

Arduino K3NG CW キーヤー 組み立て説明書 V2 版

これは K3NG OM が配布している Arduino による CW キーヤー・スケッチに対応した CW キーヤー基板で、Arduino のマイコン ATmega328 も内蔵しており、Arduino Uno や Arduino 互換基板がなくても動作します。K3NG CW キーヤーが実現している機能はとても幅広く、このうちこの基板が対応している機能は次の様になります。

このキーヤーで出来る事：

- アイアンビック A/B モードの切り替え
- Ultimatic モード (内容が良く分かりません(-_-;))
- サイドトーン (モニター音) 周波数の調整
- Bug モードのサポート
- 送信イネーブル/ディセーブルの切り替え
- 単点・長点の比率調整
- パドル反転可能
- サイドトーンのオン/オフ
- メモリー機能 (4つまで)
- チューニングモード
- スピード調整の有効・無効切り替え
- 自動スペースのオン/オフ
- 送信せずにメモリー再生
- 送信機の切り替え機能 (TX1 / TX2/TX3)
- スピード調整用 VR

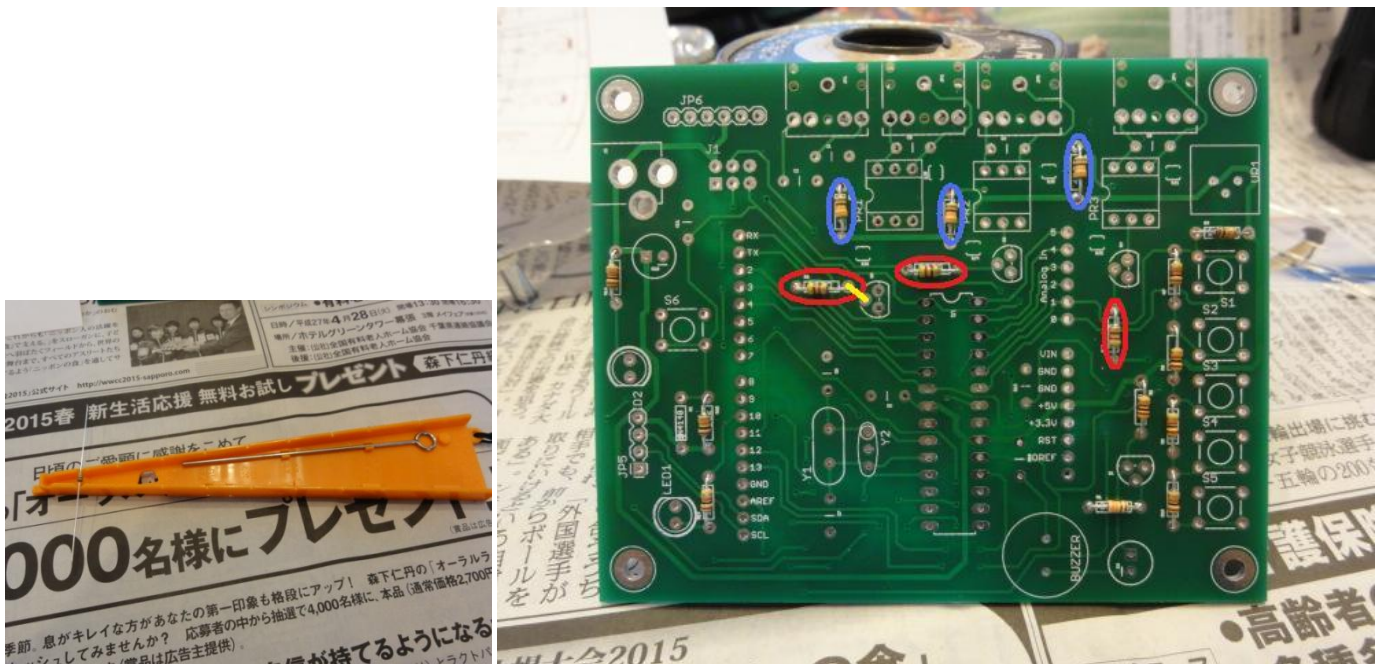
基板が完成したら USB-シリアル変換基板を通して PC に接続し、Arduino IDE を起動します。IDE からプログラムの定義ファイルを編集して、コンパイル、ダウンロードすれば直ぐに CW キーヤーが動作し始めます。今回はプログラムを事前書きこんだ ATmega328 を用意していますので、スケッチのコンパイル、ダウンロードは不要です。

I2C 接続の LCD ディスプレイを接続できる端子を設けているので、外部に LCD ディスプレイを接続する事もできます。ただし、サポートされているのは Adafruit の形式のもので。

1. 組立手順

① 抵抗の取り付け

図 1



抵抗は図のような折り曲げ治具を使って足を折り曲げるときれいに曲げられます。

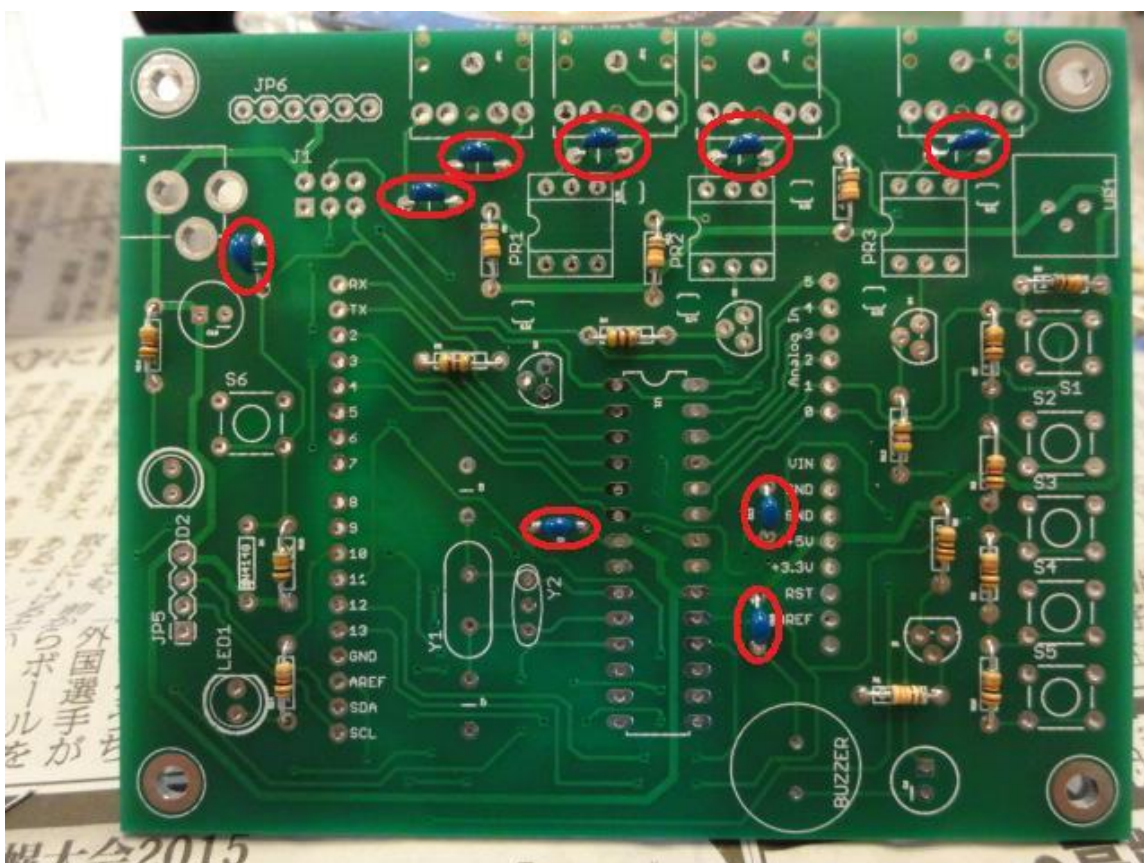
図の赤丸のところは 470Ω の抵抗を取り付けてください。青丸のところは 330Ω の抵抗を取り付けてください。

左端の赤丸のところにある黄色い線は、トランジスタを取り付けた後でジャンパ配線を行います。

それ以外の抵抗は、基板のシルクに書かれている通り $1k\Omega$ 6本、 100Ω 1本、 330Ω 1本、 $10k\Omega$ 2本、計 10本の抵抗を半田付けします。

② セラミックコンデンサの取り付け

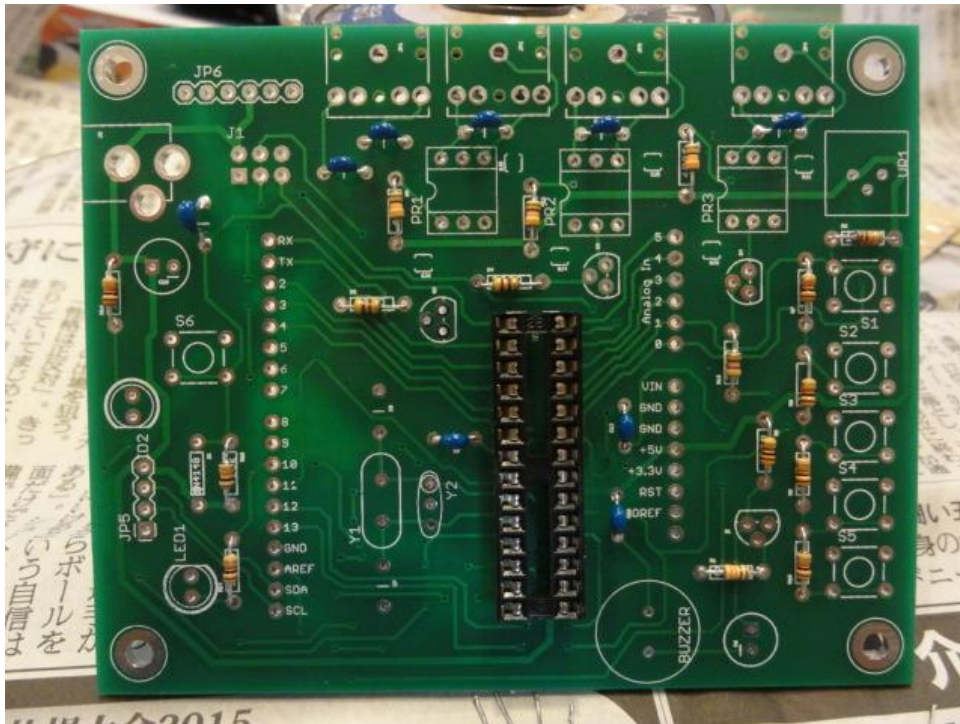
図 2



図の赤丸の位置に $0.1\mu\text{F}$ の積層セラミックコンデンサを 9 個、半田付けします。

③ IC ソケットの取り付け

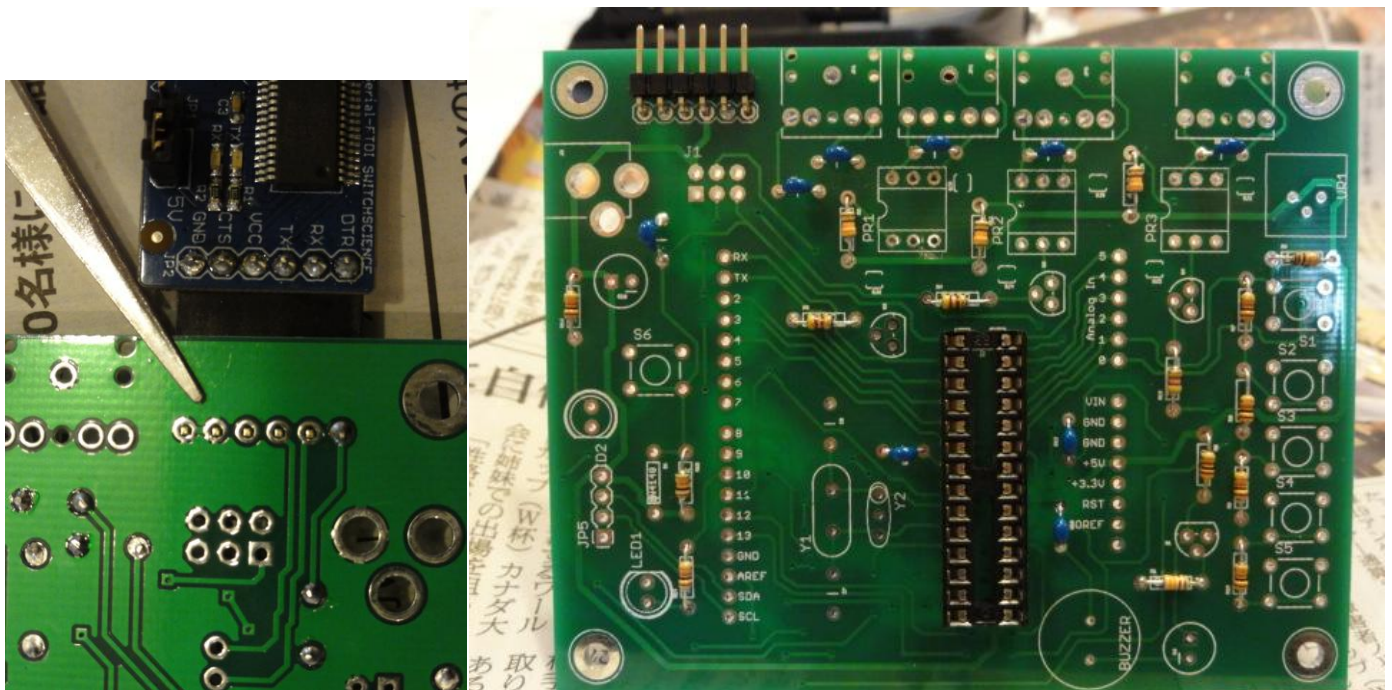
図 3



IC ソケットを取り付けます。

④ プログラム用コネクタの取り付け

図 4



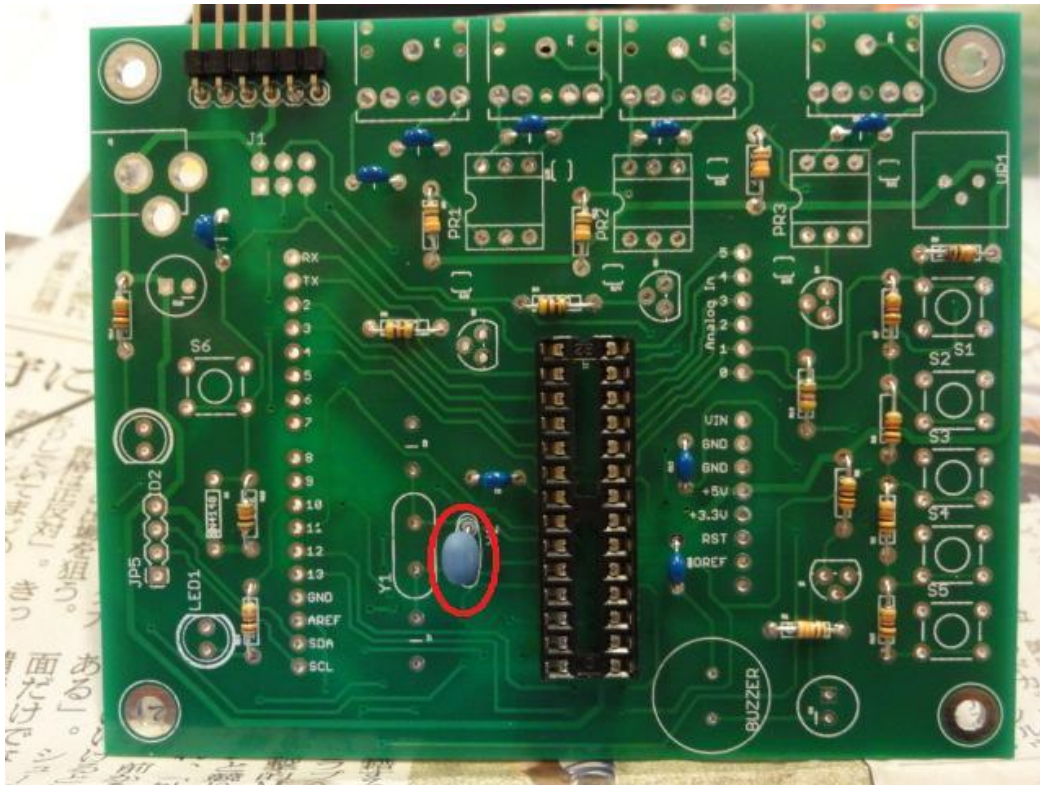
プログラム用に USB-シリアル変換基板を接続するコネクタを取り付けます。

図のように押し開くピンセット等があればそれでコネクタを固定し、半田付けします。

この時、コネクタと基板の間に USB-シリアル変換基板が入る隙間が残るように気を付けてください。

⑤ セラミック発振子の取り付け

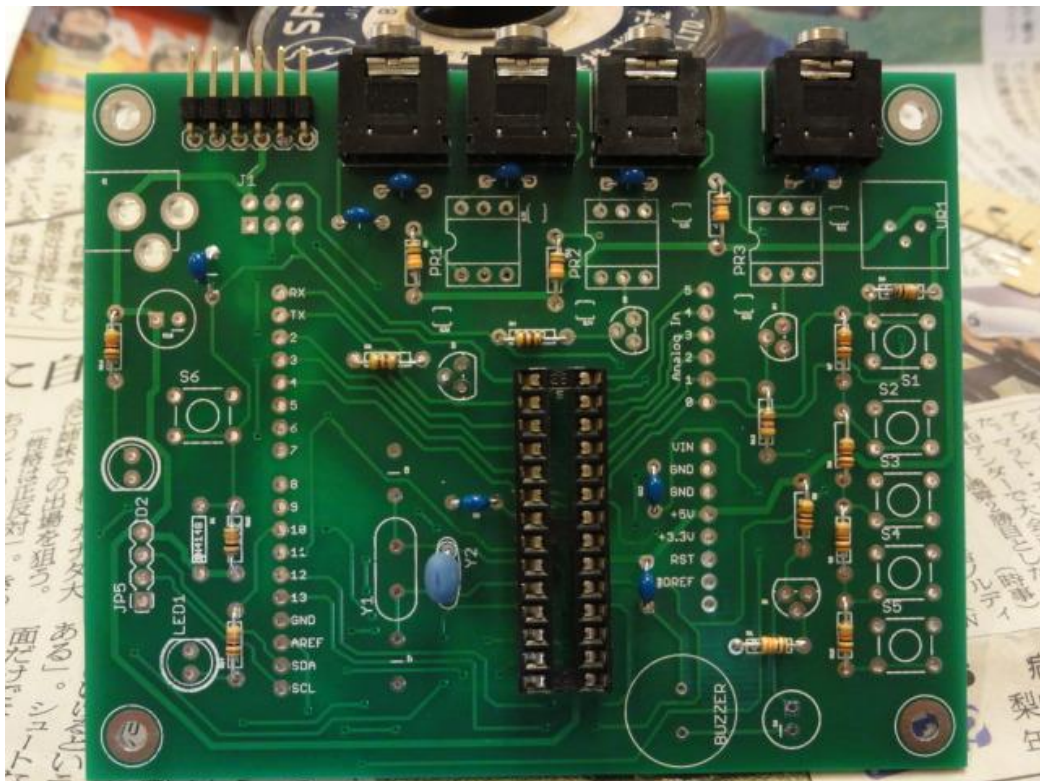
図 5



赤丸のところにセラミック発振子を取り付けます。基板は水晶発振子と 20pF のセラミックコンデンサを取り付けられるパターンも用意してありますが、今回はコンデンサも内蔵した 16MHz のセラミック発振子を使います。

⑥ 3.5mm ミニジャックの取り付け

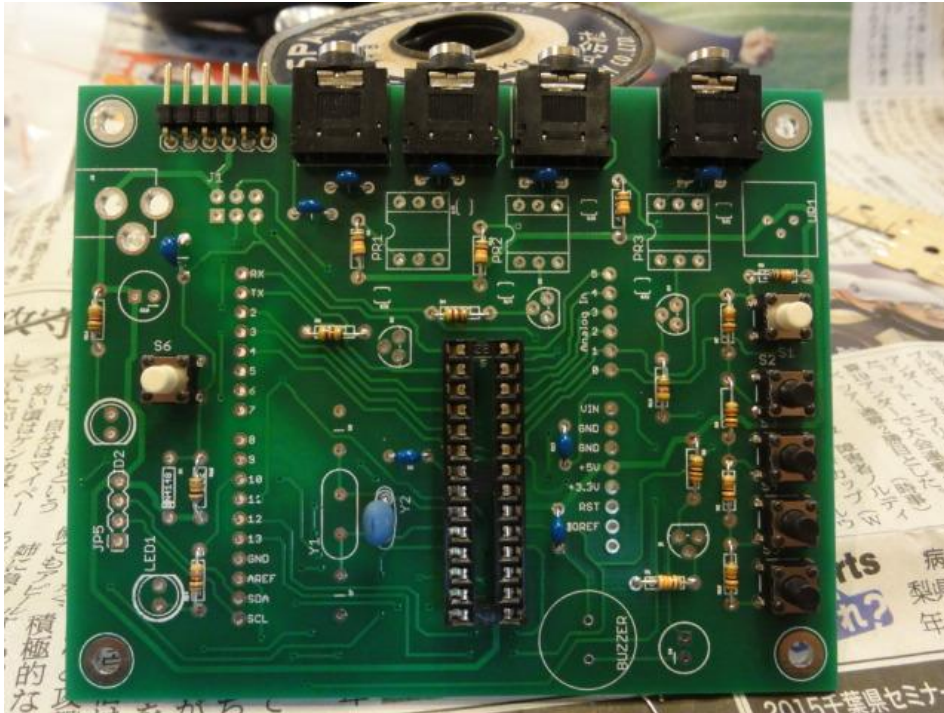
図 6



パドルと送信機のキー入力をつなぐ 3.5mm 用ミニジャック 3個とパドルをつなぐ 3.5mm 用ミニジャックを半田付けします。コネクタと同じく、手を離すとばね性で部品を押さえられるようなピンセットを使うと便利です。

⑦ タクトスイッチの取り付け

図 7



タクトスイッチを基板に挿入します。タクトスイッチの端子は爪状になっており、基板の Pad に合わせて挿入すると自分で基板に固定されます。この時、向きを間違とうまく入りません。端子の爪がきちんと Pad に合っている事を確認してから押し込みます。爪がうまく入っていないと足が折れ曲がってしまいますから注意してください。

タクトスイッチを基板と平行になるまで押しこんだら裏から半田付けします。

⑧ Photo MOS Relay、トランジスタ、ダイオードの取り付け

図 8

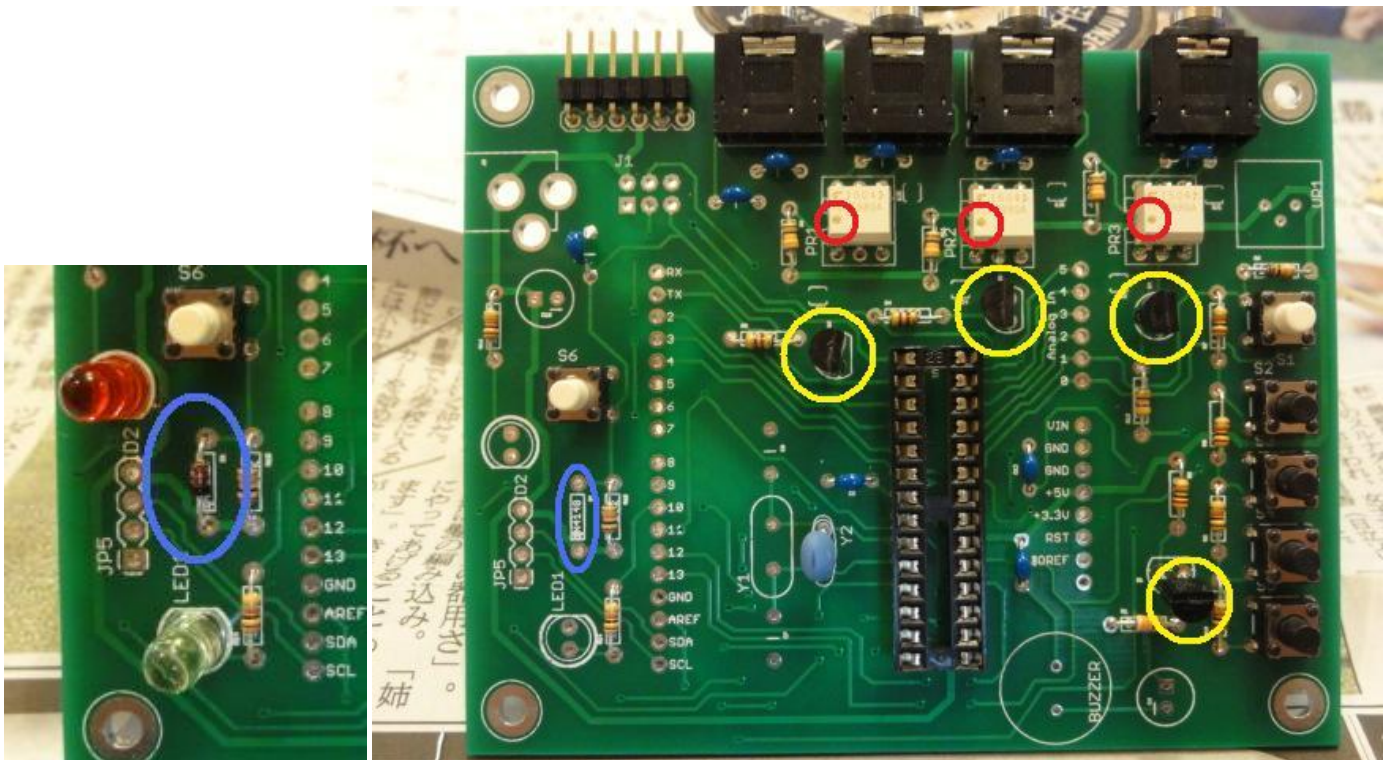


Photo MOS Relay は出力を電氣的にアイソレートしたい場合や真空管送信機など、高電圧のキーイングが必要な場合に使ってください。

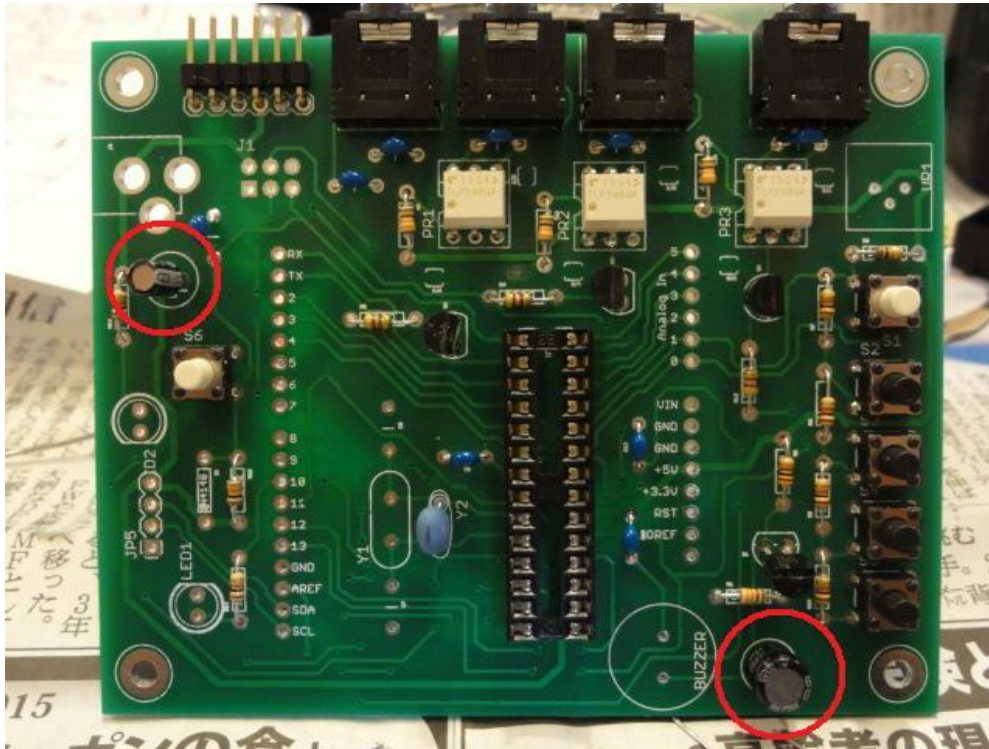
赤丸のところに 1pin マークが来るように挿入して半田付けします。

青丸のところにダイオードを半田付けします。この時、カソードが図では下向きに来るように取り付けます。

黄色の丸のところにトランジスタを半田付けします。左端のトランジスタにはベースにジャンパ配線を追加します。こちらにも向きがありますので、パッケージの平らな面がシルク印刷と合うように取り付けてください。

⑨ 電解コンデンサの取り付け

図 9



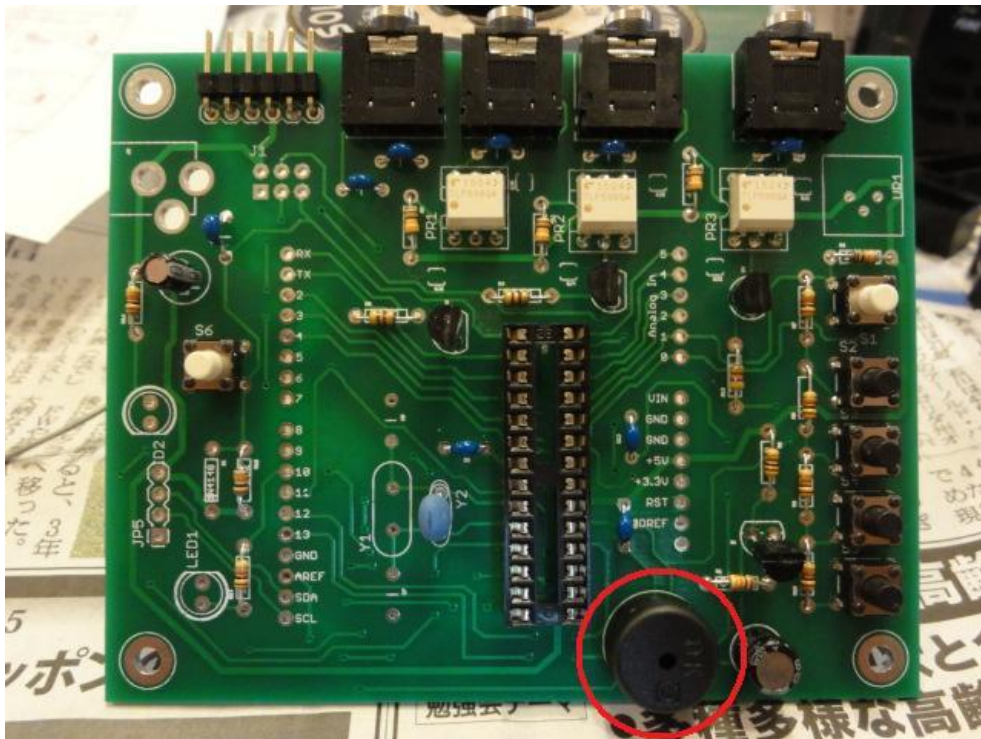
図の右下、赤丸の位置に $100\ \mu\text{F}/16\text{V}$ の電解コンデンサを半田付けします。

電解コンデンサは極性がありますので、基板上では四角い Pad に足の長い方が入るようにします。

図の左上、赤丸の位置に $10\ \mu\text{F}/16\text{V}$ の電解コンデンサを半田付けします。

⑩ 圧電スピーカーの取り付け

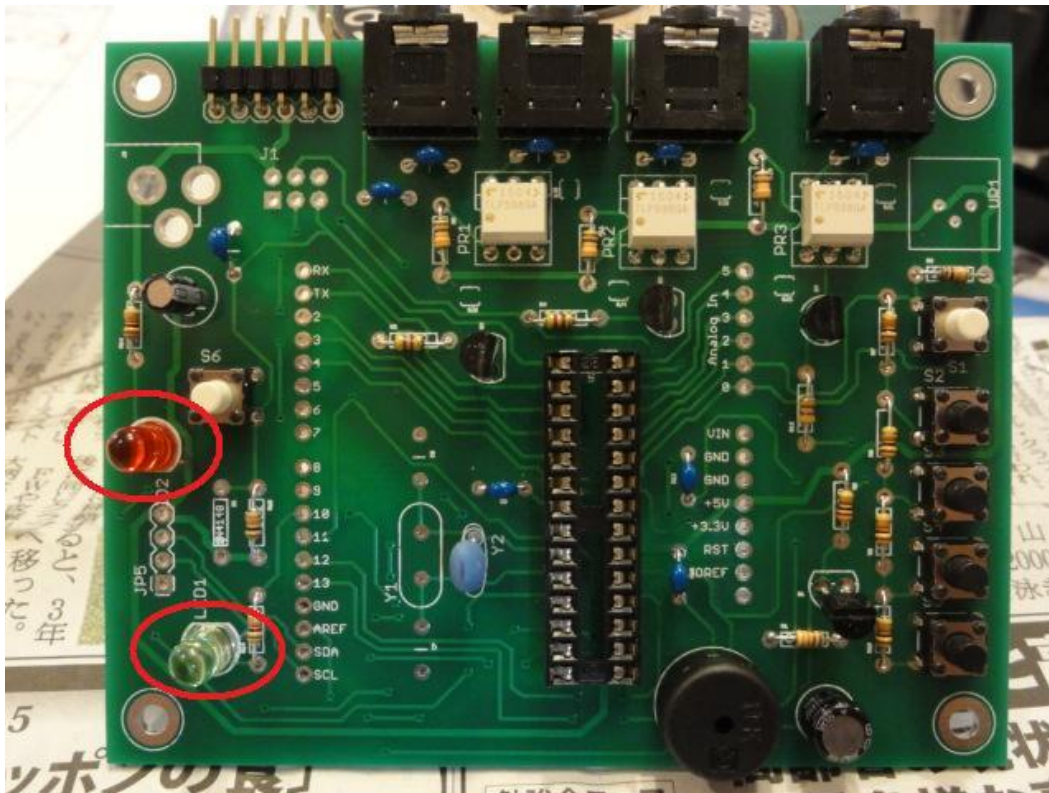
図 10



圧電スピーカーには極性がないので、そのまま半田付けします。

⑪ LEDの取り付け

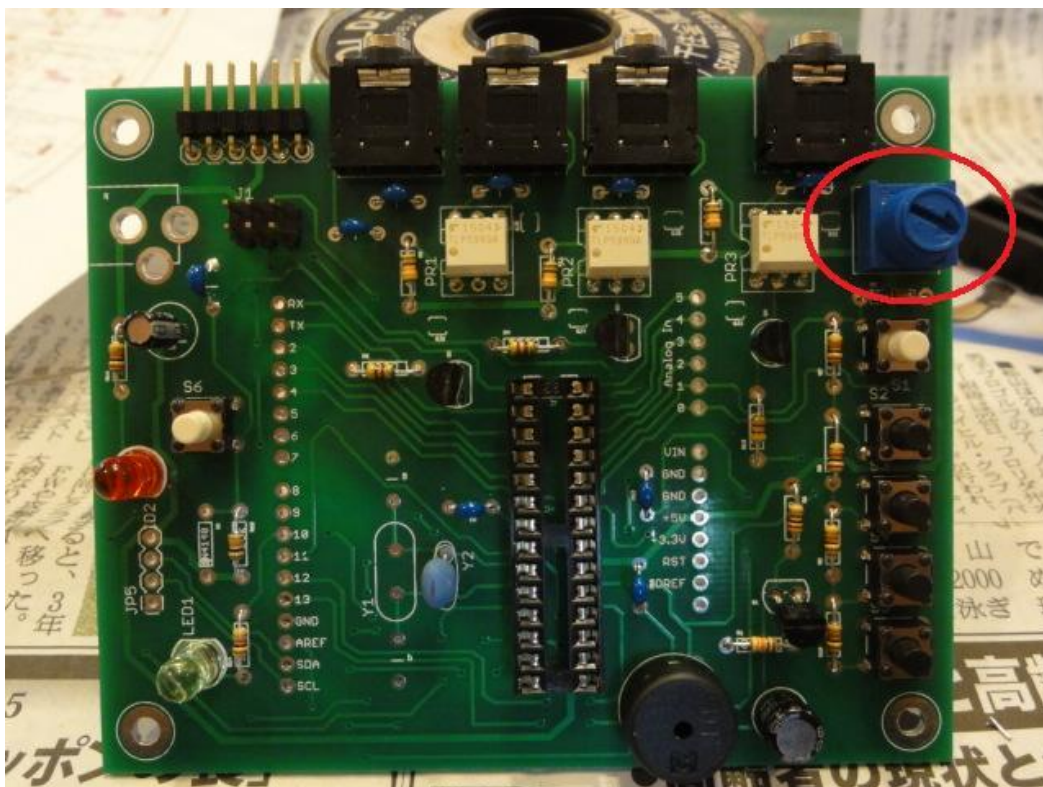
図 11



赤丸のところに赤色と緑色の LED を取り付けます。
こちらもありつけの向きがありますからシルクの切り欠き面と LED の切り欠き面を合わせて取り付けます。

⑫ VR の取り付け

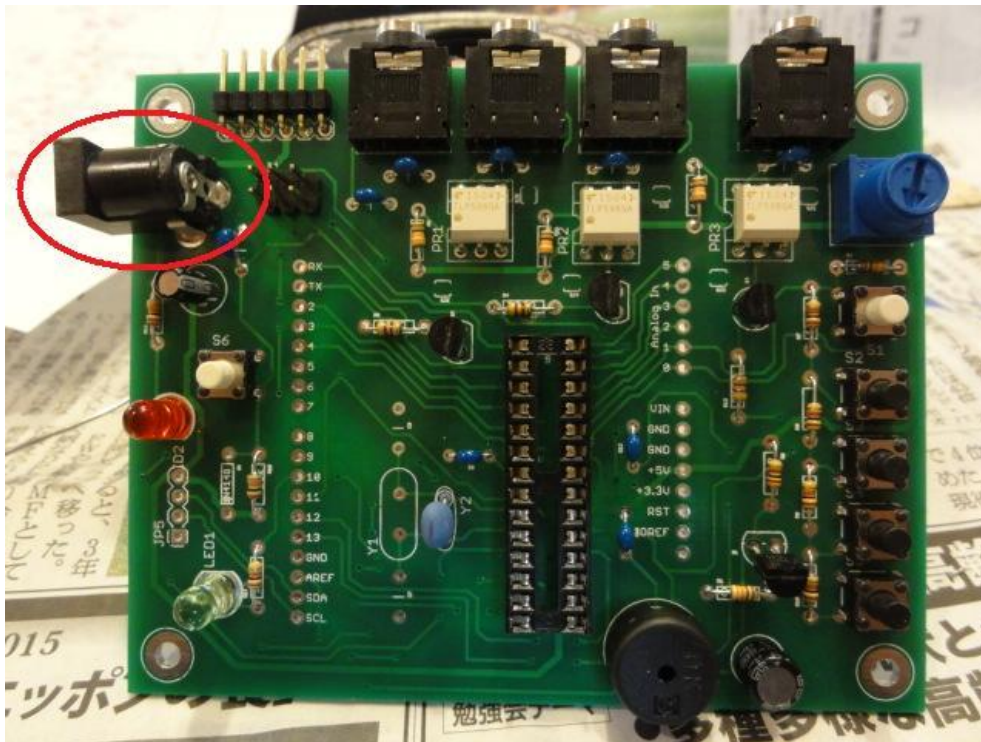
図 12



基板にスピード調整用の半固定 VR を取り付けます。

⑬ 電源コネクタの取り付け

図 13



電源コネクタを取り付けます。電源は5Vで、センタープラスです。コネクタは2.1mm標準DCジャックです。プラグは2.1mmDCプラグを使ってください。

これで完成です。

2. K3NG キーヤー・スケッチの修正箇所

K3NG OM のサイトからソース管理のサイトに移って最新版のソース・ファイルをダウンロードします。ダウンロードしたら、次の箇所を修正してコンパイルします。問題なくコンパイル出来たら Arduino 基板にダウンロードして、キーヤーを動かしてみます。

setting change log

(1) Enable Command Botton, Memory feature and Speed controll potentiometer

直すファイル : **keyer_features_and_options.h**

```
#define FEATURE_COMMAND_BUTTONS // do not enable unless you have the corresponding resistors  
connected to the analog_buttons_pin
```

```
#define FEATURE_MEMORIES
```

```
#define FEATURE_POTENTIOMETER // do not enable unless you have a potentiometer connected,  
otherwise noise will falsely trigger wpm changes
```

コマンド・ボタンの使用とメモリ機能、スピード調整 VR の使用を設定しています。

(2) Enable Command Mode LED

直すファイル : **keyer_pin_settings.h**

```
#define tx_key_line_1 11 // (high = key down/tx on)
```

```
#define tx_key_line_2 12
```

```
#define tx_key_line_3 13
```

これで送信機 3 台までをキーヤーのボタンで切り替えて使う事ができます。

```
#define command_mode_active_led 10
```

この指定により、コマンド・モードにある時は Arduino 基板上の緑色の LED が点灯します。

(3) Initial setting of WPM and Sidetone Frequency, Number of Memories and WPM setting range

直すファイル : **keyer_settings.h**

```
#define initial_speed_wpm 13 // "factory default" keyer speed setting
```

```
#define initial_sidetone_freq 1600 // "factory default" sidetone frequency setting
```

```
#define analog_buttons_number_of_buttons 5
```

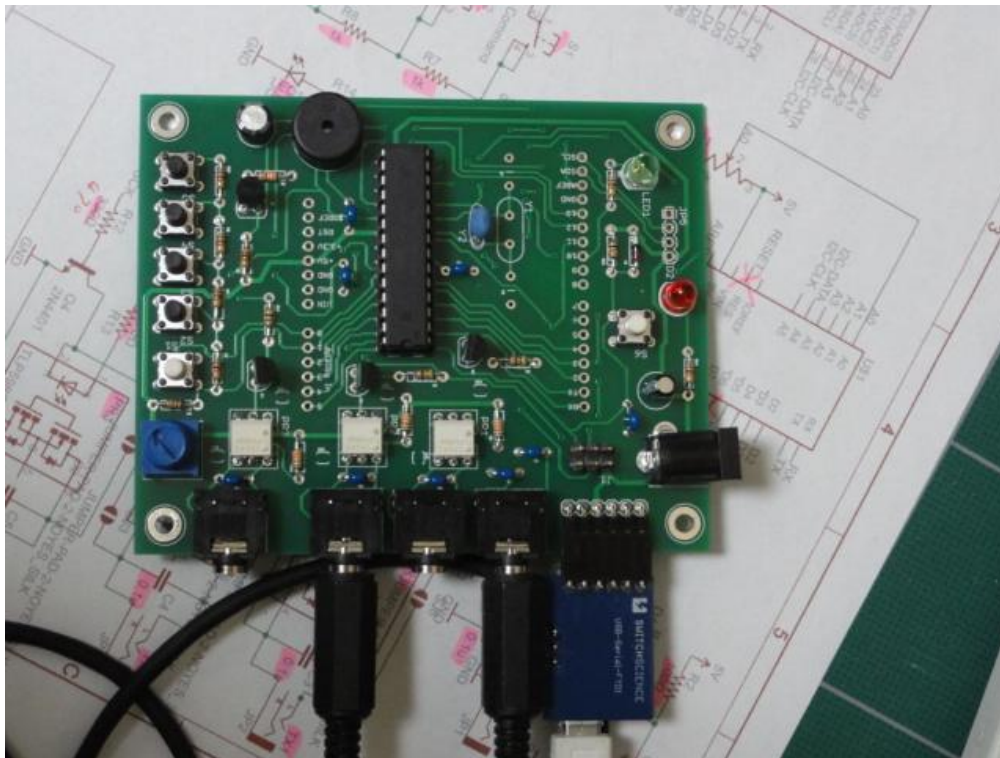
```
#define initial_pot_wpm_low_value 5 // Potentiometer WPM fully CCW
```

```
#define initial_pot_wpm_high_value 30 // Potentiometer WPM fully CW
```

ここで初期状態のスピードとスピードの可変範囲、サイドトーンの周波数を設定しています。

3. 動作確認

図 14



プログラムがダウンロードされ、起動されるとブザーから.... ..と起動音が聞こえます。ここでパドルをつないで操作するとモニターできる事を確認します。次に、送信機（トランシーバ）をつないでキーイングできる事を確認して見てください。

キーヤーの動作を変えるコマンドは [K3NG OM](http://K3NG.OM) のサイトに説明が出ています。このキーヤーでは4つまでのメモリが使えます。

4. コマンド一覧

コマンドを入力するには、コマンド・ボタンを押して、次のコードをパドルから入力します。コマンド・モードの間は基板の緑色の LED が点灯しています。

- A - Switch to Iambic A mode
- B - Switch to Iambic B mode
- D - Switch to Ultimatic mode
- E - Announce the speed in WPM
- F - Adjust sidetone frequency
- G - Switch to bug mode
- I - TX enable / disable
- J - Dah to dit ratio adjust
- N - Toggle paddle reverse
- O - Toggle sidetone on / off
- P# - Program a memory (#には数字をパドルから入力します)
- S - Alphabet Send Practice
- T - Tune mode
- V - Toggle potentiometer active / inactive
- W - Change speed
- X - Exit command mode (you can also press the command button (button0) to exit)
- Z - Autospace On/Off
- # - Play a memory without transmitting

コマンドの実行にはコマンド SW を押して、パドルからコマンドの文字を打ちます。コマンドが受け付けられると、ピープ音が ”ピッ” と鳴り、次にそれぞれのコマンドに応じた動作をします。

メモリー機能を使うには、コマンド・ボタンを押して、P1 と打つと ”ピッ” と音がするので、登録するキー・ストロークを打ちます。終わったらコマンド・ボタンを押すと、登録した内容が再生されます。再度、コマンド・ボタンを押してコマンド・モードから抜けます。メモリーした内容を再生するには、それぞれのボタンをチョンと押すと、再生されます。

コマンド・モードの状態は Arduino の D10 に接続された緑色 LED で表示されており、コマンド・モードでは点灯状態になります。

このシールドでは送信機を 3 台接続して、切り替えながら使えるようになっています。それぞれ該当するメモリーのボタンを長押しして離すと、ボタンに応じて TX1 とか、TX2 とか、TX3 とか、切り替えた結果を返してきます。これで、一つのパドルで 3 台の送信機(トランシーバ)を切り替えて使えます。