管球音響研究所

水神谷研究室

PCM1795　USB__D/A コンバータ

報告書番号：VRR001

作成２０２３年1月３１日

変更来歴

関連報告書

VRR ００２

２０２２年現在、秋葉原で入手できる部品で、PCM1795 を使用した D/A コンバータを試作します。USB 入力には COMBO384(これは通販で購入)を使用し、S/PDIF 入力は装備しません。
PCM1795,COMBO384 共に PCM/DSD に対応していますので両方再生可能としました。だたし PCM1795,COMBO384 間の接続は PCM と DSD 異なりますのでゲート IC の切り替え回路を追加しました。

COMBO384 はバスパワーで動作するので、”Isolator”IC を使用したいところですがここでのジッタ発生を無くすため抵抗のみで接続しています。USB 接続有り、５Ｖ電源 OFF の状態でマイコン電源３．３Ｖがリセット電圧超えないように３．３Ｖに１００Ωの抵抗を付けています。

PCM1795 を DSD モードにするには内部レジスタをリセット時から変更する必要があるのでシリアルコントロールに PIC16F648 を使用します。

電源は５Ｖ入力とし、ここから LC フィルタを通してアナログ５Ｖ、ドロップ型 IC で３．３Ｖ、さらに DC-DC コンバーターモジュールで OP アンプ用±１２Ｖを作ります。

USB 入力からはデエンファシス有無の情報が得られないので、デエンファシス有ディスクの対応はあきらめます。

PIC マイコンの書き込み回路を付けていますが、消去に失敗することがありますので、USB 接続タイプの ROM ライターで消去、書き込みを行いました。

基板は２．５４ｍｍピッチ、１００ｍｍ×１００ｍｍのユニバーサル基板を使用します。この為電解コンデンサーの足ピッチは２．５ｍｍに限定されます。DC-DC コンバーターの出力は容量制限があるので４７μ、それ以外は２２０μ又は４７μの２品種で設計しました。

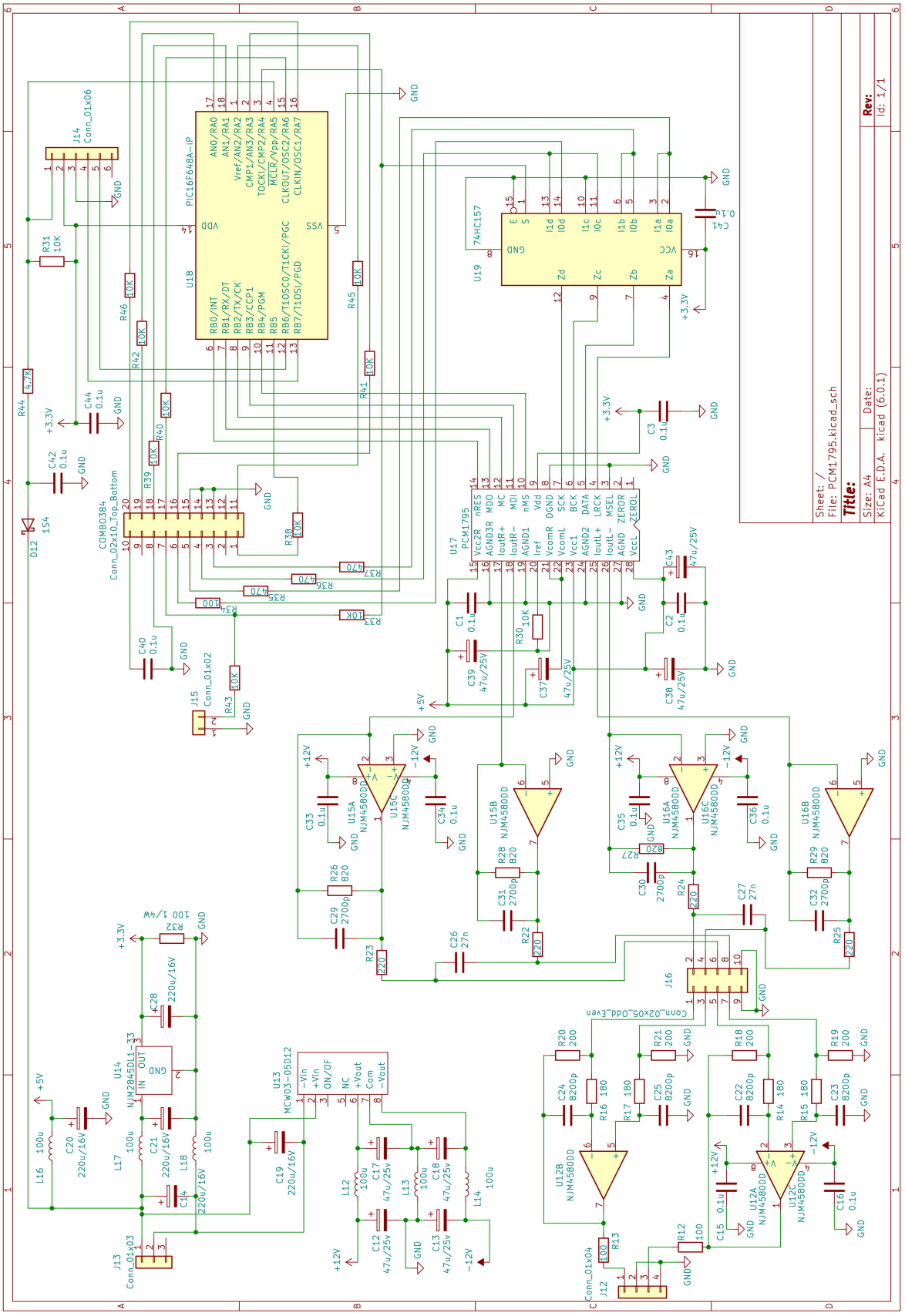
DAC 出力の正相、逆相の合成に OP アンプとトランスの両方に対応するため、R２２－２５が交換可能なよう IC 用の丸ピンソケットを使用し J16 を追加して OP アンプ回路を切断できるようにしました。本機が OP アンプで合成ですので R２２－２５に２２０Ω、J16 はジャンパーピンで接続しています。

結果：

PCM/DSD 両方再生可能な USB 入力 DAC が作成できました。

方形波再生はリップルが目立ちますが、これで普通の再生波形です。たいていの CD プレーヤはこのような応答をします。方形波の WAV データを CD-R に焼いて再生すれば観測できます。

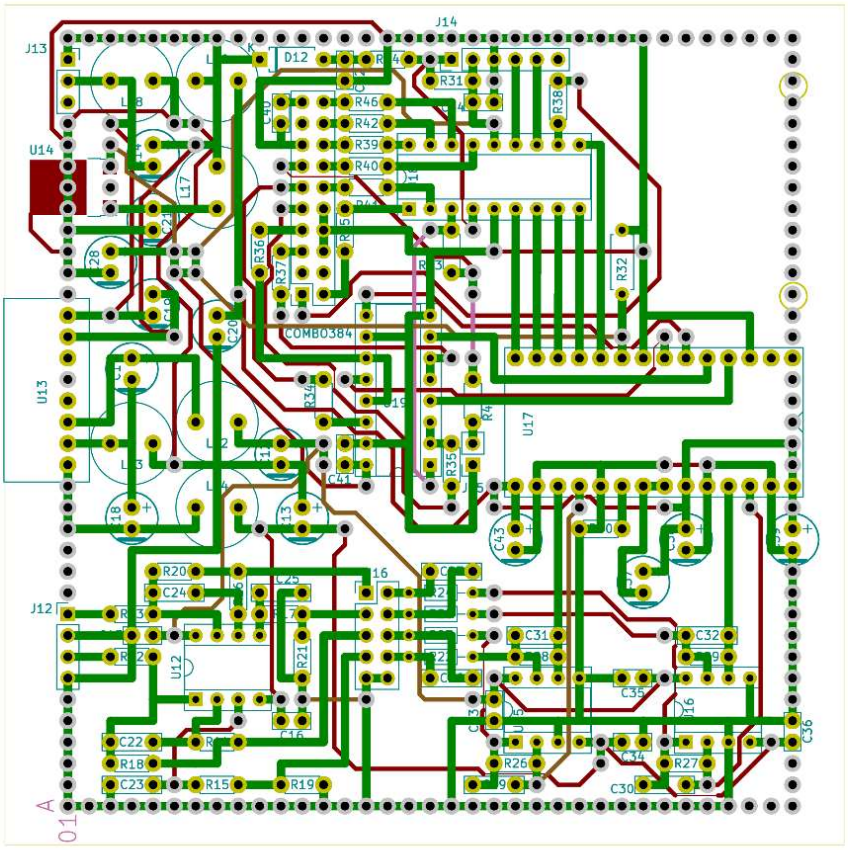
回路図：


PIC16F648A-IP
74HC157
PCM1795
NJM4580DD
MCW03-05D12
NJM2845DL1-33
COMBO384
Conn_02x10_Top_Bottom
Conn_02x05_Odd_Even
Sheet: /
File: PCM1795.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (6.0.1)
Id: 1/1

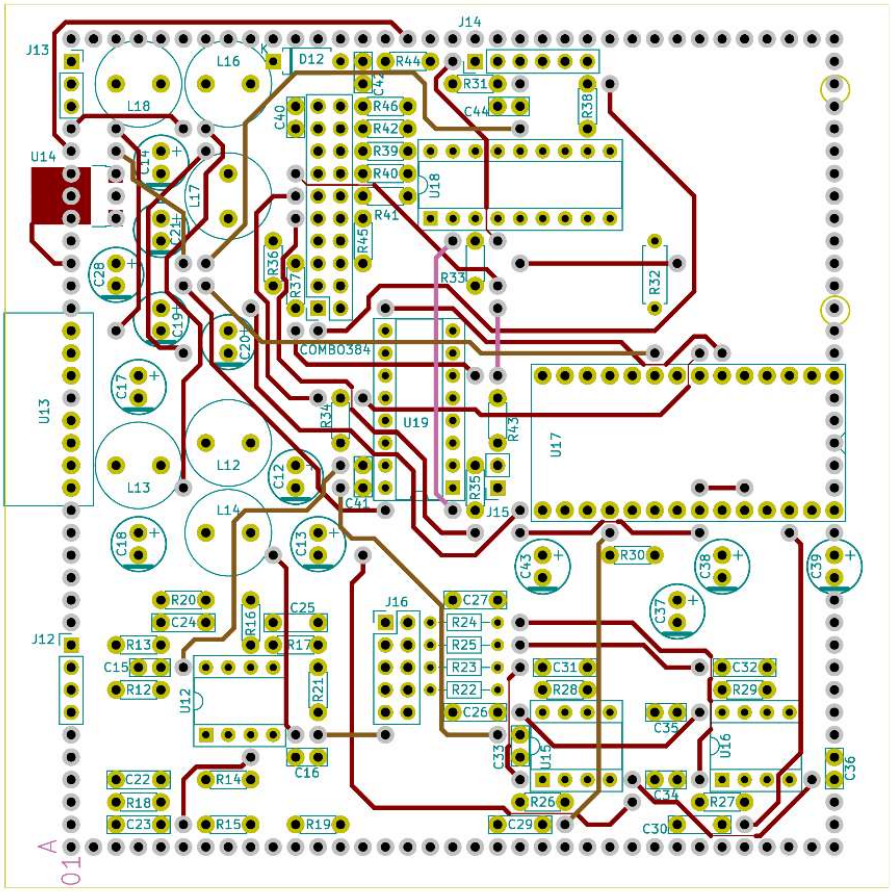
主要部品表

回路番号	品名	備考	
ユニバーサル基板	10cm*10cm	P-13160	
COMBO384 ピンヘッダ	PH-2x40SG	C-00082	
U13	MCW03-05D12	M-04265	
U17	PCM1795	I-10638	
U17 変換基板	AE-VSOP-28	P-05567	
U17 ピンヘッダ	PH-1x40SG	C-00167	
U17 ソケット	FHU-1x42SG	C-05779	
U18	PIC16F648A	I-00466	
C14	220μ/16V	P-10326	
C12	47μ/25V	P-13122	
ケース	MX3-11-12BB	タカチ	
５Ｖ電源アダプター			

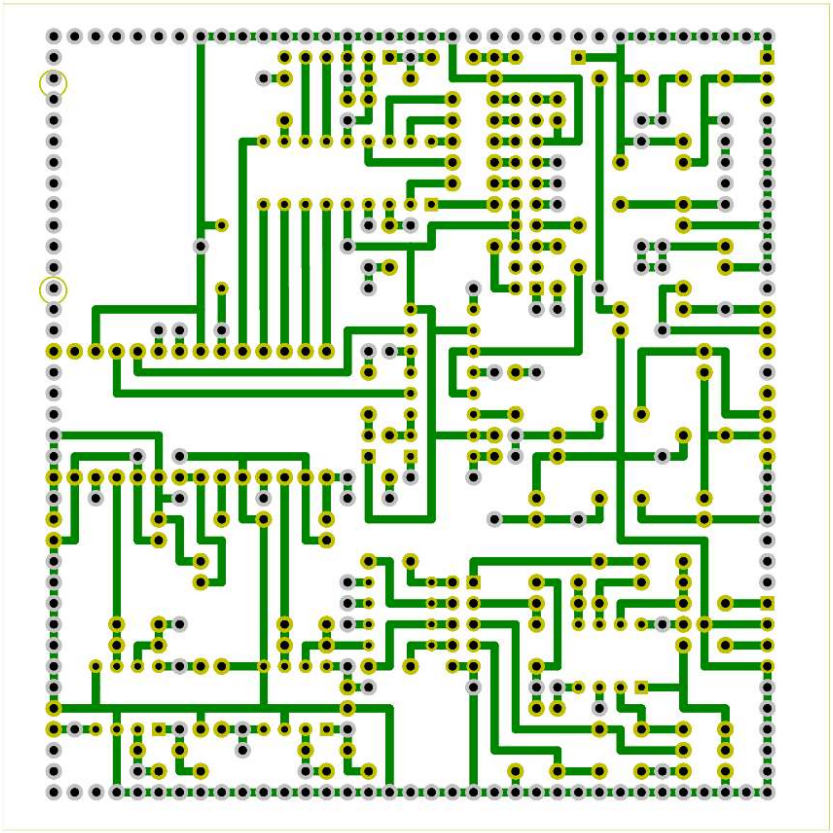
表及び裏透視パターン図（部品面、表視）



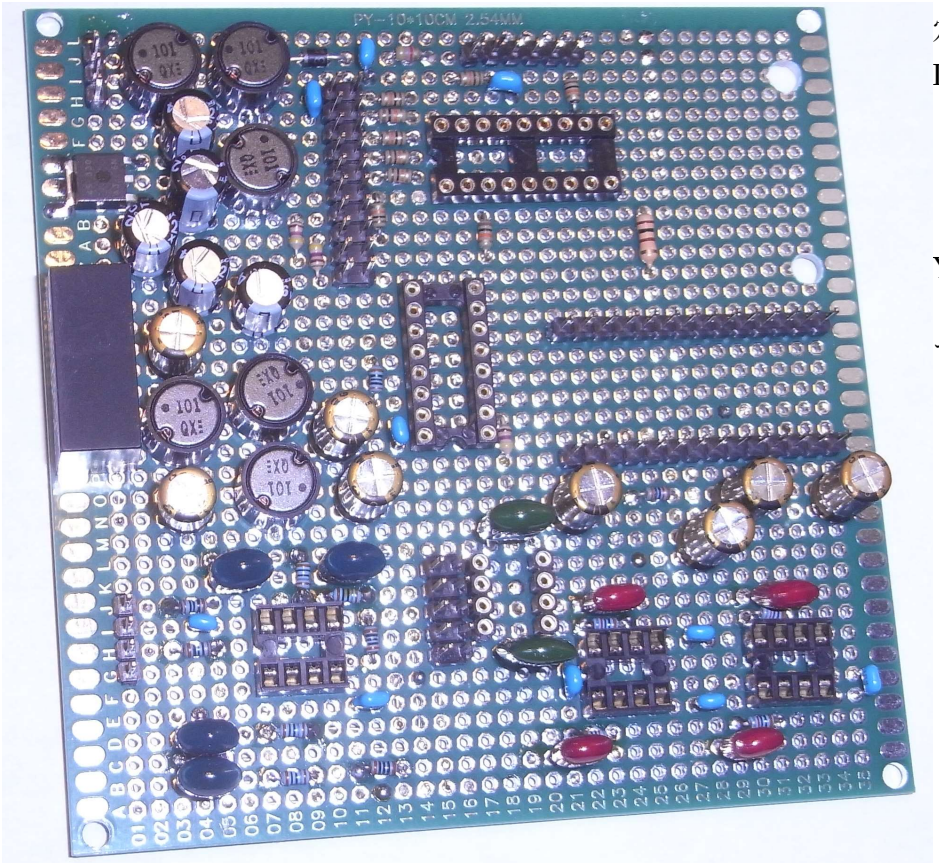
パターン図（部品面）表視



パターン図（半田面）裏視



部品配置写真



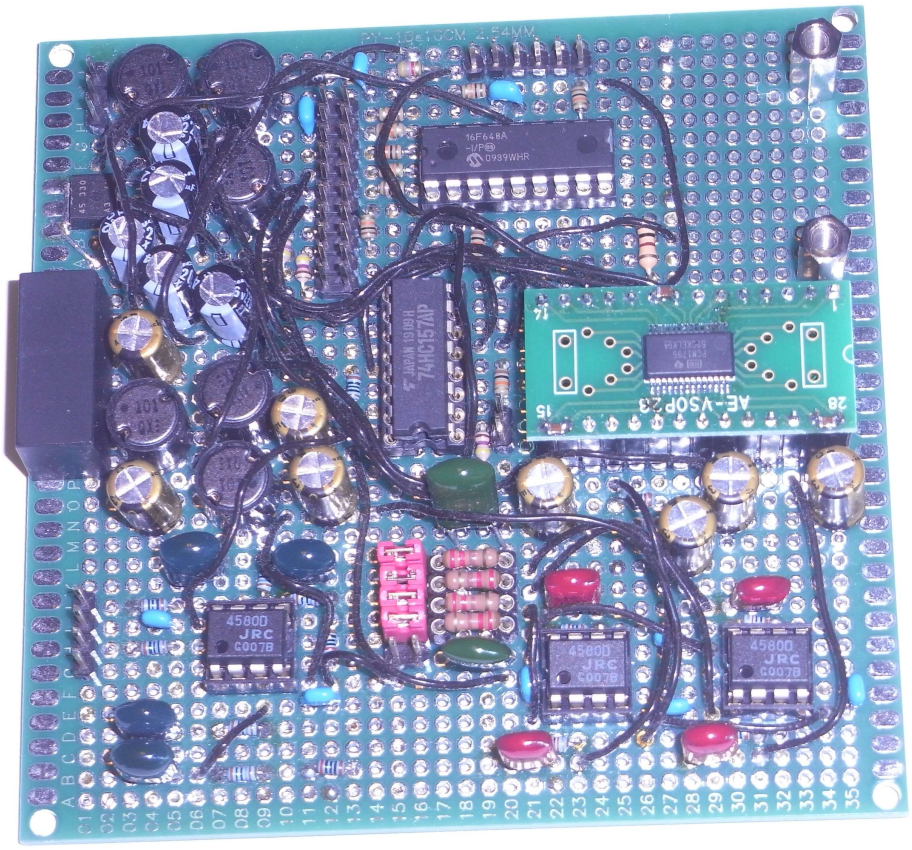
穴の座標

I-35:　4.5φ

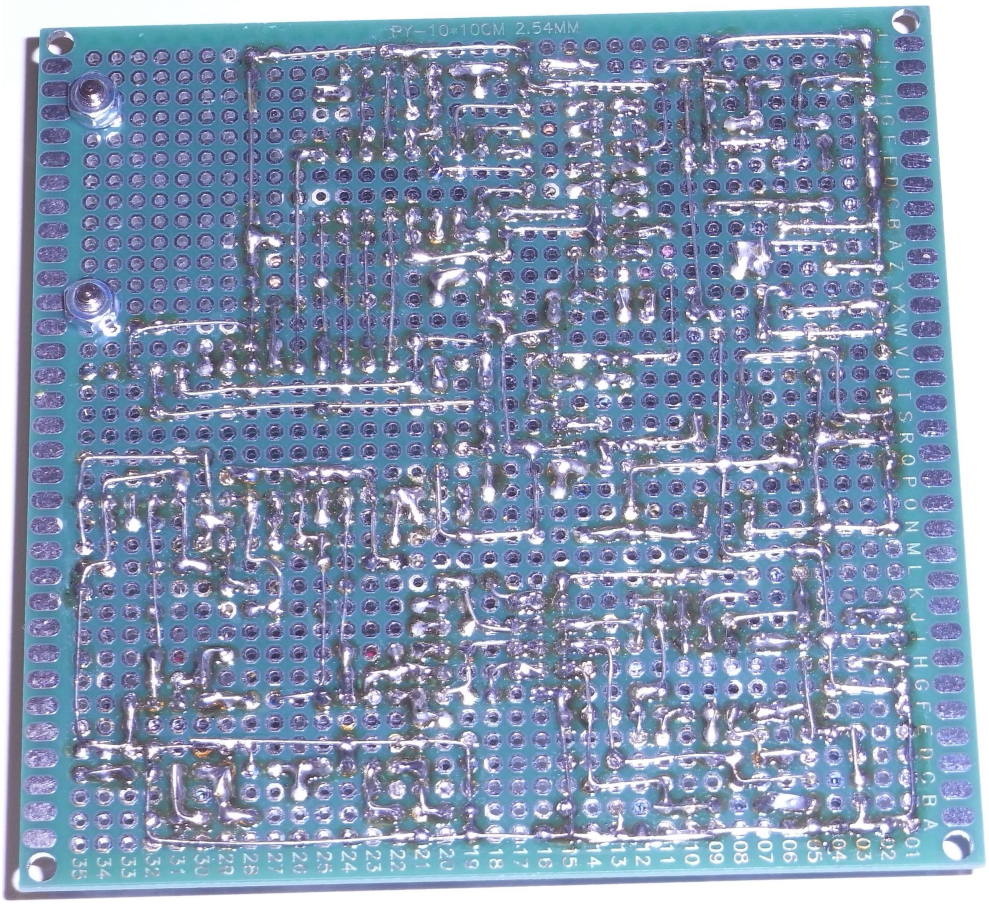
Y-35:　3.8φ

で穴を拡張しました。

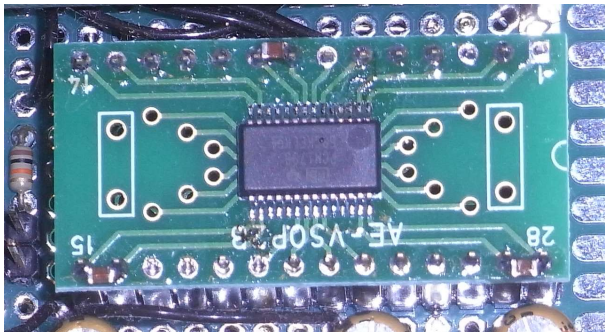
部品面配線後



半田面配線後

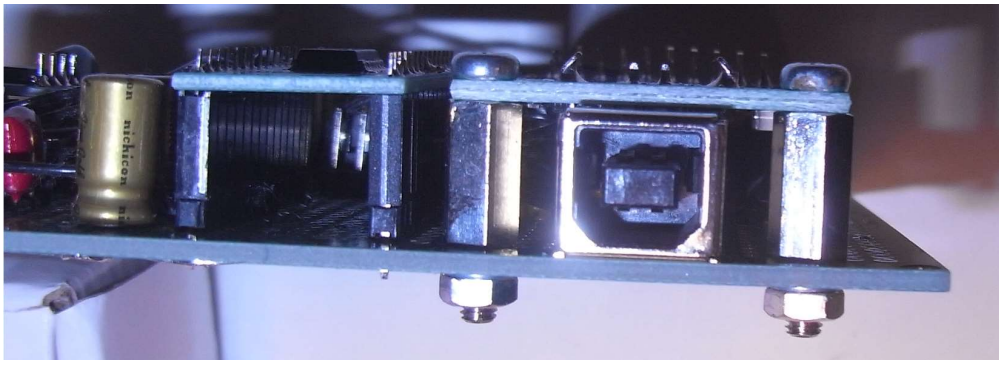


PCM1795 と変換基板



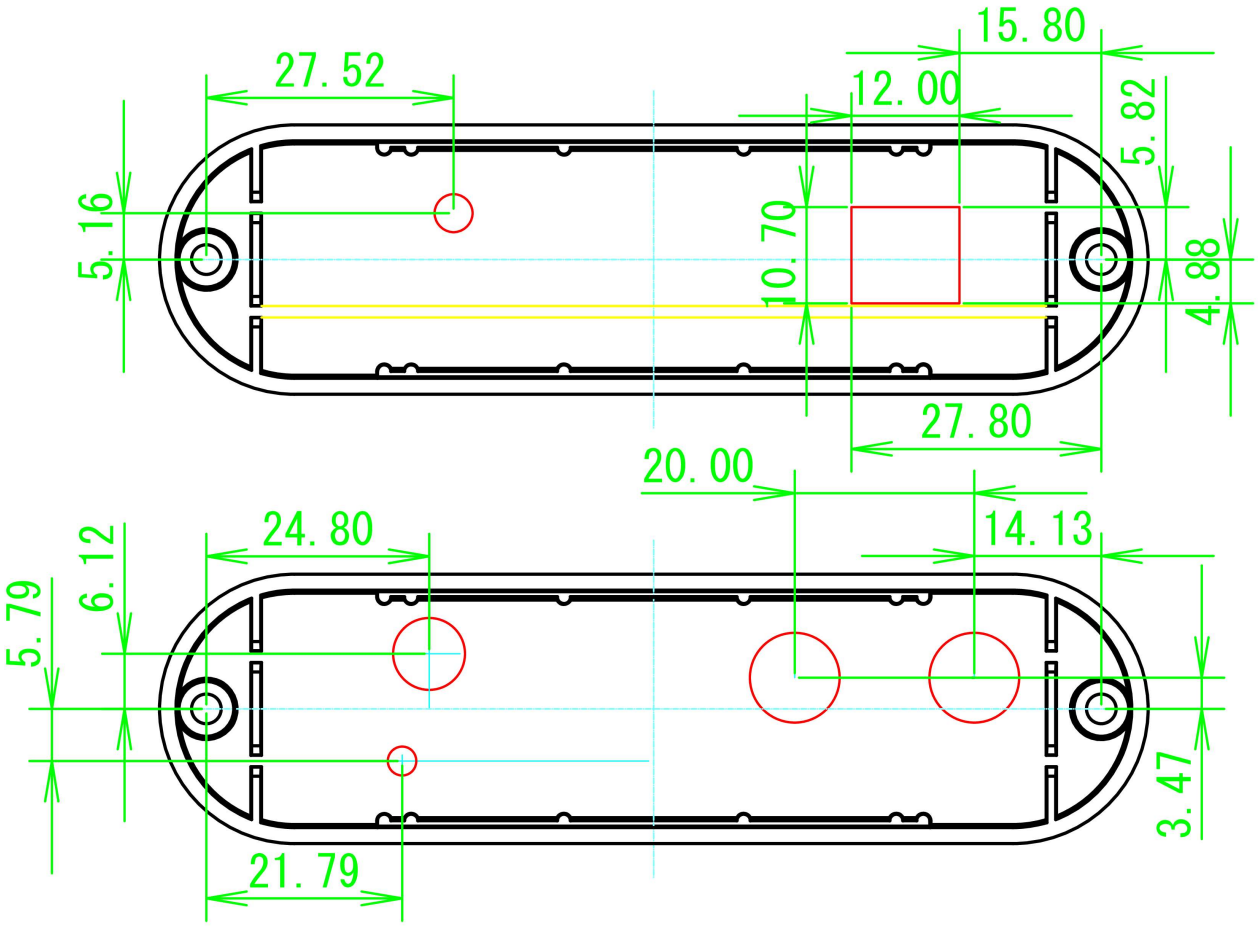
C1：１５－１６
C2：２７－２８
C3:　９－８
パスコン３個は変換基板上に配置

COMBO384 と本体基板との固定

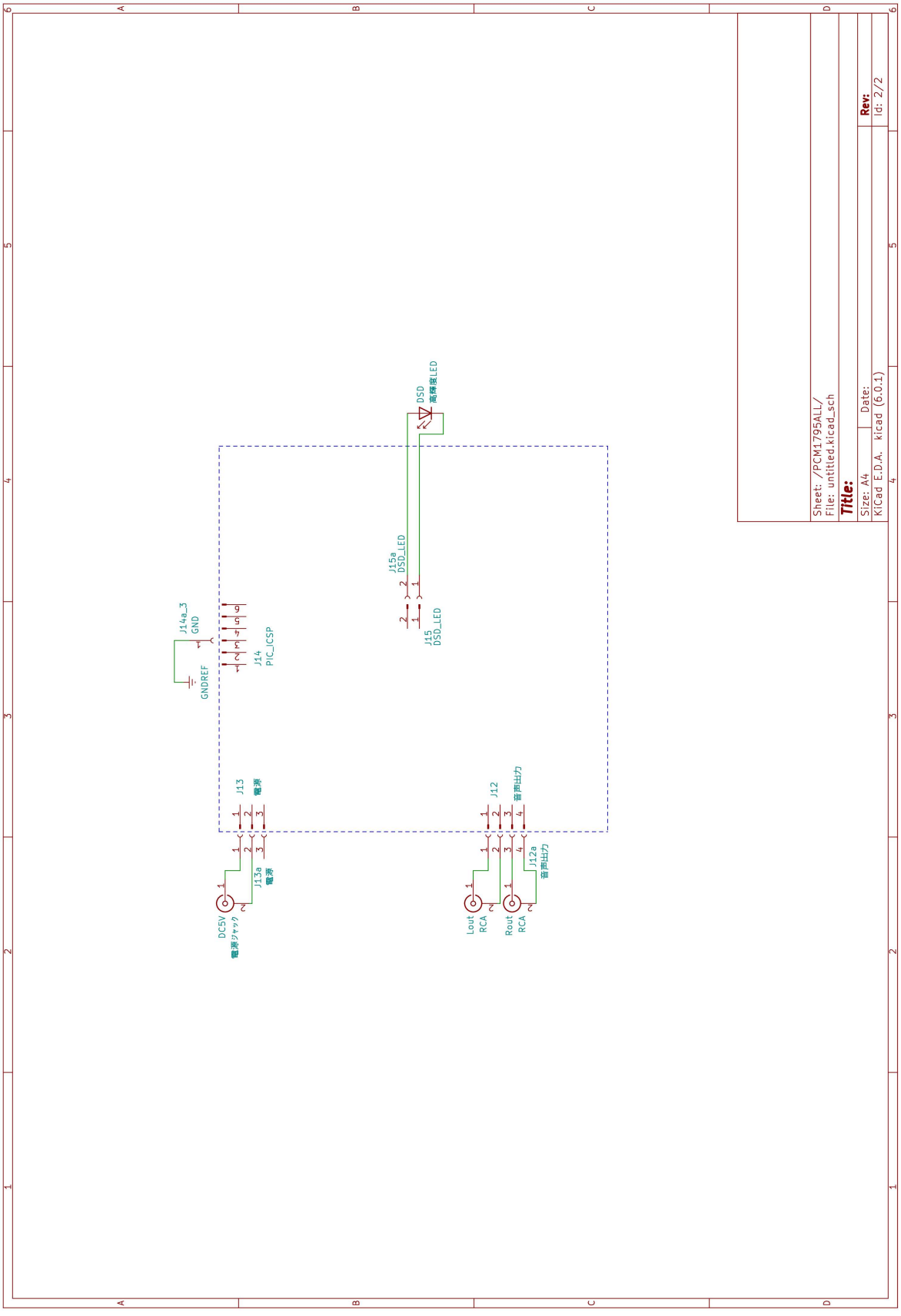


L=１１～１２ｍｍの６角スペーサーで現物合わせで固定しました。

ケース外形図

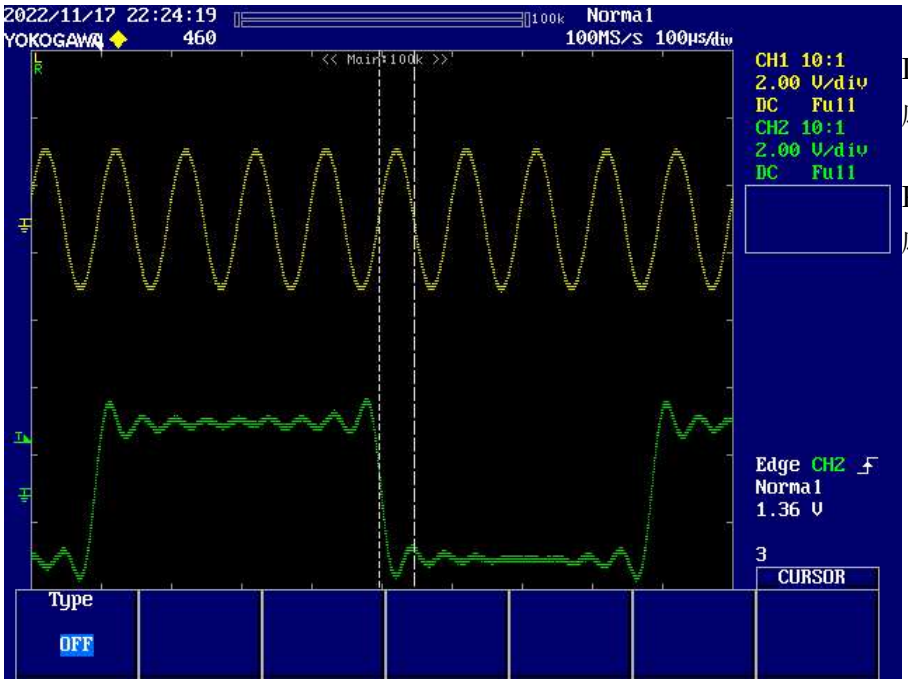

15.80
27.52
12.00
5.82
5.16
10.70
4.88
27.80
20.00
24.80
14.13
6.12
5.79
3.47
21.79

基板外部接続図


J14a_3
GND
GNDREF
J14
PIC_ICSP
J15a
DSD_LED
J15
DSD_LED
DSD
高輝度LED
J13
電源
J13a
電源
DC5V
電源ジャック
J12
音声出力
J12a
音声出力
Lout
RCA
Rout
RCA
Sheet: /PCM1795ALL/
File: untitled.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (6.0.1)
Id: 2/2

ソフトウェアー発信器で１０KHz正弦波（左）、１．２５KHz方形波（右）-3dbを発生して
Combo384 ASIO 1.03ドライバ（デバイス）でUSB出力した波形を以下にしめします。

（サンプリング周波数４４．１KHｚ）

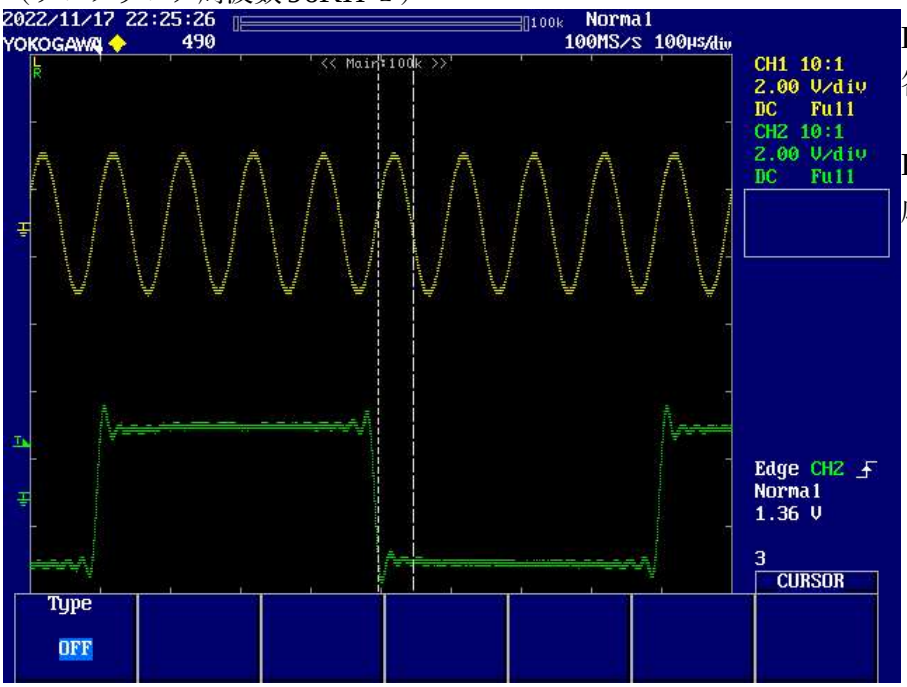


L（黄）１０K、-3db正弦波応答波形

R（緑）1.25KHz,-3db方形波応答波形

PCM1795は内部で8倍オーバーサンプリングするので１０KHｚの正弦波の再生は良好ですが、
１．２５KHｚの方形波はデジタルフィルタの影響でプリエコーとポストエコーが生じています。

（サンプリング周波数96KHｚ）

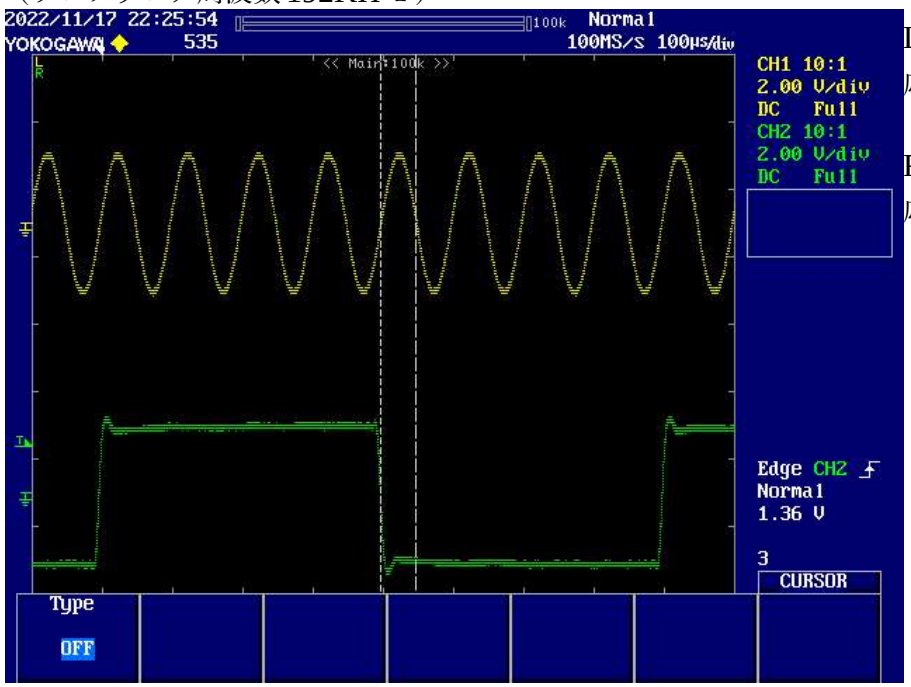


L（黄）10K、-3db正弦波応答波形

R（緑）1.25KHz,-3db方形波応答波形

サンプリング周波数96KHzになったので、方形波内のリエコーとポストエコー周波数も倍になった(44.1KHz比)。振幅が小さくなっているのはアナログフィルターの減衰領域にプリエコーとポストエコーの周波数があるからと思われます。

（サンプリング周波数192KH ｚ）

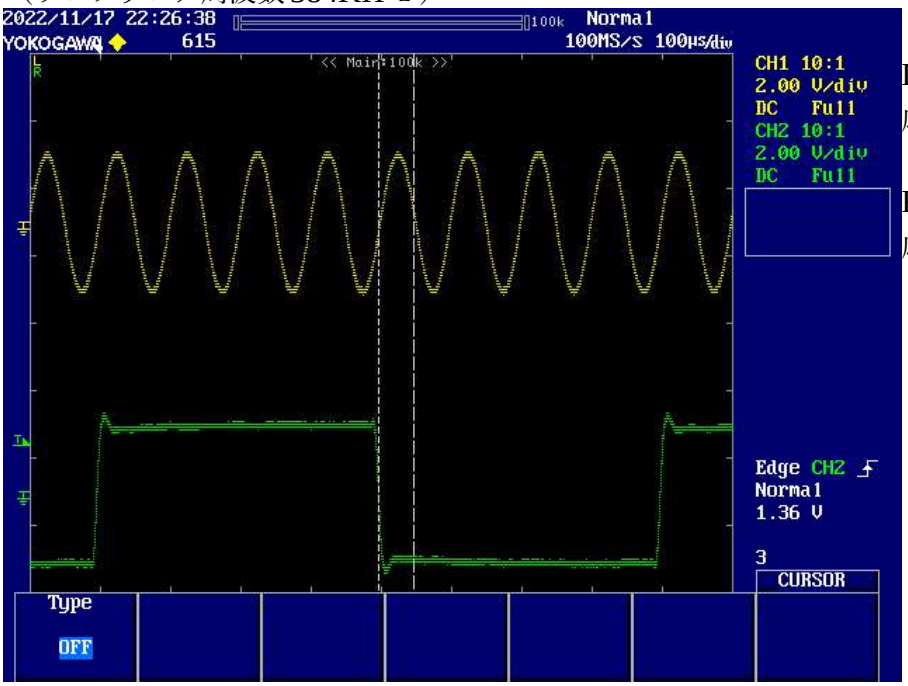


L（黄）１０K、-3db 正弦波応答波形

R（緑）1.25KHz,-3db 方形波応答波形

サンプリング周波数がさらに高くなったので、方形波内のプリエコーとポストエコーは殆ど見えなくなった。オーバーシュートとアンダーシュートが発生しているがアナログフィルターの影響と思われる。

（サンプリング周波数384KH ｚ）

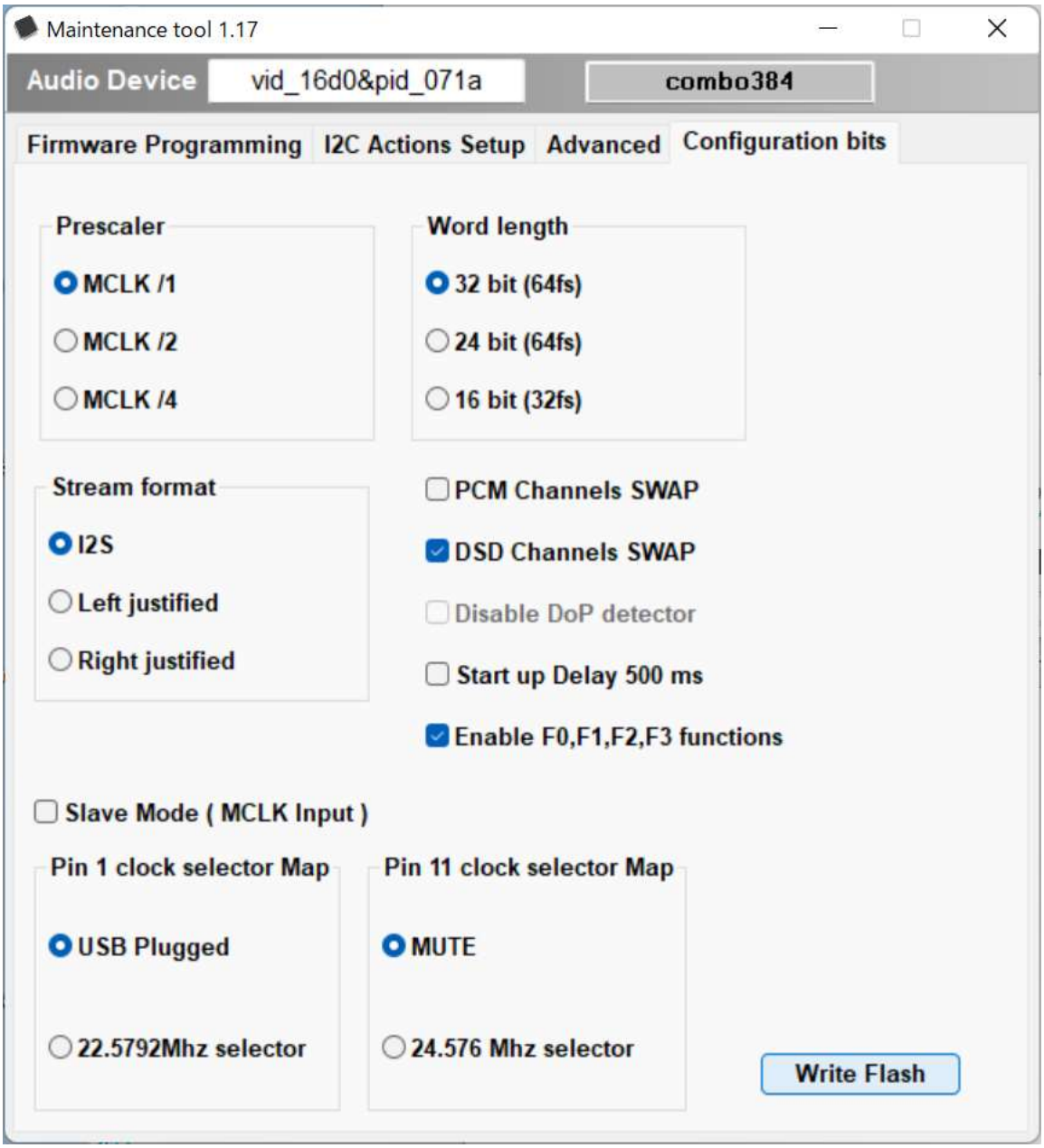


L（黄）１０K、-3db 正弦波応答波形

R（緑）1.25KHz,-3db 方形波応答波形

１９６KHz と比べて大きな変化は認められない。

COMBO384 の設定：oemtool117 デレクトリー内の「ConfigTool.exe」で設定



Configuration Bits のタブをクリックして上記の通り設定しました。

マイコンプログラム　MPLAB　X　IDE　ｖ５．５０

```
/*
 * File:   main.c
 * Author: mizukamidani
 * Created on 2022/06/13
 * MPLAB X IDE v5.50
 */
#include <stdio.h>
#include <stdlib.h>
#include <xc.h>
// CONFIG
#pragma config FOSC = INTOSCIO  // Oscillator Selection bits (INTOSC oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN)
#pragma config WDTE = OFF       // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = ON       // Power-up Timer Enable bit (PWRT enabled)
#pragma config MCLRE = ON       // RA5/MCLR/VPP Pin Function Select bit (RA5/MCLR/VPP pin function is MCLR)
#pragma config BOREN = OFF      // Brown-out Detect Enable bit (BOD disabled)
#pragma config LVP = OFF        // Low-Voltage Programming Enable bit (RB4/PGM pin has digital I/O function, HV on MCLR must be used for programming)
#pragma config CPD = OFF        // Data EE Memory Code Protection bit (Data memory code protection off)
#pragma config CP = OFF         // Flash Program Memory Code Protection bit (Code protection off)
//
#define _XTAL_FREQ 4e6  //Fosc=4[MHz]
#define nRES RB0                //PCM1795 RESET
#define MDO RB1                 //PCM1795 Mode control read-back data output
#define MC RB2                  //PCM1795 Mode control clock input
#define MDI RB3                 //PCM1795 Mode control data input
#define nMS RB4                 //PCM1795 Mode control chip-select input
#define PLUG RB5                //combo384 Cable Plugged = 1
//
#define EMPH RB7                //de-emphasis SW ON =0 ; OFF =1
//
#define MUTE RA2                //combo384 MUTE = 1
#define DSD128 RA3              //combo384 DSD64 = 0 ; DSD128 =1
#define DSD RA4                 //combo384 DSD ON = 1
//
//
char i,ADDR,SPIDATA,SAMPLErate,MODE,WORK;
char SPI(char ADDR,char SPIDATA);
void PCMset(void);
void DSDset(void);
//
//
void PICinit(){
    TRISA = 0b11111111; // RA7-F3,RA6-F0,RA5-MCLR,RA4-DSD,RA3-DSD128,RA2-MUTE,RA1-F1,RA0-F2
    TRISB = 0b11100010; // RB7-EMPHASIS,RB6-NOT_USE,RB5-PLUGGED,RB4-nMS,RB3-MDI,RB2-MC,RB1-MDO,RB0-nRES
    PORTB = 0b00000000; //
    CMCON = 0b00000111; //Comparators Off
    OPTION_REG = 0b00000000 ; //
    __delay_ms(500);
    return;
}

int main(void){
    PICinit();          //PIC I/O
    RB1 = 1;            //PCM1795 nRES=1
    __delay_ms(2);      //RESET after 1024 system clock
    ADDR = 18;          //Register 18
    SPIDATA = 0b01000000;    //32-bit I2S format data, BCK ? x64 fS
    SPIDATA = SPI(ADDR,SPIDATA);
    MODE = 0;
//
    while(1){
        //WORK = PORTA & 0b00001100 ;     //RA2-MUTE,RA3-PLUGGED
        if (MUTE == 1){
            MODE = 0;
            nRES = 0;         //PCM1795 nRES=0
            }
        else{
            nRES = 1;         //PCM1795 nRES=1
            __delay_ms(2);   //RESET after 1024 system clock
            if (DSD == 0){
                if (MODE != 1){ PCMset(); MODE = 1;}
                }
            else{
                if (MODE != 2){ DSDset(); MODE = 2;}
                }
            }
        }
    return 0;
}
//
void PCMset(void){
    ADDR = 20;        //Register 20
    SAMPLErate = PORTA & 0xC3;  //F3,F0,x,x,x,xF1,F2
    switch(SAMPLErate){
        case 0x00: SPIDATA = 0x02;  break;  //32K
        case 0x40: SPIDATA = 0x02;  break;  //44.1K
```

```
        case 0x02: SPIDATA = 0x02;  break;  //48K
        case 0x42: SPIDATA = 0x00;  break;  //88.2K
        case 0x01: SPIDATA = 0x00;  break;  //96K
        case 0x41: SPIDATA = 0x01;  break;  //176.4K
        case 0x03: SPIDATA = 0x01;  break;  //192K
        case 0x43: SPIDATA = 0x01;  break;  //352.8K
        case 0x80: SPIDATA = 0x01;  break;  //384K
    }
    SPIDATA = SPI(ADDR,SPIDATA);
    return;
}
//
void DSDset(void){
    ADDR = 20; SPIDATA = 0b00100000;       //DSD_bit
    SPIDATA = SPI(ADDR,SPIDATA);
    return;
}
//
char SPI(char ADDR,char SPIDATA){
    nMS=0;
    for(i=0;i<8;i++){
        MC = 0;
        if(ADDR >= 0x80)     //bit7 = 1 ?
            MDI = 1;
        else
            MDI = 0;
        ADDR <<= 1;
         MC = 1;
    }
    for(i=0;i<8;i++){
        MC = 0;
       if(SPIDATA >= 0x80)    //bit7 = 1 ?
            MDI = 1;
        else
            MDI = 0;
        //
        SPIDATA <<= 1;
        if(MDO == 1)
            SPIDATA = SPIDATA | 0x01;
        else
            SPIDATA= SPIDATA & 0xFE;
        //
         MC = 1;
    }
    nMS = 1;
    MC = 0;
 return SPIDATA;
}
```

内部

基板は６角スペーサで押し付けて固定しました。