聴覚刺激の素材の加工

聴覚刺激の素材として用いられるのは、コンピュータ内で発生できる信号か、デジタル・サンプリングされた音声信号である。これらの聴覚刺激の素材は、前章で得られた音響管の特性を考慮して、前もって、微分 (差分) 処理を施すことにした。この処理は、たとえば MATLAB 等を使用してコンピュータ内で行う。処理後のデータは 16bit (Short Integer) の形式とする。

5.2 MEG 装置へのトリガの生成のための聴覚刺激の素材の加工

MEG 装置へデータ取り込みタイミングを与えるためのトリガパルスを与える必要がある。これを、データレコーダ (PC-216A) の LSB を用いた TTL レベルの信号出力機能を利用して実現した。

このデータレコーダでは、TTL レベルの信号出力は 16bit データの LSB が 1 (16bit のデータとしては奇数) の時に 0 となり、LSB が 0 (16bit のデータとしては偶数) の時に 1 となる。図 6 に、LSB の値と TTL レベルの出力電圧との関係を模式的に示す。図では、100ms-126ms の区間で TTL

出力が5 Vで、他の区間では0 Vとなる場合を示している。この、LSBにより TTL 信号出力機能を利用して、聴覚刺激信号に同期した TTL レベルの信号を出力することが可能である。

そのための聴覚刺激信号の加工はコンピュータ内で行うことができる。すなわち、コンピュータ内に16bit(Short Integer)の形式で格納されている聴覚刺激信号のLSBを、MEGの要求するトリガパルスの条件を満足するように操作し、コンピュータ内に再格納する。コンピュータ内に格納されている聴覚刺激信号のLSBを操作するプログラムの例をAppendix A「LSBによるトリガー・パルス生成のための聴覚刺激加工用プログラムの例」に掲載した。

なお、ここで加工したLSBの聴覚刺激信号への影響はない。なぜなら、前述のように、ここで聴覚刺激信号の再生に用いられるソニー・マグネスケールのデータレコーダ(PC-216A)では、LSB以外の15bitだけがD/A変換に用いられるためである。

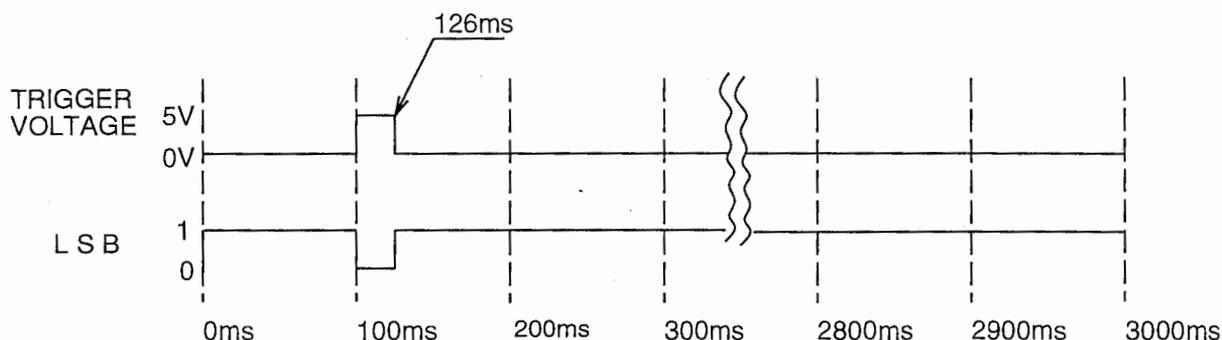


図 6: LSB 信号を用いたパルス生成

5.3 聴覚刺激の DAT テープへの録音

以上の手順で加工された聴覚刺激の素材を、実験に必要な回数だけ連結したものをコンピュータ上に作成する。このとき、使用する聴覚刺激の種類、回数、並び方などは実験目的により変更する必要がある。このように作成された「聴覚刺激素材の連結ファイル」をコンピュータとデジタル・オーディオ・テープレコーダ (DAT) とのインタフェース (例えば、DATLINK) を使用して、DAT のテープに記録する。

実験の際、この「聴覚刺激テープ」を、ソニー・マグネスケールのデータレコーダ (PC-216A) で再生する。

6 まとめ

MEG 装置を用いた実験のための、聴覚刺激信号呈示系を試作した。その際、磁気シールド室内での磁気発生を避けるために音響管による信号伝達方式を採った。次に、この音響管の音響特性を補整するために、聴覚刺激信号自体に高域強調処理を施すことを提案した。さらに、MEG 装置にデータ収集のタイミングを知らせるための、トリガーパルスを発生する機能を、聴覚刺激信号の再生に用いるデータレコーダの TTL 信号出力機能により実現した。

ここでは、上記の項目について原理を述べるとともに、実際に実験に使用される「聴覚刺激テープ」の作成方法について述べた。なお、実際に、この聴覚刺激信号呈示系と「聴覚刺激テープ」を用いて行った、正弦波と母音 /a/ に対する脳磁界測定の前備実験については [7] を参照されたい。

謝辞

今回の実験をすすめるにあたり、協力、助言をいただいた、ATR 人間情報通信研究所、第4研究室の草川 直樹氏、党 建武氏、大西 通博氏、第1研究室の河原 英紀氏、相川 清明氏、山田 玲子氏、住友金属ハイコリティライフ研究所の林 昇氏に感謝いたします。また、本報告書の第一著者、小幡 靖に本実験の実習の機会を与えてくださった、奈良先端科学技術大学院大学の渡辺 勝正先生、ATR 人間情報通信研究所の東倉洋一社長に感謝いたします。

参考文献

- [1] 南部 篤: 脳磁場計測 (MEG) によるヒトの感覚運動機構の解析, 神経進歩, Vol.38, No.2, pp.225-237 (1994.4)
- [2] 中里 信和・吉本 高志: 磁気脳波 (MEG) の臨床応用, 神経進歩, Vol.38, No.2, pp.238-246 (1994.4)
- [3] 中里 信和・吉本 高志: SQUID 脳磁計の臨床応用, 電子情報通信学会誌, Vol.78, No.2, pp.258-263 (1995.3)
- [4] 今田 俊明・川勝 真喜・益子 拓徳・小谷 誠: 聴覚誘発脳磁界反応と刺激呈示感覚, 医用電子と生体工学, Vol.33, No.1, pp.15-23 (1995.3)
- [5] 今田 俊明: 非侵襲的生体観測技術 - 損傷せずに健常な人の機能を観測する方法 -, 電子情報通信学会誌, Vol.78, No.2, pp.309-312 (1995.3)
- [6] 鈴木 陽一・浅野 太・曾根 敏夫: 音響系の伝達関数の模擬をめぐって (その 2), 日本音響学会誌, Vol.45, No.1, pp.44-50 (1989)
- [7] 小幡 靖・正木 信夫: 正弦波と母音 /a/ に対する脳内活動の比較 - 磁気脳波計測装置を用いた予備実験 -, ATR テクニカルレポート (1995)

A LSBによるトリガー・パルス生成のための聴覚刺激加工用プログラムの例

以下は、MEG装置へのトリガーパルスを生成するために、聴覚刺激の素材のファイルのLSBを加工するためのプログラムのソース・リストである。

聴覚刺激の素材のファイルは16bitで48kHzでサンプリングされたものとする。素材ファイルの長さは問わない。

パルスは100ms時点から26msの間は1 (TTLレベルで5V)であり、その他の区間では0 (TTLレベルで0V)となるように、それぞれの区間のLSBを0と1に書き換えている。

使い方は、「gcc odd_even_odd.c -o o_e_o」とコンパイルとした場合、「o_e_o ファイル名1 ファイル名2」とする。ただし、ファイル名1がもとのファイル名、ファイル名2が変換したあとのファイル名である。

```
#include <stdio.h>
#include <string.h>

#define MAX_AMP    32767 /* max amplitude */
#define MIN_AMP   -32768 /* min amplitude */
/**** 48kHz のとき、一秒分のデータは、48000 個の点で表現される。 *****/
#define NUM_OF_POINT_OF_FIRST_ODD_PART 4800 /*** 100ms / 48kHz ***/
#define NUM_OF_POINT_OF_EVEN_PART      1248 /*** 26ms / 48kHz ***/

#define MAX_ARRAY    256

main(int argc, char *argv[])
{
    FILE *input_fp, *output_fp;
    int  c1, c2, flag, count;
    short int amplitude;
    char  source_filename[MAX_ARRAY] = "";
    char  target_filename[MAX_ARRAY] = "";

    strcat(source_filename, argv[1]); strcat(target_filename, argv[2]);
    printf(" source_filename = %s\n", source_filename);
    printf(" target_filename = %s\n", target_filename);
    input_fp = fopen(source_filename, "r");
    output_fp = fopen(target_filename, "w");

    flag = 0; count = 0;
    while( flag != 1)
    {
        if ((c1 = fgetc(input_fp)) != EOF)
        {
            count++;
            if ((c2 = fgetc(input_fp)) != EOF)
            {
                amplitude = c1 * 256 + c2;
            }
        }
    }
}
```

```

if ( ( count <= NUM_OF_POINT_OF_FIRST_ODD_PART )
    && ( amplitude % 2 == 0 ) )
{/** 0ms から 100ms の偶数に 1 を足す。 ***/
    if( amplitude < MAX_AMP )
        amplitude++;
    else /** 振幅の最大値を越えないように。 ***/
        amplitude--;
}
else
{
    if ( ( count > NUM_OF_POINT_OF_FIRST_ODD_PART
        && count <= NUM_OF_POINT_OF_FIRST_ODD_PART
            + NUM_OF_POINT_OF_EVEN_PART )
        && ( amplitude % 2 == 1 || amplitude % 2 == -1 ) )
    {/** 100ms から 126ms の奇数から 1 を引く。 ***/
        if( amplitude > MIN_AMP )
            amplitude--;
        else /** 振幅の最小値を越えないように。 ***/
            amplitude++;
    }
    else
    {
        if ( ( count > NUM_OF_POINT_OF_FIRST_ODD_PART
            + NUM_OF_POINT_OF_EVEN_PART )
            && ( amplitude % 2 == 0 ) )
        {/** 126ms からあとの偶数に 1 を足す。 ***/
            if( amplitude < MAX_AMP )
                amplitude++;
            else /** 振幅の最大値を越えないように。 ***/
                amplitude--;
        }
    }
}
c2 = amplitude % 256; c1 = ( amplitude - c2 ) / 256;
fputc(c1, output_fp); fputc(c2, output_fp);
}/** if ( c2 ) ***/
else
    flag = 1;
}/** if ( c1 ) ***/
else
    flag = 1;
}/** while ( flag ) ***/
fclose(input_fp); fclose(output_fp);
printf(" count = %d\n", count);
}

```