

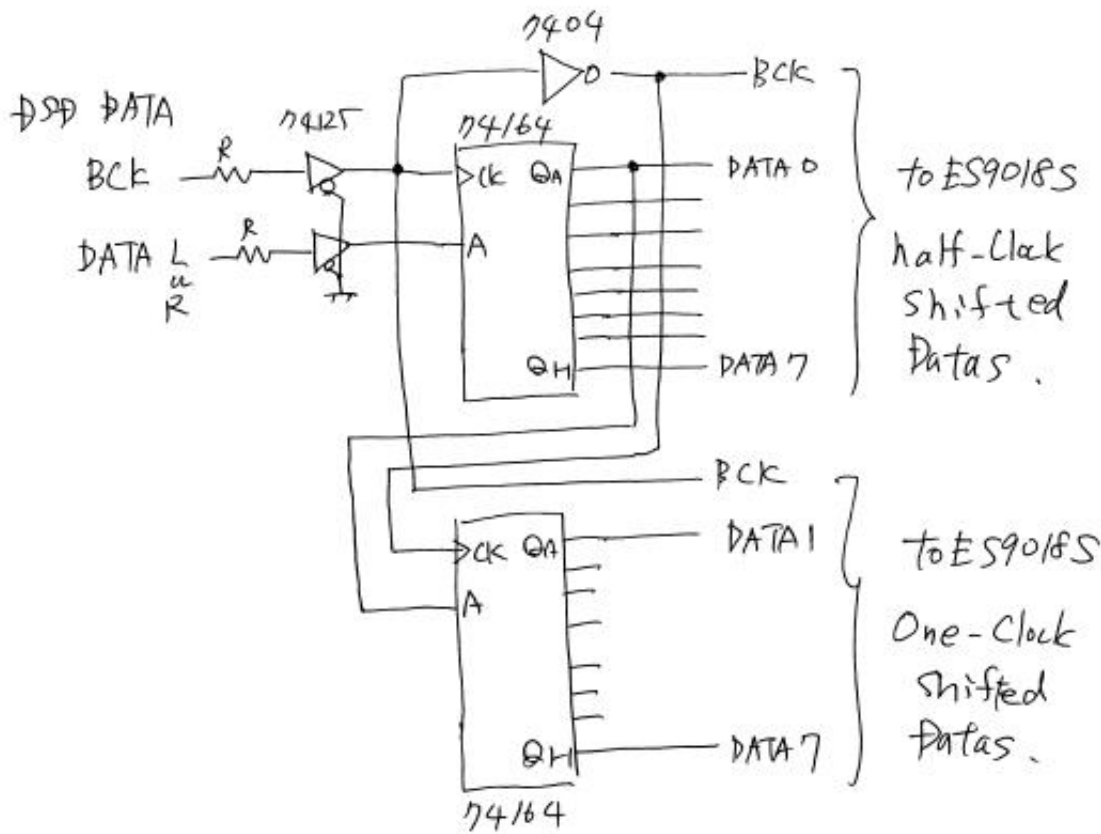
# ES9018SでDSDシフトパラは実現するか？の巻き 2012.2.5

前編は[こちら](#)。

DSDの信号が容易に得られる環境になってきたので、ちょっとやりたいことができました。  
DSD信号は1ビット信号ですから、信号を少しずつシフトして加算することにより、  
フィルターの役目を果たすことができます。さらに、加算処理で素子固有のノイズの平均化を  
計ることもできそうです。  
この構想は最初にES9018を検討したときにPCM信号でやることも考えてみましたが、  
ちょっと大変そうだったので棚上げしてましたが、1ビットのDSD信号ならそれほど難しいことはありません。

0. 5クロック、1. 5クロック……7. 5クロックずらした信号と  
1クロック、2クロック……8クロックずらした信号は  
下記の回路図で作成することができます。

シフトレジスタの74164とインバータの7404があれば構成できます。

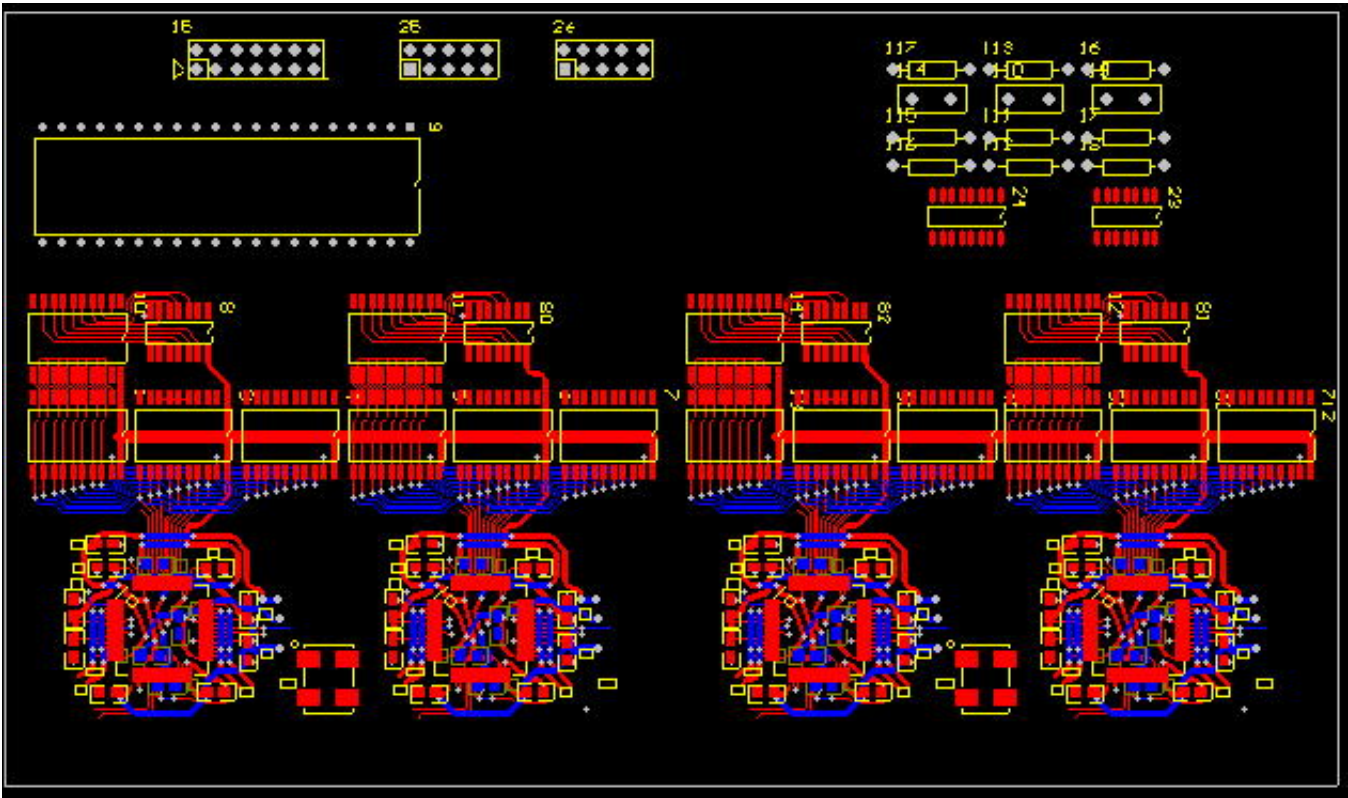


# データシフトを実現するための回路

さて、この回路を組み込むとどんな感じの規模になるか、少し部品をレイアウトして検討してみましょう。  
多彩な入力構成ができるように

- 1. SPDIF入力
- 2. PCM入力 (I2S, RJ, LJ)
- 3. DSD入力 (単純パラ、シフトパラ)

を可能にすることを前提として部品を配置してみました。  
IVステージはありませんが、DAC部分だけでは幅175mm程度の基板に収まりそうです。



基板の幅は約175mm程度。極端に大きくなることは避けられそうです。

このくらいの規模となると、おもわずDAC1704-4Dを思い出してしまいました。  
結構作るとなると、思いっきりが必要になりそうです。

なんとなく頓挫しそうな予感(笑)。でも、考えることが楽しいです。

福島の人からこんな石があることを教えてもらいました。

ES9018SでDSDシフトパラは実現するか？の巻き - 福島の人

2012/02/07 (Tue) 09:23:58

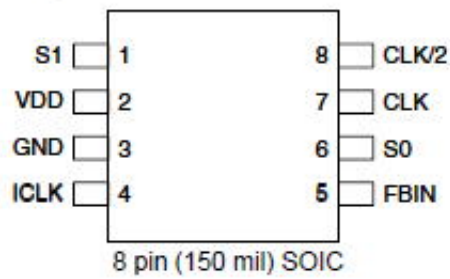
ヒロさんへ、  
とても興味ある内容だと思います。探していたら,ICS570  
というICを見つけました。入力と同期して32 逡倍までの  
クロックを発生させるものです。Digikeyなどで探すと  
同じような 使い方ができ、もっと性能の 高い物(シッタ  
性能など)もあるかもしれません。  
ご参考まで。

いつかのBBSでPCM(I2S)入力ではシステムクロックはfsの逡倍でないとLOWESTでロックしないという書き込みが  
ありましたが、ちょうどこの石があれば実現できそうです。ちょっとマニュアルをみてみましょう。

## すごくシンプルなIC

最初はマイコン制御でも必要かな？とおもっていましたがものすごくシンプルなICです。  
入力クロックを与えて、S0,S1を設定すれば自動的に逡倍の出力が得られるものです。こりゃ便利です。  
ただし、入力クロックの周波数に範囲があります。  
最初はマニュアルをみて何の数字？かとおもいました。まったく表に単位が書いてありません。  
MHzでしょうが……技術屋の資料としてはちょっと失格です。

Pin Assignment



Clock Multiplier Decoding Table

(Multiplies Input clock by amount shown)

S1	S0	FBIN from CLK		FBIN from CLK/2		IC570B (3.3 V)	
		CLK	CLK2	CLK	CLK2	ICLK Input Range	FB from CLK/2*
#1	#6	pin #7	pin #8	pin #7	pin #8		
0	0	Power Down and Tri-State				-	
0	M	x3	x1.5	x6	x3	3.75 to 28	
0	1	x4	x2	x8	x4	2.75 to 19	
M	0	x8	x4	x16	x8	2.5 to 9.5	
M	M	x6	x3	x12	x6	2.5 to 12.5	
M	1	x10	x5	x20	x10	2.5 to 7.5	
1	0	x1	/2	x2	x1	11 to 75	
1	M	x16	x8	x32	x16	2.5 to 5	
1	1	x2	x1	x4	x2	5.5 to 37.5	

0: connect directly to ground

ちょっと計算

PCM入力では192fs以上のシステムクロックが要求されます(オーバサンプリングを使用しないなら24fs以上)。  
マニュアルには **> 192fs**とありますから、厳密には192fsを超える値が必要ですが、おそらく192fsでも大丈夫でしょう。

で、すこし計算してみました。  
用はBCKは64fsなので、BCKの3倍以上のシステムクロックが生成できればよいことになります。  
上の表からBCKを入力周波数として可能な逡倍数をみてみました。

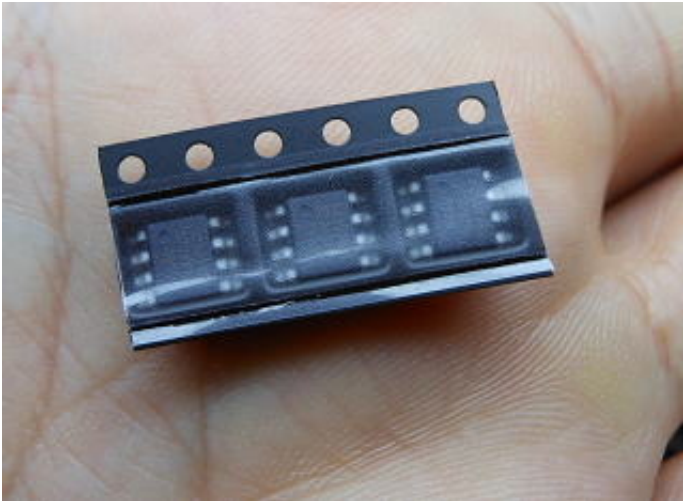
fs (kHz)	Bck(MHz)	必要クロック	BCKを入力として可能な逡倍数
----------	----------	--------	-----------------

		(MHz)	(括弧内は出力周波数が100MHzを超える)
44.1	2.8224	>8.4672	×4～×32
96	6.144	>18.433	×3～×16、(×20)
192	12.288	>36.864	×6、(×12)
384	25.476	>73.723	×3、(×4、×6)

いずれの周波数でも使用可能な通倍の組みあわせがありますね。  
 これは一度試したくなります。  
 ということで、買い物ついでにGETしてみましょう！

## ICがとどきました。 2012.2.10

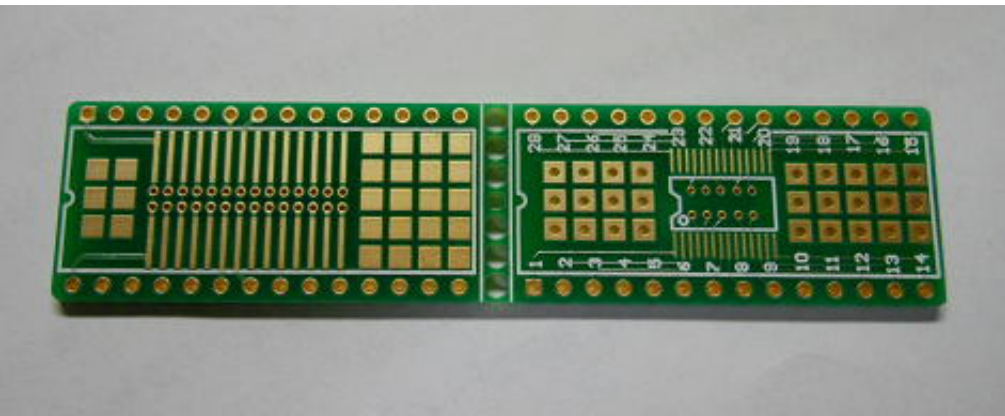
購入したのは3. 3V版のICS570Bです。予備も含めて3個ばかり入手しました。  
 やっぱりDIGIKEYは早いです。



3個ばかり購入。さて、遊んでみましょう。

変換基板が必要ですが、なにかと使う機会も多いので基板の空きスペースがあったときに  
 つくりました。SOIC(1.27mm)とSSOP(0.65mm)の2枚つづりのものです。もちろん今回は  
 左側のSOICのものをつかいます。



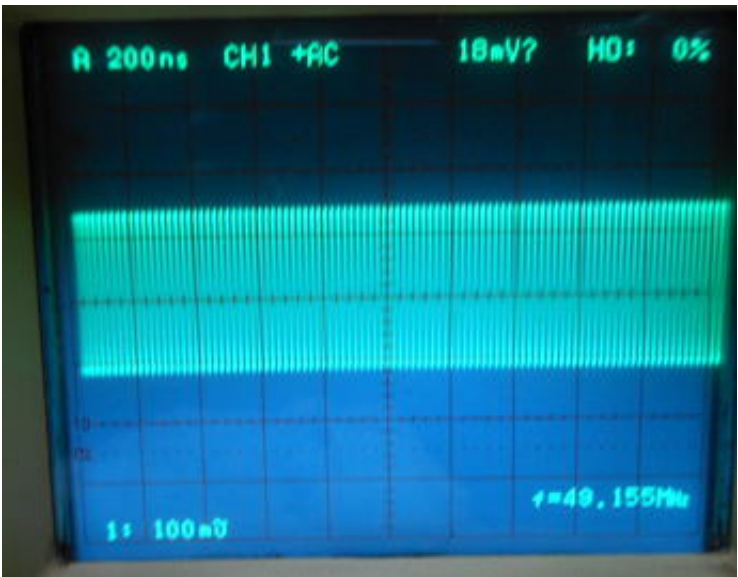


いつか作った変換基板。結構このタイプのものは使います。

## 実装してみましょう！

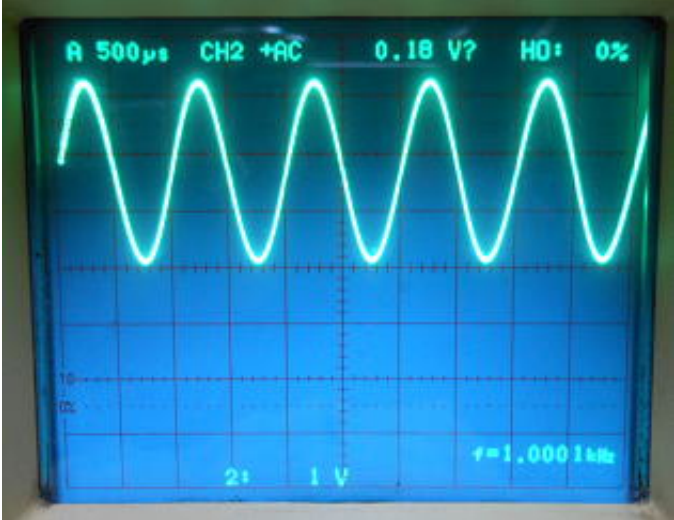
変換基板にICを載せて、その他の配線をします。面倒なのでDAC9018SのICの上に基板をおいて適当にやっつけました(笑)。入力確認は $f_s = 192\text{kHz}$ でしますので、入力クロックは $64f_s = 12.288\text{MHz}$ になりますから、倍率は8倍(98.304MHz)にしています。パターンカットはしたくなかったので、もともとついている100MHzの水晶発信器の1番ピンをGNDに落として出力をDisenableにしておきます。こうすれば、ICS570との出力はぶつかりません。





発振はもんだいなくしています。

入力データに正弦波を入れています、綺麗な波形がでています。  
ロックがはずれると波形がとまった状態が観測されることにはなりますが、  
そんなこともなく安定して出力がでています。



ロックははずれないようです。

ただし、液晶の周波数表示はかなり暴れます。液晶にはDPLLの値を表示していますが、このDPLLの値は  
 $f_s = (DPLL \times \text{入力クロック}) / (2^{32})$   
であらわされますが、書き直すと  
 $DPLL = f_s / \text{入力クロック} \times (2^{32})$   
すなわち

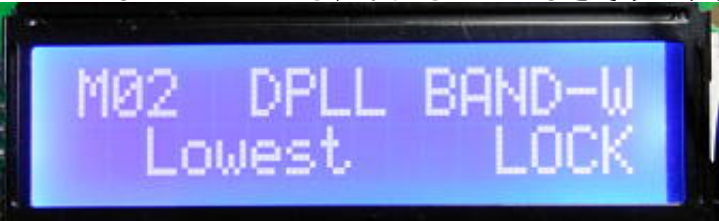


$DPLL = (f_s / f_s) / (64 \times 8) \times (2^{32})$   
となりfs/fsの項があります。この演算はかなり不安定になります。



周波数表示は暴れてしまいます。

表示は暴れますが、LOWESTでも問題なくロックします。  
しばらくながめていても、外れることはなさそうです。



LOWESTでもロックが外れません。

バラックでくみましたが、アドオン基板なんかあると便利です。  
周波数を判別して自動的に倍率を変更すれば、32～192kHzおよびDSD64～DSD256にも対応できそうです。

## こんな石もありました。

Digikeyで探していたらこんな石もありました。  
ジッタクリーナの機能と周波数乗算の両方があります。



# Si5326

## ANY FREQUENCY PRECISION CLOCK MULTIPLIER/JITTER ATTENUATOR

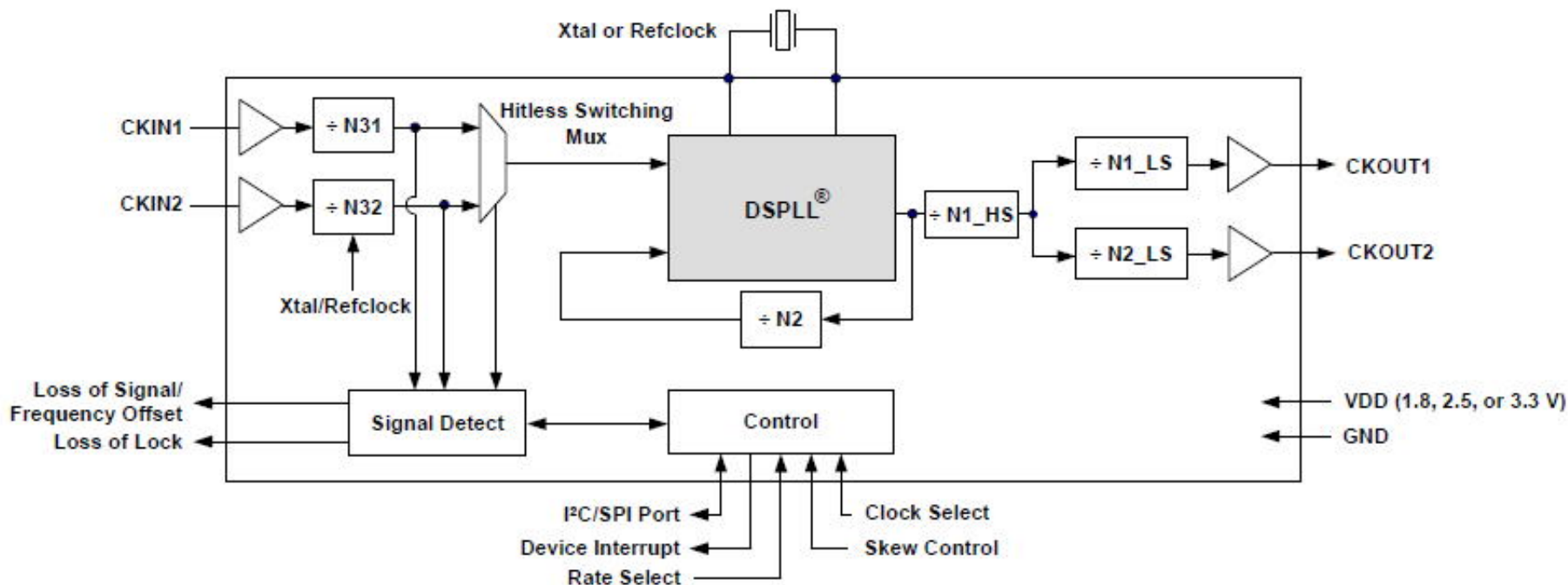
### Features

- Generates any frequency from 2 kHz to 945 MHz and select frequencies to 1.4 GHz from an input frequency of 2 kHz to 710 MHz
- Ultra-low jitter clock outputs with jitter generation as low as 0.3 ps rms (50 kHz–80 MHz)
- Integrated loop filter with selectable loop bandwidth (60 Hz to 8.4 kHz)
- Meets OC-192 GR-253-CORE jitter specifications
- Dual clock inputs with manual or automatically controlled hitless switching (LVPECL, LVDS, CML, CMOS)
- Dual clock outputs with selectable signal format
- Support for ITU G.709 and custom FEC ratios (255/238, 255/237, 255/236)
- LOL, LOS, FOS alarm outputs
- Digitally-controlled output phase adjustment
- I<sup>2</sup>C or SPI programmable
- On-chip voltage regulator for 1.8 ±5%, 2.5 ±10%, or 3.3 V ±10% operation
- Small size: 6 x 6 mm 36-lead QFN
- Pb-free, ROHS compliant



Ordering Information:  
See page 65.

Pin Assignments



探せば色々とでてきますね。

しかし、この石すこしお高いです。でもSPIあるいはI2Cで制御できるのでマイコンのピン数は少なくて済みます。ICS570+Si5317の組み合わせはお財布にやさしいのですが、いかんせんSi5317はハードウェア制御なので、制御のためにマイコンのピン数がたくさん必要です。いづれにしてもアドオンなんてできるのだろうか？

# アドオンするなら

他の部品と干渉することのないエリアを確保しつつ、かつ各種の制御線を横取りしないといけないので、結構エリアは決まってきます。

最低限必要になる石を考えると、ICS570のみをつかう場合だと、ICS570(これがないとはじまりません) PICマイコン(機能的には8PのPIC12F675で足りるかな？無理かな？)

74393(クロック分周用。1/256分周)

の3つでたりそうです。

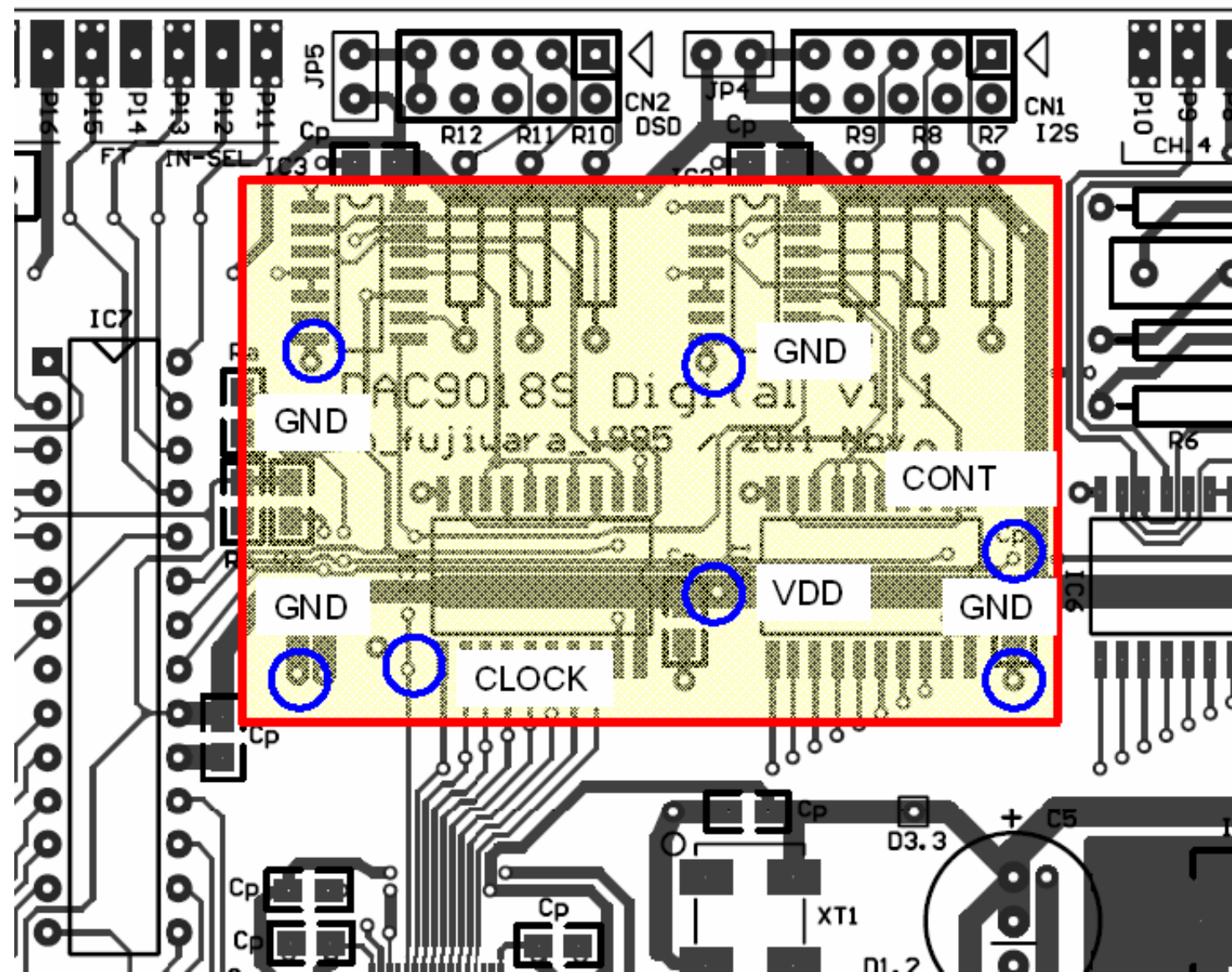
なぜ74393がいるかというと、PICのカウンタに入る周波数は192kHz時で約12MHzになります。

PIC自体は最大周波数が約16MHz対応可ですが、これはあくまでも5V動作時の値なので3.3Vでは少し低くなること。

それと、384kHz時には25MHzになるので、完全にオーバーしてしまうからです。

マイコンが必要な理由は、入力周波数を判定して、それにみあった倍率を設定する必要があります。

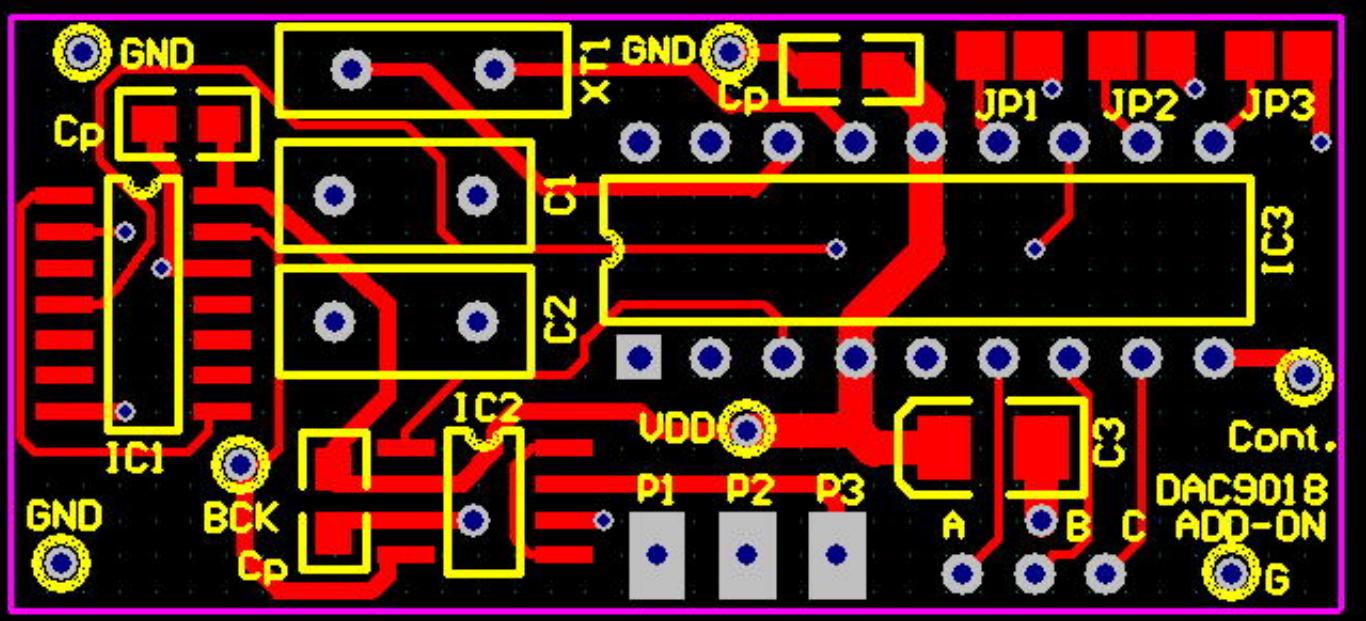
まあ、この3つくらいなら搭載できるでしょう。



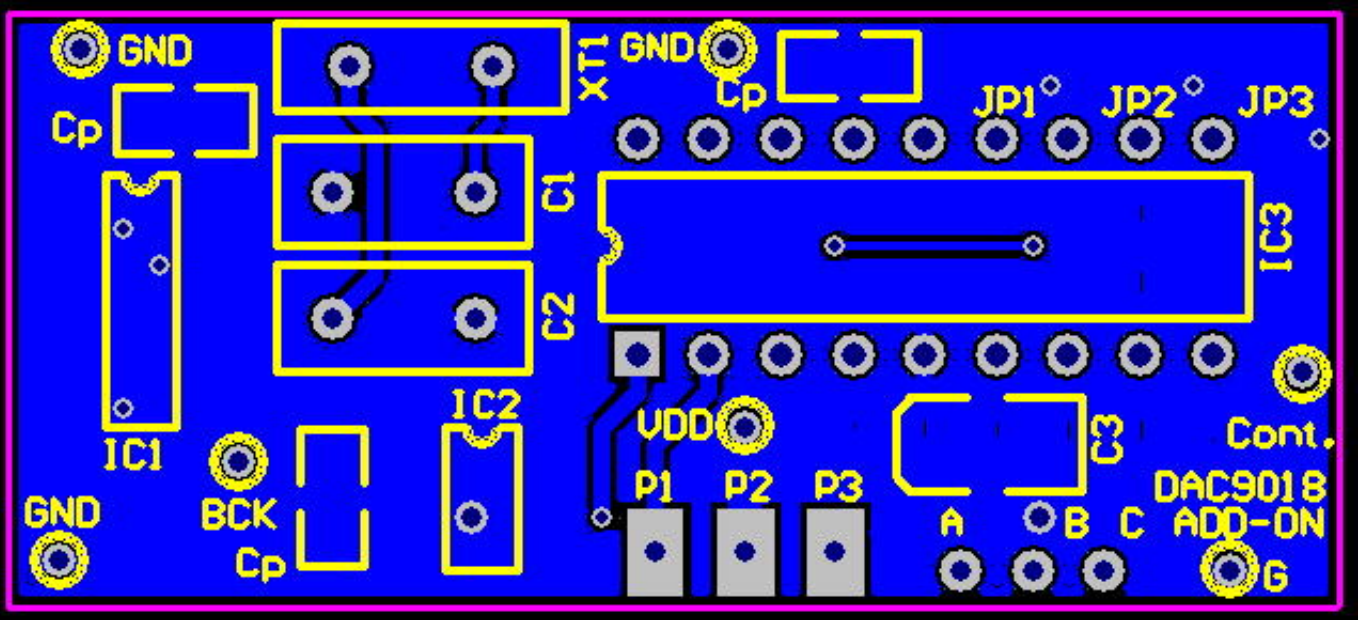
アドオンする場合のエリアです。

こんな感じのアドオンカードです。PICの内蔵の発信器の精度だと、すこし周波数の判別精度が落ちる可能性があるので、水晶振動子も搭載できるようにしました。





パターン面(部品面)



ベタアース面(半田面)

まずはアートワーク描いてみました。

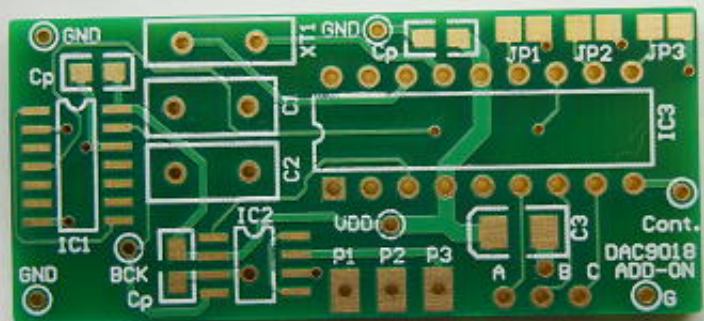
# どうしようかな？ 2012.2.18

シフトパラをやってみたいところだけど、ES9018Sを4個つかって16シフトパラ(半クロックずらし)って、かなり豪勢です。豪勢だけならいいけど、IV変換にOPアンプが使えない問題もあります(前編参照)。音色を決める大きな要因はアナログ部分ですから、柔軟にアナログ部を変えられるようにしたほうが面白そうです。となると、8シフトパラにしたほうが、自作の面では面白いかもしれません。そうすれば2個のES9018で済みますので基板サイズとしても適度なサイズで収まります。折角やるなら半クロックシフトパラと、全クロックシフトパラの切り替えもできるといいかな。さらに、DAC基板を2枚組み合わせれば16シフトパラもできるような拡張性をもたせるのもいいかな？

なんか、構想ばかり膨らんで全然前に進まないです(笑)。

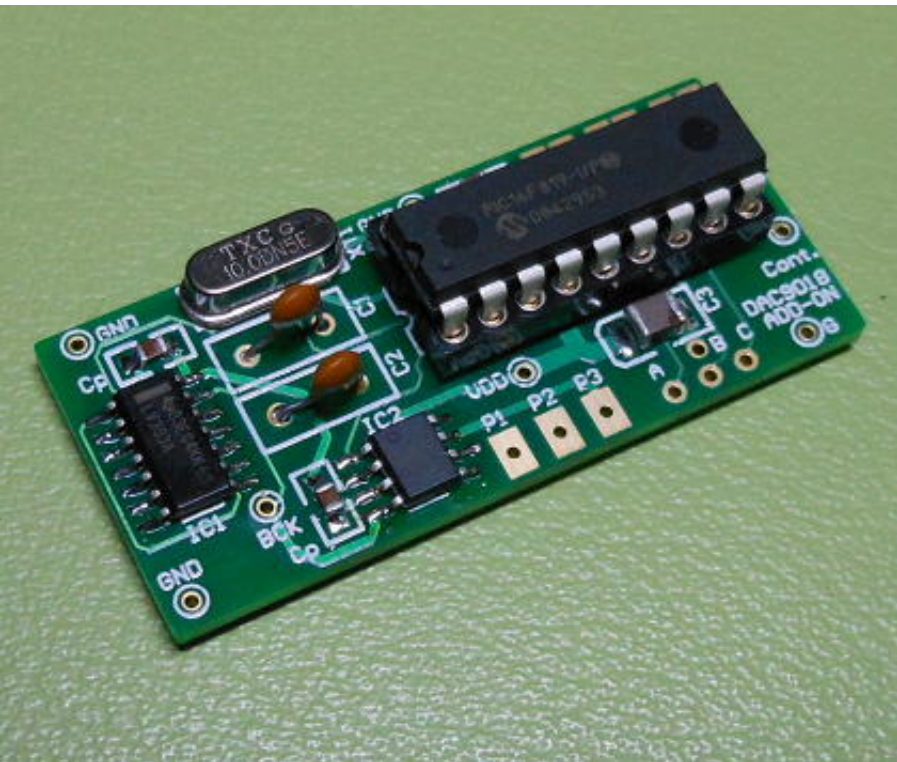
## そうこうしているうちに基板が到着 2012.2.28

いろいろな基板を試作している中に今回のDAC9018S用のアドオン基板をいれていました。サイズはDIP-ICのサイズから判るように小さいです。



基板が完成しました。

部品点数も少ないの半田付け自体は10分もあれば終わるでしょう。まずは表面実装部品をとりつけて、その後にスルーホール部品を取り付けて完成です。

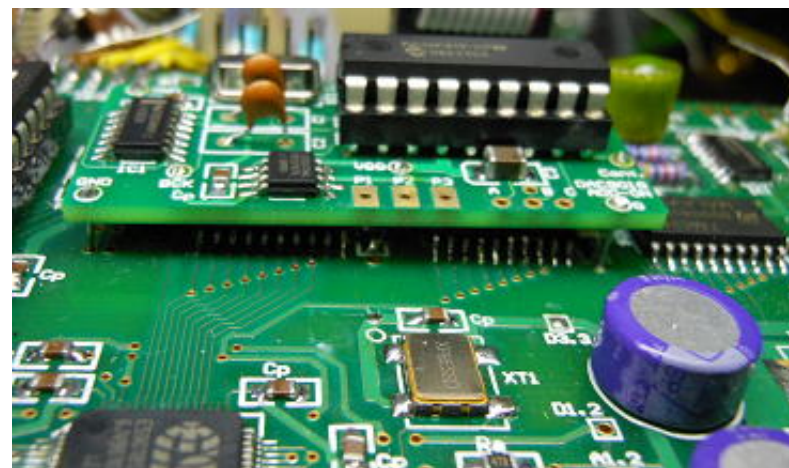
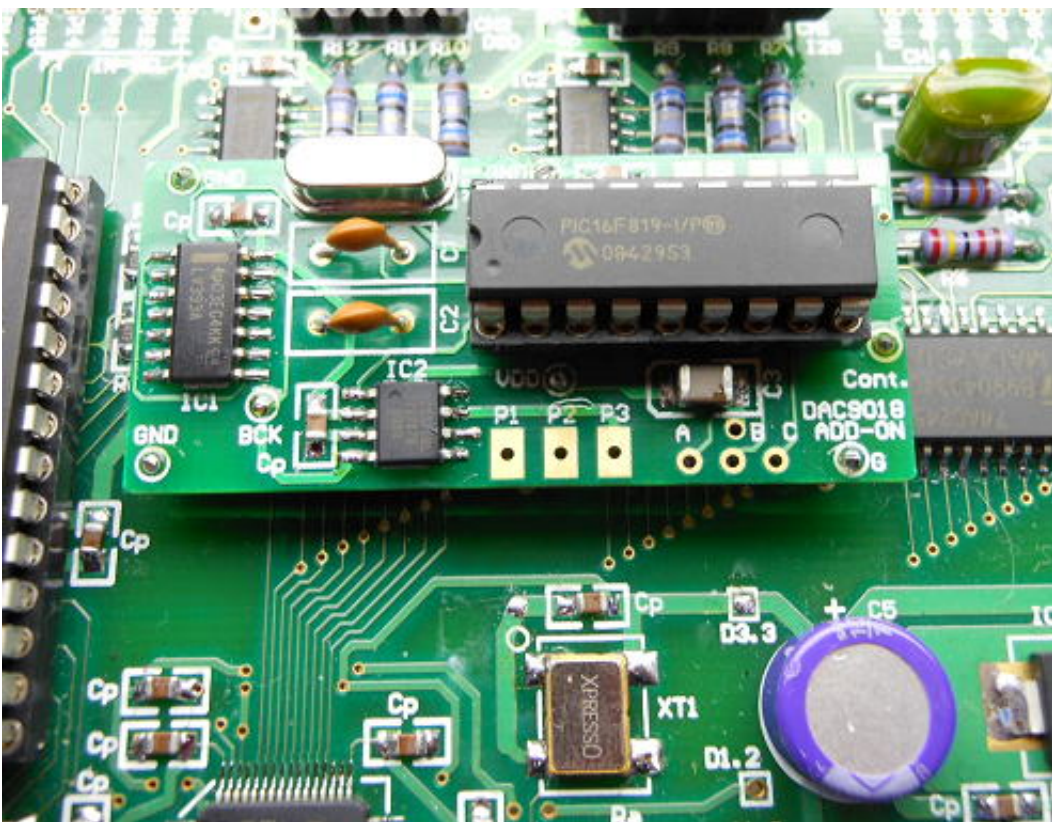


完成したのでお出かけ用の写真をパチリ！

## メイン基板と接続しましょう！

メイン基板と今回の基板との接続は部品のリード線の切れ端をつかって接続します。  
電源ラインについてはメイン基板も大きめのスルーホール穴径なので大丈夫ですが、  
信号線については小さい穴径なのでラッピング線などの細い線でメイン基板と接続します。  
結構たくさんの線（電源線5本、信号線2本）で結合することになるので、  
しっかりと取り付きます。

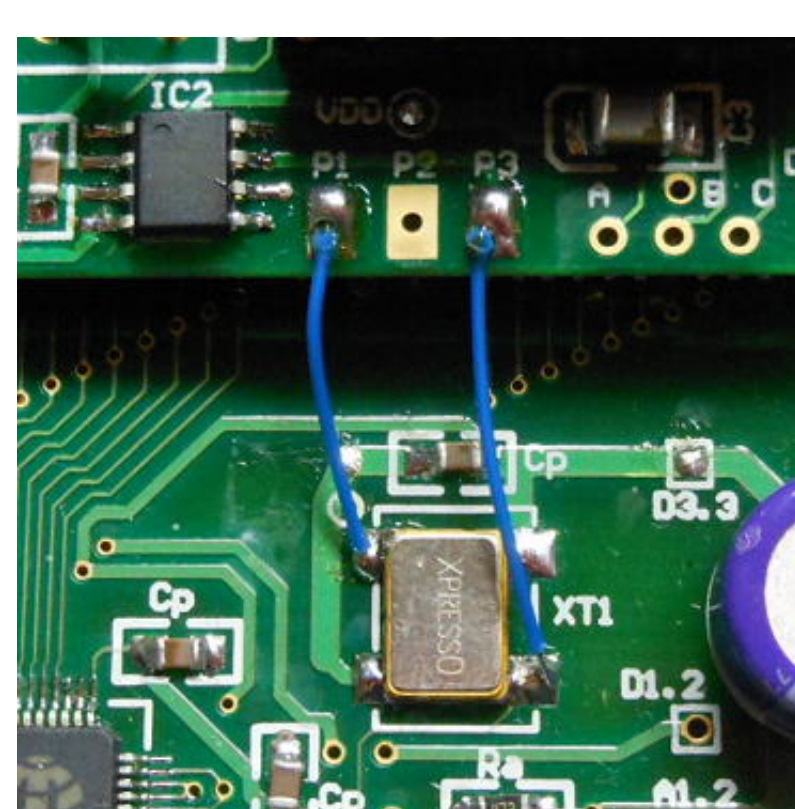




メイン基板と今回基板ドータボードとの接続の様子)

メイン基板との接続はP1,P3端子を使って水晶であるXT1に接続します。水晶のPIN1は出力制御端子になっているので、この端子にドータボードから制御信号を送り込みます。





ドータボードと水晶との接続。

## ソフトウェアは

基板自体のハードは簡単ですが、ソフトは少し複雑です。下記のシーケンスで動かすことになります。

### 1. 入カラインの判別

これはメイン基板 (DAC9018S) が SPDIF、PCM、DSD のどの入力モードで動いているか検知します。

### 2. 周波数解析

メイン基板が PCM、DSD で動いている場合は BCK 信号の周波数を計測します。  
周波数計測の精度が必要なことからドータ基板には水晶発振子を搭載しています。  
このとき周波数が 32kHz、44.1kHz、・・・、192kHz、384kHz などの規定値を外れる場合は  
既存の水晶発信器 (100MHz) をつかうようにしています。

### 3. クロック逡倍の設定

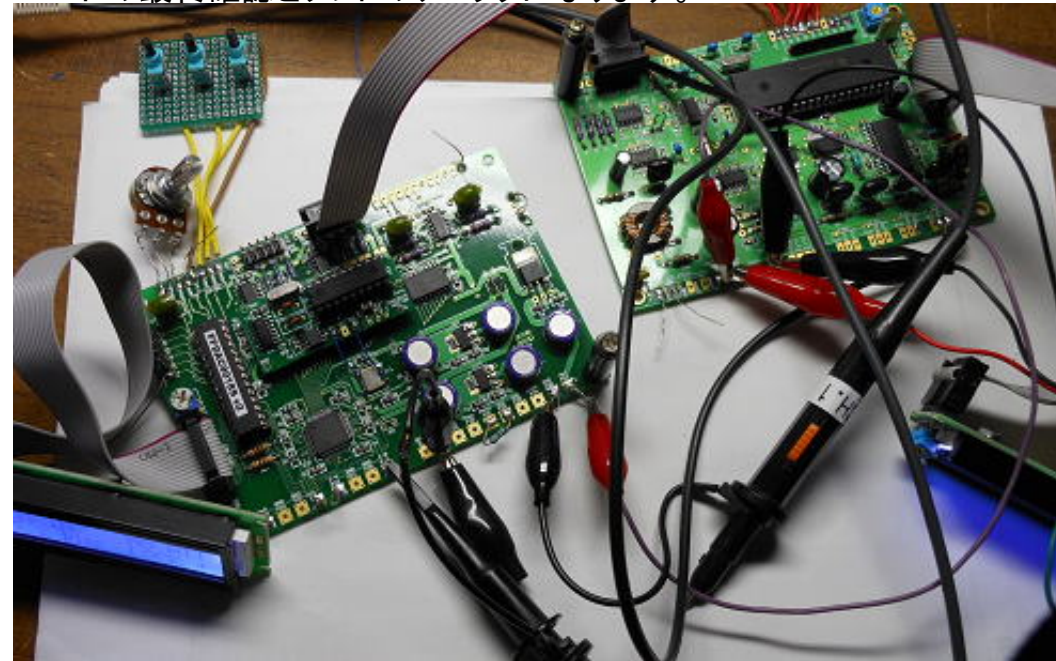
計測した周波数から 100MHz にもっとも近くなる逡倍数を設定して、ICS570 に設定信号を送ります。

## 4. 1に戻ります。

こんな感じのプログラムを組みます。フラッシュメモリーの使用率は30%程度でした。

## 動作確認！

最後にFFASRCと接続して動作確認です。もともと動くことは確認済みですから、どちらかといえばハードの最終確認とソフトのチェックになります。



動作確認の様子。

## LOWESTでもロックしますね。

ソフトのチェック確認も兼ねて、色々な周波数の信号を入れてDPLLがロックするかどうかを確認しておきましょう。

44. 1～192kHzで問題なくロックします。非常に安定しています。

ソフトは各種の周波数に対応できるように

**16、22.05、32、44.1、48、64、88.2、96、128、176.4、192、256、384kHz**

を標準周波数として設定しています。逆にこれ以外の周波数が入った場合は内部水晶発信器に切り替えます。

しかし、これだけ変化のある設定はないでしょう(笑)。



44.1kHz時



96kHz時



192kHz時

## お楽しみの試聴！

いつものようにIV基板に接続して、耳蛸ソースを試聴です。しかし、毎度のことながらバラックでの配線も大変です。ほんとにそろそろケースに入れなくっちゃ(笑)。

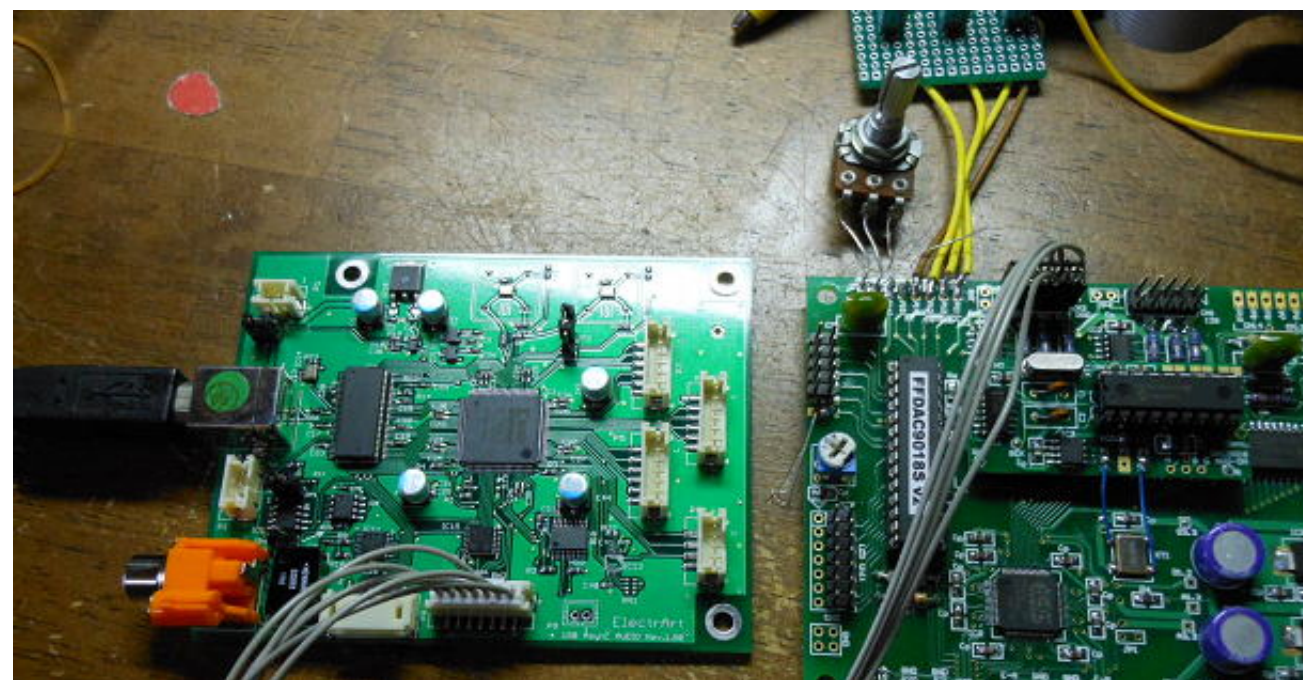
おお！

プラシーボ全開！(笑)。これいいですね。やはりLOWESTでロックするという安心感がそう感じさせるのでしょうか。とにかく細かい音がよく聞こえます。いつまでも音楽を聴きたくなる感じになります。でも、なぜシステムクロックの通倍ならLOWESTでもロックするのだろう？よく考えてみないといけなそうですね。



## DSD入力は？

エレクトさんのUSB基板を再度接続してDSDでの動作も確認しておきました。  
こちら問題ないようです。表示は本来の入力値とはことなりますし、また表示も荒れますが  
ロックが外れることはなさそうです。



DSDでの動作も確認しておきました。



DSD128での再生の様子。表示周波数は異なります。

## 備忘録

このドータボードの入力周波数(BCK)と出力周波数の関係を少し整理しておきましょう。  
基板端子とあるのは、3本の端子があるので現在の受信周波数を分析して3ビット出力でだすようにしています。



44.1～192kHzの主要な周波数では基板端子は1～6に設定して、それ以外では7になるようにしています。  
入力はSPDIFであれば、内部クロックをつかいますので、このときはゼロ(0)になるようにしています。

	Fs (kHz)	BCK (MHz)	通倍率	SCK (MHz)	基板端子 (A～C)
1	16.00	1.024	32	32.7680	7
2	22.05	1.4112	32	45.1584	7
3	32.00	2.048	32	65.5360	7
4	44.10	2.8224	32	90.3168	1
5	48.00	3.072	32	98.3040	2
6	64.00	4.096	20	81.9200	7
7	88.20	5.6448	16	90.3168	3
8	96.00	6.144	16	98.3040	4
9	128.00	8.192	12	98.3040	7
10	176.40	11.2896	8	90.3168	5
11	192.00	12.288	8	98.3040	6
12	256.00	16.384	6	98.3040	7
13	384.00	24.576	4	98.3040	7

さて、そろそろリリースかな？

Re: ES9018同期入力時にロック周波数表示がぶれる原因 - Bunpei NEW

2012/03/03 (Sat) 17:33:08

同期マスタクロックのヒロ様による音質評価結果を拝見し、ひと安心しました。  
ところで、fs=352.8kHz (PCMのDXD)には非対応なのですね。2Lの無料公開DXDソースがこれで聴けなくて残念！

編集

削除

あ、ちょっと横着してました。実は上記の表は厳密ではなく実際には下表のようになります。  
256kHzまでは正確に判定していますが、300kHz以上は大まかに判定しています。  
なにが起こるかわかりませんので……。

	Fs	BCK	通倍率	SCK	基板端子
--	----	-----	-----	-----	------

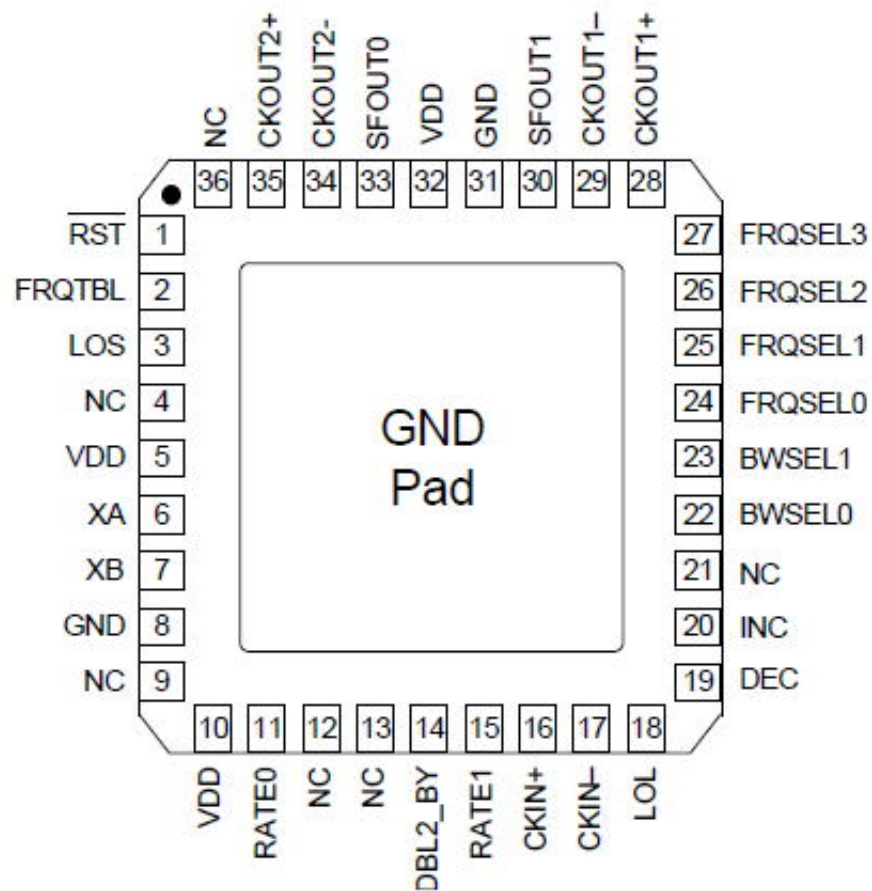
	(kHz)	(MHz)		(MHz)	(A～C)
1	16.00	1.024	32	32.7680	7
2	22.05	1.4112	32	45.1584	7
3	32.00	2.048	32	65.5360	7
4	44.10	2.8224	32	90.3168	1
5	48.00	3.072	32	98.3040	2
6	64.00	4.096	20	81.9200	7
7	88.20	5.6448	16	90.3168	3
8	96.00	6.144	16	98.3040	4
9	128.00	8.192	12	98.3040	7
10	176.40	11.2896	8	90.3168	5
11	192.00	12.288	8	98.3040	6
12	256.00	16.384	6	98.3040	7
13	300-390	19.2-25	4	76.8-100	7
14	>390	>25	2	>50	7

## ちょっとお遊び 2012.4.7

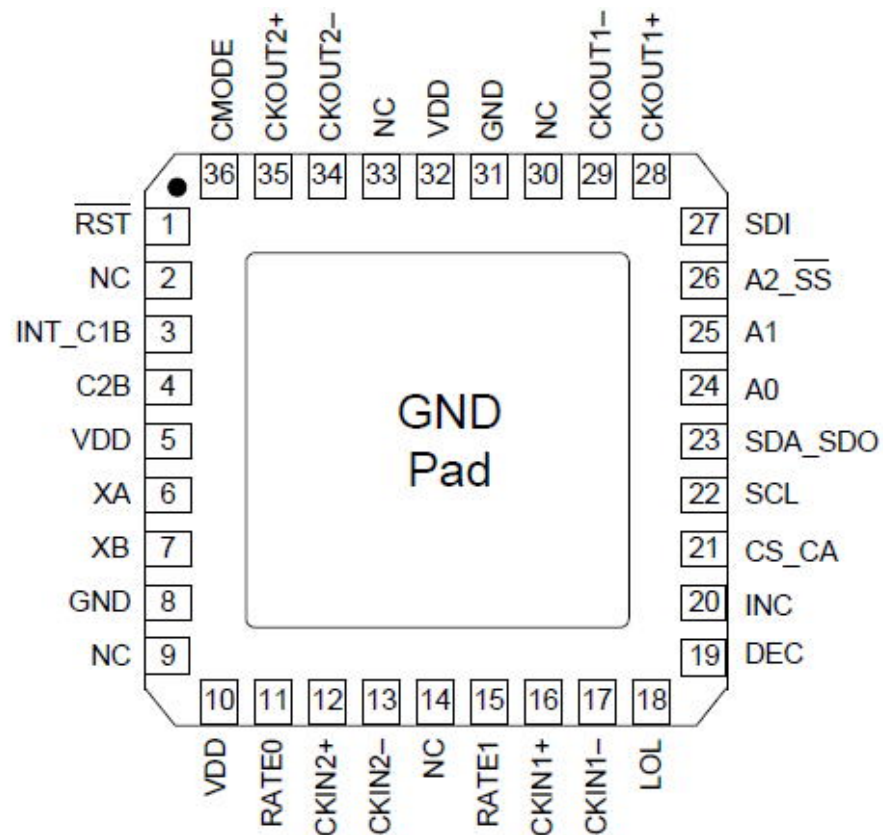
Si5326を買ってみました。結構高いICなので、ジッタクリーナとしてつかうならSI5317を間違いなくつかうでしょうね。でも、なんとなく面白そうなのでDIGIKEYに注文するときに併せて発注しました。というか、どうしても送料無料(7500円以上)にならなかったなので、無理やり押しこんだのですが(笑)。

### こりゃ便利！

なにが便利かといえば、Si5326ってSi5317とピン配置に共通点が多いです。ということは、ジッタクリーナ基板をつかって評価することができそうです。

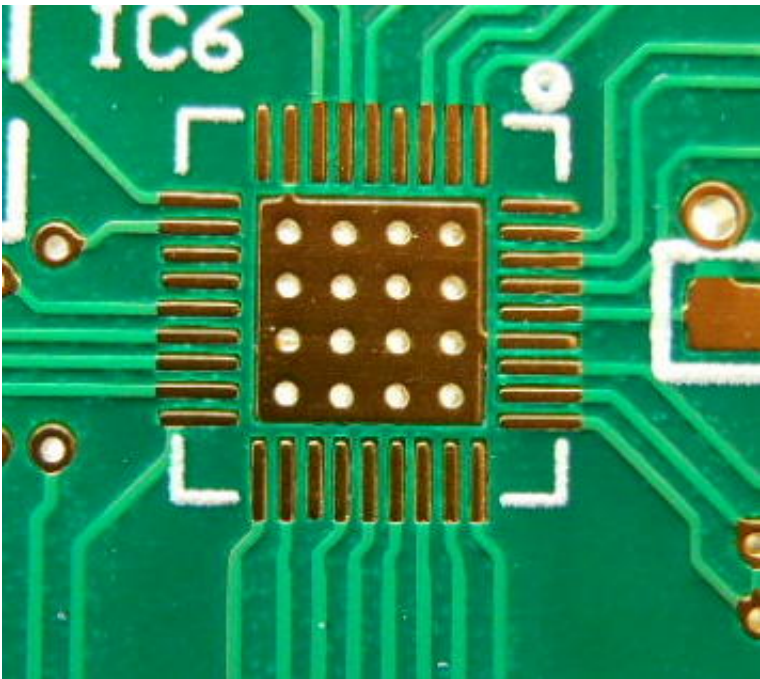


Si5317のピン配置



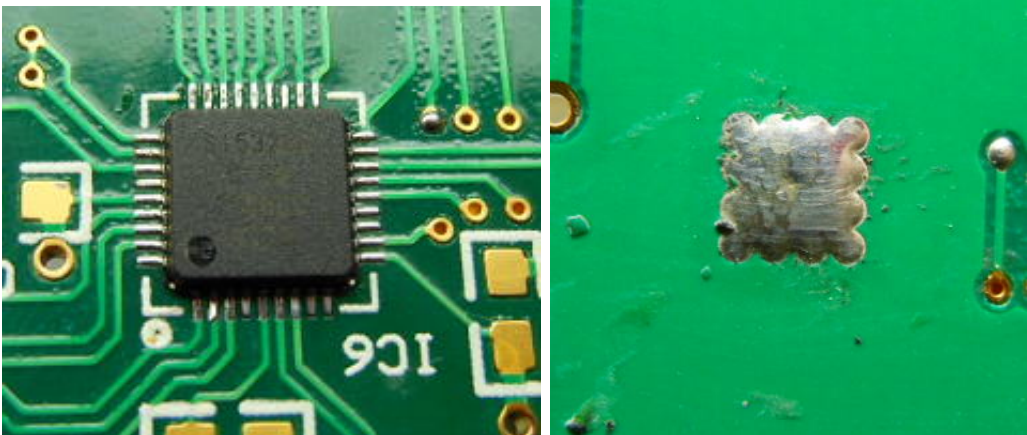
Si5326のピン配置

ということで、いきなり実装していきましょう。  
 使用した基板は試作版のジッタクリーナ基板ですから、サーマルパッドの部分はリリース版とはちがって小さいパッド径になっています。



試作版でのサーマルパッドの穴径は少し小さいです。

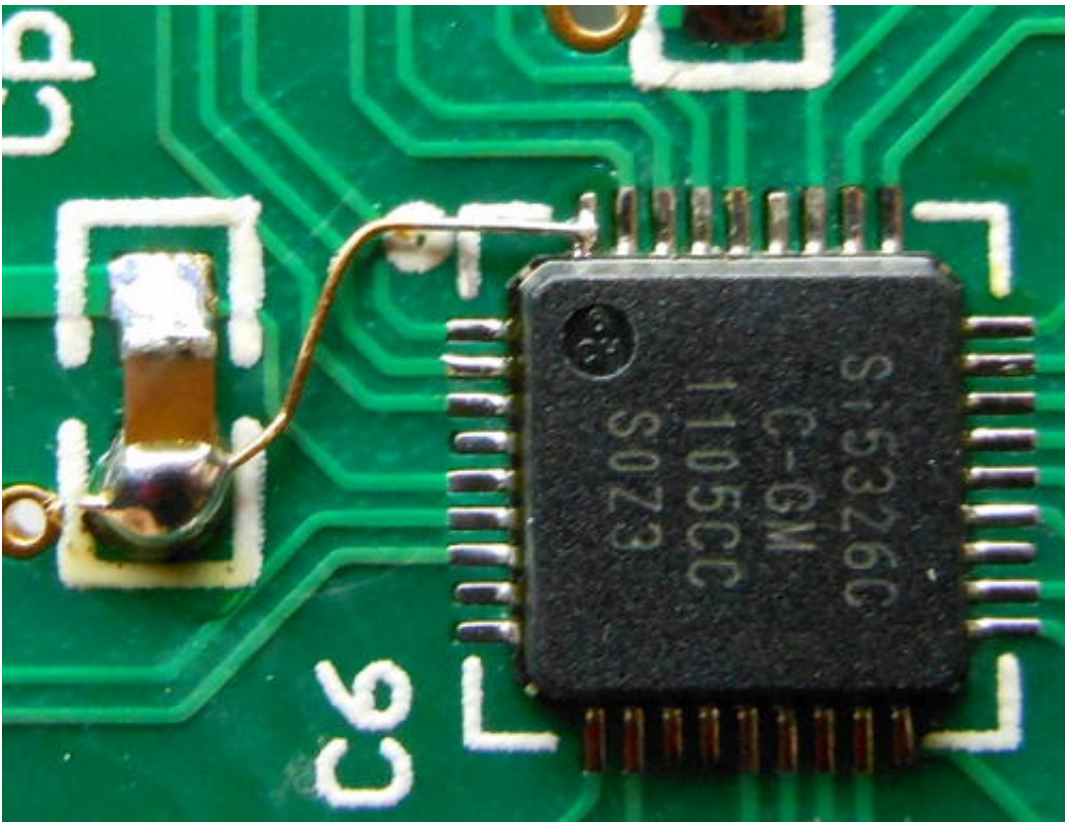
半田付け自体はあまり難しいものではありません、裏面からの半田付けは容量の大きなコテをつかってじっくり、こんがりと仕上げてやります。



半田付け完了です！

SI5317とピン配置に共通点が多いからといっても、もちろん異なるところもあります。あるピンはGNDへの接続が必要だったりしますが、ラッキーなことに端ピンだったので簡単にジャンパーを飛ばすことで対応ができました。これが中央のピンだったら、結構難しいです。





必要なピンをGNDに接続しますが、端ピンなので簡単でした。

さて、動作に必要なソフトを書いていきましょう。うまく動くかな？

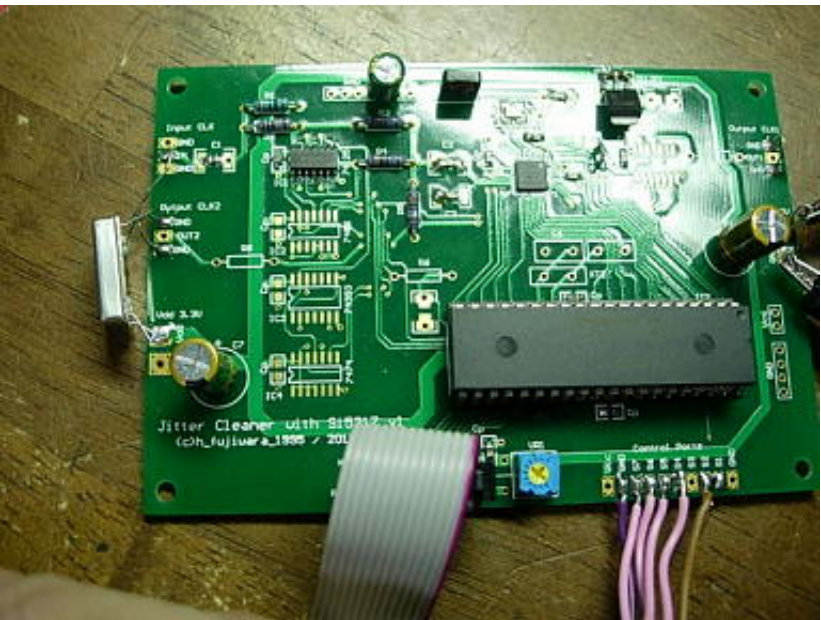
---

しかし、また日曜日から2泊の出張…… 最近全然趣味が進まなくてストレス溜まるナ～。

---

# さてさて

まずは必要最小限の部品をとりつけて動作評価です。またSi5326は内部の設定レジスターの数が多いので、スイッチとLCDをとりつけて、任意のレジスタに任意の値を書き込めるようにして評価です。

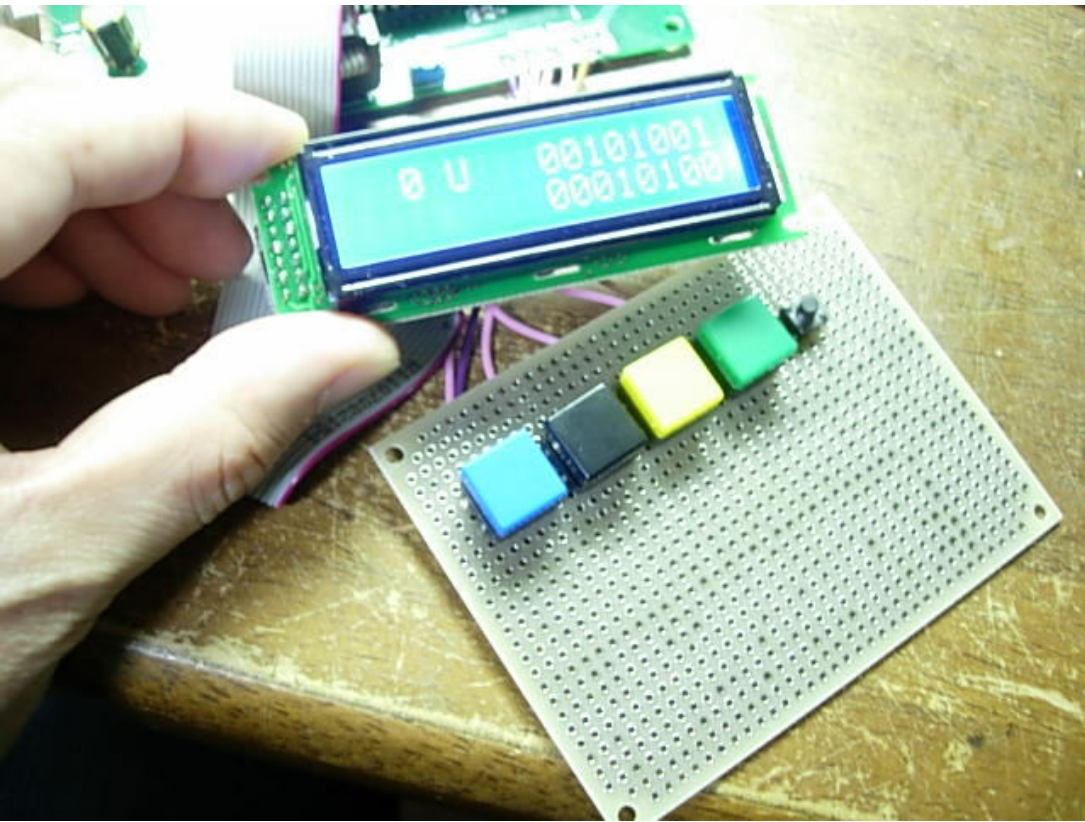


必要最小限の部品を搭載して動作確認の準備です。

## うまく動かない？

まず、LCDの表示や書き込みは問題なしです。ということはSi5326自体は動いているようです。というのもデータ通信にはI2Cをつかうので、素子が動いていなければ通信エラーが生じます。

さて、入力に適当な周波数信号を入れますが、なぜか有効な入力と判断しません。で、入力周波数の範囲を変えてやると(高くすると)、入力信号が有効となることがわかりました。なぜかある入力の範囲なら入力は有効になります。ただし、DSPLLはロックしません。つぎに入力が有効な範囲にしておいて、レジスターの設定値を変更するとDSPLLがロックする値があることがわかりました。  
ん……よくわからないな。  
なにか設定しわすれているレジスタがあるのかな？もうわかんない。



動作チェックをころもみるも、うまく動きません……

水晶を変えてみるかな？  
これは苦戦するかも……

# シフトパラにもどる。

SI5326にはすこし見切りをつけました・・・、いや見切りをつけられたかも(汗)。  
さて夜の夜長でようやく構想が固まってきました。まずシフトパラは行うとしてもES9018を4つつかうのは少し多すぎるので、  
2個使ってモノラル使いで8シフトデータ入力がバランスがいいのではと勝手に思っています。それよりもクロック関係を  
充実させた方が効果が高いかもしれません。ひとつは逡倍クロックでPCM入力でもLOWESTでロックすることはわかりましたので、  
ICS570は搭載しましょう。

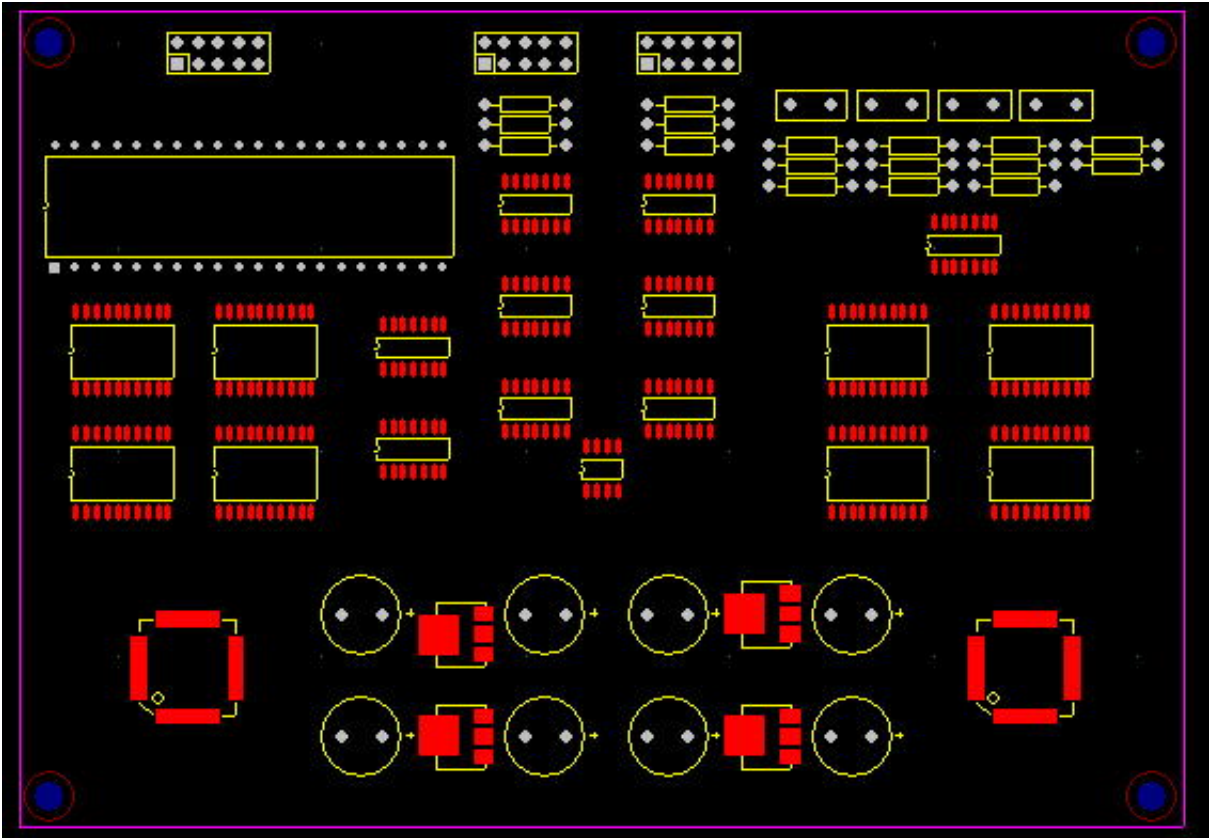
さて、まず基板サイズですがこれについてはDAC9018Sのサイズだとおさまらないので、もうすこし大きめのメモリーバッファ  
などと同じサイズまで拡張しましょう。  
搭載するマイコンですが、どのくらいのIOが必要になるか考えてみると

LCD制御	6
ES9018制御	I2C 2
RESET	1



動作モニタ	4
ロジック制御	5
入力、フィルタ切り替え	4
BW切り替え	3
予備	$\alpha$
計	$25 + \alpha$

ということで28PのPIC16F886では $\alpha$ がなければ収まりますが、無理はやめましょう。  
 ここは40PのPIC16F887を搭載した方がよさそうです。ということで必要になりそうなICを搭載してみて、密度感をみてみましょう。



こりゃいっぱいだけどなんとかなるかな。

しかし40PのDIPの存在感が大きいな。これがもっと小さくなると配置に余裕ができるのですが、QFPなどはマイコンのいライターがないので使いにくいです。

ちょっと欲をだして

ICS570をつかえばPCMでもLOWESTでロックすることは確認できていますが、ICS570自体の出力



はそれなりにジッタが乗っています(約200ps)。そこで、さらにジッタクリーナを通して(300fs程度と1/1000)、供給すればもっとよくなるのではとの期待があります。さらにSPDIFについても100MHzの内部クロックでロックはしますが、なかにはしない場合もあるようです。ということはロックしている場合でもかなりぎりぎりの状態でロックしているのでは？という懸念があります。そこで、SPDIFでもクロック周波数を抽出して、それをベースにして逡倍クロックをつくれれば、さらに安定してLowestでもロックするのは？という期待です。で、SPDIFでクロック信号を取り出すためにはどうするか？これはもったいないですがCS8416などのDAIを搭載します。もったいないといってもES9018に比べると随分安いですし、入力アンプの代わりになりますし、それで性能があがるなら文句なしです。

ジッタクリーナにはSi5326をつかいたいところですが、じつはこれが意外と高いのと、さきの検討でうまく動かせていません。ということもあり、少し部品点数が増えるのですがSI5317をつかいたいと思います。Si5317はハードウェアコントロールなので、制御のためのマイコンのピン数がより多く必要になります。CS8416はI2C制御が可能なので、IOは増えません。SI5317の制御でどのくらい増えるか考えてみると、12本程度増えそうです。となると40PのPIC16F887でも無理だなあ？

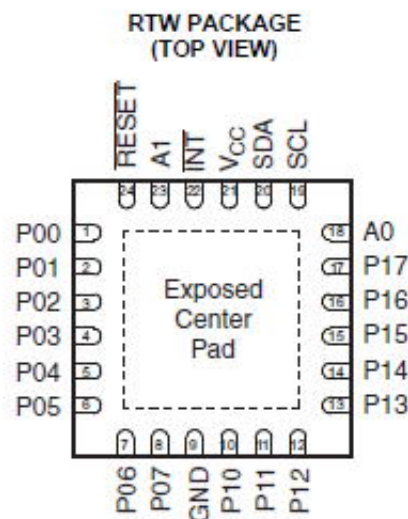
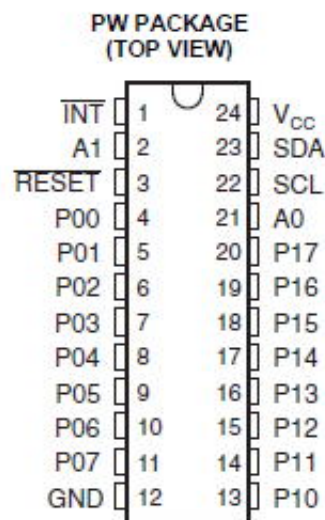
どうしよう？

1個のPICで納めるためのものは難しくなるので、クロックコントロールとDACコントロールを2つのCPUに分けることが考えられますが、ソフトがかなりややこしくなりそう。ロジックICを追加してIO点数を増やすことを考えてみましょう。いろいろと探しているとI2C制御が可能なIOエキスパンダがあることがわかりました。これをつかうとIOが簡単に増やせられます。さらによいことは、制御線はI2Cなので2本だけで済むので、基板内の配線のとりまわしが簡単になります。40PのPICから各ICに配線を延ばすとなると、かなり配線のための基板面積が必要になってきますからね。

# LOW VOLTAGE 16-BIT I<sup>2</sup>C AND SMBus LOW-POWER I/O EXPANDER WITH INTERRUPT OUTPUT, RESET, AND CONFIGURATION REGISTERS

## FEATURES

- Low Standby-Current Consumption of 3  $\mu$ A Max
- I<sup>2</sup>C to Parallel Port Expander
- Open-Drain Active-Low Interrupt Output
- Active-Low Reset Input
- 5-V Tolerant I/O Ports
- Compatible With Most Microcontrollers
- 400-kHz Fast I<sup>2</sup>C Bus
- Polarity Inversion Register
- Address by Two Hardware Address Pins for Use of up to Four Devices
- Latched Outputs With High-Current Drive Capability for Directly Driving LEDs
- Latch-Up Performance Exceeds 100 mA Per JESD 78, Class II
- ESD Protection Exceeds JESD 22
  - 2000-V Human-Body Model (A114-A)
  - 1000-V Charged-Device Model (C101)
  - 200-V Machine Model (A115-A)



The exposed center pad, if used, must be connected as a secondary ground or left electrically open.

こんなIOエクspandaがあるんですね。便利そうなのでつかってみましょう。

# 具体的に基板配置を考えてみましょう。

IOエキスパンダをつかうので、思い切ってPICは28PのPIC16F886に変更しました。  
これで基板の面積はだいぶ余裕がでます。

さて、ここいらで機能を整理しておきましょう。

## 入力は以下の4種類

1. SPDIF入力(4ch)
2. PCM入力(I2S, RJ, LJ)
3. DSD(通常)
4. DSD(シフトパラ)

こんな感じでしょうか。SPDIF入力についてはES9018での切り替え機能は使わずに、外部で切り替えるようにします。  
その理由はSPDIF信号からクロックを抽出することを考えると、信号線は1つにまとめておいた方が便利です。  
CS8416がSPDIFの入力切り替えの役割も担います。

## クロックジッタ機能搭載

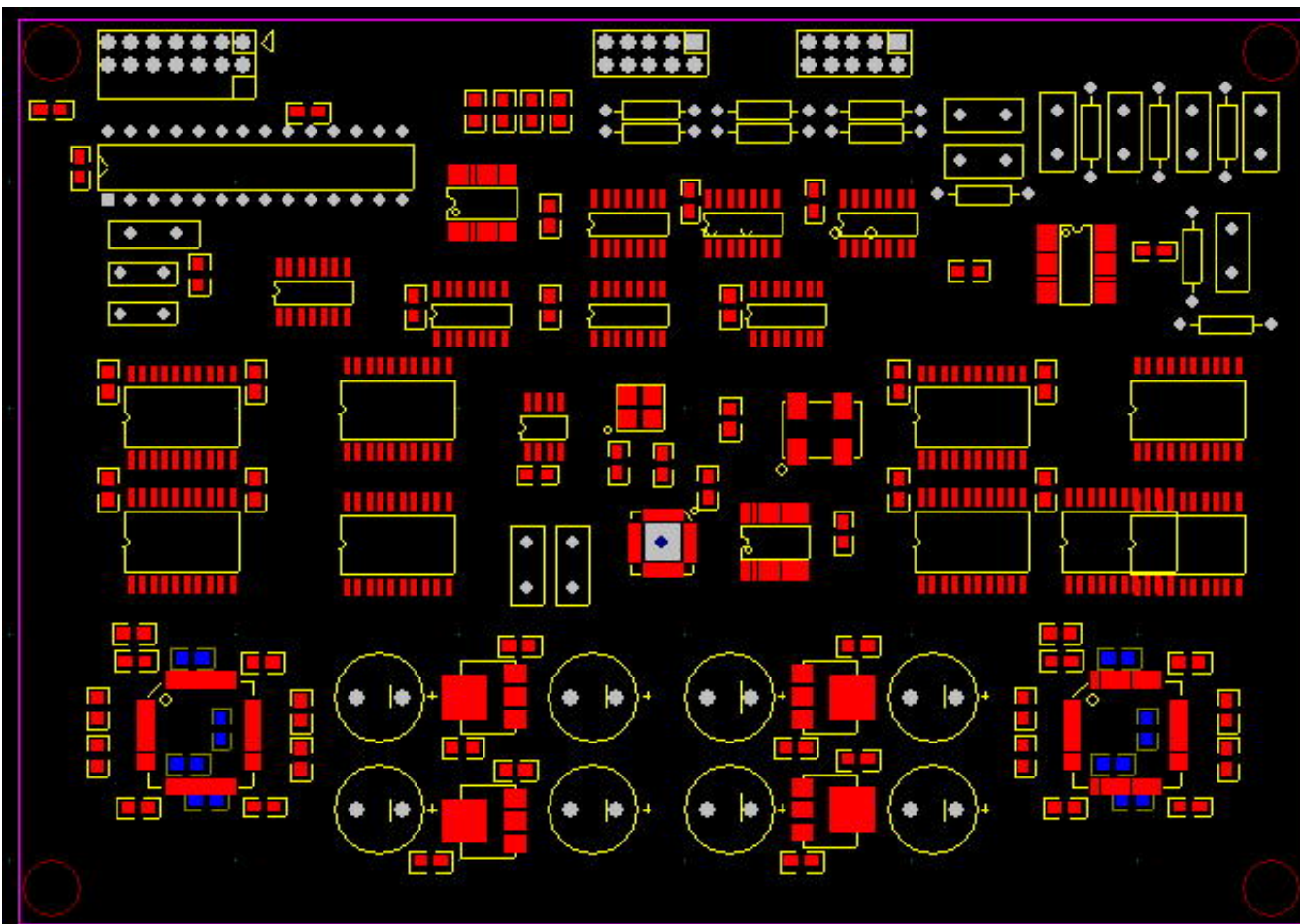
これがICS570+SI5317をつかいます。PCM入力に加えてSPDIFでも、BCKの逡倍からシステムクロックを生成できるようにします。もちろん外部入力がないときも考えて100MHz水晶も搭載します。

## ES9018はモノラル使用。

これに限定します。2枚使いも考えられますが、そこまで拡張することもないでしょう。

## さて、ざっと部品を配置してみましょう。

部品はのりそうですが、配線できるだろうか……。  
企画倒れにならないか心配になってきました。

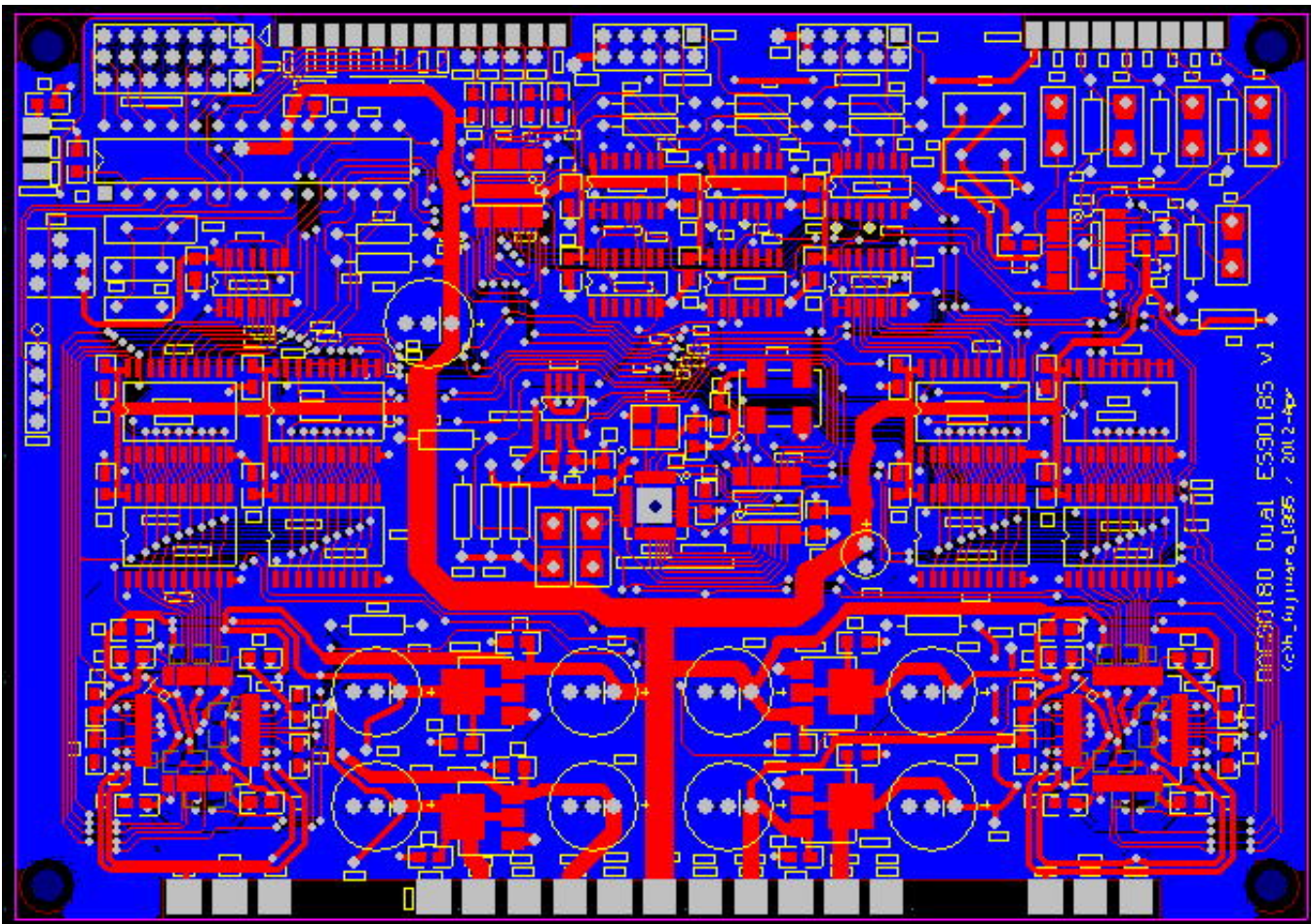


こりゃ結構な部品点数になりそうです。配線できるかな～。

なんとかなりそう！

コードネームはDAC9018Dです。





一通り完成？

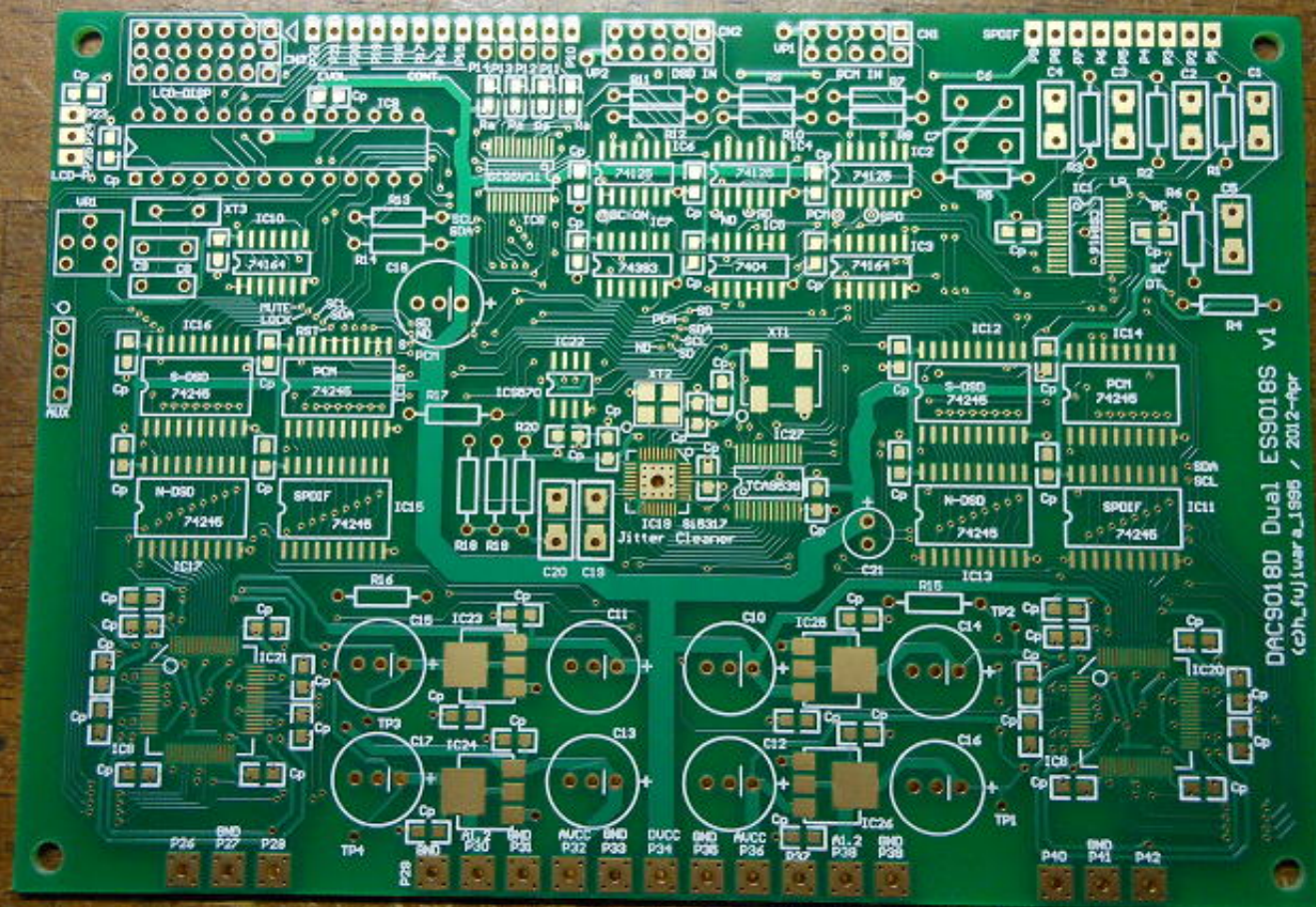
いままでにない配線の密度です。メモリーバッファを越えるかな？  
さて、一通り見直して製作してみましょう！

来週は国内ロードに海外ロードに大忙し……

基板到着してました。 2012.4.25

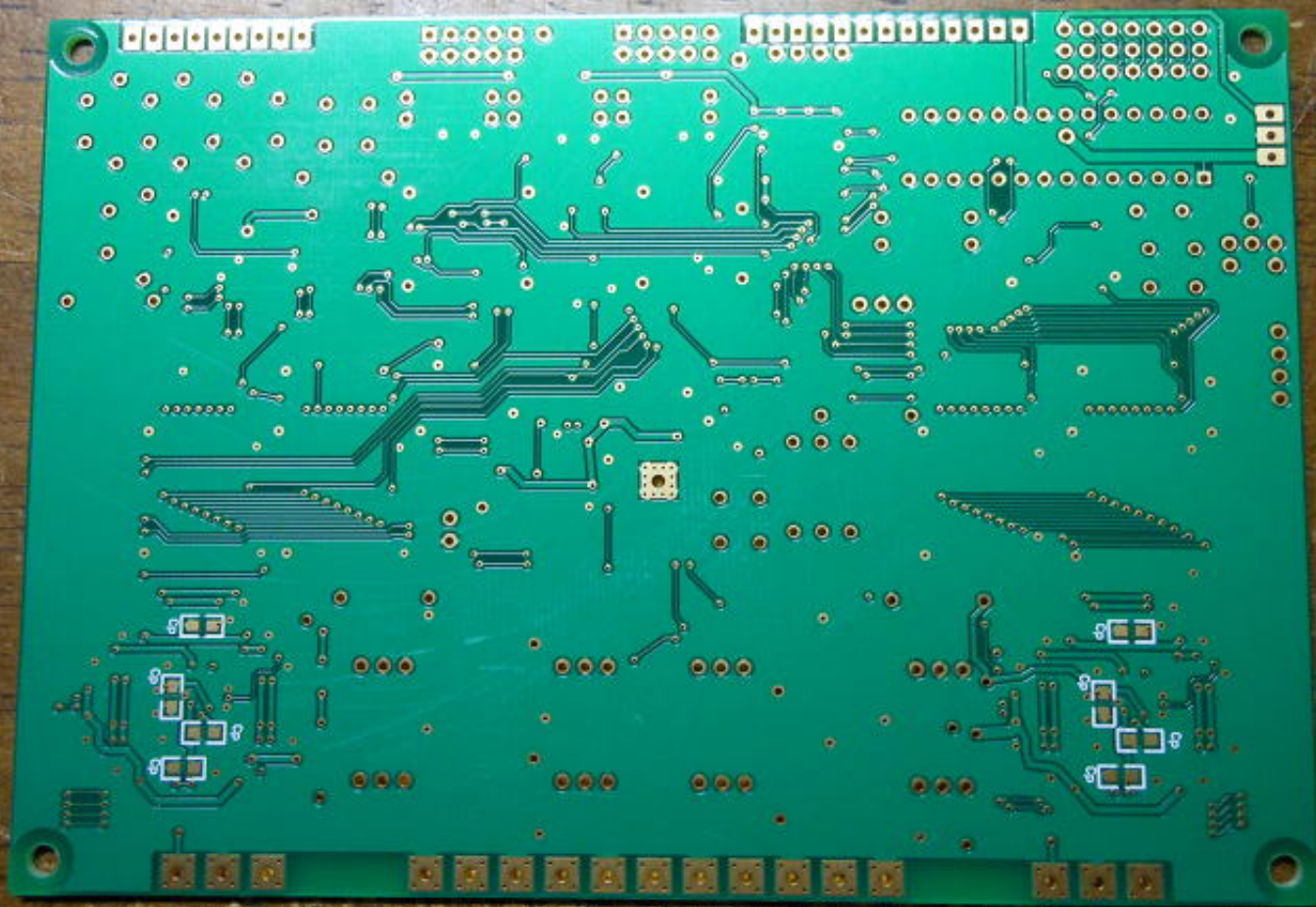
帰国したら基板が到着してました。





かなりの実装密度になりそうです。

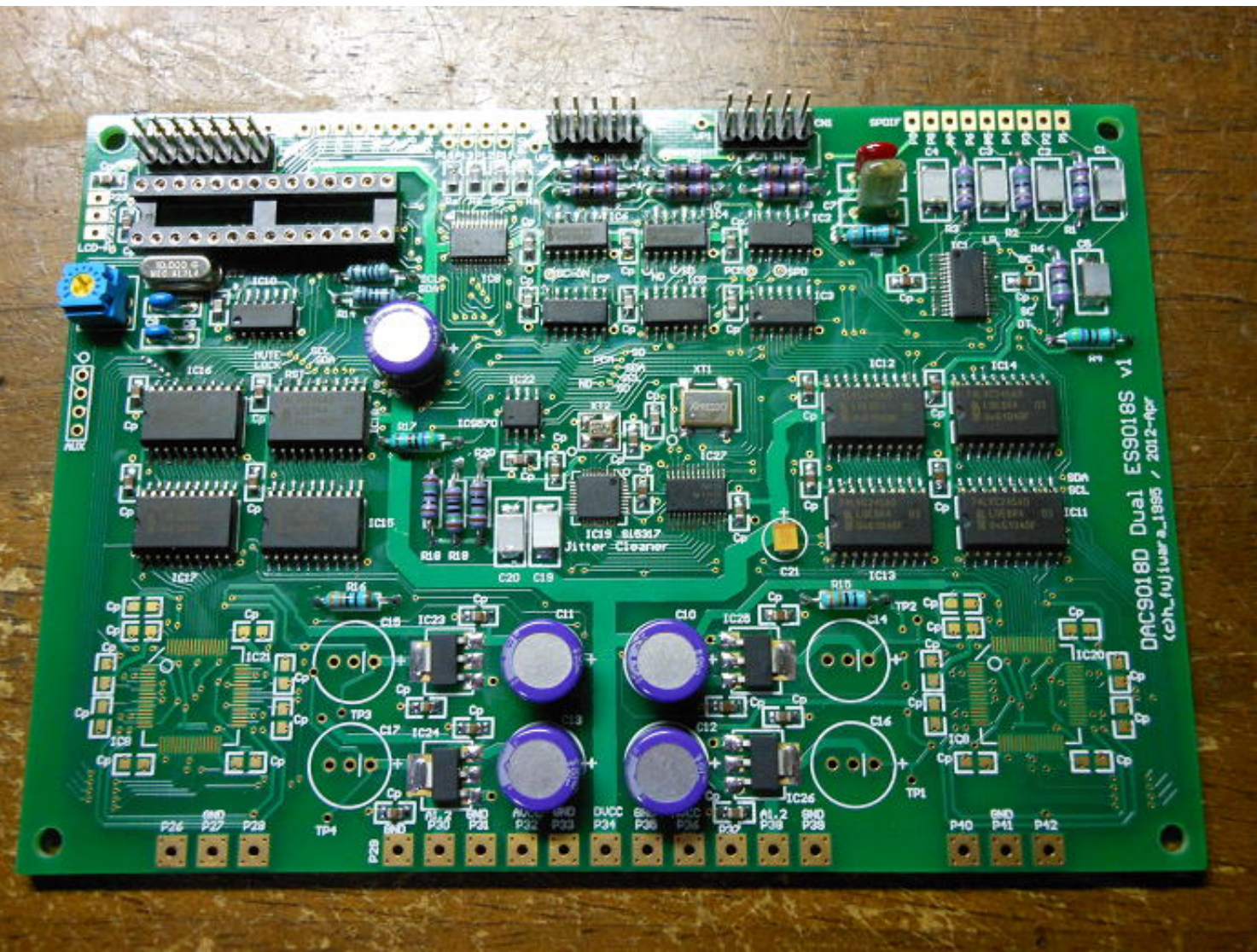




半田面がベタGND面になっています。

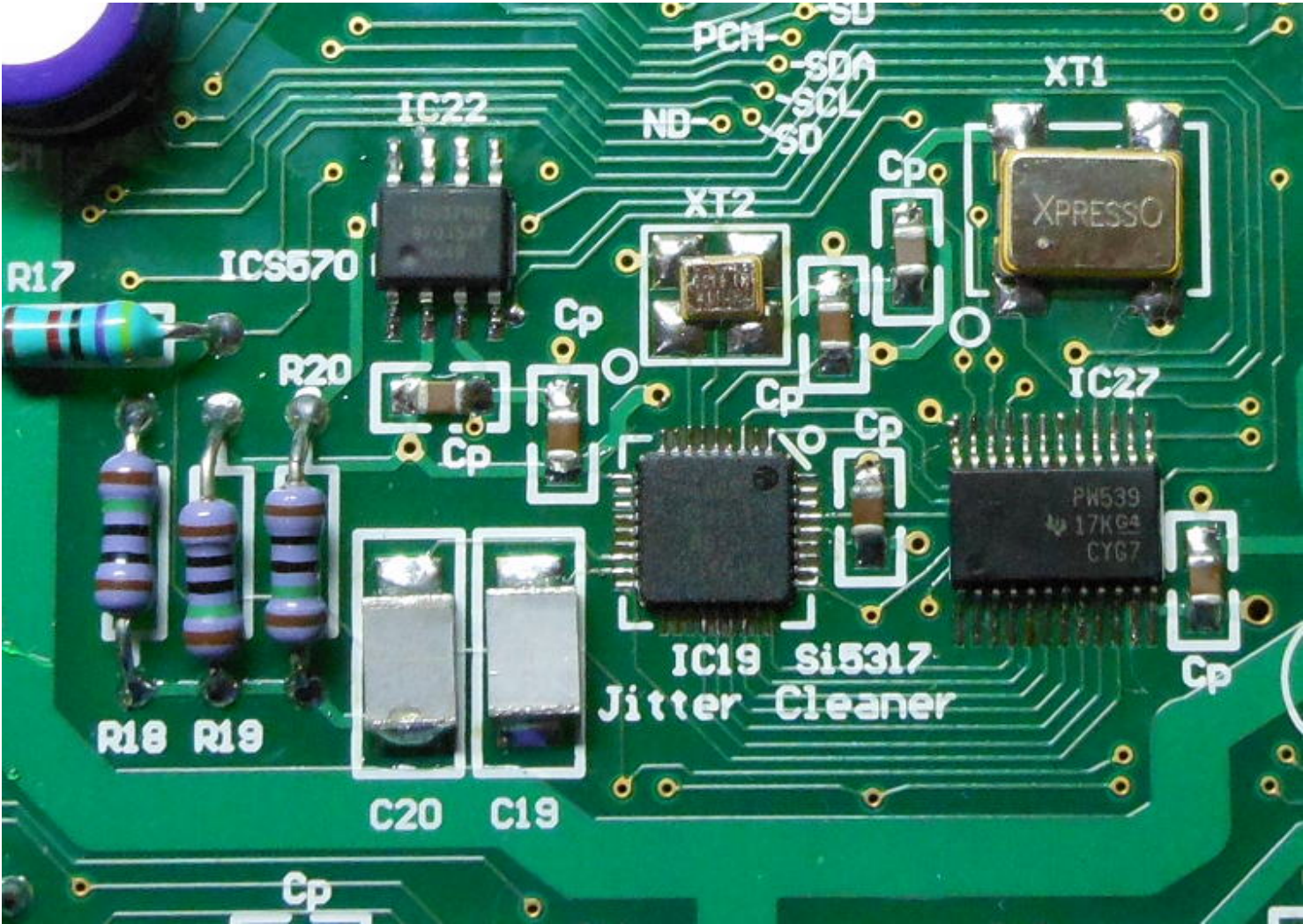
ストレス発散(?)も兼ねて半田付けです。でも、ここは冷静にキーパーツとなるES9018S以外をとりつけます。まずはこの状態で動作を確認していきましょう。フィルムコンの一部は表面実装部品にも対応できるようにしたので、手持ちのSMDパーツをとりつけました。基板の背の高さをできるだけ抑えようと考えています。





DAC素子以外は半田付け完了です。





この基板の特徴のひとつのシステムクロック用のジッタクリーナ部分です。

以下は備忘録です。

PICマイコンのIO定義

A0(AN0)	I	P21-EVOL	B0	O	LCD-D6	C0	I	BCK(1/64)
A1	IO	AUX1	B1	O	LCD-D7	C1	IO	SCL(I2C)
A2	IO	AUX2	B2	O	LCD-D4	C2	IO	SDA(I2C)
A3	IO	AUX3	B3	O	LCD-D5	C3	O	ES9018S RESET

A4	O	LCD-E	B4	I	P18	C4	O	CS8416/IO-RST
A5	O	LCD-RS	B5	I	P17	C5		N.C
A6		XTAL	B6	I	P16	C6	I	ES9018S RIGHT LOCK
A7		XTAL	B7	I	P15	C7	I	ES9018S RIGHT MUTE

IOエクспанダ0のIO定義(adrs=0xea)

P00	O	BCK-ON	P10	O	SELECT DATA CONVENTIONAL DSD
P01		NC	P11	O	SELECT DATA SPDIF
P02		NC	P12	O	SELECT DATA SHIFT-PARA DSD
P03		NC	P13	O	SELECT DATA PCM
P04	I	P11	P14	O	SELECT BCK CONVENTIONAL DSD
P05	I	P12	P15	O	SELECT BCK SHIFT-PARA DSD
P06	I	P13	P16	O	SELECT BCK SPDIF
P07	I	P14	P17	O	SELECT BCK PCM

IOエクспанダ1のIO定義(ジッタクリーナ周辺)(adrs=0xe8)

P00	IO	SI5317 FRQSEL3	P10	I	ES9018S LEFT MUTE
P01	IO	SI5317 FRQSEL2	P11	I	ES9018S LEFT LOCK
P02	IO	SI5317 FRQSEL1	P12	IO	ICS570 #6
P03	IO	SI5317 FRQSEL0	P13	IO	ICS570 #1
P04	IO	SI5317 BW1	P14	O	100MHz XTAL CONTROL
P05	IO	SI5317 BW0	P15	O	SI5317 SFOUT0
P06	I	SI5317 LOL(LOSS OF LOCK)	P16	O	SI5317 RESET
P07		NC	P17	O	SI5317 LOS(LOSS OF SIGNAL)

DEVICE ADRS  
 ES9018 RIGHT CHANNLE 0x90  
 ES9018 RIGHT CHANNLE 0x92  
 CS8416 0x20

# 一気にくみたてちゃえ！ 2012.4.28

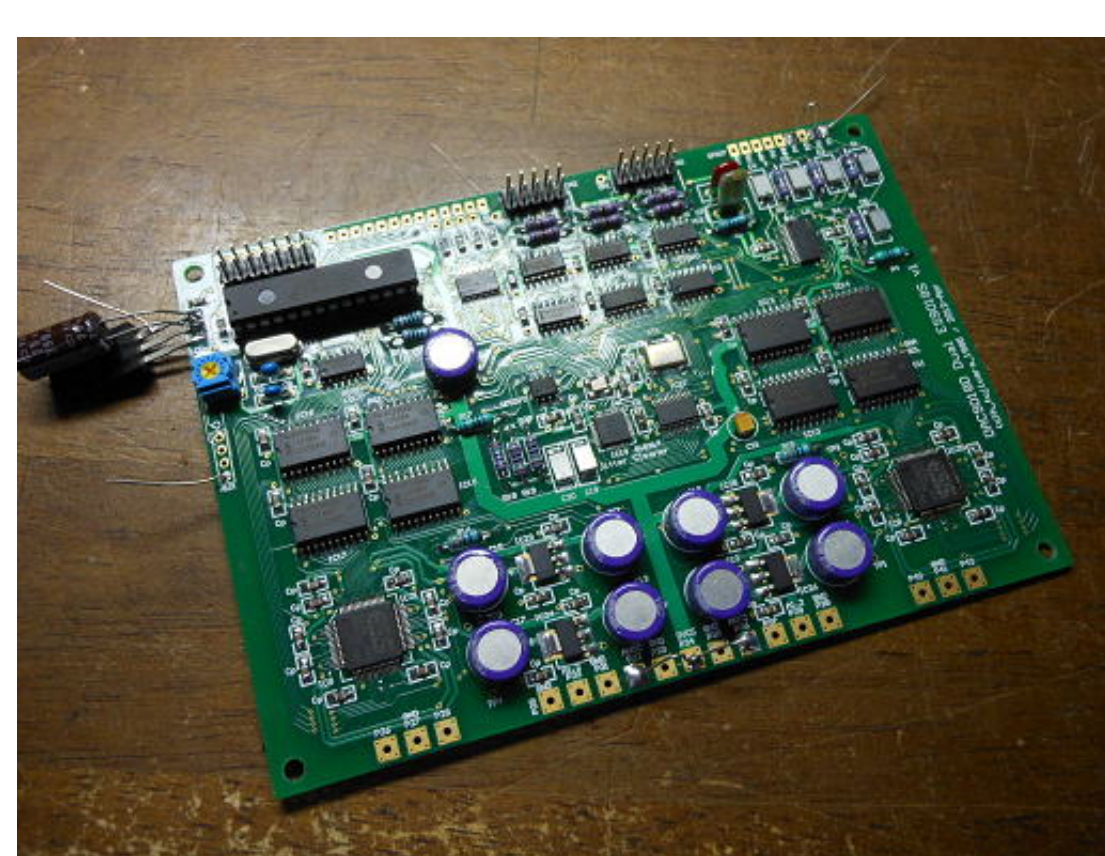
大型連休に突入です。何日かは仕事の都合で出勤しないといけなく、まあ出来るときに作業を進めましょう。どうせES9018を実装しないど、最終の動作確認ができないので一気にとりつけました。



とりつけたES9018S. 急いでいるとはいえ半田付けの確認は入念に。

基板が一通り完成しました。この基板は基本的には3.3Vの単一電源で動作します。LCDも3.3V用のモノを用いたところですが、手持ちに5Vのものしかないので、3.3Vの電圧レギュレータを基板端子に接続して、LCDには5V、基板には3.3Vの電源を供給するようにしました。



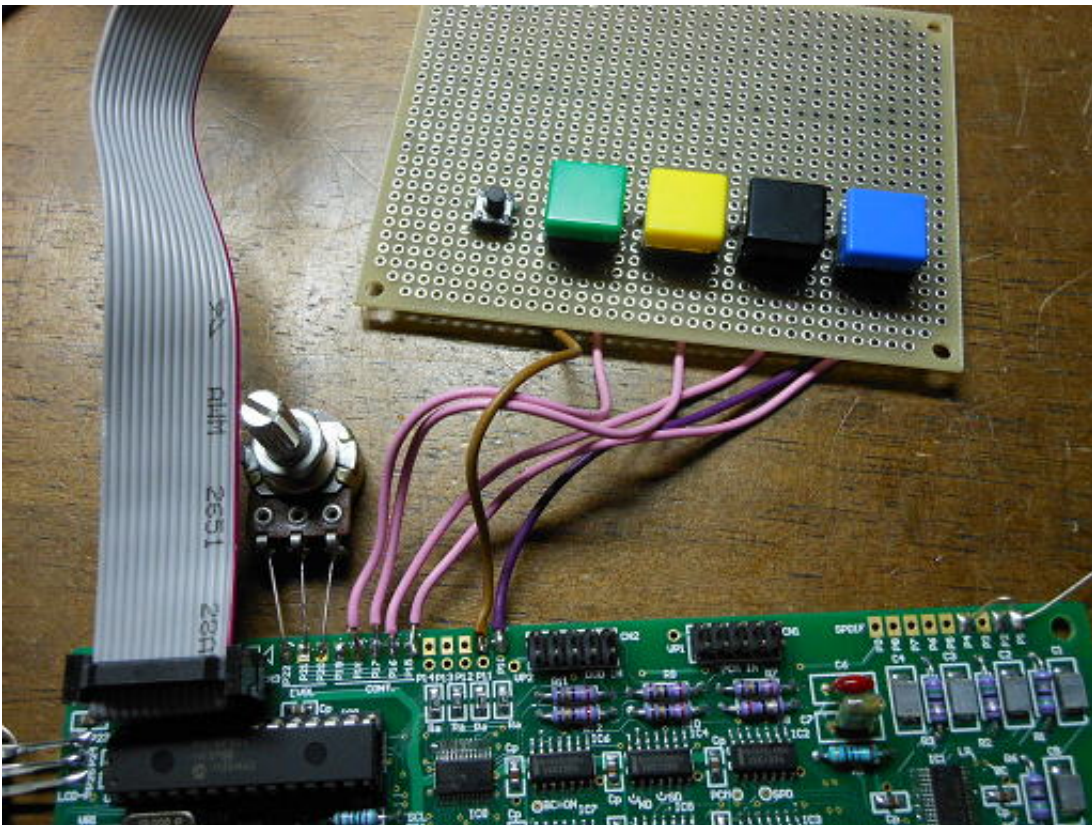


一応完成した基板です。

## 動作確認！

どんどん動作確認をしていきましょう。まずはソフトで色々と項目が切り替えられるようにスイッチをとりつけました。あとは電子ボリューム用のVR, そしてLCD用のフラットケーブルです。



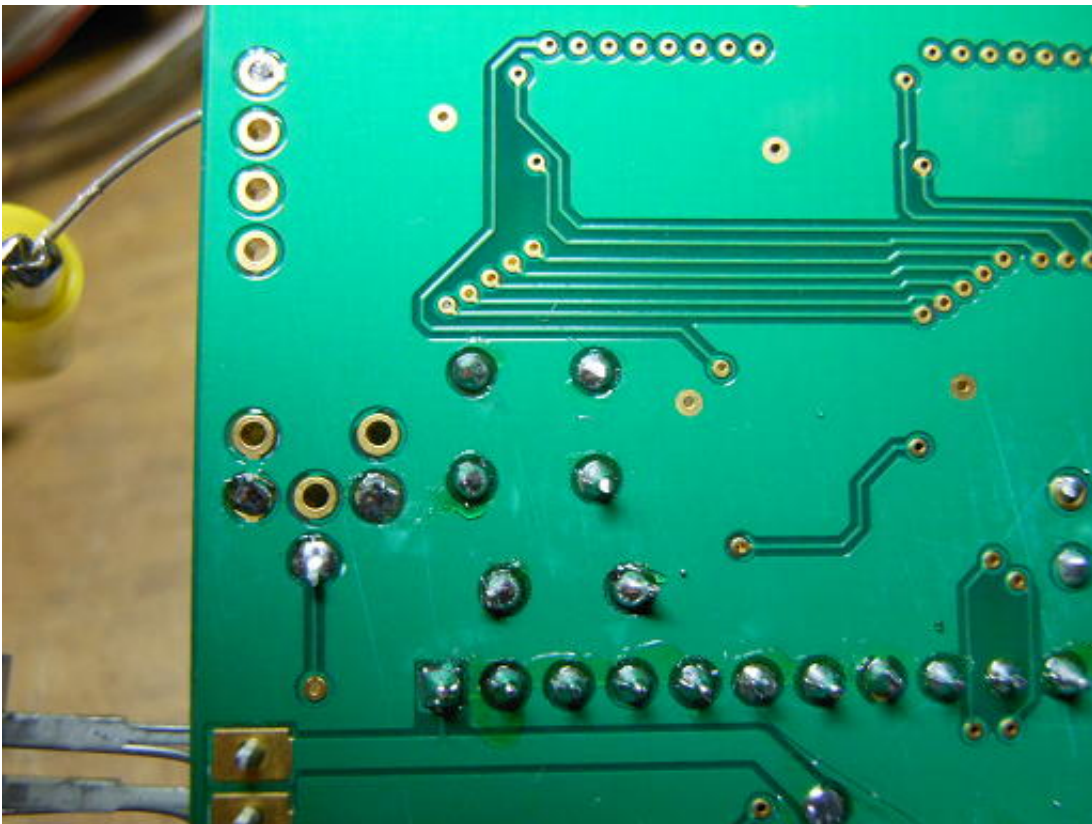


周辺のスイッチ類を取り付けて動作確認の準備が進行中です。

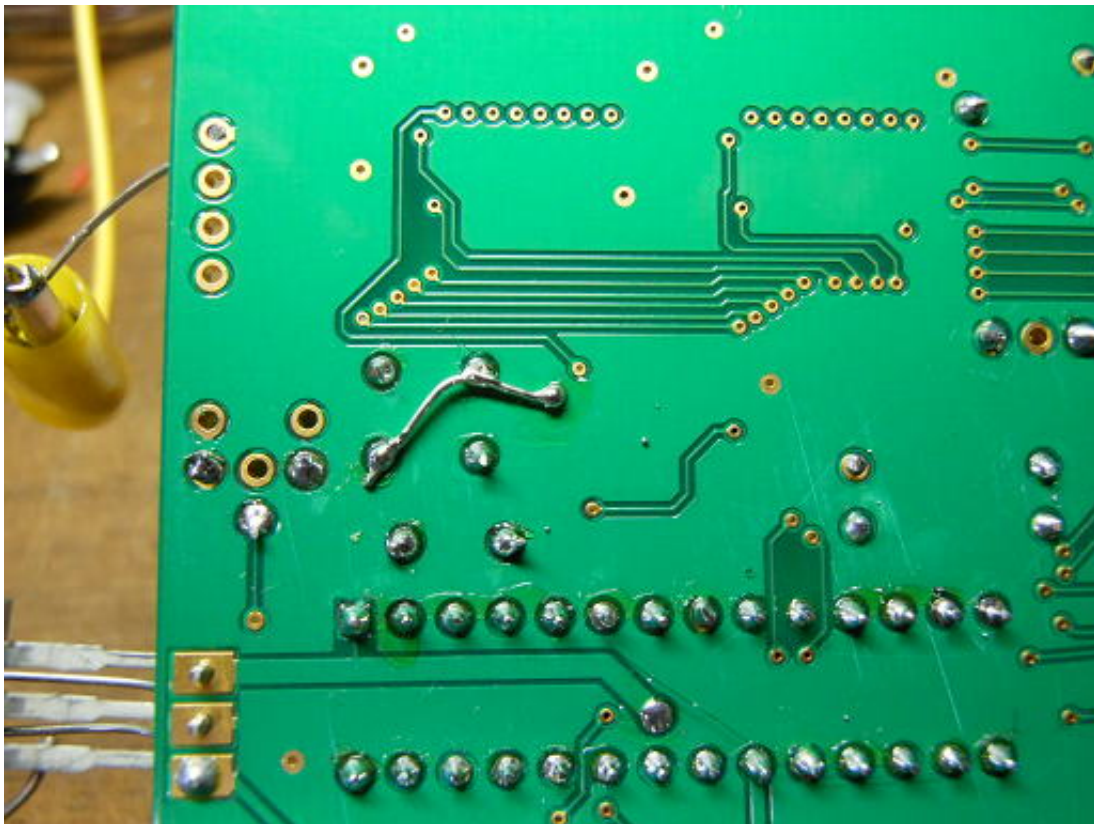
間違い箇所は一杯でるでしょうけど、まあそれは想定範囲内です。

## 間違いその1.

偶然発見しました。基板の裏を眺めていてベタに接続されていないランドが固まってあったので、確認したら水晶発振子に接続されるコンデンサ(22pF)の片側がGNDにおちいていませんでした。これは修正は簡単です。近くにあるGNDのパッドへ接続して修正しました。



修正前(2箇所がGNDに接続されていない)



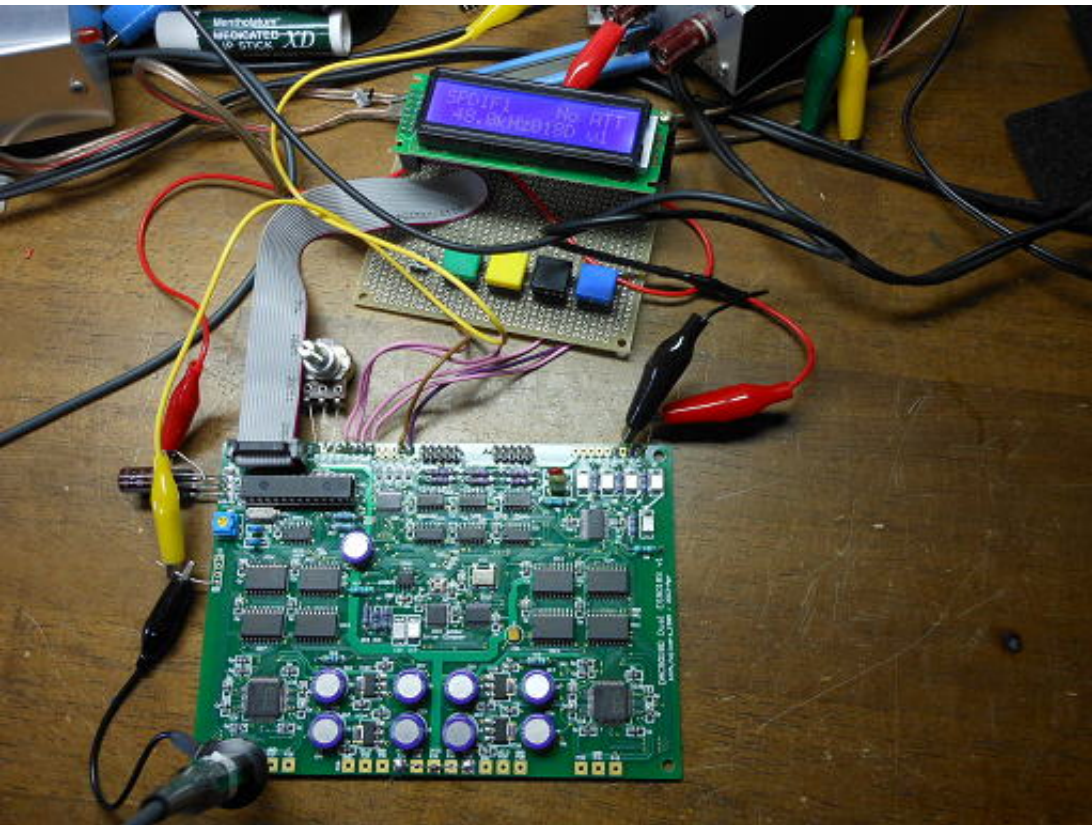
修正後

## まずはSPDIF入力から

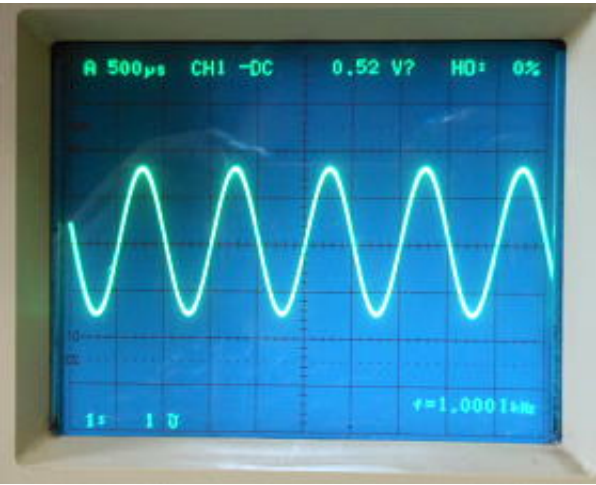
簡単なソフトを書いてSPDIF入力の確認です。簡単といっても、入力アンプ用につかったCS8416の制御コードもありますし、IOエクspandaやES9018の制御コードも含むので、結構なボリュームになります。

プログラムを書いたPICを基板に差し込んで、電源ONです。いつでも緊張の瞬間ですね。でも、波形がでたときはものすごくうれしいです。自作ならではの喜びです。





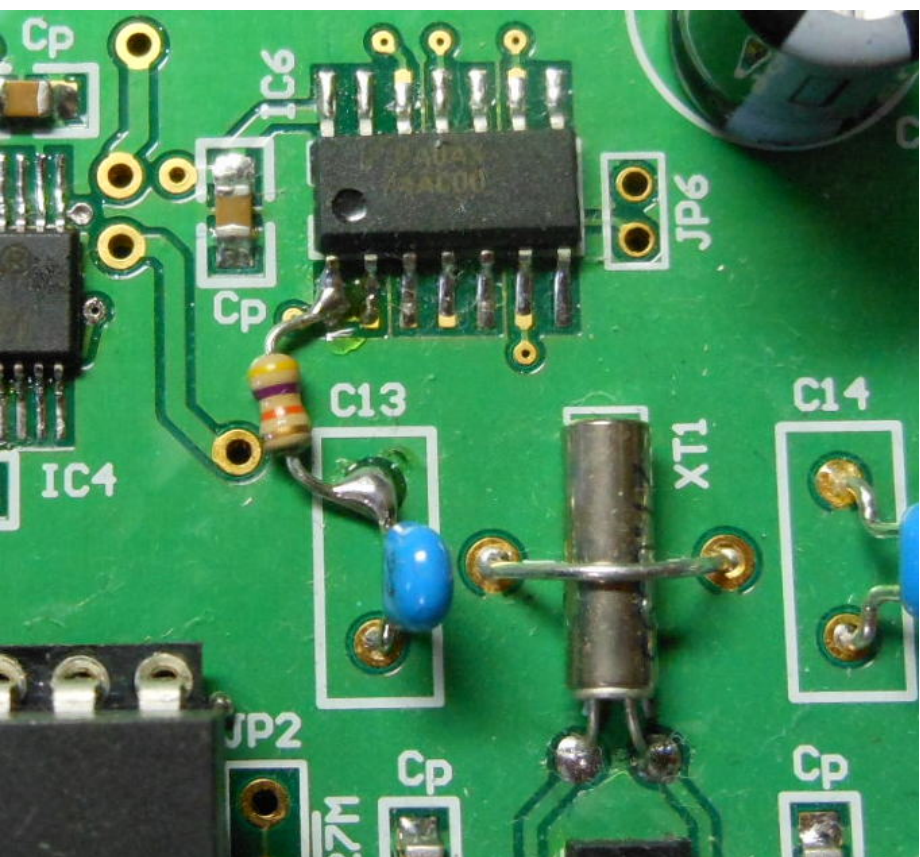
SPDIFの入力での動作確認できました。



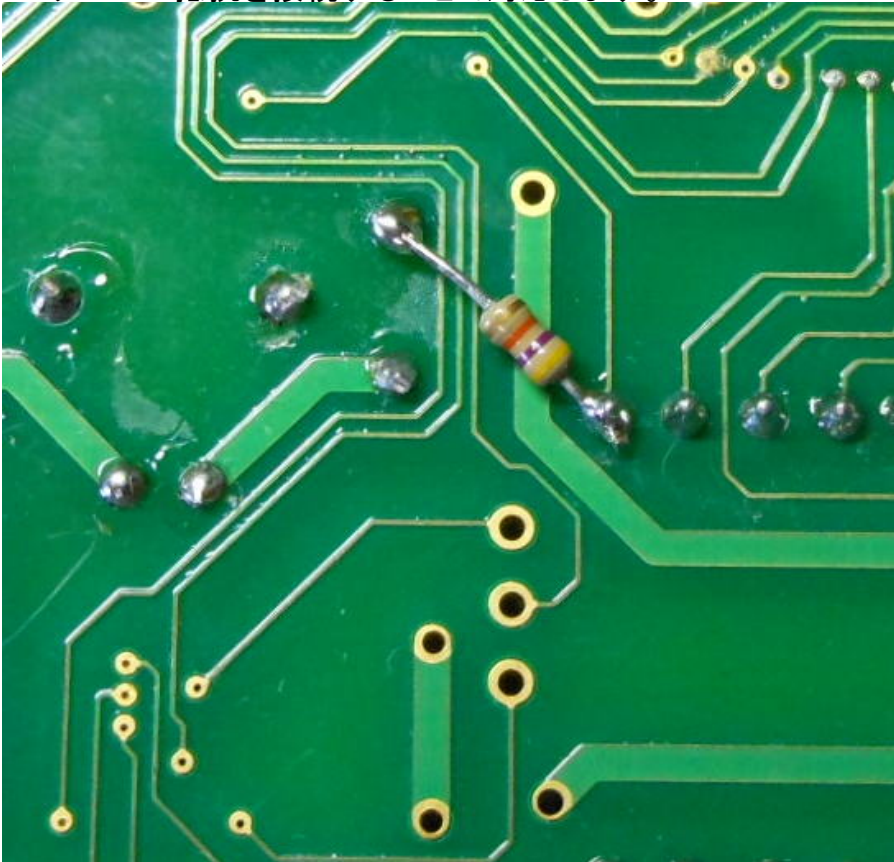
無事に信号がでていますね。

つぎはPCM信号(I2S)での確認

PCM信号の生成にはFFASRCを活用します。このFFASRCですがBBSで出力周波数が1.5倍になるという報告もありました。その動作の不安定さの解消にはPICマイコンのPIN21をGNDへプルダウンすることが有効なので、私の基板でもその対処をしました。修正例を下記の2点にあげています。部品面から修正する場合はIC6のPIN1あるいはPIN2から47KΩ程度の抵抗で近くのGNDパターンに接続します。裏面から修正する場合は、PICのPIN21から近くのGNDパターンへ抵抗を接続することで対応します。

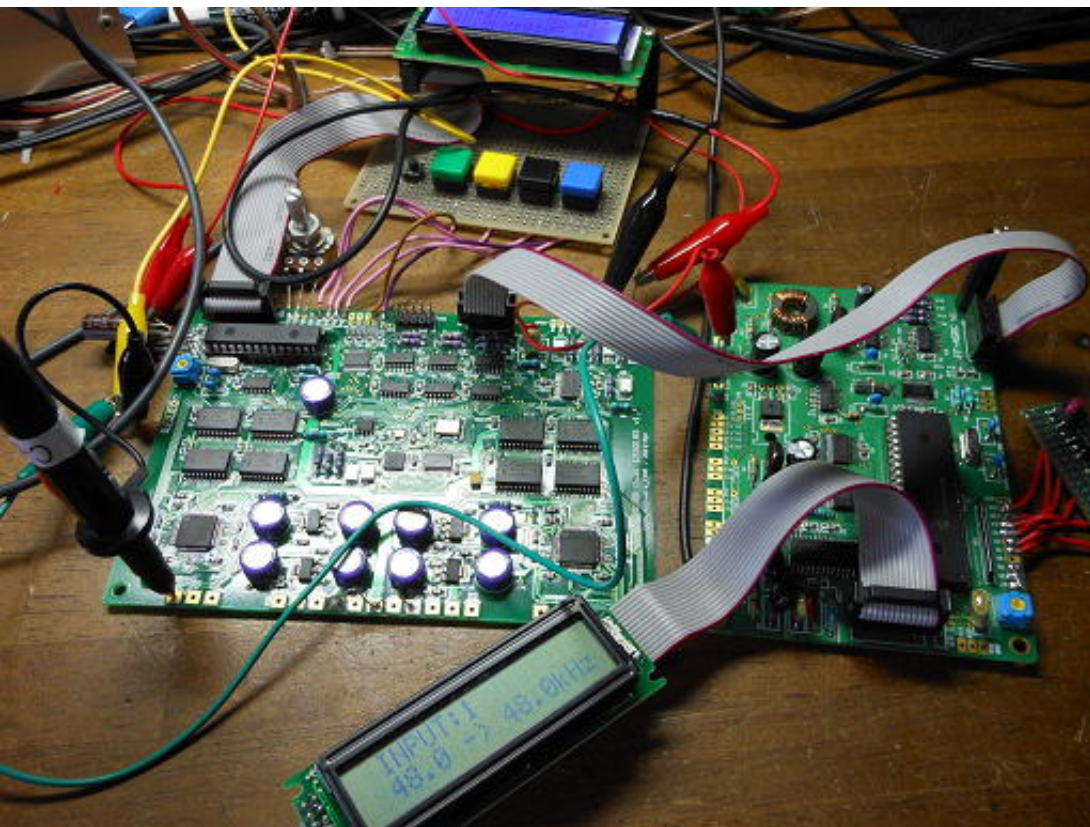


FFASRCの修正例。部品面から行う場合。



FFASRCの修正例。半田面(裏面)から行う場合。



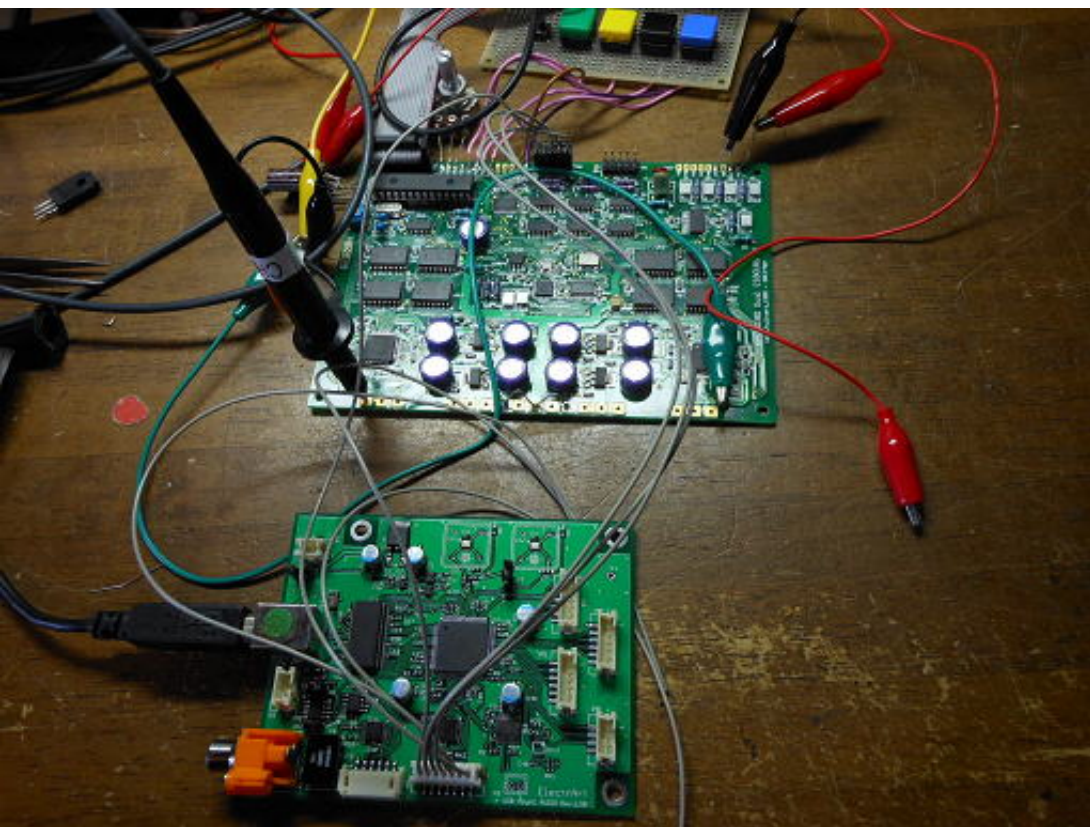


FFASRCと接続してPCM入力の確認中の様子。

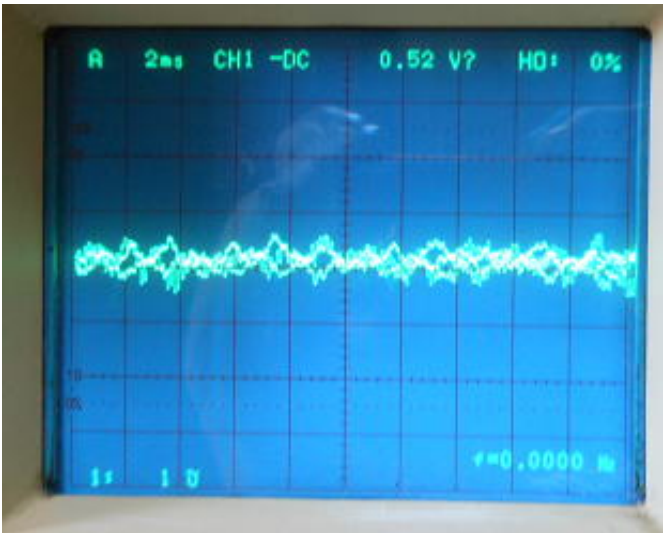
PCM信号入力での動作もあっさり確認できました。

### DSDでの入力確認！

DSDの出力にはエレアトさんのUSB-AUDIOをつかいます。同様に、FFASRCから接続を変更して動作確認です。こちらは無事、動作することが確認できました。



DSD入力での動作確認中の様子。

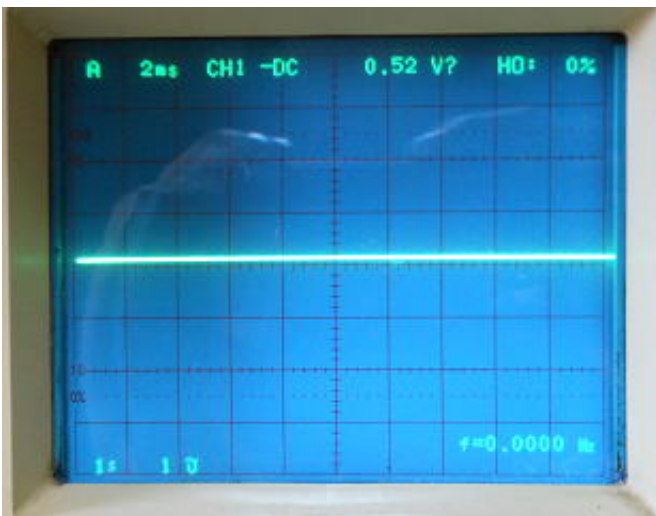


DSD入力時でも動作確認できました(音楽信号入力)

# 最後はシフトパラDSDでの確認です。

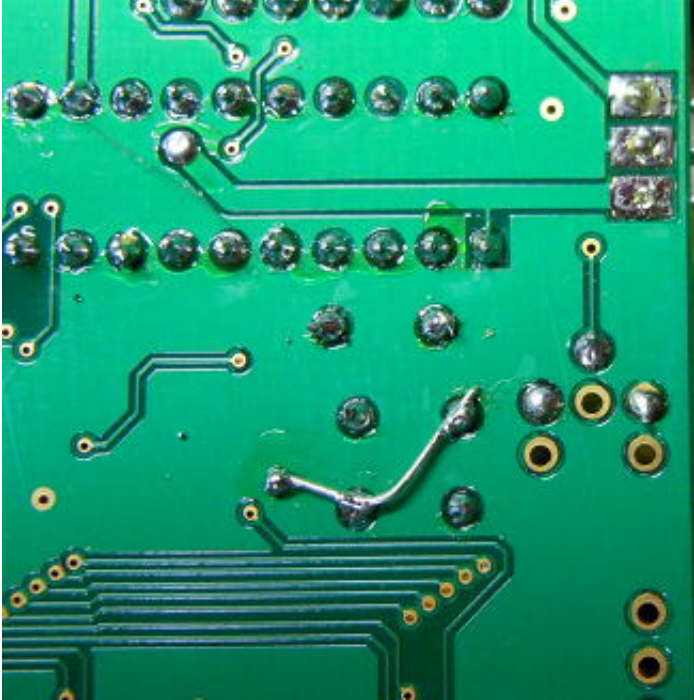
左チャンネルはすぐに動作していることを確認できましたが、右チャンネルが音沙汰なしです。  
あれ、何を間違えたのかな？調べてみるとシフトレジスタの74164の出力がでていません。  
確認したら電源がきていませんでした。あちゃ、ボンミスです。



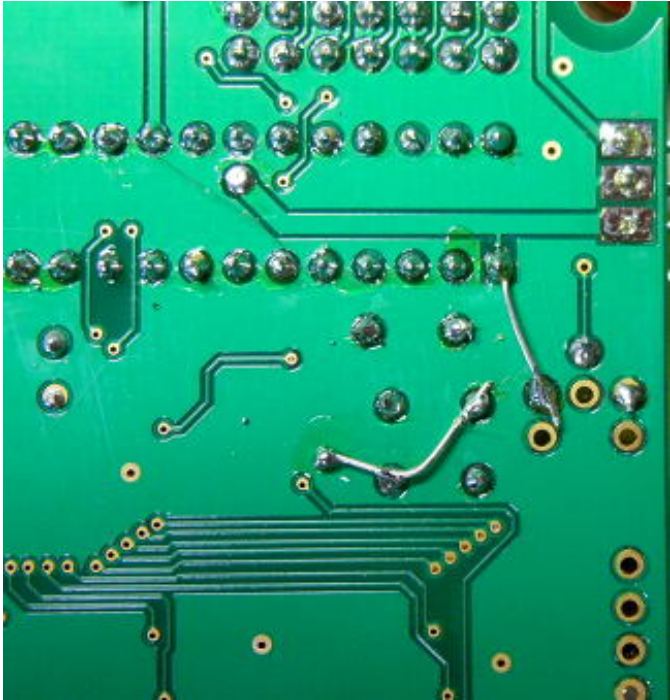


シフトパラDSDでは左チャンネルから出力がでない……

でも修正は簡単です。74164のVCCから接続されているところを追って行って、簡単にジャンパーできそうなところを探し、そこでジャンパーしました。



修正前



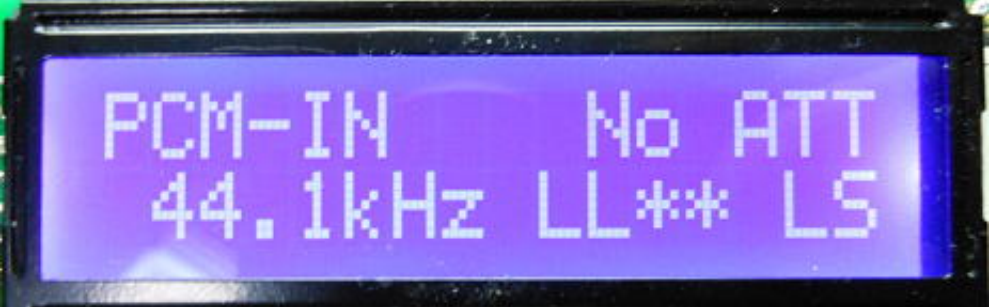
修正後。1本ジャンパー線が増えています。

これでシフトパラDSDも動作することを確認です。

つぎはシステムクロックの通倍器とジッタクリーナの動作確認です。  
ちょっとしたプログラムを書かないといけないので時間かかりそうです。  
でも、どんどん進めましょう！

## システムクロックのジッタクリーナ動作確認

なかなか、写真であらわしにくいので割愛(笑)。  
ロックの有無の確認のためLCDに状態表示をさせるようにしました。  
左下のLL \* \* はES9018の左右のロックの状態とMUTE状態を示しています。  
おなじくLSはSi5317のLOCKおよびSIGNALの状態を示しています。



frontページの表示

さて、一気にソフトを組み上げましょう！

ちなみに、このDACはかなりおおぐらいです。3. 3Vラインの電流は0. 5Aを超えます。  
ひとえにジッタクリーナがおおぐらいですね。



かなり消費電力が大きいです。

すこし長くなってきたので[PART2](#)に続きます。

[\(後編へ\)](#)