

# F2A 発振器 マニュアル

J11BXM/Dialectech®

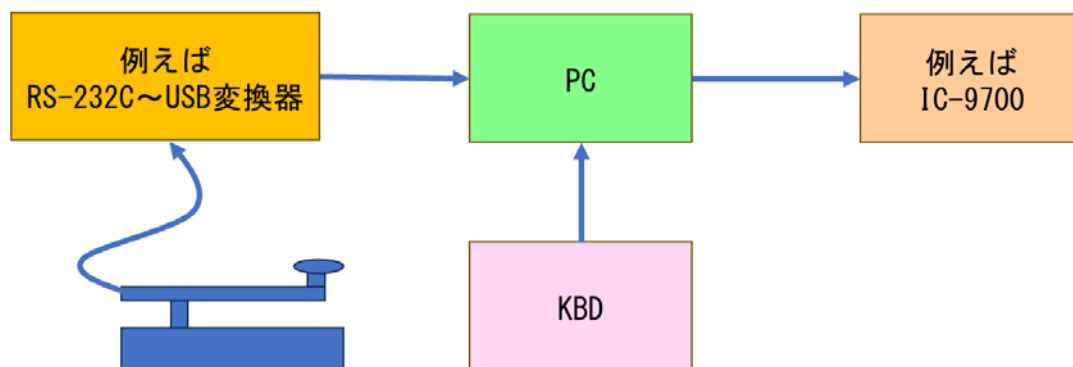
2024/10/17

## 目次

F2A 発振器 マニュアル .....	1
1. 本プログラムを動作させる際の構成と動作.....	2
2. 操作面からの仕様.....	3
2.1. プログラムの起動.....	3
2.2. プログラム実行時に表示される UI(User Interface)とその意味 .....	3
2.2.1. 操作タブ .....	3
2.2.2. 設定タブ .....	6
2.3. 運用手順 .....	7
3. プログラムの構成について .....	8
3.1. 構成概要 .....	8
3.2. 全体のオブジェクト構成.....	9
3.2.1. App オブジェクト (2) .....	9
3.2.2. Main オブジェクト (1) .....	9
3.2.3. CUIController オブジェクト (5) .....	9
3.2.4. CComCenter オブジェクト (8) .....	9
3.2.5. CKeyHandler オブジェクト (4) .....	10
3.2.6. CSendReveiveController オブジェクト (3) .....	10
3.2.7. CToneGenerator オブジェクト (6) (7) .....	10
4. 現時点で未だ実装していない機能と発生している問題点 .....	11

## 1. 本プログラムを動作させる際の構成と動作

本プログラムは PC に USB 接続されたキー情報、若しくはキーボード中の「↑」「↓」などを電鍵として入力し、同 PC に USB で接続された iCOM 製のリグに対して F2A 信号を送出するようになっています。



iCOM 製のリグに限ったのは、CI-V を使って PTT 制御をするためです。YAESU 系のリグは CAT を使えば制御できるので、プログラムを変更すればよいと思います。

IC-9700 の場合、一本の USB ケーブルで接続すれば、2 つの COM ポートが開き、そのうちひとつは CI-V でコマンドを送ることができ、もうひとつのポートは AUDIO チャンネルが生成されます。本プログラムはその両方のチャンネルを使って F2A の送信をします。

USB 接続されたキーとは、JI1JDI さん作成の DitDahChat に接続する RS-232C～USB コンバータ経由で接続されたキーなどがこれに当たります。

具体的には、RS-232C～USB コンバータの RS-232C 側で、4 番 PIN と 5 番 PIN の間に電鍵を繋がればよいことになります。RS-232C ポートが実装されている PC を使っている場合には、USB を経由せずとも、そのポートの 4 番 PIN と 5 番 PIN の間に直接電鍵を繋がればよいことになります。

本プログラムは右図のように、キー入力があると、キーが押されたときには、

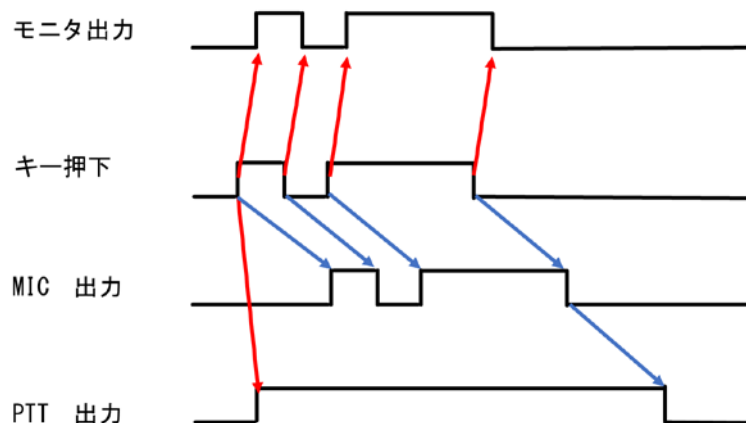
- ・モニター
- ・MIC 出力
- ・PTT 出力

を行うように動作します。

この際、キー押下と同

時に PTT を ON にします。また、モニターとして PC に接続されたスピーカーにも発信音を出力します。一方、指定した時間遅れの後に発信音はリグのマイク入力に送られます。

キー入力が途絶えたら、指定した時間の後、PTT を OFF にします。



## 2. 操作面からの仕様

### 2.1. プログラムの起動

本プログラムは java で書かれていますので、作成された実行ファイルは、jar になっています。よって、これを起動するには、

```
"C:¥jdk-21.0.2¥bin¥java" -jar C:ProgramLocation¥f2a¥target¥f2a.jar
```

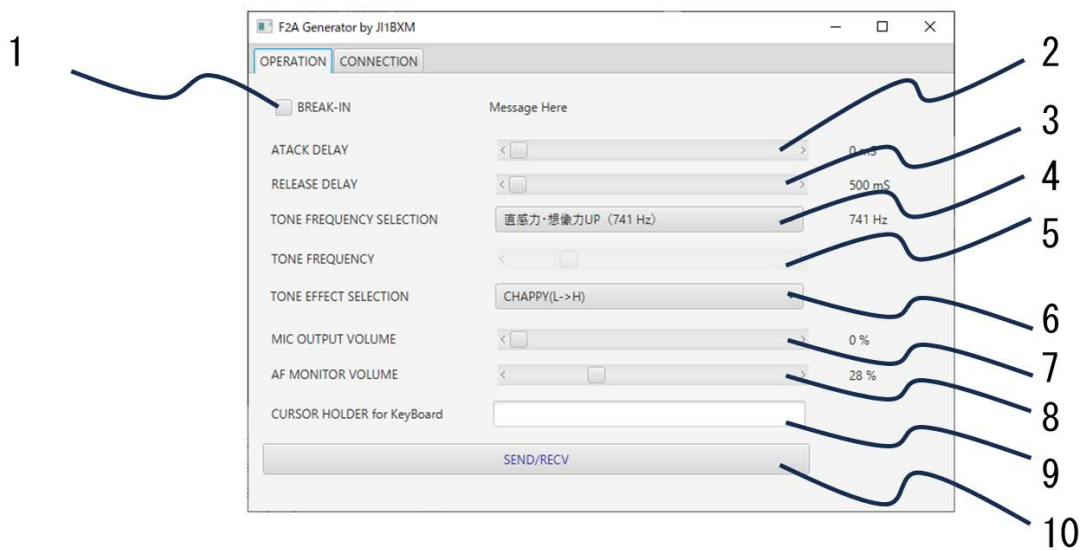
のようにします。

### 2.2. プログラム実行時に表示される UI (User Interface) とその意味

プログラムを実行すると、2つのタブを持つダイアログボックスが表示されます。このダイアログボックスを通して各種操作を行います。

#### 2.2.1. 操作タブ

操作タブ上には次のような選択肢等による操作 UI が表示されます。

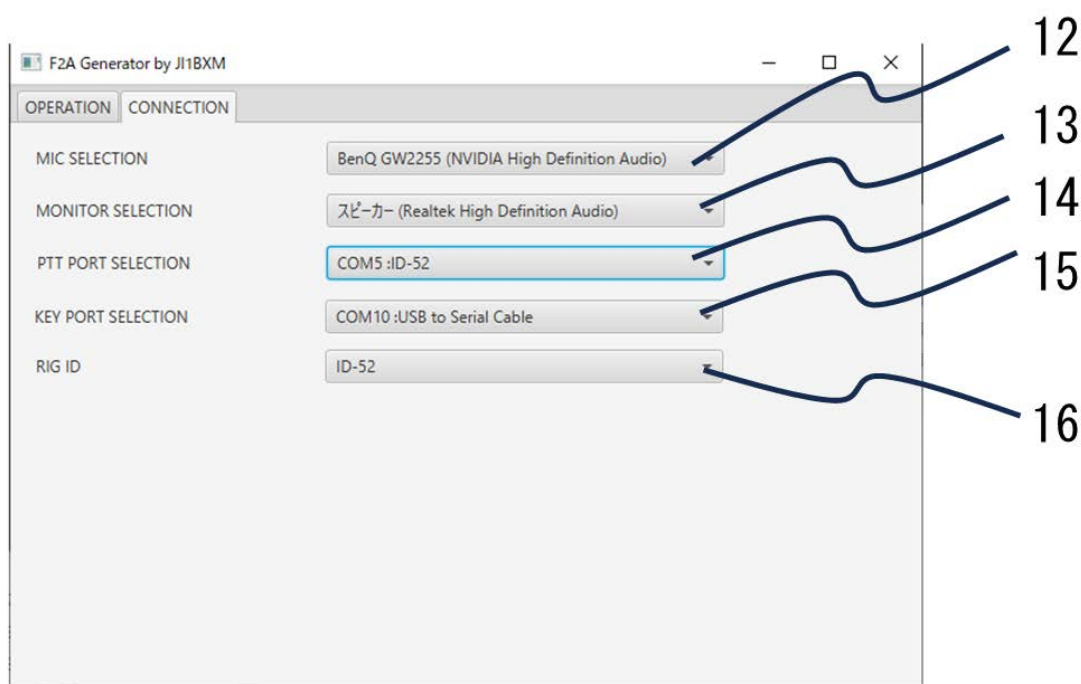


1	BREAK-IN。即ち、PTT 操作についての指定。チェックがなければ SEND/RCV ボタン (10) で PTT を操作することになります。チェックがあれば、セミブレイクイン動作をすることになります。
2	ATAACK DELAY。これは、キーを押してからどの程度遅れてトーンを発生させるかを指定します。100mS 単位で、2 秒まで遅らせることができます。ブレイクイン動作の際には、PTT オンしてからトーンが出るまでの遅れ時間となりますので、相手局のスケルチが開くまでの凡その時間を指定することになります。短すぎると相手局で頭切れのように聞こえ、長すぎると、相手局のスケルチが開いてから音が出るまでの時間が長すぎてイラつかせることになります。
3	RELEASE DELAY。これはブレイクイン動作の際、キーを離してから PTT をオフにするまでの時間を指定します。短すぎると、ひとつの伝文の途中でキャリアが途絶え、相手方のスケルチが ON/OFF が頻繁に繰り返されることになります。一方、長すぎると、伝文が終わってもなかなかキャリアが落ちずに、相手が伝文を送る順番で長く待たせることになります。
4	TONE FREQUENCY SELECTION。プリセットされた音程群からトーンとして発振する音程を指定します。現在プリセットされている音程は、所謂ソルフェジオ周波数になっています。仮にこれ以外の音程を発生させたいのであれば、選択肢の最後にある、「10Hz 毎」を選択します。これにより、次の TONE FREQUENCY スライドバーでの操作が有効になり、2kHz までの好きな周波数で発振させられます。
5	TONE FREQUENCY。TONE FREQUENCY SELECTION セレクトボックス (4) で「10Hz 毎」を選択したときに操作が有効となり、
6	TONE EFFECT SELECTION。発信音を効果音にする選択肢。

	<p>NORMAL は普通の正弦波となる。</p> <p>CHAPPY(H-&gt;L)は次第に高音から低音になる CHAPPY 音となる。</p> <p>CHAPPY(L-&gt;H)は次第に低音から高音になる CHAPPY 音となる。</p> <p>GRADUAL_ATTACK は発信開始部分での振幅が次第に大きくなる音となる。</p> <p>WHITE NOISE は、MARK が正弦波ではなく WHITE NOISE となる。</p>
7	MIC OUTPUT VOLUME。リグに接続したオーディオポートへの出力音量調整ができます。
8	AF MONITOR VOLUME。キーイング時に PC のスピーカーから出すモニター音の音量調整ができます。
9	キーボードの「↑」「↓」でキーイングするときには、カーソルをここにフォーカスしておきます。この操作でフォーカスが移動することがありません。
10	SEND/RECV。BREAK-IN チェックボックスにチェックが入っていないときに、送信・受信を切り替えるボタン。また、BREAK-IN にチェックが入っているときには、送信時にピンク色に着色されて送信中であることがわかります。

## 2.2.2. 設定タブ

設定タブ上には次のような選択肢等による設定 UI が表示され、各種設定をすることができます。



12	MIC SELECTION。現在 PC に接続されている出力ポートが全て列挙されているので、そのなかから、リグのマイクに相当するポートを選択する。
13	MONITOR SELECTION。現在 PC に接続されている出力ポートが全て列挙されているので、そのなかから、PC のモニタースピーカーに相当するポートを選択する。
14	PTT PORT SELECTION。リグの CI-V でコマンドを送ることができる PORT を指定します。
15	KEY PORT SELECTION。PC に接続するキーのポートを指定します。通常、COM ポートになっているので、PC に RS-232C のポートが実装されている場合には、それが含まれていますし、USB～シリアル変換ケーブルで USB に接続されている場合には、その PORT が表示されているはずなので、それを指定します。
16	iCOM のリグの名称を指定します。iCOM のリグでは、CI-V でアドレスの指定をすることになっていますので、そのアドレス（標準値）を選択することになります。もし、リグの CI-V アドレスをリグ側で変更していた場合には、通信できなくなります。

### 2.3. 運用手順

本プログラムを正しく動作させるためには、以下の手順に沿ってプログラムを起動させる必要があります。

1. P CとリグとをU S Bケーブルで接続する。
2. リグの電源を入れる。
3. P Cの電源を入れる。

ここまでの順序はいつでも良いです。続いて、

4. P Cで本プログラムを起動する。  
(プログラム起動時点でのみ、C O M、A U D I Oの接続状況を検査しますので、上記1～3の手順をプログラム起動後に行っても、正しく接続できません。)
5. 設定タブを選択して、5つのパラメータを設定する。  
(ここでしたパラメータは記録されますので、次に本プログラムを起動した際には、この手順は不要になります。)
6. 操作タブを選択して、操作を行う。

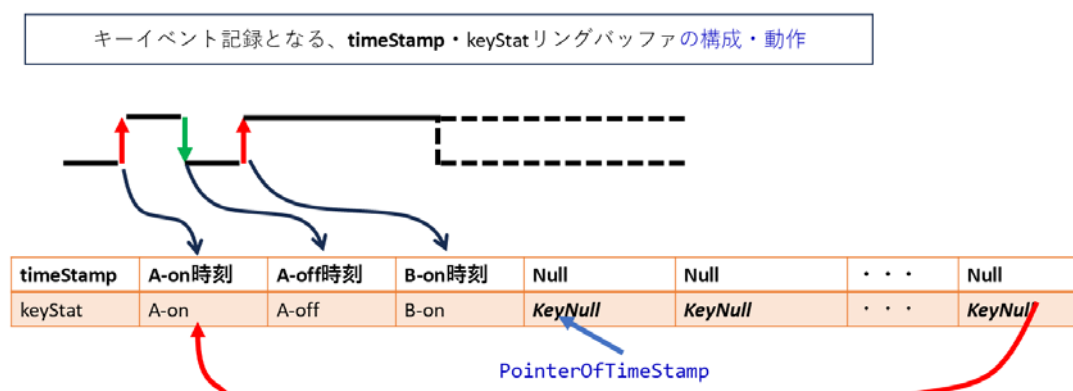
### 3. プログラムの構成について

#### 3.1. 構成概要

本プログラムでは、キー入力にあわせて発振し、モニター音としてPCのスピーカから出力するとともに、キーのON/OFFにあわせてPTTを操作します。また、キーから所定時間遅れてリグのMICに発信音を出力します。

この時間遅れをどのように実現するかは、プログラムの根幹にかかる設計事項となります。

本プログラムでは、キー入力があったときには、キーのONとOFFをそれぞれイベントとみて、そのイベントが生成した時刻をmS（ミリ秒）の単位でイベント履歴リストに記憶していきます。



イベント履歴リストはリングバッファで実現します。そして、イベントの内容を記録するkeyStat配列と、そのイベントの時刻を記録するtimeStamp配列とで構成しています。リングバッファですので、keyStat/timeStamp双方が配列を使い切ると、また最初から記録していくことになります。

この履歴リストを、

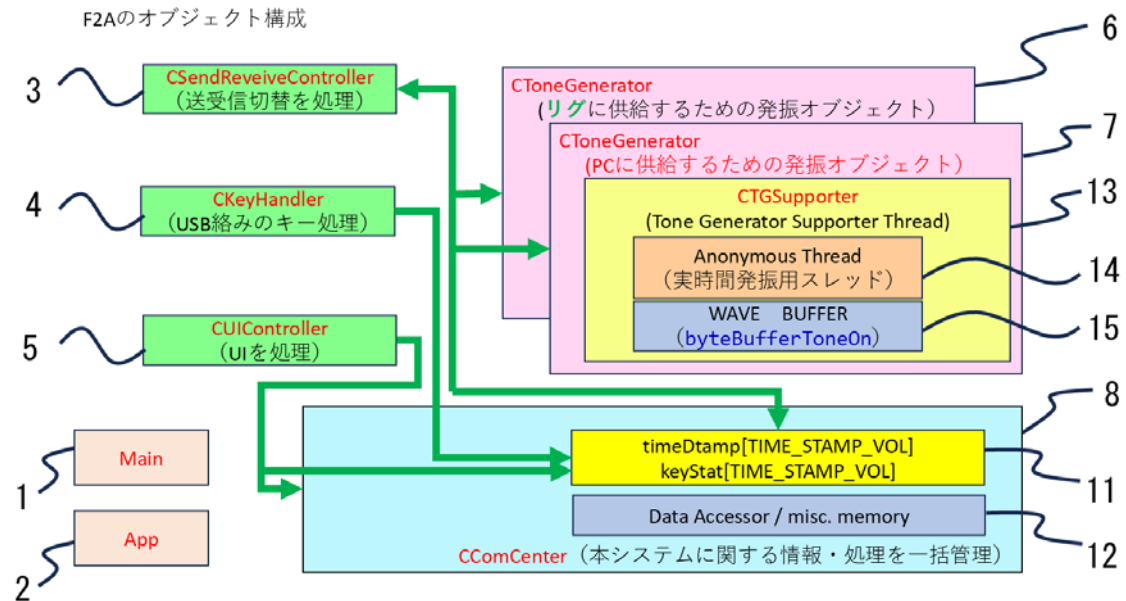
- 1) PTTを管理するオブジェクト、
- 2) モニターに宛てた音を生成するオブジェクト、
- 3) リグに宛てた音を生成するオブジェクト、

からなる3つのオブジェクトが参照し、それぞれが必要なタイミングでPTTのON/OFF、PCのモニターへの発信音の断続、リグへの発信音の断続を独自に行うようになっています。



## 3.2. 全体のオブジェクト構成

前節でも紹介したように、主として3つのオブジェクトがイベント履歴リストを参照する体系をとりますが、キー入力からこのイベント履歴リストにイベントを登録するオブジェクトが当然必要になります。また、UI を処理するためのオブジェクトなどが必要になります。これらを図示すると下図のようになります。



以下、各オブジェクトについて簡単に説明します。なお説明中（番号）とあるのは、上図中の番号に対応します。

### 3.2.1. App オブジェクト (2)

これは、JavaFx の仕様の都合で実装したオブジェクトです。Main オブジェクトを起動します。

### 3.2.2. Main オブジェクト (1)

JavaFx の UI 画面の表示、その他常駐オブジェクト (TONE 生成等) を生成し動作させるためのオブジェクトです。

### 3.2.3. CUIController オブジェクト (5)

JavaFx の UI を担当するオブジェクトです。UI 上タブがありますが、これらを一括して本オブジェクトが差配します。

### 3.2.4. CComCenter オブジェクト (8)

JavaFx では UI Thread のほかに別スレッドで Worker Thread を用意する必要がありま

す。この際、UI Thread と Worker Thread との間の通信手段が必要になります。これ以外にも本プログラムでは、イベント履歴リストを中心に複数のオブジェクトが独自の動作をしますので、これらの情報センターとして一括して情報を管理する必要があります。本オブジェクトはこのような情報センターとしての機能を持ちます。

このため、本オブジェクトには、timeDtamp、keyStat (11) のほかに、多くのデータアクセス (12) を持っています。

本オブジェクトは、Singleton で実装しています。

### 3.2.5. CKeyHandler オブジェクト (4)

CKeyHandler オブジェクトは、

### 3.2.6. CSendReveiveController オブジェクト (3)

CKeyHandler オブジェクトは、PTT の制御を行うオブジェクトです。このため、このオブジェクトは、CComCenter オブジェクト内のイベント履歴リスト (11) のチェックをループしています。そして、キーオンのイベントがあったときには即時に PTT を ON に、キーオフのイベントがあったときには、そこから Release Delay で設定された時間経過していればそのときに PTT を OFF となるように CI-V コマンドを送信します。

CI-V のコマンドを送信するようになっていますが、CAT のコマンドを送るようにすれば、YAESU のリグにも使えるようになるはずです。

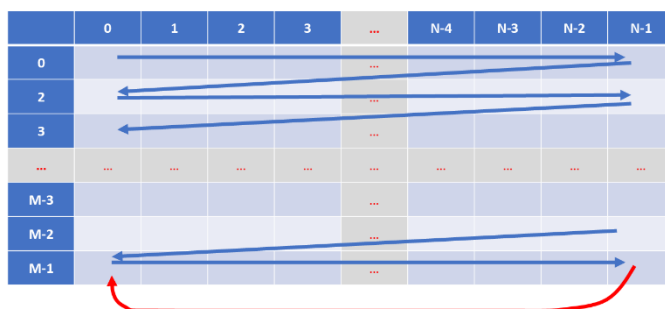
### 3.2.7. CToneGenerator オブジェクト (6) (7)

CToneGenerator クラスからは2つのオブジェクトを生成します。ひとつは PC に接続されたスピーカに対して TONE を送ります。もうひとつはリグの MIC に TONE を送ります。

本オブジェクトは、内部で CTGSupporter オブジェクト (13) を生成して利用します。CTGSupporter オブジェクト (13) 内では WAVE BUFFER (15) があり、ここに波形データを記憶させ、この波形データを出力するようになっています。

右図は WAVE BUFFER の構成を表したものです。WAVE BUFFER は有限範囲の二次元配列になっていて、1次元は信号の1波分の量が用意されます。もう1次元は、所定数 (現在は128個) となっています。そして、TONE を生成する際には、図中青矢印の

WAVE BUFFERとなる、byteBufferToneOnの構成・動作



順にデータを読み出していきます。そして最後の行まで再生されたら、赤矢印のように、その行のみを繰り返すようになっています。列が1波長分となっていますので、同じ行を繰り返しても、位相飛びがおこらないようになります。

実装にあたっては、内部で Anonymous Thread (14) を生成して、この Thread が WAVE BUFFER 内の波形データから波形データを取り出して、出力するようになっています。

ここで、WAVE BUFFER の使い方について簡単に触れます。

キーイング情報は CComCenter オブジェクト内のイベント履歴リスト(11)を参照して、現時点で TONE を出すかどうかを判断しています。

#### 4. 現時点で未だ実装していない機能と発生している問題点

現時点で未だ実装していない機能と発生している問題点を列挙します。

- (1) break-in については、記録されていませんので、break-in を選択しても、プログラムを再起動したときには、break-in が選択されていない状況になります。
- (2) IC-9700 ではとりあえず動作しています。
- (3) ID-52 で、Break-in は効くものの、マイクからも外部マイクからも変調をかけられません。IC-9700 でこれらからいずれも変調を掛けられるのとは対照的なので、ID-52 側の問題かと思いますが、まだ icom に問い合わせしておりません。
- (4) その他のリグでは、CI-V のアドレスのみ正しいですが、実験しておりません。オブジェクトの構成は、設計資料のところに PowerPoint 版、設計資料-Structure の下に PNG 版が置いてあります。
- (5) WSJT-X を動作させ、終了したあとに本プログラムを(何回)動作させても、Break-In が正常に動作しません。
- (6) 本プログラムを動作させ、終了したあとに(何回) WSJT-X を動作させても WSJT-X は正常に動作しません。
- (7) (6)(7) の場合には、リグの電源を落として、さらにリグを再起動させると正常に動作するようになります。現時点では原因を掴んでおりません。
- (8) 使い方などは追ってマニュアルを作成するつもりです。
- (9) DitDahChat で USB-シリアルポートの DTR/DSR を断続させるインターフェースを利用している方は、CKeyHandler.java 内で現在コメントアウトしてある 43～56 行のところにコメントアウトを外し、現在の 57～69 行の部分をコメントアウトすれば同じインターフェースでキーイングできます。
- (10) [OPERATION タブ]内の[CURSOR HOLDER for KeyBoard]をマウスでクリックしてからキーボードの「上矢印」か「下矢印」を押下すればキーボードを使ってキーイングできます。

以上