

ADX-S キットの組み立て説明書

CRKITS.COM

2023/8/13 V1.5: 正式リリース 7バンド化

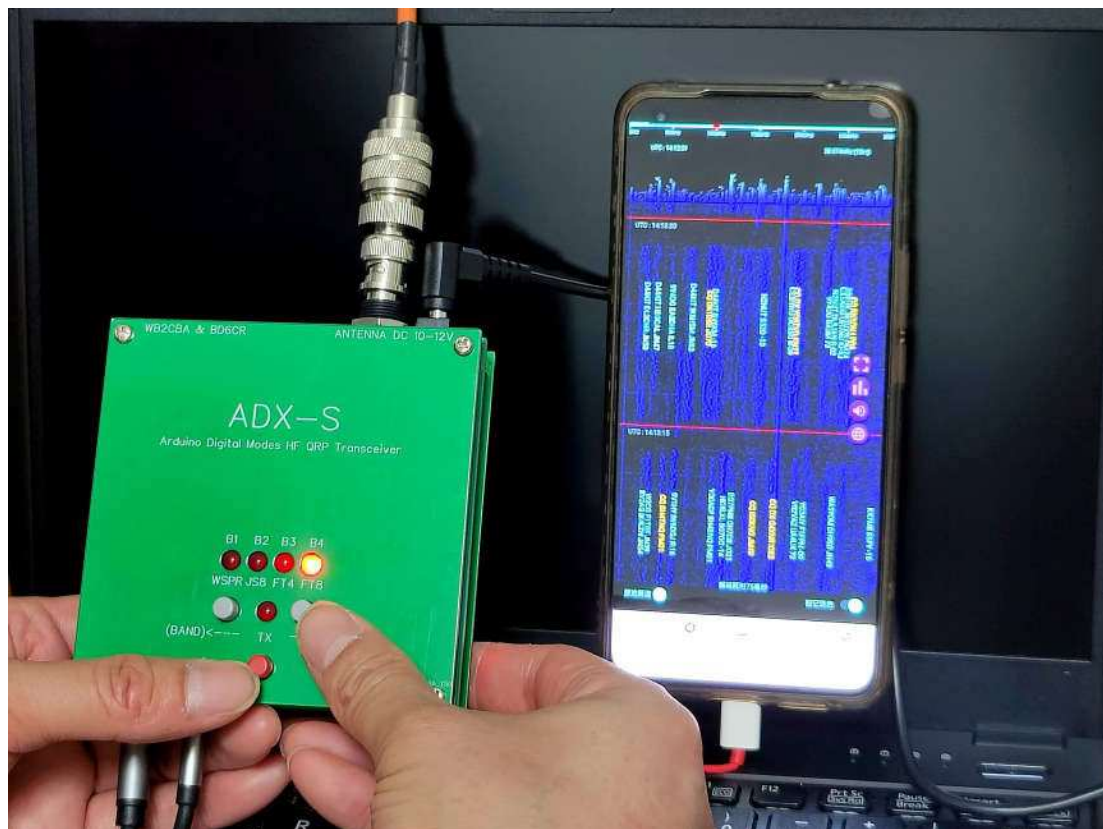


写真: Jack Liang, BG7IKK - EFHW アンテナを使用して 18,000 km 以上の DX 交信を達成した

変更履歴

2023/3/26 V1.0: 正式リリース

2023/3/29 V1.1: 正式リリース修正版

2023/4/2 V1.2: 正式リリース 修正版 2

2023/4/29 V1.3: 正式リリース、CAT 追加

2023/7/24 V1.4: 正式リリース、改造 追加

2023/7/24 V1.5: 正式リリース、7 バンド化 追加

(2023/4/1 V1.1: 日本語版 by exJA5GHK)

(2023/4/29 V1.3: 日本語版 by exJA5GHK)

(2023/7/25 V1.4: 日本語版 by exJA5GHK)

(2023/8/14 V1.5: 日本語版 by exJA5GHK)

目次

ADX-S キットの組み立て説明書.....	1
目次.....	2
仕様.....	3
部品表.....	3
動作原理.....	5
キットの組み立て.....	8
ステップ 1: ダイオード.....	10
ステップ 2: +12V 電源.....	12
ステップ 3: Arduino Nano モジュールと、+5V、+3.3V 電源.....	13
ステップ 4: CD2003 の電源.....	15
ステップ 5: 5351 モジュール.....	16
ステップ 6: LED とボタン.....	18
ステップ 7: TX ドライバ 74ACT244.....	20
ステップ 8: TX 部の組み立て.....	21
ステップ 9: オーディオジャック.....	23
ステップ 10: CD2003 受信 IC.....	24
ステップ 11: LPF バンドフィルター基板.....	30
ステップ 12: ADX-S のケース.....	32
設定とキャリブレーション.....	35
ADX-S の使い方.....	37
オープンソースファームウェアの修正とアップデート.....	39
CAT サポート.....	40
改造について.....	41
4 バンドから 7 バンド化への改造.....	43
追加ドキュメント.....	43
参考情報と謝辞.....	44

ADX (Arduino Digital Xcvr) は、uSDX の設計者の一人として有名な Barb、WB2CBA の設計による Arduino ベースのデジタルモード HF QRP トランシーバです。BD6CR はこれに変更を加え、スーパーヘテロダイン方式の ADX-S としました。受信性能が大幅に向上し、また回路の改良により、とても初心者向けのデジタルトランシーバキットになりました。組み立てはほぼ 3 時間で終わります。また、最低限測定器として周波数カウンタが必要ですが、アマチュア無線のトランシーバでも OK です。ADX-S は、PC 上では WSJT-x または JTDX を、Android スマートフォンやタブレットでは FT8CN を使用可能です。WSPR TX の場合、Android アプリの WSPR TX も使用可能です。

仕様

寸法: 97 mm x 97 mm x 22 mm (突起部を除く)

重量: 150g、LPF バンドフィルタ基板 1 個を含む

電源: 10-12V(DC IN ジャック) または 5V(USB ポート)

電源電流(受信時): 12V 時 約 60mA

電源電流(送信時): 12V 時 約 600mA (15m 帯、10m 帯では低下)

送信出力: 12V 時 約 3.5W、5V 時 約 0.3W (15m 帯、10m 帯ではこの約 1/2)

スプリアス抑圧比: 約 -45dBc

使用可能モード: WSPR、JS8、FT4、FT8 を含むデジタルモード、上側側波帯(USB)

使用可能バンド: 40m、20m、15m、10m、LPF バンドフィルタ基板の取り換えが必要

オーディオコネクタ: MIC (サウンドカードの MIC ジャックへ)、SPK (サウンドカードの PHONE または SPK ジャックへ)

部品表

最初に構成部品リストに従って構成部品を確認し、不足している部品があれば連絡してください。

メインボード

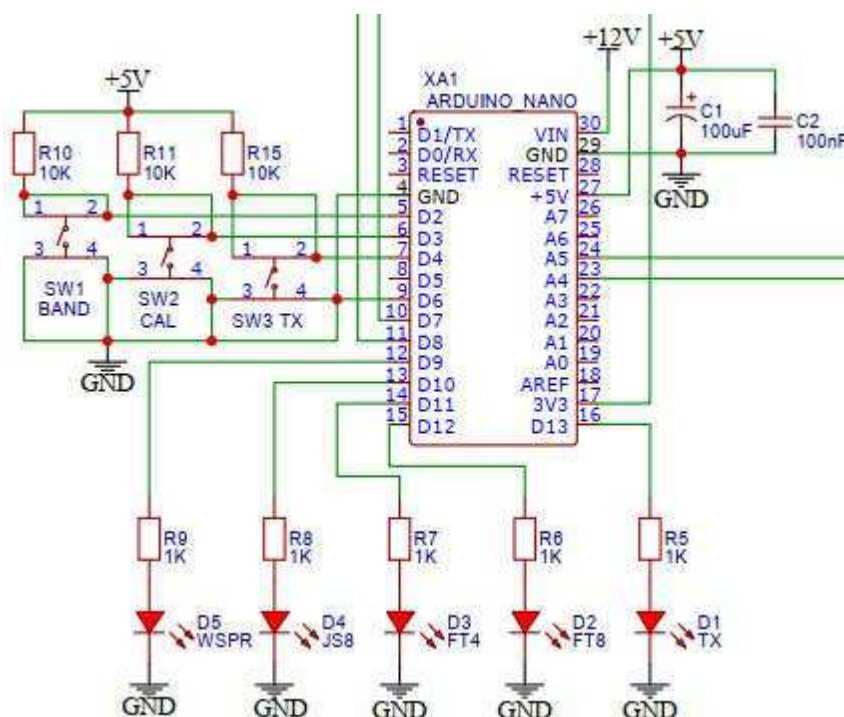
部品	数量	部品番号
100μF/16V	3	C1,C16,C24
10μF/16V	2	C13,C18
560pF, 620pF or 680pF	1	C11
10nf	4	C4,CS,C12,C17
100nF	13	C2,C3,G6,C8,C9,C10,C14,C15,C19,C20,C21,C22,C23
470nF or 1μF	1	C7
5.1pF	1	C25
PFB455JR セラミックフィルタ	1	FL1
SPK 3.5mm オーディオコネクタ	1	CON1
MIC 3.5mm オーディオコネクタ	1	CON2
TX 5mm 赤 LED	1	D1
FTS 5mm 赤色 LED	1	D2
fT4 5mm 赤色 LED		D3
1S8 5mm 赤色 LED		D4
WSPR 5mm 赤色 LED		D5
1N4148	3	D6,07,08
1N5817	1	D9
43-47V 1W ツェナーダイオード	1	終段の保護用. テストポイント RFO と GND 間に接続
BNC	1	J1
DC-005	1	J2
1μH	1	L
100μH	1	L2
8S170	4	Q1,Q2,Q3,.,Q4
1M or 470k	1	R1
2.7k	1	R2
100	1	R14
4.7k	1	R2
10k	5	R3,R4,R10,R11,R15
1k	6	R5,R6,R7,R8,R9,R13
UP(BAND) タクトスイッチ 灰色	2	SW1
DOWN(CAL) タクトスイッチ 灰色		SW2
TX タクトスイッチ 赤色	1	SW3
CD2003GB_GP	1	U1
SI5351_モジュールまたは同等品	1	U2
74ACT244	1	U3
IPF JBANO_MODULE ソケット	2	U4_A,U4_B
Arduino Nano または同等品 (プログラム書き込み済)	1	XA1
メイン PCB	1	
上面パネル PCB	1	
底面パネル PCB	1	
ネジ	4	
スタンドオフ M3x12+6	4	
スタンドオフ M3x6 6	4	
スタンドオフ M3x5	4	

バンドフィルター部

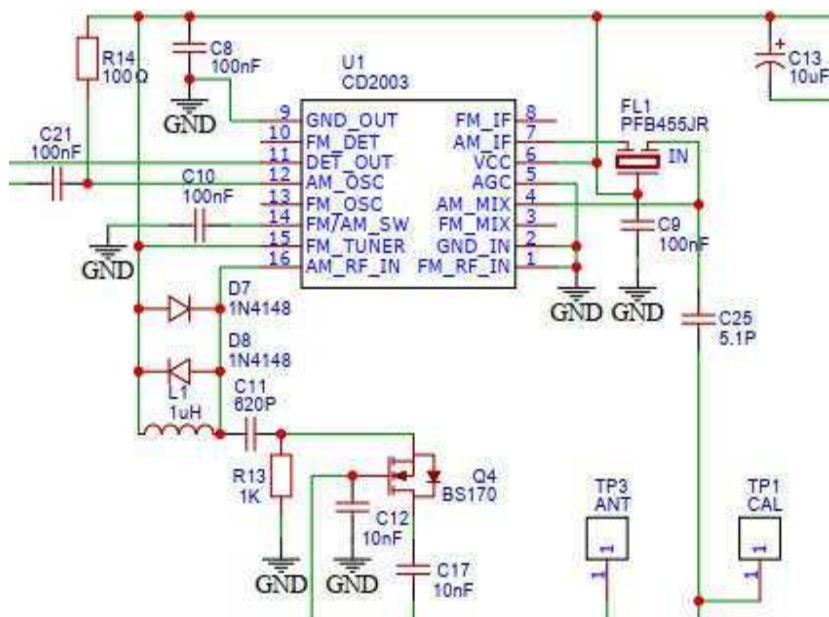
LPF バンドモジュール										
バンド	C1	C2	C3	C4	C5	C6	C7	L1	L2	L3
40	120+10	470	470	1000	300	1000	100n	12T/黒	15T/赤	10T/赤
20	68	470		470	150	470	100n	12T/黒	10T/赤	7T/赤
15	68	300	15	300	100	300	100n	12T/黒	11T/黄	7T/黄
10	39	240		240	75	240	100n	12T/黒	9T/黄	6T/黄

動作原理

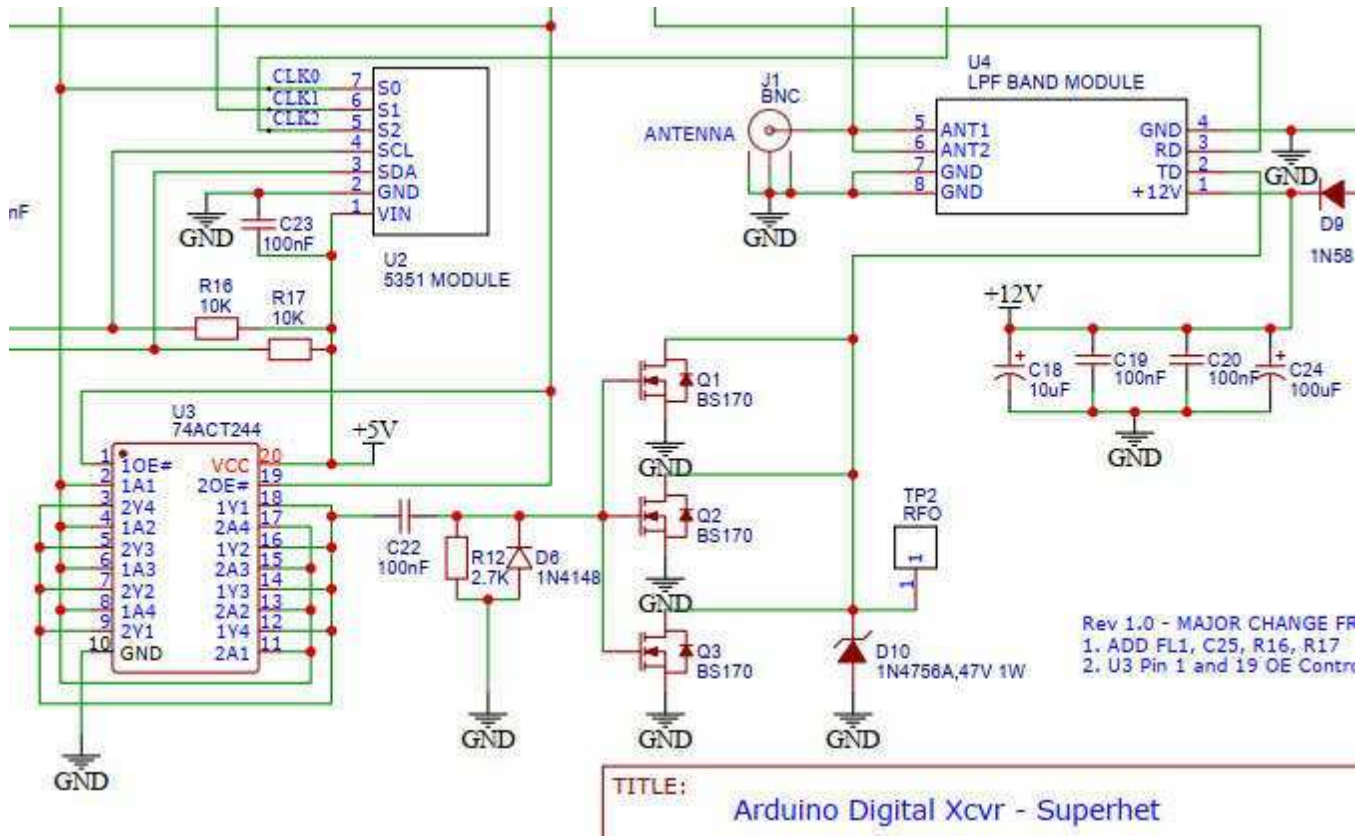
Arduino Nano モジュール (XA1) は制御コアです。ボタンへの応答、LED の駆動、5351 モジュールを駆動などの処理の他に、SPK インターフェイスからのオーディオ信号の周波数を識別して FSK 変調を行います。したがって、ADX-S の送信信号は D4D のような両側波帯ではなく、単側波帯です。Arduino Nano は、受信信号に対して追加の処理を行わないので、受信は純粋にアナログです。Nano はメインボードに安定した 5V を供給します。



5351 モジュール (U2) は、I2C インターフェイスを介して Arduino Nano によって制御されます。S0-S2 から異なる周波数を持つ 3 つのチャンネルの方形波を出力します。そのうち、S0 は送信信号、S1 は受信局発振器、S2 は受信局発振器です。キャリブレーションモードでは、S2 はキャリブレーション用の CAL テストポイントに 1MHz の信号を出力します。



ADX-S の送信部は ADX のオリジナル設計と同じです。5351 モジュール (U2) の CLK0 信号は 74ACT244 (U3) により駆動および増幅され、最終的に 3 個の BS170 パラレル接続で構成されるクラス E モードの出力段へ送られます。そして、LPF バンドフィルタ基板 (U4) でフィルタされ、J1 からアンテナへ出力されます。送受信の切り替えは、最初に CLK0 または CLK1+CLK2 の出力切り替え、それに RX フロントエンドのための RXSW ラインによる切り替え、さらに 74 ACT244 #OE ピンによる送信部の動作/停止、により行います。



ADX-S のマン・マシン インターフェイスは、5 個の LED と 3 個つのキーを使用しており、ハードウェアの点では ADX と一致しています。このうち 4 つの LED は、バンド (B1 ~ B4) とモード (WSPR、JS8、FT4、FT8) を同時に表示します。

バンドは電源投入時の点滅で、モードは固定点灯により表示します。モードを切り替えると、周波数テーブルに基づいた周波数に切り替わります。TX LED は TX 時、またバンド設定モード時に点灯します。通常の動作モードでは TX ボタンによりマニュアル送信しますが、バンド設定モードでは現在の設定の確認と保存のために使用します。ADX と ADX-S の主な違いは、バンドの変更方法です。通常モード時に 2 個のボタンを同時に押すのではなく、電源投入時に BAND ボタンを押すことにより切り替えます。

キットの組み立て

いよいよキットの組み立てを始めましょう。このキットはすべてリード部品ですので、目に優しいでしょう。パッドが小さいので、先の細いはんだゴテを使用されることをお勧めします。ハンダは帰途に含まれていません。太さ 0.8mm、錫(Sn) 63%、鉛(Pb)のヤニ入りハンダを用意してください。



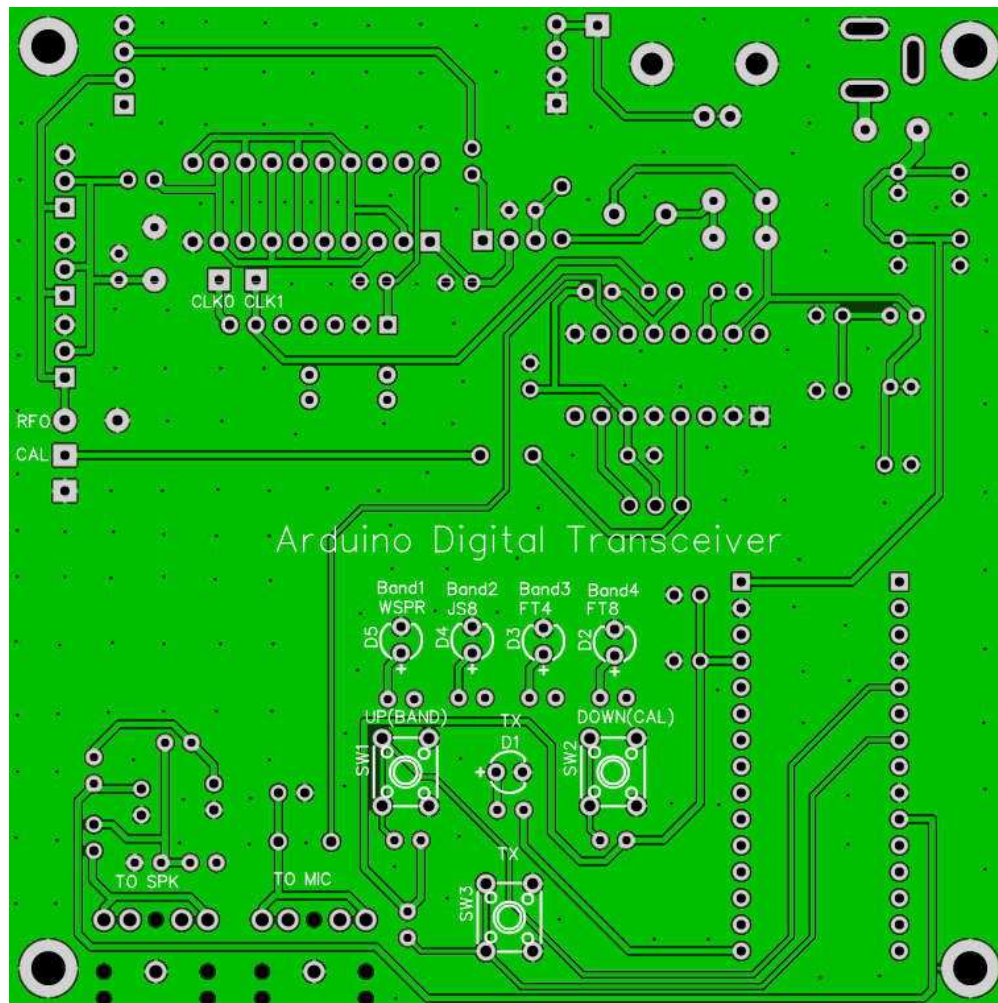
キットはプラスチックケースに入っています。ケース内には、メイン基板、底面カバー基板、メイン基板の部品一式、4 個のバンドフィルタ用部品とオーディオケーブル一式、が収められています。

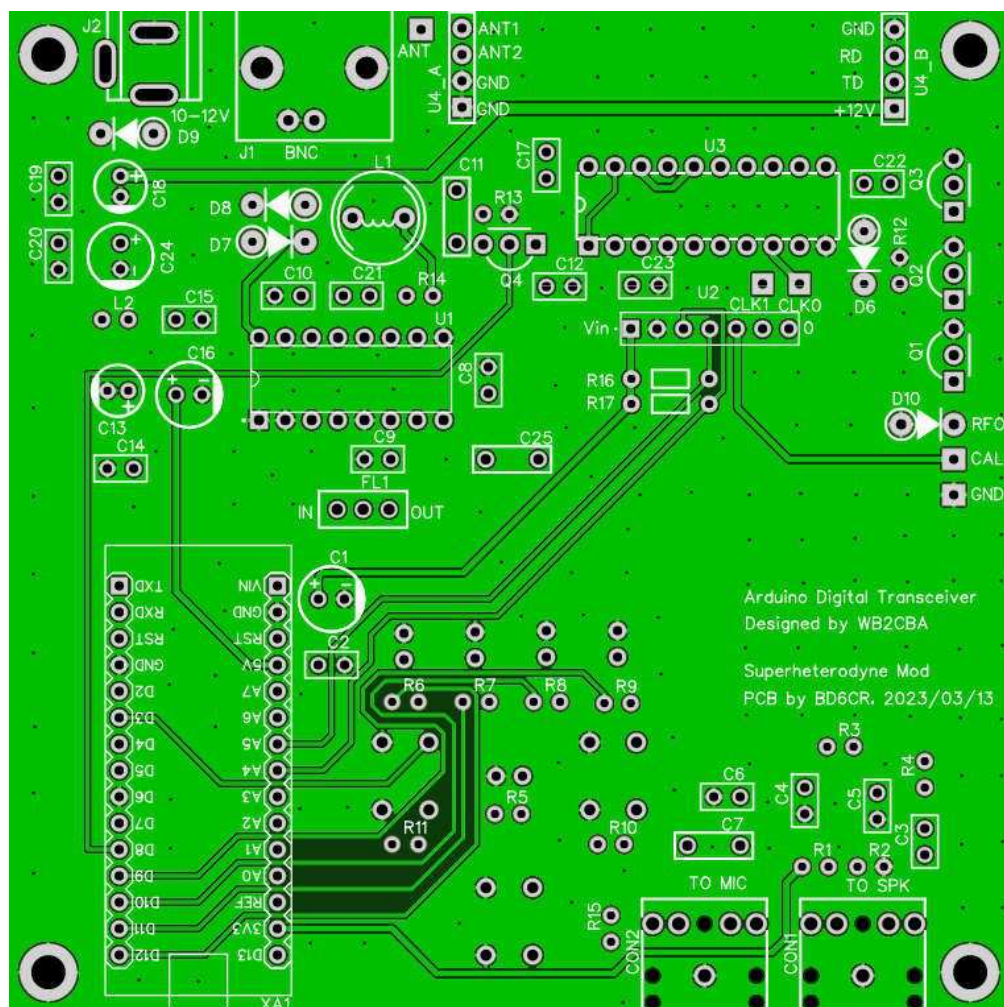


メインボードにとりかかりましょう。完全な回路図と部品表については、IO グループの File セクションからダウンロードしてください。メンバーでない場合は、登録が必要です。

<https://groups.io/g/crkits/files/ADX-S>

次に現在の PCB の写真を示します。リビジョンの日付を確認してください。

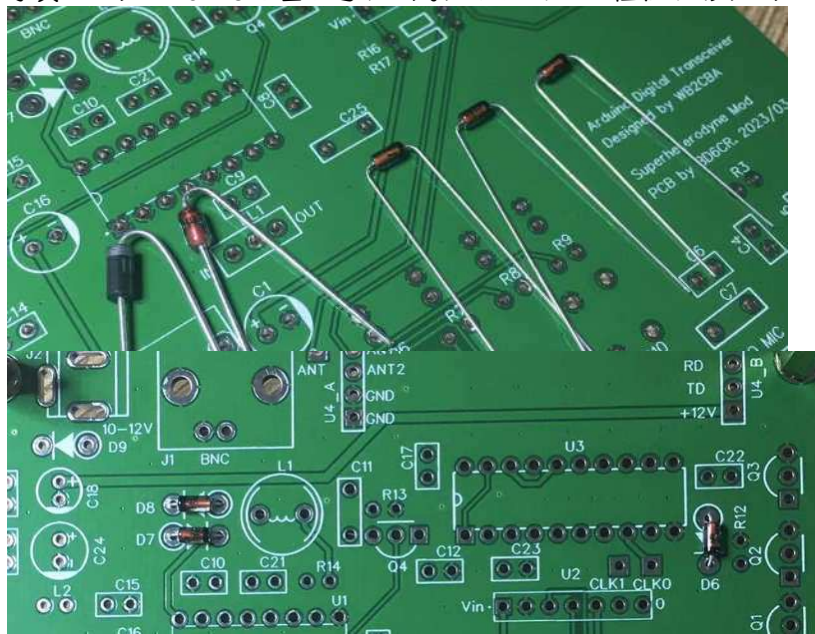




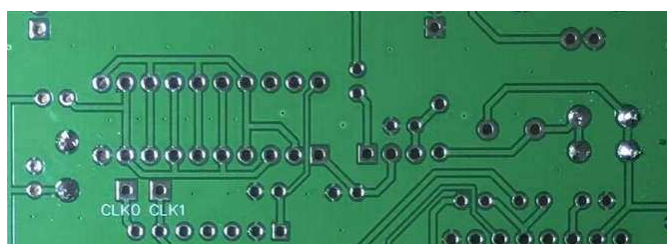
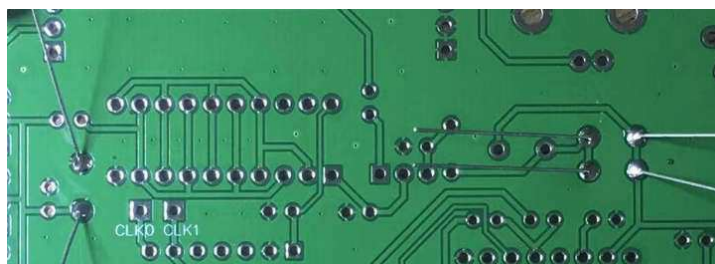
ステップ1: ダイオード

以下の写真のようにダイオードのリード線を曲げます。左から右へ、D9 - 1N5817、D10 - 1N4756A、D6-D8 - 1N4148 です。

写真のように D6-D8 を差し込みます。ダイオードには極性がありますので、間違えないでください。

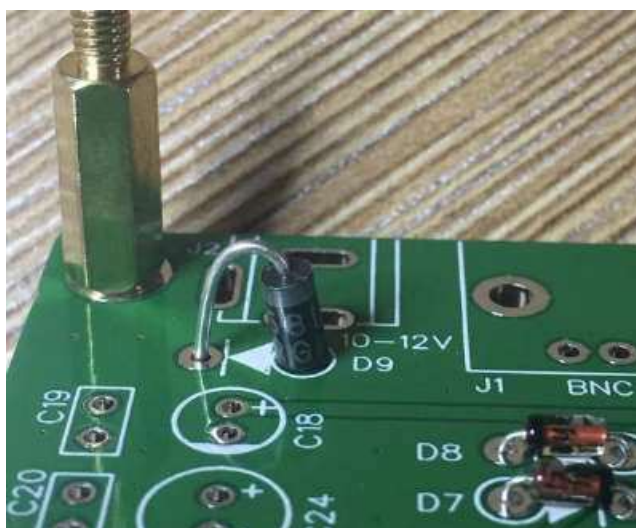


ハンダ付けのために PCB を裏返したら、抜け落ちないようにリード線を曲げてください。温度調節が可能なハンダごてを使用しているなら、350°Cに設定します。鉛入りハンダにも鉛フリーハンダにも適温です。各パッドを 3 秒間でハンダ付けします。長時間ハンダごてをあてないでください。



ニッパーを使用してリード線をカットします。

D9 を装着し、ハンダ付けします。向きを間違えないでください。

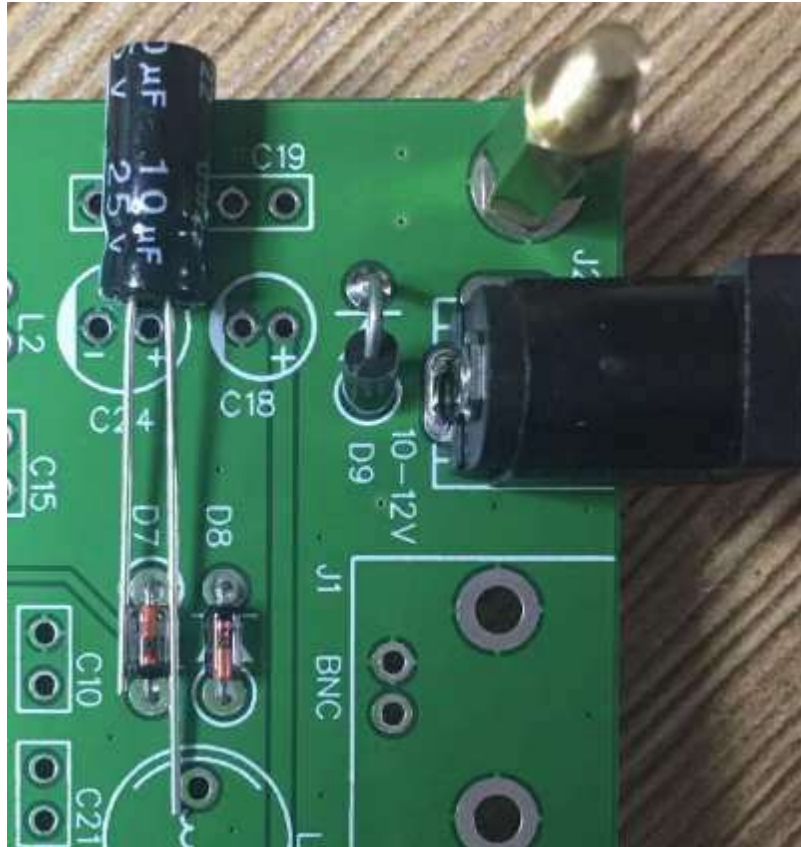


D10 を装着し、ハンダ付けします。向きを間違えないでください。



ステップ 2: +12V 電源

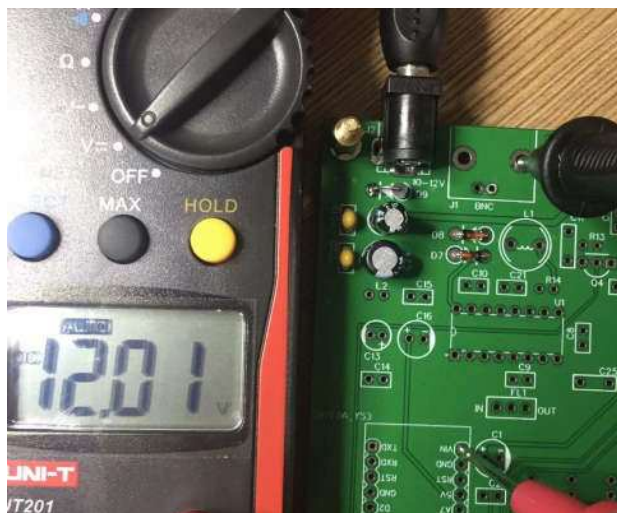
次の写真は C18 - 10uF です。長いリード線を +のマークのあるパッドに差しこみます。向きを間違えないでください。



C24 - 100uF も同様です。C18、C19-100nF、C20-100nF、それに C24 をハンダ付けします。

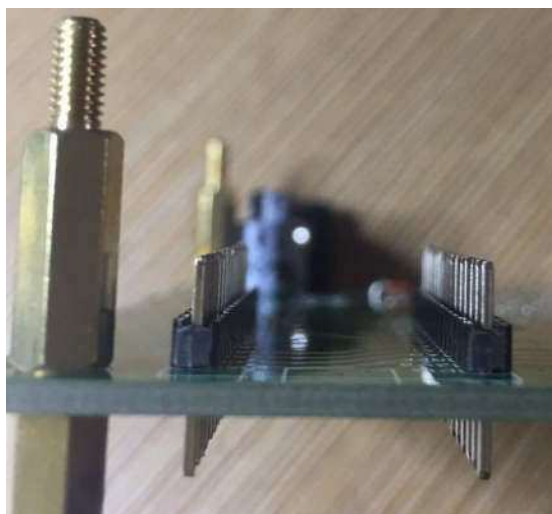


次に、12V が XA1 - Arduino Nano モジュール - の VIN ピンとつながっているかどうか、テストします。DVM を使用して、12V 前後かどうか確認します。赤いテスターリードを VIN パッドに、黒いテスターリードを BNC コネクタのグランドパッドに当てます。電源プラグに 12V 電源(センター+)を接続し、測定します。今のところ良い感じですね。



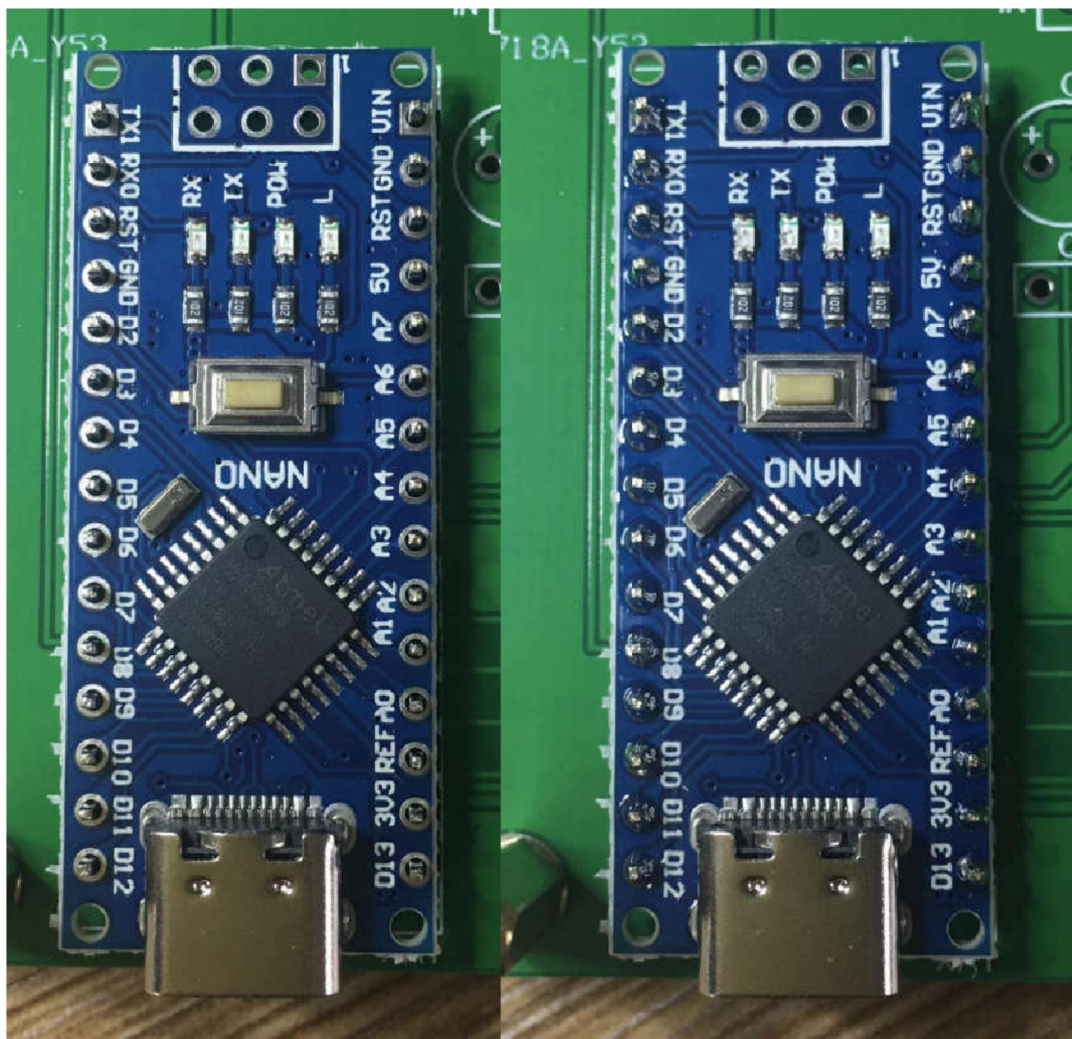
ステップ 3: Arduino Nano モジュールと、+5V、+3.3V 電源

XA1 - Arduino Nano モジュールは、メイン基板に対して+5V と+3.3V を供給します。そのため、Arduino Nano モジュールをハンダ付け後に電圧を確認します。

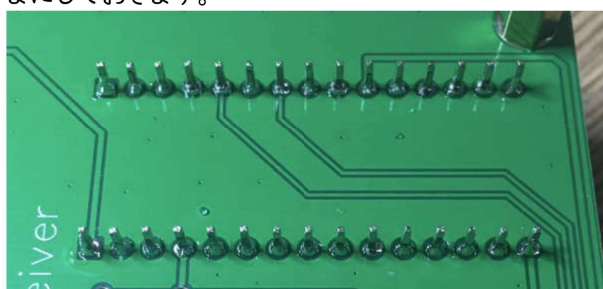


Arduino Nano モジュールをメイン基板にハンダ付けするために、ピンヘッダーを使用します。短い方のピンを上側にして、15 ピンのピンヘッダーを取り付けます。

Arduino Nano モジュールをメイン基板に差し込みます。USB 端子がメイン基板の端の方にあることを確認します。四隅のスルーホールに注意して、30 ピンをすべてハンダ付けします。



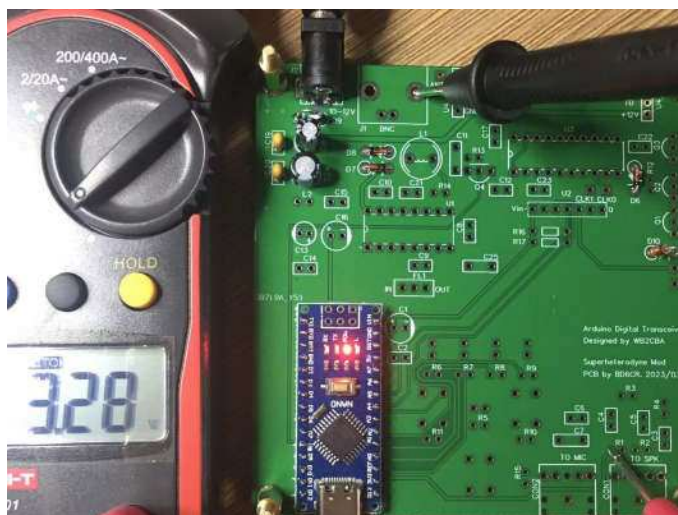
メインボードを裏返し、ピンヘッダ—のピンをすべてハンダ付けします。長いピンはカットしないでそのままにしておきます。



次はテストの番です。12V 電源(センター+)を供給して、赤いテストリードを C1- 100uF の+パッドに当て、電圧を測ります。4.93V なら、Arduino Nano モジュール上の 5V レギュレータの仕様範囲内です。このステップを飛ばさないでくださいね。レギュレータが破損したら、Arduino Nano モジュールのハンダ付けをやり直すのは非常に困難です。

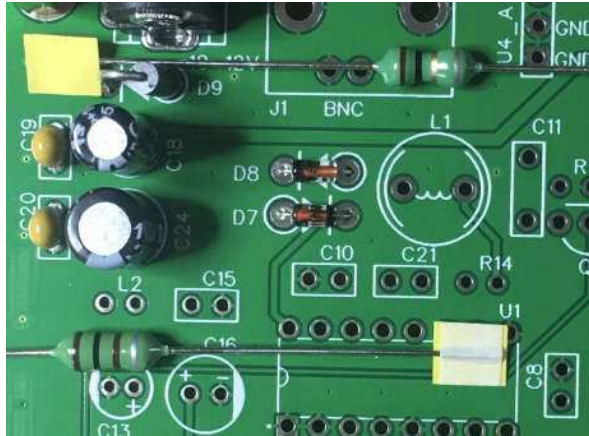


R1 - 1M の左側のピンで 3.3V かどうか測定します。大丈夫そうですね。このステップを飛ばさないでください。レギュレータが破損したら、Arduino Nano モジュールのハンダ付けをやり直すのは非常に困難です。

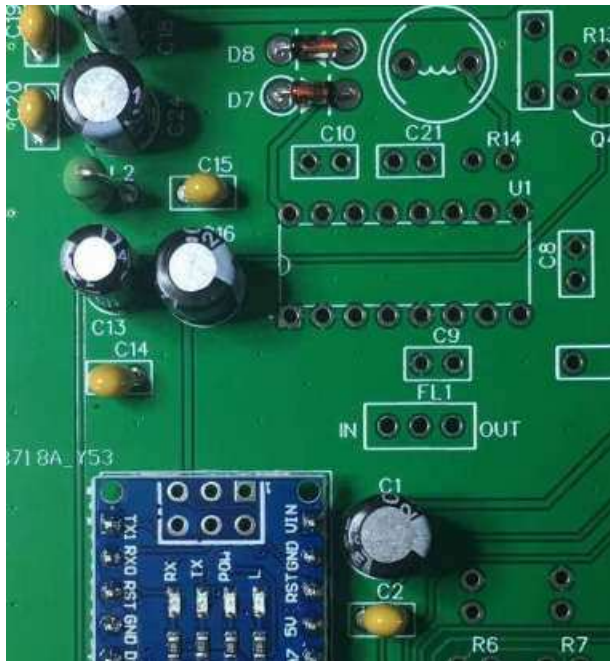


ステップ 4: CD2003 の電源

CD2003 のようなアナログデバイスには、電源ラインの LC フィルタが必要です。以下の写真に、L1 - 1uH と L2 - 100uH のインダクタを示します。



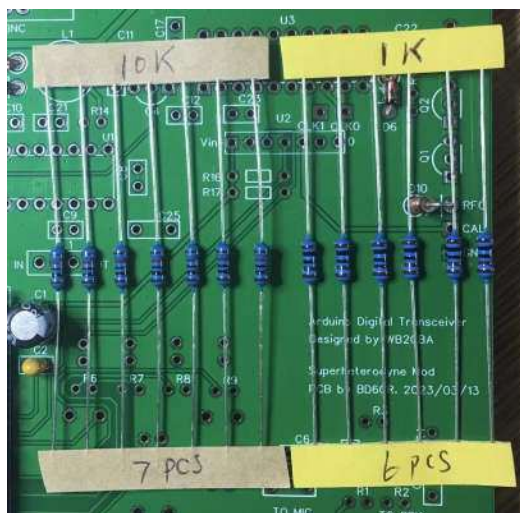
L2, C13-10uF, C1, C16-100uF, C2, C14, C15-100nF をハンダ付けします。C1、C13 それに C16 の極性を間違えていないか確認してください。



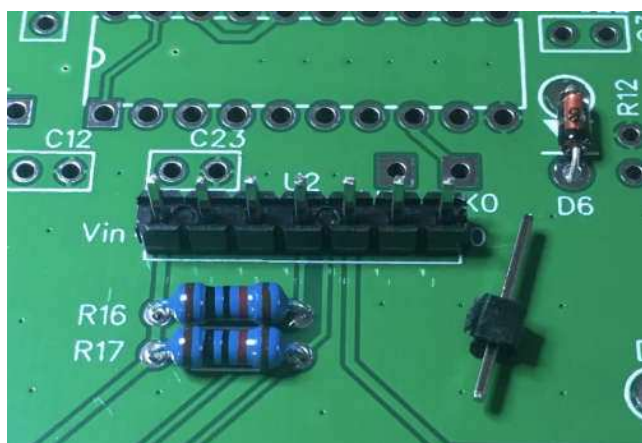
もう一度 12V 電源(センター+)を供給し、U1-CD2003 のピン 15 に+5V が供給されているかどうか確認します。

ステップ 5: 5351 モジュール

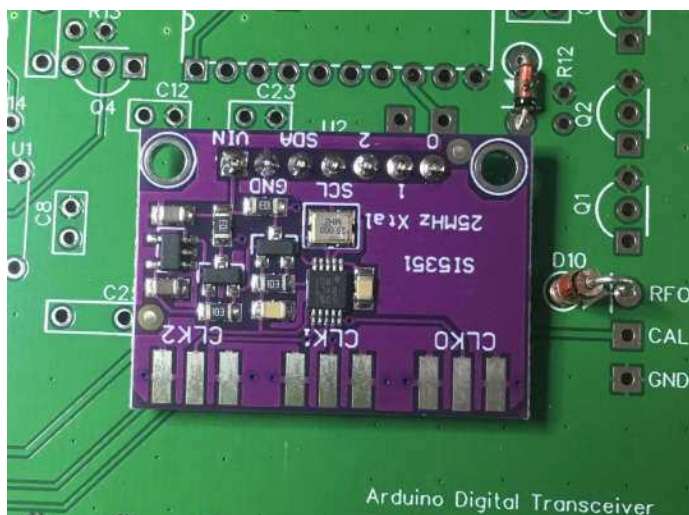
抵抗のカラーコードを確認して、このステップ用に 10K の抵抗 2 本を用意します。



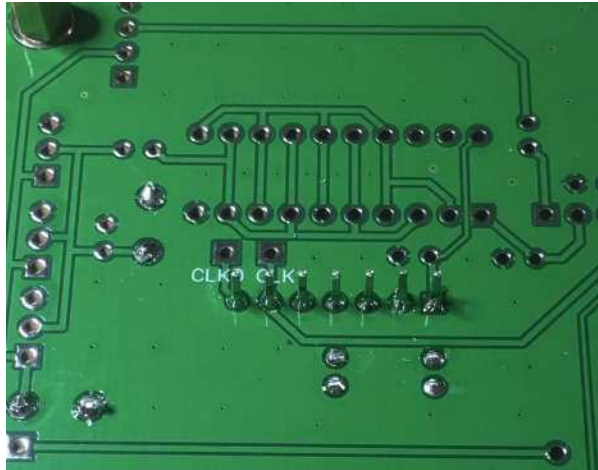
R16, R17-10K を PCB に密着して取り付けハンダ付けします。5351 モジュールには 8 ピンのヘッダーが用意されていますが、7 ピンしか使用しません。ですから、1 ピンだけ切り離した後、短い方のピンを上にして、7 ピンのヘッダーを差し込みます。



5351 モジュールをピンヘッダーに差し込み、ハンダ付けします。モジュール上のチップは、Silicon Labs SI5351 の中国製の互換チップの場合がありますが、十分に動作します。



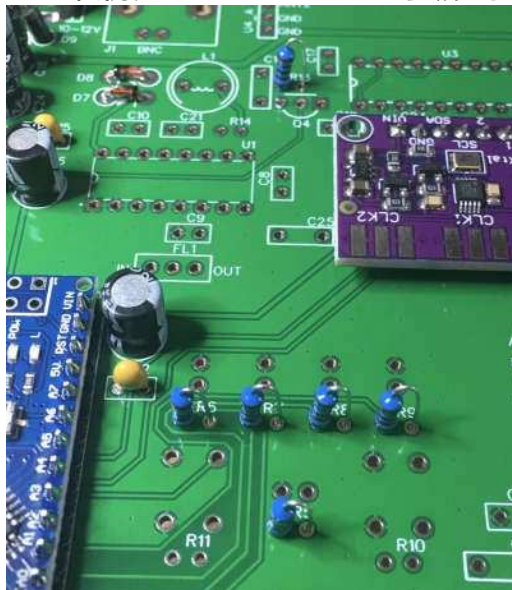
メイン基板を裏返して、ピンヘッダーをハンダ付けします。ピンが長いからとカットしないでください。



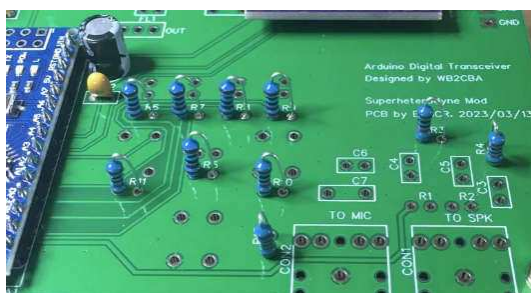
これで次のステップに進み、5351 モジュールをテストします。

ステップ6: LED とボタン

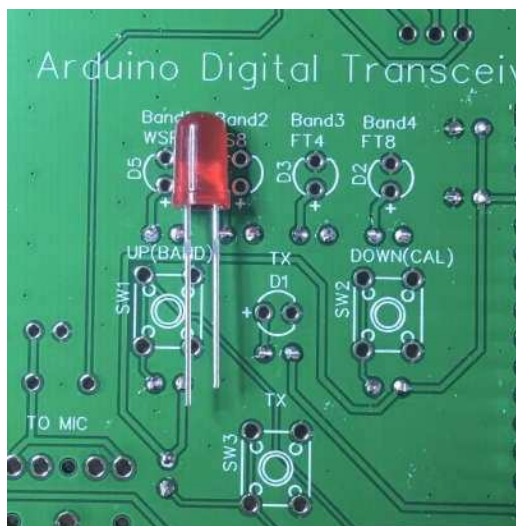
1K の抵抗、R5-R9 と R13 をメイン基板に差し込み、ハンダ付けします。



10K の抵抗、R3, R4, R10, R11, R15 をメイン基板に差し込み、ハンダ付けします。



LED の長いリード線は D1 の+のマークのあるパッドの方です。



D1 を基板に密着するように差し込み、ハンダ付けします。

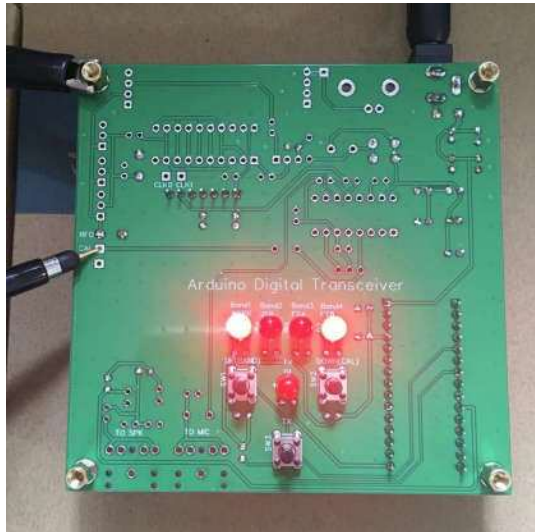


D2-D5 それに SW1-SW3 を、できるだけ基板に密着するように差し込み、ハンダ付けします。



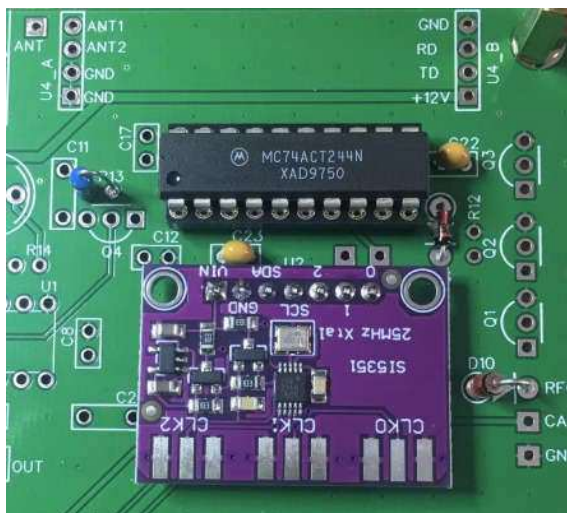
SW2-DOWN(CAL)を押しながら、電源を供給し、B1 と B4 の LED が点灯するのを待ちます。次に、テストポイント CAL に周波数カウンタ、またはオシロスコープなどの計測器の入力を接続し、1MHz の方形波が出力されているかどうか確認します。クリップを CAL のテストポイントに接続する前に、必ず先にグランドをクリップしてください。周波数がズれている場合は、キャリブレーションを行います。SW1-UP(BAND)または SW2-DOWN(CAL)を押して、1MHz 近くになったことを確認したら、SW3-TX を押

して設定を保存します。オシロスコープで表示される周波数はあまり正確でないこと、また周波数を表示するためには大きな入力信号を必要とする周波数カウンタがあること、に注意してください。



ステップ7: TXドライバ 74ACT244

C22, C23-100nF、それに U3-74ACT244 を差し込み、ハンダ付けします。74ACT244 にはソケットを使用することをお勧めします。IC のリードは正確に直角ではないことが多いので、両手で IC チップを持ち、直角になるよう平らな場所に押し付けます。そうすると、ソケットに差し込むのが簡単です。IC のリードピンを 1 ピンずつ曲げたりしないように。



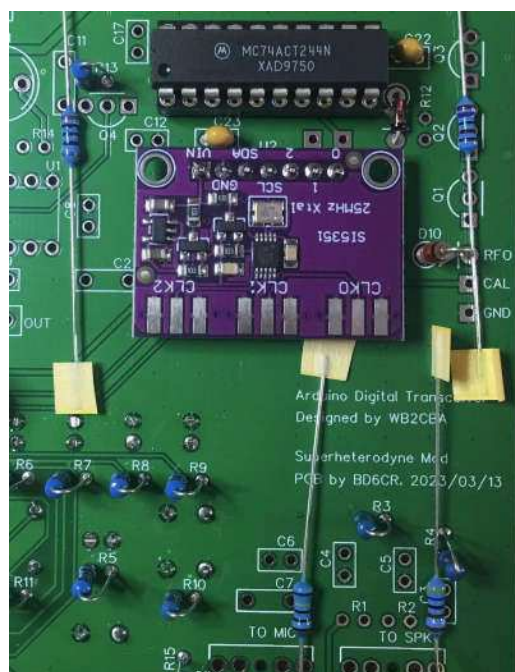
ここで TX 送信部のテストを行います。放射アンテナの代りに、DVM のテストリードを Q3-BS170 のセンターピンに当てます。



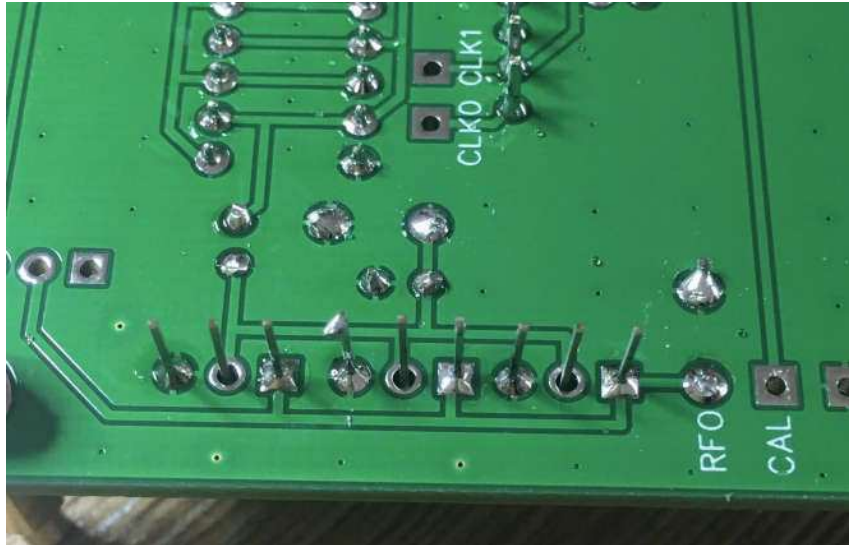
SW3-TX ボタンを押し送信にして、受信機を近づけると、強力な信号が聞こえます。私の TS-590 トランシーバでは、S9+60dB でした。LED の表示から 20m 版であり、モードが FT8 だったので、周波数を 14.074MHz にあわせました。

ステップ 8: TX 部の組み立て

抵抗 4 本、R1-1M、R2-4.7K、R12-2.7K、それに R14-100R のカラーコードを確認して、基板に取り付けて、ハンダ付けします。

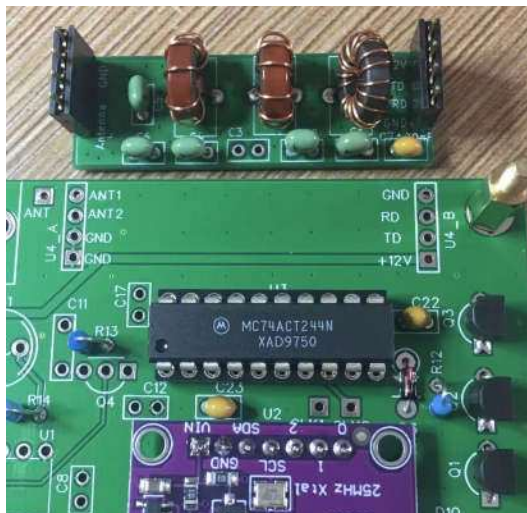


Q1-Q3 - BS170 を取り付け、センターピン以外のリード線をすべてハンダ付けします。静電気に弱いのでリード線に触らないでください。



ハンダごてのアースがとれているかどうか確認します。そうでなければ、ハンダごての電源コードを抜いてから、センターピン3箇所をハンダ付けします。

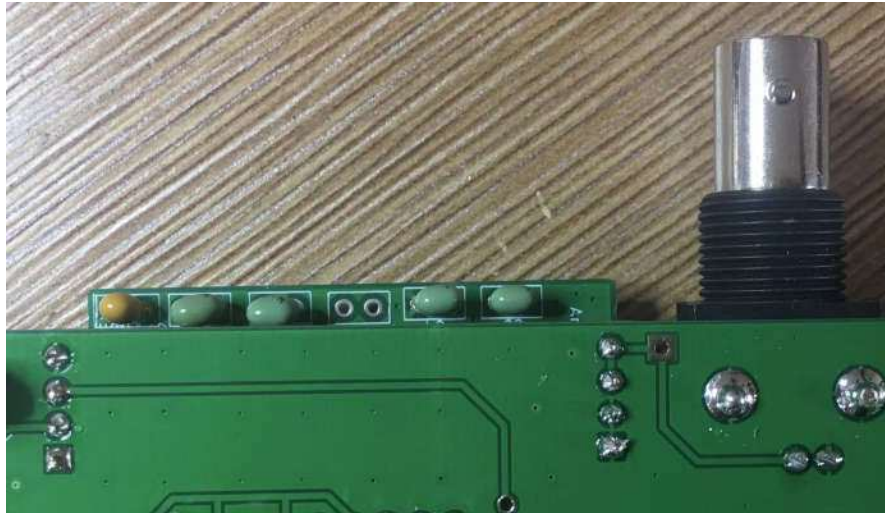
LPF バンドフィルタ基板が用意できているなら、確認のため、まず LPF バンドフィルタ基板にピンヘッダーを差し込んでみてください。



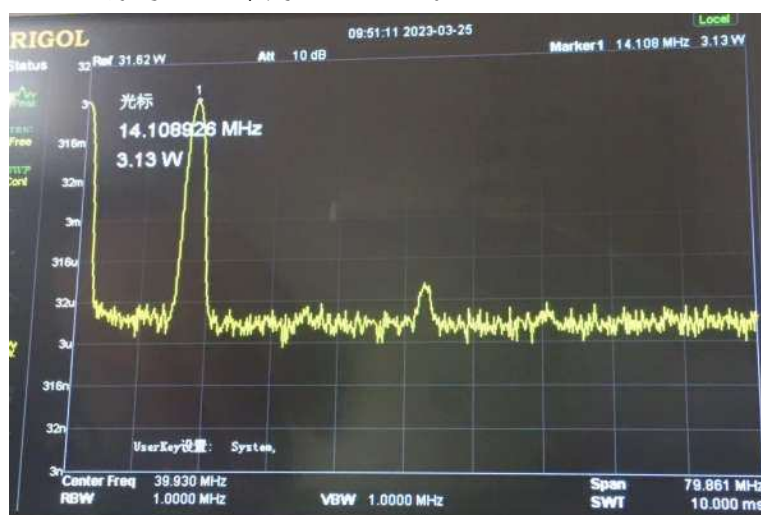
メイン基板にピンヘッダーを差し込み、ハンダ付けします。



メイン基板に J1-BNC を差し込み、ハンダ付けします。基板の高さに対する制限があるので、浮いていないように確認してください。



これで TX 送信部の組み立ては終了しました。TX のテストの前に、LPF バンドフィルタ基板と Arduino Nano モジュールのバンド設定が一致しているかどうか確認します。一致していなければ、SW1-UP(BAND)を押しながら電源を投入して一致させます。バンドを変更するには、SW1-UP(BAND) または SW2-DOWN(CAL) を押します。デフォルト設定では、B1 - 40、B2 - 20、B3 - 15、B4 - 10 です。SW3-TX を押してバンド設定を確認します。ここで、TX を RF 電力計、お持ちならスペクトラムアナライザに接続します。TX に対して 50Ω の負荷が接続されていることを確認してください。私の場合、テスト中の TX とスペクトラムアナライザの間に 45dB のアッテネータを接続し、SW3-TX ボタンを押した時に、RF 出力レベルとスペクトルを確認しました。RF 出力は 3.13W、スプリアスは約 -50dBc でした。RF 出力は電源電圧により上下します。スペクトラムアナライザが 35dBm 付近を示しているなら、50Ω 負荷時 3W 前後になります。送信時の電源電流が 600mA 付近なのを確認してください。



ステップ9:オーディオジャック

C3、C6-100nF、C4、C5-10nF、C7-1uF、CON1-TO SPK、それに CON2-TO MIC を差し込み、ハンダ付けします。

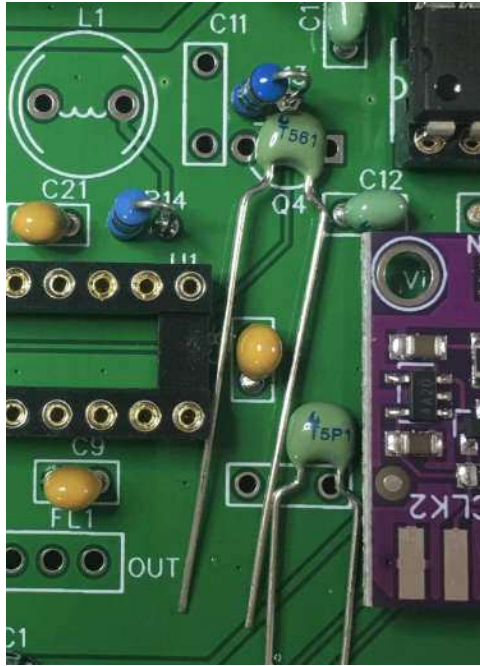


iPhone 上の Sonic Tools のようなオーディオ信号発生器を使用して、CON1-TO SPK ジャックにオーディオ信号を入力します。ボリュームを最大にすると、TX 状態に変わり、TX LED が点灯します。このテストを実行する間、50Ω 負荷抵抗が J1-BNC に接続していることを確認してください。

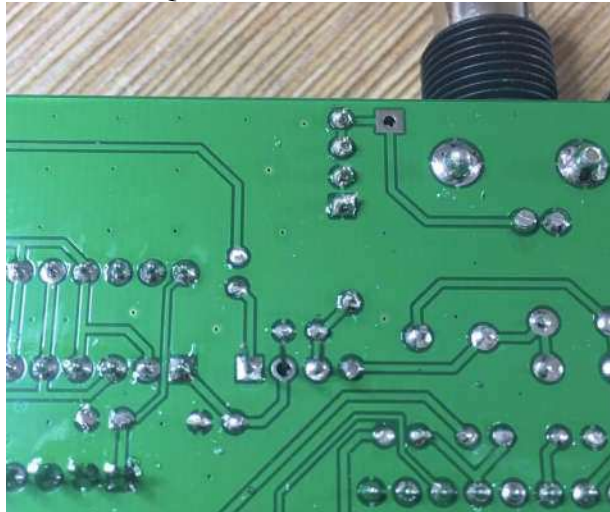


ステップ 10: CD2003 受信 IC

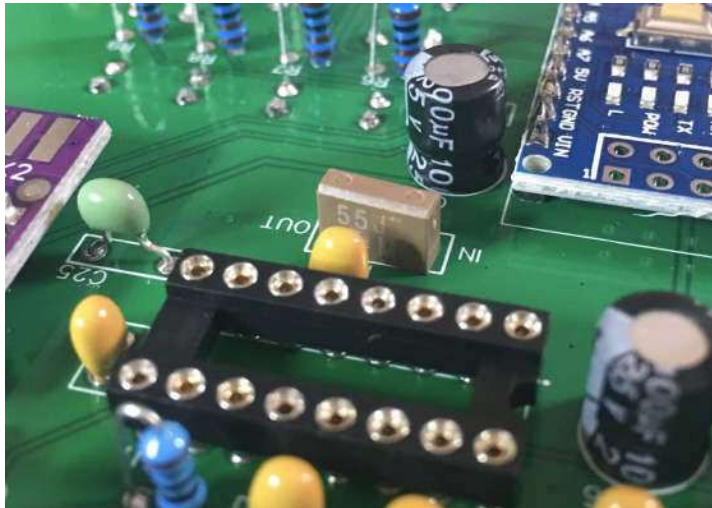
次の写真は、C11-560pF, 620pF or 680pF と C25-5.1pF を示しています。同じように見えますね。



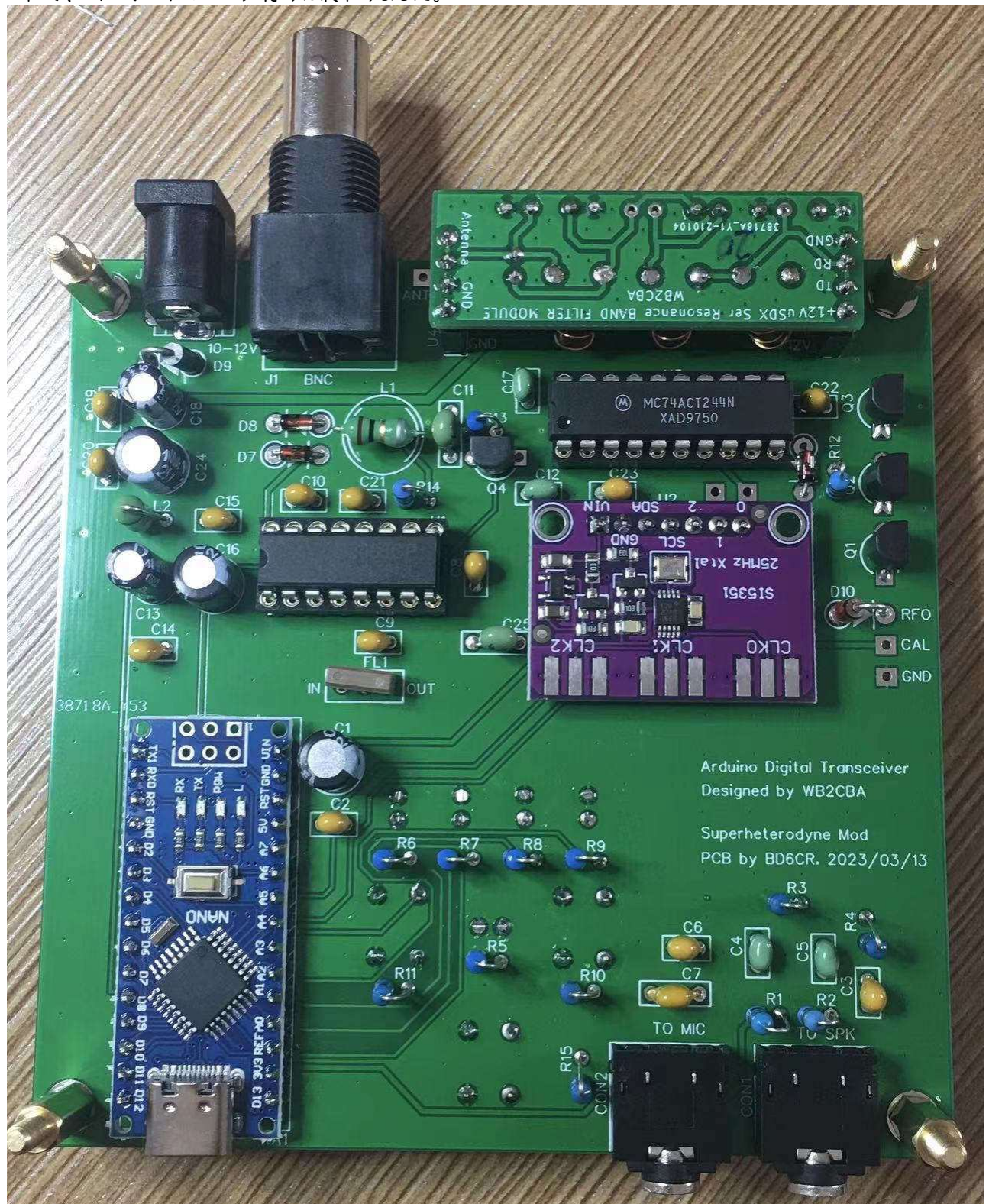
次の写真は、Q4-BS170 のセンターピンが静電気にさらされていないかどうかを示しています。



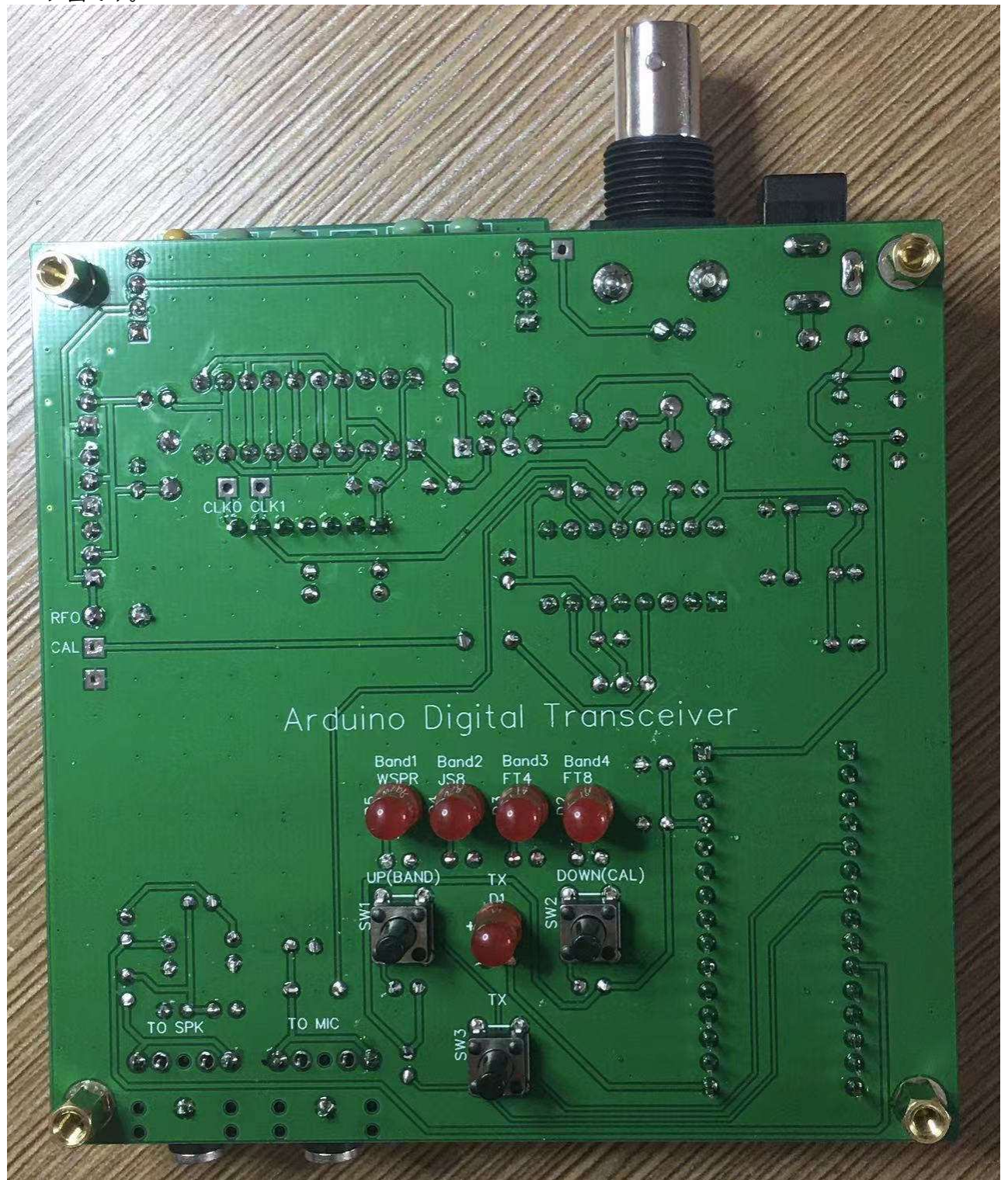
次の写真は、FL1 のマーキングが CD2003 側かどうかを示しています。



残りの部品、L1-1uH、Q4-BS170、FL1-PFB455JR、U1-CD2003、C8、C9、C10、C21-100nF、C12、C17-10nF、C11-560pF または 620pF または 680pF、そして C25-5.1pF を取り付けハンダ付けします。これで、メインボードのハンダ付けは終わりました。



ハンダ面です。



ここでトップカバーを取り付け、ネジで固定します。スイッチのキャップを取り付けます。赤色はSW3-TX用です。



ボトムカバーを取り付け、M3×6 スタンドオフで固定します。

トップ/ボトムカバーの重なり具合やスタンドオフが正しく組み立てられているかどうか、を確認します。回路ショートが気になるようでしたら、電源に接続する前に再度確認してください。



これで、メインボードの組み立てが終了しました。

WB2CBA & BD6CR

ANTENNA DC 10-12V

ADX-S

Arduino Digital Modes HF QRP Transceiver

B1 B2 B3 B4



WSPR JS8 FT4 FT8



(BAND)<--- TX --->(CAL)



SPK

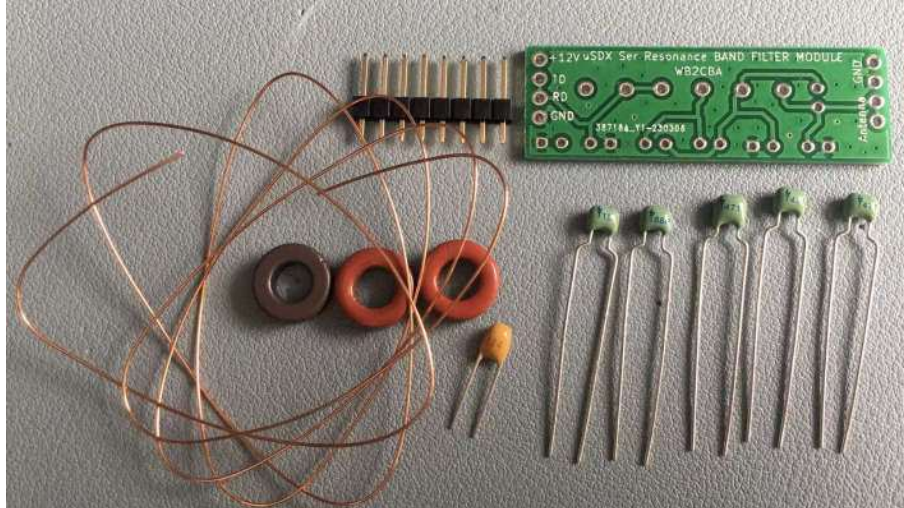
MIC

Arduino USB

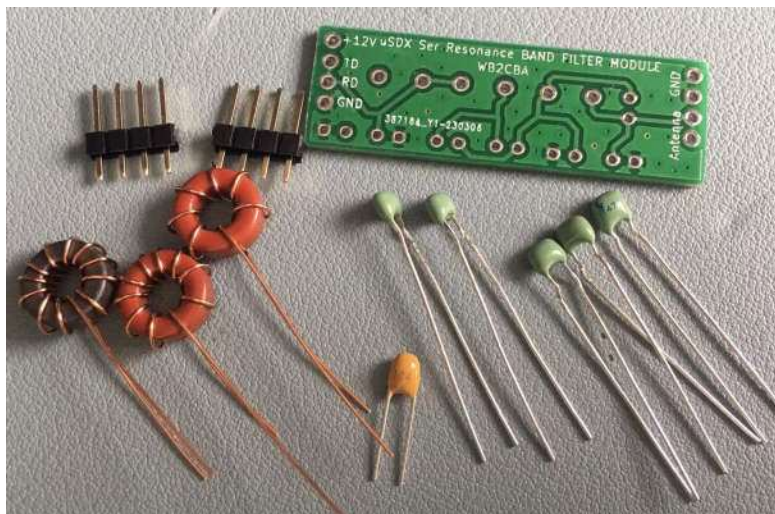
38718A_Y50

ステップ 11: LPF バンドフィルター基板

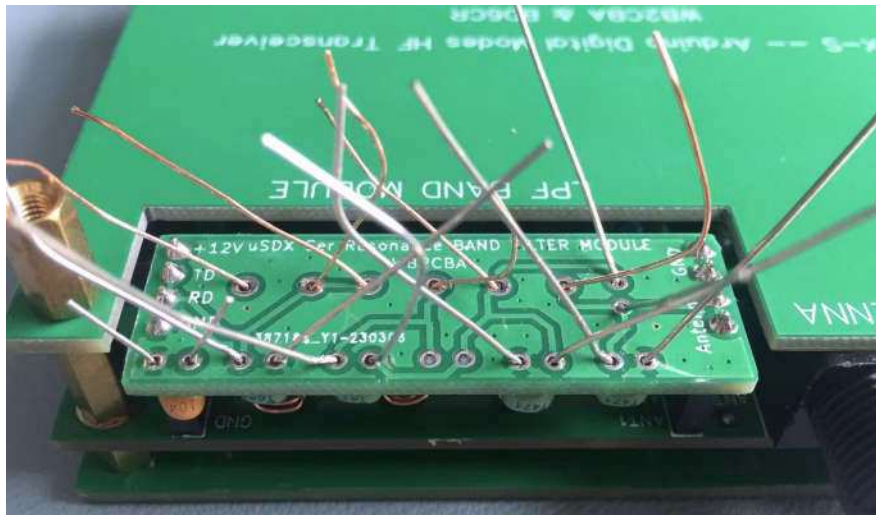
20m LPF バンドフィルター基板の例です。取り付ける部品はこれだけです。10m や 15m 版の場合は、赤色のトロイダルコアは黄色になります。



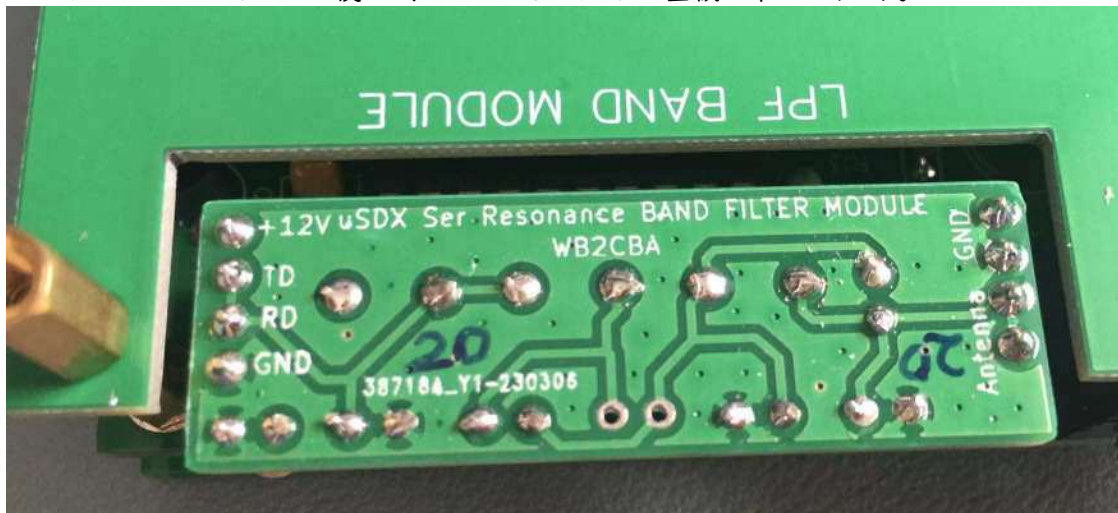
部品表に従って、トロイダルコイルを巻き、ピンヘッダーとキャパシタを準備します。黒色のトロイダルコイルは 12 回巻き、赤色のトロイダルコイルは 10 回巻きと 7 回巻きです。エナメル線の端のエナメルをはがしてからハンダ付けします。紙やすりを使ってもいいでしょう。必要なエナメル線の長さは、1 回巻く毎に 1.5cm+5cm です。12 回巻きなら、長さは 23cm になります。



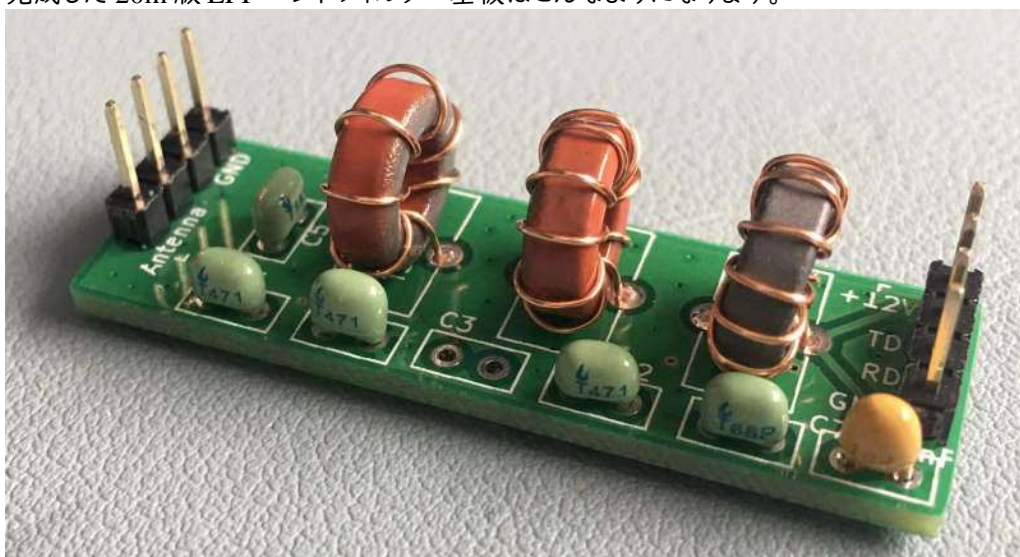
サポートとして、メインボード上のソケットを使用することをお勧めします。部品を正しく差し込み、ハンダ付けします。C7-100nF が黒色トロイダルコイルの側に、赤色トロイダルコイルが 3 個のトロイダルコイルの中央にあることを必ず確認してください。オリジナル設計の ADX の回路図と部品表が、PCB と一致しないことがあります。ADX-S の部品表を再度確認してください。



マーカーペンまたはラベルを使って、LPF バンドフィルター基板に印をつけます。



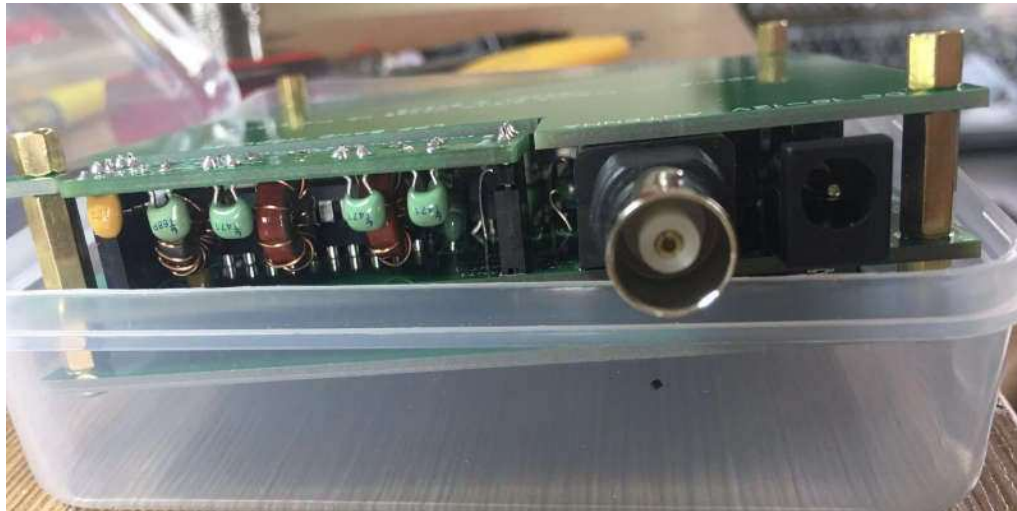
完成した 20m 版 LPF バンドフィルター基板はこんなようになります。



ステップ 12: ADX-S のケース

キット到着後にプラスチックケースがお気に入りを感じたら、PCB のサンドウィッチも十分素敵に見えるのですが、ケースとして使用してみてください。壊れていた場合は、BOM に表示されていないのですが、遠慮なく申し出てください。

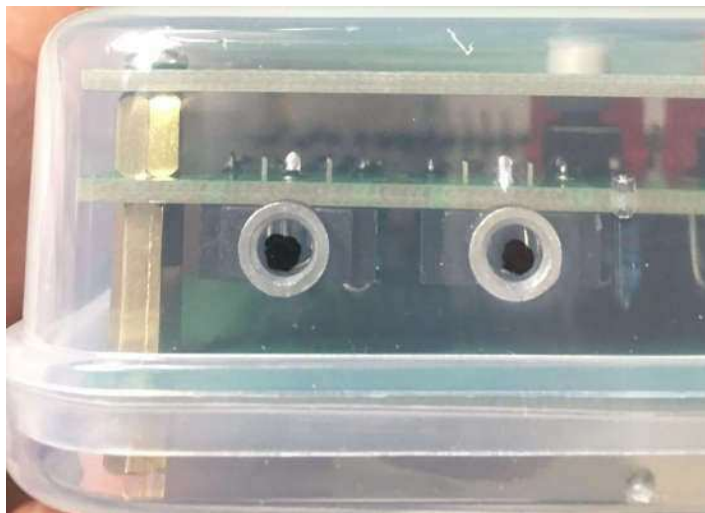
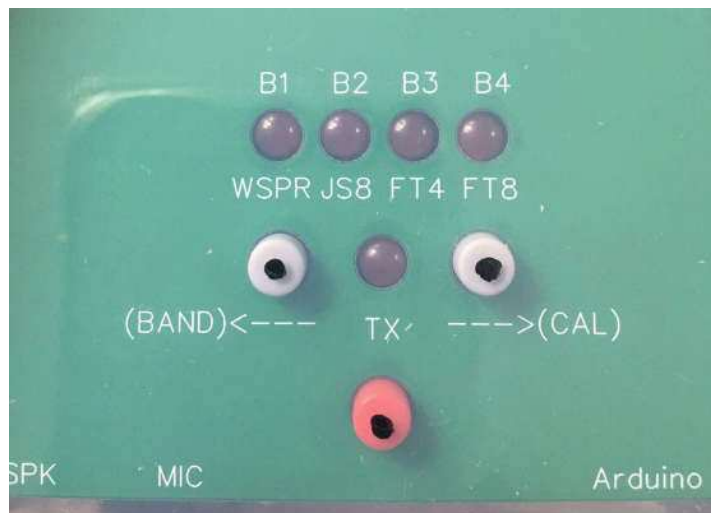
BNC 用の穴の位置を決めてマークを付けます。ケースの上縁が BNC の穴の上限です。ADX-S を上下逆さまに組み込むことをお勧めします。そうすると、LPF バンドフィルタ基板の交換が簡単になります。



最初にドリルで小さな穴を開けてから、穴を大きくして、BNC コネクタが出るようにします。



ケースの底の方にボードを入れると、オーディオジャック、電源ジャック、それに USB ポートの穴の位置が決まります。そして、ボードのボタンが底部に当たっていると思います。穴開け位置をマークします。



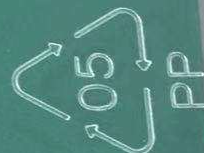
USB ポートの位置は短い線でマークし、2 個の穴を開け、それを大きくしてつながるようにします。穴開けの終わったケースは次の写真のようになります。

WB2CBA & BD6CR

ANTENNA DC 10-12V

ADX-S

Arduino Digital Modes HF QRP Transceiver



B1 B2 B3 B4

WSPR JS8 FT4 FT8

(BAND)<---

TX

--->(CAL)

38718A Y50

SPK

MIC

Arduino USB

設定とキャリブレーション

必要な設定が2ヶ所あります。BAND 設定と MODE 設定です。BAND 設定の方法が ADX のオリジナルの設計から変更されていることに注意してください。

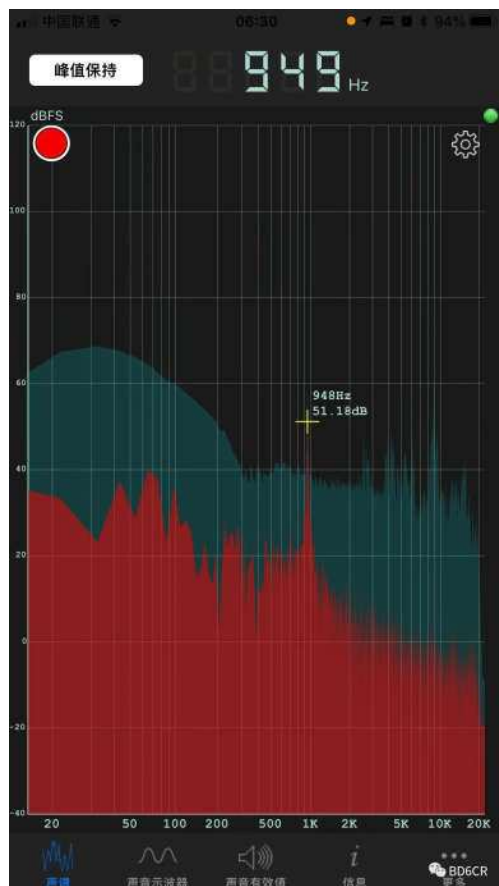
送信する前に、LPF バンドフィルター基板と Arduino Nano のバンド設定が一致していることを必ず確認してください。BAND 設定を確認するには、W1-UP(BAND)ボタンを押したまま電源を投入します。LED の点滅が終わった後、SW1-UP(BAND) または SW2-DOWN(CAL) を押して BAND を選択します。デフォルト設定では、Band1 - 40、Band2 - 20、Band3 - 15、Band4 - 10 です。SW3-TX ボタンを押してバンド設定を保存します。

MODE 設定は通常の動作中でも変更可能です。LED が現在の MODE を表示しています。左から右へ、WSPR、JS8、FT4 そして FT8 です。MODE 変更時に周波数も変更します。BAND と MODE の組み合わせによりキャリアの周波数が決まります。例えば、20m バンドで FT8 モードを選択すると、キャリアの周波数は 14.074MHz になります。ファームウェアにある周波数の表を次に示します。

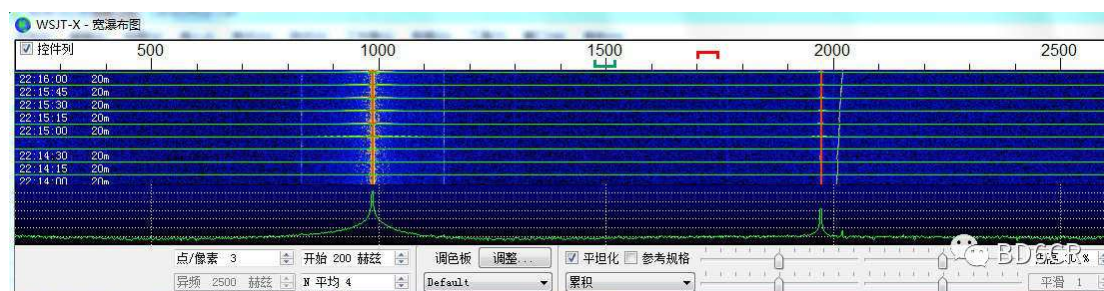
(kHz)	40 meter	20 meter	15 meter	10 meter
WSPR	7038.6	14095.6	21094.6	28124.6
JS8	7078	14078	21078	28078
FT4	7047.5	14080	21140	28180
FT8	7074	14074	21074	28074

キャリブレーションが必要なのは、周波数のキャリブレーションです。5351 モジュールにある 25MHz の水晶発振器の発振周波数には誤差があります。SW2-DOWN(CAL)ボタンを押しながら電源を投入すると、キャリブレーションモードになり、B1 と B4 の LED が点灯します。ウォームアップのために2分間そのままにしておきます。標準的なキャリブレーション方法は、CAL テストポイントに分解能 1Hz の周波数カウンタを接続し、SW1-UP(BAND) または SW2-DOWN(CAL)を押してできるだけ 1MHz に近くなるようにした後、SW3-TX ボタンを押して設定を保存します。キャリブレーションモードを終了するために、一度電源を切った後再度投入します。

周波数カウンタをお持ちでない場合は、周波数誤差の小さいアマチュア用のトランシーバまたは受信機を使用します。周波数キャリブレーションを実行するのは 10m バンドを選択してください。そうすれば、低いバンドの周波数は自動的に正確になります。例えば、10m バンドで FT8 モードを選択し、28074kHz USB に周波数をあわせませ。10m LPF バンドフィルター基板を装着しており、バンドの設定が正しいことを必ず確認してください。少なくとも 5W のダミーロード、または同調するアンテナを BNC コネクタに接続します。SW3-TX ボタンを押すと、約 1kHz のオーディオ音が聞こえると思います。iPhone 上の Sonic Tools などのオーディオスペクトラムアナライザのアプリを使用して、オーディオ音の周波数を測定します。オーディオケーブルを使用しているなら、<http://www.qsl.net/z11an/Software/Spectrum3.zip> をダウンロードして、PC 上で実行することも可能です。電源投入してキャリブレーションモードに入り、一方向に 10 ステップ繰り返した後、SW3-TX を押して設定を保存します。電源を再投入し、受信機で周波数測定を行います。周波数を正確に合わせるには何度か実行する必要がありますが、正確な 1kHz の音になるまで繰り返します。



tinySA をお持ちの場合、信号発生機能を使用して、28075kHz、(30dB または 40dB の)アッテネータを追加して-85dBm の信号として入力し、WSJT-x または JTDX モードでの ADX-S の受信スペクトルをモニタします。電源投入してキャリブレーションモードにして、次に示すようにスペクトラムを確認します。



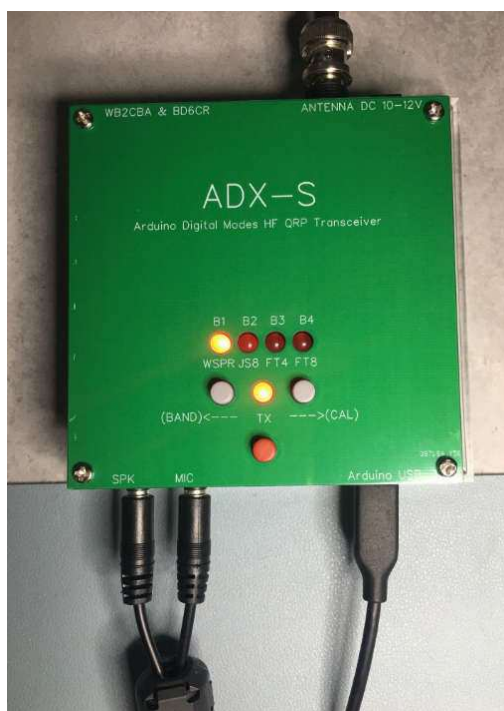
nanoVNA をお持ちの場合、LPF バンドフィルタ基板を細かく調整することが可能です。これにより、RF 出力をより大きくしながら、高調波を抑えることが可能になります。元の設定でも悪くはないので、経験者だけに通用する選択肢です。

ADX-S の使い方

最初に LPF 小型基板を ADX-S に差し込み、電源投入時に BAND ボタンでセットします。LPF バンドフィルタ基板が正しい向きかどうか、ボトムカバーを取り付けていないときは特に注意して、確認してください。さもないと、LPF バンドフィルタ基板が壊れることがあります。LPF バンドフィルタ基板上のキャパシタはメイン基板の外側に向いています。バンドフィルタ基板を正しく差し込んだら、電源投入時に BAND ボタンを長押しし、現在のバンドが表示されたら、UP または DOWN キーで切り替え、TX ボタンで設定を確認します。

ADX-S に接続する電源は 10-12V です。最大値は、リチウムイオン電池の 3 本分、または 12.6V です。出力電力は QRP の範囲内です。電源電流は 400-600mA であり、JS8、FT 4 それに FT8 動作に最適な QRP 電力です。送信部で発生する熱により周波数変動する可能性があり、WSPR にはあまり向いていません。

USB ポートを使用して 5V で電源を供給することにより、QRPP に切り替えることが可能です。12V 電源と USB ポートを同時に接続しないでください。Arduino Nano モジュールに損傷を与える可能性があります。出力電力は QRP 電力の約 1/10 に低下します。QRPP では発熱が小さいので、WSPR としても適しています。もちろん、できるだけ遠くのラジオ局に接続するために QRPP パワーに挑戦する友人もいて、とても楽しいです。



ADX-S には VOX 用プログラムが組み込まれているため、2 本のオーディオケーブルで接続するだけで機能します。一本のオーディオケーブルで ADX-S の MIC または TO MIC 端子とサウンドカードの MIC 端子を、もう一本のオーディオケーブルで、ADX-S の SPK または TO SPK 端子とサウンドカードの SPK 端子を接続します。逆に接続すると、受信スペクトルに奇妙な縦線が現れ、また送信に切り替えできなくなります。ADX-S は、大きめのオーディオ入力レベルが適しています。過変調の問題について気に掛けることはありません。ADX-S のオーディオ出力が低い場合は、システムがサポートしている場合は MIC の感度を調整してみてください。例として WSJT-x について説明すると、Radio は [None]

に、PTT 方法は[VOX]にそれぞれ設定します。ログを正しく記録するには、メイン インターフェイスで適切な帯域周波数とモードを手動で選択してください。



設定インタフェースでは、Radio は[None] を、PTT 方法は[VOX]を選択します。



メインウィンドウでは、正しい動作周波数を表示するよう、バンドとモードを手動で設定します。例えば、20m、WSPR を設定すると、周波数は 14.095600 に移動しますが、これは ADX-S のファームウェアの設定と同じです。

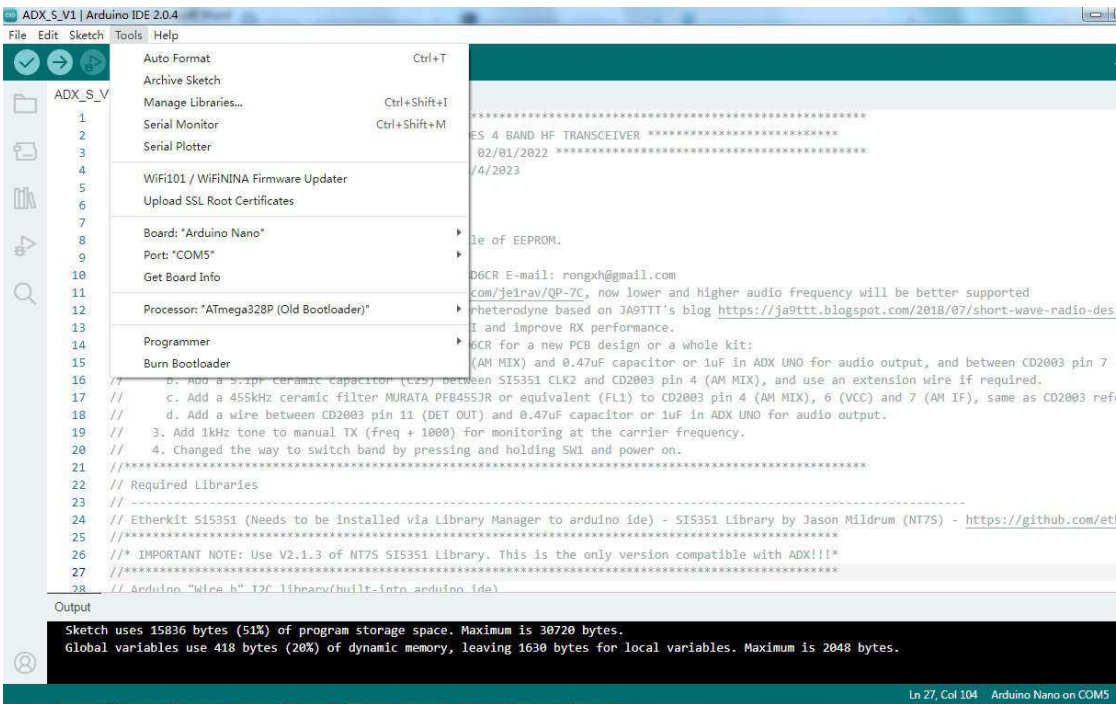
交信を成功させるためには、時刻と周波数を校正する必要があります。ブラウザ中で time.is を入力して、コンピュータの内蔵時計が正確であることを確認します。1 秒以内なら良好です。ADX-S により実際の QSO をする前に、周波数の校正をしてください。送信出力は QRP の範囲内ですので、まず強力な送信局、少なくとも SNR が -5dB 以上ある局、を呼ぶことから始めます。その後、さらに弱い信号の局を呼びます。フルサイズのアンテナがあれば、DX 交信を行うことも夢ではありません。

もし受信部が強力な放送局による BCI を受ける場合は、マグネチックループアンテナのような同調型のアンテナを使用するか、アンテナとの間に Z マッチチューナのような ATU を追加してください。それでもだめだった場合は、アンテナと ADX-S の間に一時的に固定アッテネータ、例えば -20dB、を挿入して、効果があるかどうか確認してください。効果があったなら、受信部を改造して、入力にアッテネータと、ON/OFF を切り替えるスイッチを追加します。

オープンソースファームウェアの修正とアップデート

Arduino のネイティブ コード ADX_S_Vx.ino はオープンソースですので、自分で変更してコンパイルすることが可能です。デフォルトでは、事前に Arduino Nano モジュールのファームウェアは書き込み済みであり、B1-B4 はそれぞれ 40 20 15 10 メートルバンドにプリセットされています。コードを参照して個人的な変更を加えたい場合は、<https://www.arduino.cc/en/software> から Arduino IDE をダウンロードしてください。

12V 電源を落とした後、まず TypeC ケーブルを使用して Arduino Nano ボードを PC の USB ポートに接続します。USB ポートよりボードに電源が供給されます。Arduino IDE のボードマネージャでは、Arduino Nano を選択し、COM5 などの正しいポートを選択します。Arduino Nano モジュールが青色の場合は、プロセッサで ATmega328P (旧ブートローダー) を選択してください。Arduino IDE にファームウェアをロードした後、コンパイルし、Arduino Nano モジュールにアップロードします。5351 ライブラリが見つからないというエラーが表示された場合は、Sketch - Include Library - Add .ZIP Library からライブラリ Si5351Arduino-master をロードする必要があります。バージョンの互換性を確保するために、ソースファイルとライブラリファイルを同時にダウンロードしてください。



セラミックフィルタの通過帯域と中心周波数には誤差があるため、ビート周波数を正確に調整した方がよいでしょう。そうすれば、SNR が向上し、干渉が低下します。そのためには、ソースコードを検索して、異なるビート周波数、2ヶ所を見つけてください。一つは 464570 で、他の一つは 447430 です。今のところ、447430 は 20m バンドでしか使われていません。500-1kHz ステップで調整し、その前の設定値の結果と比較します。そうやって、自分のキットに最適の設定値を見つけます。これが、オープンソースのキットの醍醐味の一つです。変更オプション: 3pF のキャパシタをお持ちでしたら、5.1pF と交換し、最適化を行ってみてください。通常、3pF の方が 455kHz により近くなります。

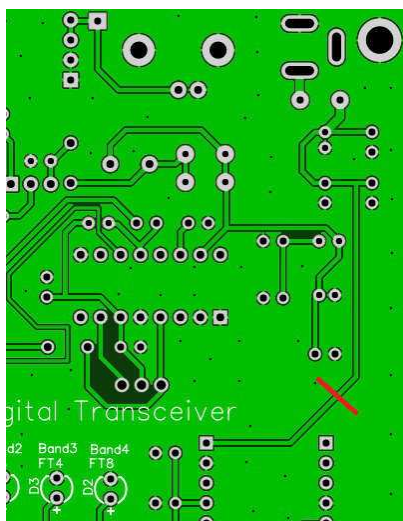
SPECIFICATIONS				
Part Number	3dB Band Width (kHz)	Selectivity	Insertion Loss (dB) max.	Composition
		+9kHz off (dB) min.		
PFB455JR	5.5 ± 1.5	17 (23)	6 (3)	2 Elements

CAT サポート

CAT は、Computer Aided Tuning の略語です。この機能を使用することにより、PC のソフトウェアにより無線機を制御することが可能になります。Arduino モジュールのファームウェアは出荷時に CAT を直接サポートしているわけではありませんが、ファームウェア V1.2 以降(ソースコード <https://groups.io/g/crkits/files/ADX-S> から)ダウンロードして、再プログラムすることにより、CAT 機能を使用可能です。サポートしている CAT コマンドは限られていますが、WSJT-x または JTDX 上で動作します。CAT の導入はよく考えられています。CAT コマンドを受信するまでは、ADX-S はももとのノーマルモードで動作します。CAT コマンドを受信すると、ADX-S は B2 と B3 の LED を点灯し、CAT モードに変わったことを表示します。これで、ボタン操作は無視され、CAT による制御しか受け付けません。このモードから抜けるには、電源を再投入します。

ファームウェアは、CAT 周波数を検出し、マニュアルで設定されたバンドと比較します。別のバンドである場合、受信は可能ですが、送信はできません。これは、出力トランジスタの保護のために追加した機能です。

繰り返しますが、12V 電源と USB ポートを同時に接続しないでください。Arduino Nano モジュールに損傷を与える可能性があります。CAT モードではいつも USB ポートを接続するので、USB の 5V による QRPP で使用するか、または PCB の配線をカットすることにより 12V 電源と USB ポートを同時に接続し、3-5W の QRP レベルで送信可能になります。



WSJT-x または JTDX の設定は、Kenwood TS-2000 です。正しいシリアルポートを選択し、ボーレートを 115200 に設定します。その他の設定は変更の必要はありません。動作確認のために Test CAT ボタンを押します。TX にするためにはオーディオ信号が必要なので、Test PTT は機能しません。

周波数同期の問題:動作開始時に、周波数表示ウィンドウ中に周波数同期が外れたことを示す赤色のバックグラウンドが表示されることがあります。MODE を切り替えた後、再度 sync に切り替えてください。

送信中に CAT を切り離すと、RFI が発生することがあります。USB ケーブルにフェライトコアを追加するか、またはアンテナから遠ざけてください。

通常、Arduino の再プログラミング後に周波数のキャリブレーションを再実行する必要はありません。

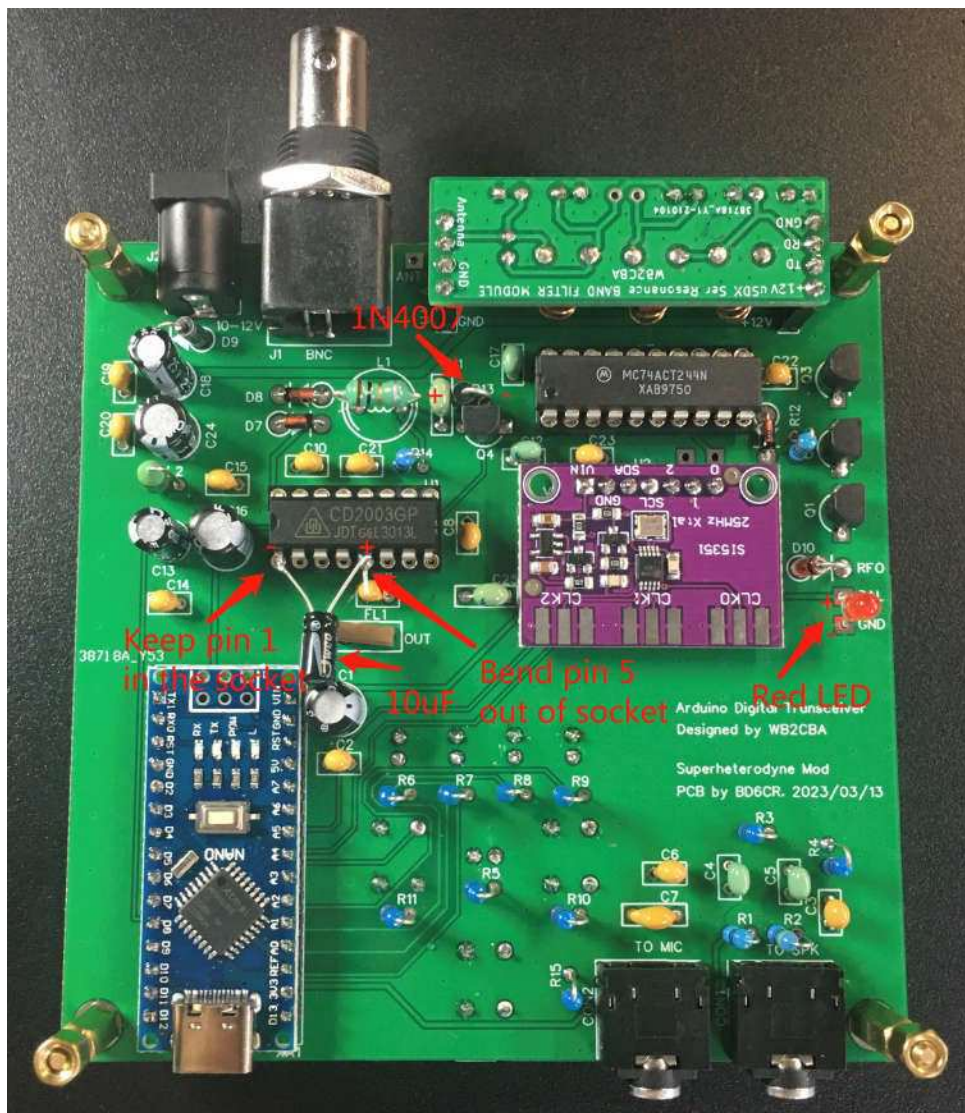
改造について

デコーディング特性を向上させるために、製作中に受信部に以下のような改造を行うこともお勧めします。

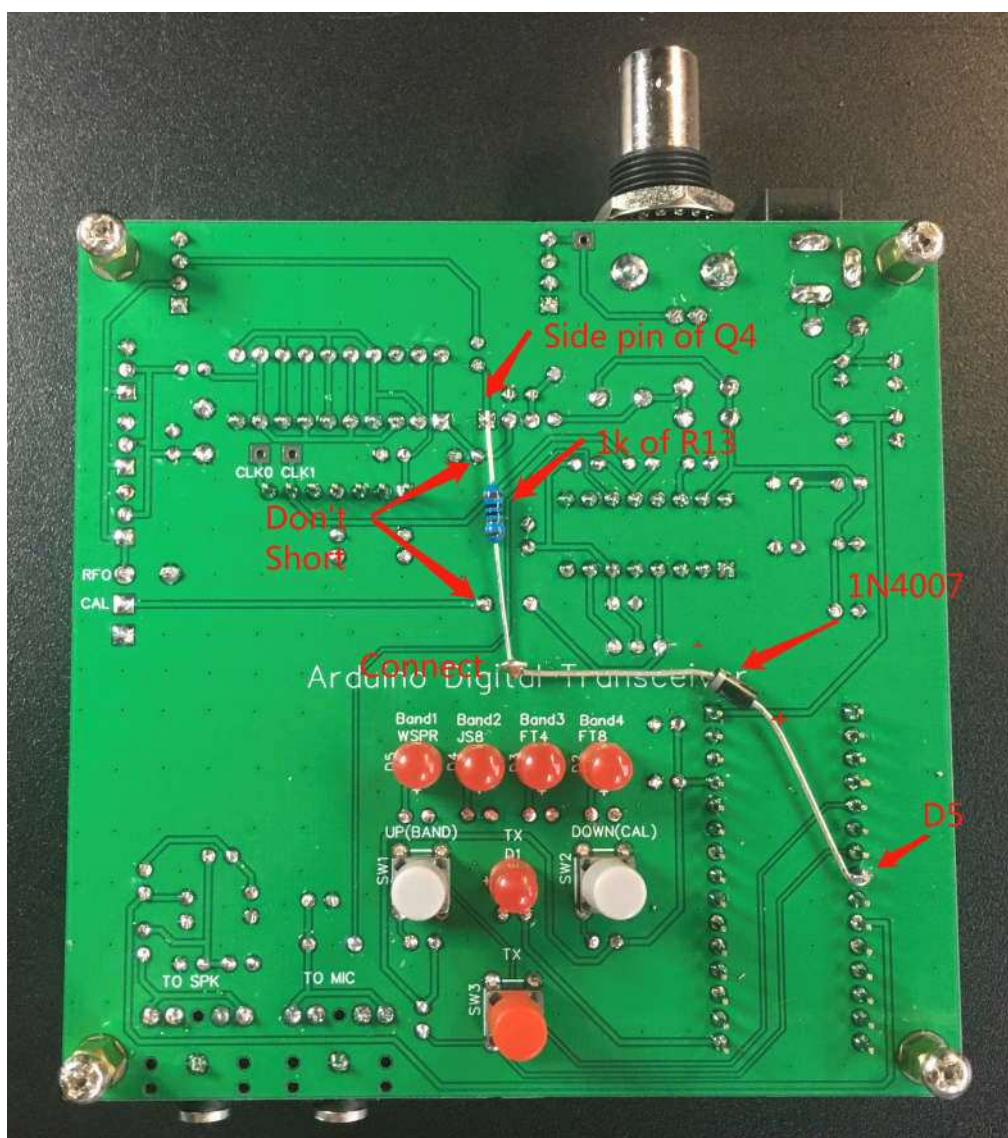
- RX フロントエンドの固定減衰器を制御するボタンの追加
- CD2003 の AGC 機能の追加
- RX 部受信感度の向上のために、BFO の振幅の低減とファームウェアの更新
- CAT モードで AM 放送を受信可能にする(高インピーダンスのヘッドホンを推奨します)

改造のための部品表です。

- ダイオード IN4007 を 2 個
- キャパシタ 10 μ F を 1 個
- 赤色 LED を 1 個 (3mm Φ でも 5mm Φ でも OK です)



赤色 LED は CLK2(BFO)の振幅を低減するために使用しますが、同時に CLK2(BFO)が出力されていることを示します。10 μ F により CD2003 の AGC 機能が復活します。R13 の代わりに 1N4007 を付ければ、RX フロントエンドの減衰器になります。



R13 は元々は 1k の抵抗でした。ボタン UP と DOWN を同時に押すと、D5 ピンを H または L に切り替えますが、これが ATT の切り替え機能として動作します。

この機能を使用するためには、ファームウェアを V1.2RX 以降に更新する必要があります。

<https://groups.io/g/crkits/files/ADX-S> からファームウェアをダウンロードし、Arduino IDE により更新します。

使用方法:

	SW1 (UP)	SW2 (DOWN)	SW3 (TX)	SW1 とSW2を同時
通常動作モード	モード変更	モード変更	TX	ATT の on/off
CAT モード	-5kHz	+5kHz	AM モードの ON/OFF(改造により追加された赤色 LED が、BFO の on/off を示します)	ATT の on/off

WSJT-x の周波数リストに、AM 放送周波数を追加してください。

4 バンドから7 バンド化への改造

標準の4バンドは40m、20m、15m それに10m です。今回、30m、17m と12m を追加するアップグレードキットを用意しました。ファームウェアはV1.3以降に更新する必要があります。新ファームウェアでは、前章で説明したように、受信部の改造も同様に追加されます。それにより、受信部の選択性が向上します。ありがたいことに、アップグレードキットには改造用の部品が含まれています。

新ファームウェアでは周波数テーブルは次に示すように更新されます。

(kHz)	40m	30m	20m	17m	15m	12m	10m
WSPR	7038.6	10138.7	14095.6	18104.6	21094.6	24924.6	28124.6
JS8	7078	10130	14078	18104	21078	24922	28078
FT4	7047.5	10140	14080	18104	2114	24919	28180
FT8	7074	10136	14074	18100	21074	24915	28074

電源投入時のLEDの点滅により、以下に示すようにバンド設定を表示します

B1 – 40m 変更なし
B1&B2 – 30m 新規追加
B2 – 20m 変更なし
B2&B3 – 17m 新規追加
B3 – 15m 変更なし
B3&B4 – 12m 新規追加
B4 – 10m 変更なし

バンドフィルタの部品表を以下に示します。

LPF BAND MODULE										
Band	C1	C2	C3	C4	C5	C6	C7	L1	L2	L3
40m	120+10	470	470	1000	300	1000	100nF	12T/BLACK	15T/RED	10T/RED
30m	100	330	330	680	220	680	100nF	12T/BLACK	12T/RED	8T/RED
20m	68	470		470	150	470	100nF	12T/BLACK	10T/RED	7T/RED
17m	91	360		360	120	360	100nF	12T/BLACK	11T/YELLOW	7T/YELLOW
15m	68	300	15	300	100	300	100nF	12T/BLACK	11T/YELLOW	7T/YELLOW
12m	47	270		270	62	270	100nF	12T/BLACK	10T/YELLOW	6T/YELLOW
10m	39	240		240	75	240	100nF	12T/BLACK	9T/YELLOW	6T/YELLOW
Unit: pF by default										

追加ドキュメント

本ドキュメントは、他のドキュメントと共に、以下のフォルダにアップロードする予定です。

メンバーでない場合は、登録が必要です。

<https://groups.io/g/crkits/files/ADX-S>

または、スマートフォンでこのQRコードをスキャンしてください。

参考情報と謝辞

Barb, WB2CBA

<https://antrak.org.tr/blog/adx-arduino-digital-transceiver/>

<https://github.com/WB2CBA/ADX>

Burkhard Kainka, DK7JD

[https://www.elektronik-](https://www.elektronik-
labor.de/HF/FT8QRP.html)

[labor.de/HF/FT8QRP.html](http://elektronik-
labor.de/HF/SDRtxFSK2.html) [http://elektronik-](http://elektronik-
labor.de/HF/SDRtxFSK2.html)

[labor.de/HF/SDRtxFSK2.html](http://elektronik-
labor.de/HF/SDRtxFSK2.html)

JE1RAV

<https://github.com/je1rav/QP-7C>

JA9TTT

<https://ja9ttt.blogspot.com/2018/07/short-wave-radio-design-2.html>

QRPGUYS

<https://qrpguys.com/qrpguys-digital-fsk-transceiver-iii>

Jason Mildrum, NT7S

<https://github.com/etherkit/Si5351Arduino>

Peter Parker, VK3YE

<https://www.youtube.com/watch?v=FGH0kUvkqX0>

BD4AHS, BG7IKK, BH3RQW, BD6EC, BG7IBQ, BD8TE, BG8NSK, BA7MPY、その他、に感謝します。