＃＝命令文

# TurboHAMLOG入力とＱＳＬカード印刷出力一覧表　2

　付属の定義ファイルに手を加えて**自分専用の定義ファイル**にしたら、**必ず名前を変えて保存**して下さい。（インストール時に同名のファイルは上書きされますので）

◎**各命令について**（条件命令について，各項目について）

以下に示す命令以外はコメントと見なします。各命令は、行頭から記述して下さい。

行頭から記述されてない場合は、コメントとなります。 **#**の後に空白を入れてもかまいません。

参考資料アップ場所　　http://mackey.my.coocan.jp/jo1ndg-qsl.html

**文字列の印字命令**

**#Print**

文字列を印字します。印字位置の座標と文字列をカンマで区切ってください。

カンマで区切る考え方は、CSVファイルの１行と同じ要領です。

文字列は、基本的にダブルクォーテーションで括って下さい。ダブルクォーテーションそのものを印字したい場合は、""のように続けてください。

なお、文字列をダブルクォーテーションで括らなくても動作しますが、その場合は文字列中にカンマを入れたり、右端にコメントを入れることはできません。

文字列の中では、!マークから始まる各項目用変数が使用できます。（座標は０，１ミリ単位）

　構文　#Print <横座標>, <縦座標>, "文字列"

【例】　#Print 310,455,"交信時間は!TH:!TMでした"

※上の例では、左から31ミリ、上から45.5ミリの位置から交信時間を印字します。

【例２】　#Print -10000,455,"おやすみなさい。"

横座標を **-10000**とすると、直前の#Print命令による文字列の次に続けで表示します。これによりフォントを変えることもできます。

上の例では、【交信時間は23:50でした。おやすみなさい。】のようになります。

**#PrintA**

文字列が長くて用紙からはみ出してしまう場合に、自動的に改行して印字します。

また、文字列中に<BR>という文字があれば、その位置で改行します。<Br>や<br>では改行しません。

改行したときの行間隔を0.1ミリ単位で指定します。それ以外は、**#Print**と同じです。

　構文　#PrintA <横座標>, <縦座標>, <行間幅>, "文字列"

【例】　#PrintA 220,965,10,"QSO有難うございました。<BR>CU AGN"

※用紙の右端で自動的に改行させる場合、印刷前にプリンターの用紙設定ではがきを指定しておく必要があります。

**#PrintC**

指定した範囲の中央に印刷します。いわゆるセンタリングです。

　構文　#PrintC <横座標>, <縦座標>, <範囲>, "文字列"

【例】　#PrintC 650,390,150,"!FR"　　；7MHzでも1200MHzでも真ん中に印字する。

※文字列の長さが<範囲>で指定した長さよりも長かった場合は、**#Print**と同じです。

**#PrintK**

文字列を均等割り付けして印字します。それ以外は、**#Print**と同じです。

　構文　#PrintK <横座標>, <縦座標>, <均等割付する幅>, "文字列"

※<均等割付する幅>が小さいと、密着割付となります。

**#PrintL**

文字列を斜め、縦、逆さま等、角度を変えて印字します。角度は0～360度が指定できます。

文字のサイズは、 **#Print**で印字したときと微妙に違う場合があり得ますが、それ以外はほぼ **#Print**と同じです。

フォントは、「ＭＳ ゴシック」、「ＭＳ 明朝」のように、フォントサイズの変更可能な **TrueTypeフォント** を使って下さい。

　構文　#PrintL <横座標>, <縦座標>, <角度>, "文字列"

**#PrintR**

文字列を右寄せで印字します。<横座標>は、文字列の右端の位置を指定します。

　構文　#PrintR <横座標>, <縦座標>, "右寄せ文字列"

## フォントの指定(F)

**#FontName**

印字フォントの名前を指定します。フォント名は、ダブルクォーテーションで括って下さい。

【例】　#FontName="ＭＳ ゴシック"

**#FontSize**

印字フォントのサイズを指定します。

【例】　#FontSize=13

**#FontColor**

印字フォントのカラーを10進数または16進数で指定 します。

文字や罫線のカラーは、Ｃ言語の１６進数 0xFF形式で指定することができます。

16進数で指定した場合、３バイトがそれぞれ青，緑，赤のRGBカラーの輝度を表します。

**0xFF0000**という値は最高の輝度、つまり**純粋の青**を示し、 **0xFF00**は純粋の緑を示し、**0xFF**は**純粋の赤**を示します。

**0x00**は**黒**、**0xFFFFFF**は**白**です。ちなみに、0xFFは 10進数に直すと255です。

※16進数とは、 **0**～ **9**, **A**～ **F**の１６の値で表す数値で、 **F**の次が１桁繰り上がって**10**となります。

どの数字がどの色にあたるかは難しいので、**フォント設定ダイアログ**の中の**色**で設定して下さい。

**#FontStyle**

印字フォントのスタイルを指定します。（ボールド、イタリック等）

これら設定した**フォントの内容**は、次の指定が現れるまで有効です。

また、個々に指定可能です。フォントは、定義ファイル中 **#Print**命令よりも上の行で、**最低限一度は必ず指定**してください。

## 次のデータを読む(N)

**#Read**

用紙を排出せずに次のデータを読み込みます。タックシールのように、一度に複数の局宛に印刷する場合に使えます。

**#Readc**

読み込んだ次のデータのコールサインが前回のものと違っていれば、この命令より下の行を無視して用紙を排出します。

このことにより、１枚のQSLカードに同一局との複数の交信を印字することができます。

ただし、**JARL指定順**か**JARL指定逆順**でなければコールサインで並んでいないためうまく動作しません。

また、クイックQSL印刷では正常に動作しません。

読み込んだデータのコールサインは、QSLマネージャー経由の場合は、QSLマネージャーのコールサインです。

**#Readd**

**#Readc**と同じですが、読み込んだデータのコールサインは、QSOした相手のコールサインです。

**#Reade**

同じく、読み込んだ次のデータのCodeが前回のものと違っていれば、この命令より下の行を無視して用紙を排出します。

**#Readj**

読み込んだ次のデータのコールサインが前回のものと違っていれば、指定した行、またはラベルにジャンプします。

ジャンプの要領は **#Goto**と同じです。

このことにより、１枚のQSLカードあるいはタックシールに同一局との複数の交信を印字することができます。

ただし、**JARL指定順**か**JARL指定逆順**でなければなりません。クイックQSL印刷では正常に動作しません。

読み込んだデータのコールサインは、QSLマネージャー経由の場合は、QSLマネージャーのコールサインです。

【例】　#Readj *760　；違うコールサインだったら、行頭に *760 と書いてある行にジャンプ

**#Readk**

**#Readj**と同じですが、読み込んだデータのコールサインは、QSOした相手のコールサインです。

**#Readf**

同じく、読み込んだ次のデータのCodeが前回のものと違っていれば、指定した行、またはラベルにジャンプします。

※定義ファイル中に#Reade か #Readf命令がある場合は、JARL指定順＋Code順＋レコード番号順でソートされます。

※DX局の場合、JARL指定順を「（QSLマネージャー）＋フルコールサイン」と読み替えてください。

## コールサインが連続している場合に印刷

**#MaxCall**

同一局と何回か交信した場合に、１枚のQSLカードに複数データを印刷することがあります。

この命令でコールサインが連続しているデータのみ抽出し、１枚のカードに印刷できるデータ数を指定します。

もちろん、前述の **#Readc**命令や **#Readj**命令などと併用する必要があります。

印刷指定したレコード番号の範囲中に、単一のコールサインのデータは無視します。

定義ファイルの上の方で一度だけ指定してください。

【例】　#MaxCall=5　；１枚のカードに２～５局印刷する。

例えば、５を指定して同一コールサインが７ある場合は、１枚目に５データ、２枚目に２データ印刷できます。

例えば、４を指定して同一コールサインが５ある場合は、１枚目に４データを印刷し、残りの１データは無視します。この残りの１データは、別の通常の定義ファイルを使って印刷することとなります。

**JARL指定順**か**JARL指定逆順**を指定しておく必要があります。

## 用紙の排出（改ページ）

**#Exit**

下の行の命令を無視して、ここで定義ファイルの実行が終わり、用紙を排出します。

**#ExitX**

そのまま用紙を排出します。下の命令は次の用紙に印刷されることになりますので、QSLカードの両面印刷に利用できます。

両面印刷するには、プリンタに両面印刷の機能があることが必要です。

あらかじめプリンタの設定で**はがき**の**両面印刷**にセットしてください。

また、**#Size**命令は使用しないでください。

**#ExitZ**

下の行の命令を無視して、ここで定義ファイルの実行が終わります。

用紙排出をしません。

ただし、結果的に印刷データがまったく無かった場合は、何も印刷せずに用紙を１枚だけ排出します。

## 印刷の縦横変換(T)

**#Yoko**

この命令があると、カードを全体的に縦横変換して印刷します。

定義ファイルの上の方で一度だけ指定してください。

事前にプリンタの設定で用紙を葉書サイズに指定しておかないと、正しい位置に印字しないことがあります。

## 指定行へジャンプ(P)

**#Goto**

指定したラベルのある行へ処理をジャンプさせます。つまり、目印のある行までジャンプです。

無限ループ防止のため、上の行へはジャンプできません。ラベルは、行頭から *123 のように **＊ と適当な数字**を使用します。

８桁以下の数字であれば何でも構いません。

【例】　#Goto *950　　；行頭に *950 と書いてある行にジャンプ

行番号を絶対行で指定することもできます。

これより下の行の命令を無視し、強制的に用紙を排出させるには、最下行よりも大きい数字を指定します。

【例】　#Goto 107　　；107行目にジャンプ

## 用紙サイズの指定(W)

**#Size**

印刷イメージを確認するときに使います。数値は０．１ミリ単位で指定します。印刷時には影響ありません。

構文　　#Size <横の長さ>, <縦の長さ>

通常は、はがき縦サイズのイメージで表示しますので、この命令を使うことはありません。

## 罫線の印字(L)

**#LineS**

**罫線の種類(S)**を指定します。スピードバーに表示されている、罫線ボタンをクリックすると、命令が挿入されます。

横スクロールバーが罫線の太さ、●印が罫線の色です。罫線に関する命令では、次の#LineSが現れる行までこの設定が有効です。

罫線を印字する場合は、必ずこの命令で**最低限一度は**罫線の種類を指定して下さい。

#LineS Pen, Style, Color

Pen

罫線の太さを0.1ミリ単位で指定する（太さは、あまり正確ではありません。）

罫線の太さをピクセル単位で直に指定したい場合は、この数値をマイナスにしてください。

この場合は、印刷イメージと実際の印刷では、解像度の違いから太さが大きく異なります。

Style

罫線の種類（0=実線, 1=破線, 2=点線, 3=一点鎖線, 4=二点鎖線）

Color

罫線の色を指定する。カラーは、10進数か16進数の数字。

文字や罫線のカラーは、Ｃ言語の１６進数 0xFF形式で指定することができます。

16進数で指定した場合、３バイトがそれぞれ青，緑，赤のRGBカラーの輝度を表します。

**0xFF0000**という値は最高の輝度、つまり**純粋の青**を示し、 **0xFF00**は純粋の緑を示し、**0xFF**は**純粋の赤**を示します。

**0x00**は**黒**、**0xFFFFFF**は**白**です。ちなみに、0xFFは 10進数に直すと255です。

※16進数とは、 **0**～ **9**, **A**～ **F**の１６の値で表す数値で、 **F**の次が１桁繰り上がって**10**となります。

**※注意※**　Windowsの仕様のため、実線以外の罫線を指定する場合は、罫線の太さを最低にしてください。

【例】　　#LineS 1, 0, 0

**#Line**

斜め罫線(L)を印字します。#Print同様、0.1ミリ単位で指定します。

#Line x1, y1, x2, y2

x1

左からの開始位置を0.1ミリ単位で指定する

y1

上からの開始位置を0.1ミリ単位で指定する

x2

左からの終了位置を0.1ミリ単位で指定する

y2

上からの終了位置を0.1ミリ単位で指定する

**#LineR**

四角形を描く(R)

#LineR x1, y1, x2, y2 ※ パラメータは#Lineと同じ

**#LineC**

角の丸い四角形を描く(C)

#LineC x1, y1, x2, y2 ,x3, y3 ※ パラメータは#Lineと同じ

x3

横方向の丸みの長さを指定します。

y3

縦方向の丸みの長さを指定します。

**#LineX**

横罫線(X) を印字します。

#LineX x1, y1, Length

x1

左からの開始位置を0.1ミリ単位で指定する

y1

上からの開始位置を0.1ミリ単位で指定する

Length

罫線の長さを0.1ミリ単位で指定する

**#LineY**

縦罫線(Y)を印字します。

#LineY x1, y1, Height

x1

左からの開始位置を0.1ミリ単位で指定する

y1

上からの開始位置を0.1ミリ単位で指定する

Height

罫線の長さを0.1ミリ単位で指定する

**#Ellipse**

円や楕円の罫線を印字します。

#Ellipse Left, Top, Width, Height

Left

左からの開始位置を0.1ミリ単位で指定する。

Top

上からの開始位置を0.1ミリ単位で指定する。

Width

円や楕円の幅を0.1ミリ単位で指定する。

Height

円や楕円の高さを0.1ミリ単位で指定する。省略した場合はWidthと同じ長さ（真円）

## QSL発行済みマーク(M))

**#Mark**

ここで指定した文字（半角）が、QSOデータにQSL発行済みのマークとして、QSLの入力項目のうち２文字目に書き込まれます。

QSL印刷ウインドウのQSLマークよりも優先されます。=は入れても入れなくても動作します。

""の場合は、QSL発行済みマークを記入しません。

" "の場合は、QSL発行済みであってもスペースを書き込んで未発行とします。

【例】 ? User

#Mark="U" ；HAMLOGユーザーであれば**U**をQSL発行済みマークとする。

【例】 ? DXST

#Mark = "" ；DX局のときはQSL発行済みマークを書き込まない。

【例】 ? Qsl2 "E"

#Mark="W" ；ハムログ電子QSL発行済みであった場合は、**E**を**W**に書き換える。

## JPEGイメージの印刷(J)

**#Jpg**

デジカメ写真など、JPEG画像ファイル（*.jpg）を読み込み、印刷します。次のいずれかの構文で指定します。

```
#Jpg x1, y1, Width, Height, "Photo1.jpg"
#Jpg x1, y1, Width, 0, "Photo1.jpg"
#Jpg x1, y1, 0, Height, "Photo1.jpg"
```

x1

左からの印字位置を0.1ミリ単位で指定します。

y1

上からの印字位置を0.1ミリ単位で指定します。

Width

JPEG画像の幅を0.1ミリ単位で指定します。

Widthがゼロの場合は、Heightを指定してください。比例して適正な幅で印字されます。

Height

JPEG画像の高さを0.1ミリ単位で指定します。

Heightがゼロの場合は、Widthに比例して適正な高さで印字されます。

FileName

JPEG画像のファイル名を指定します。ファイル名は、ダブルクォーテーションマークで括って下さい。

フォルダの指定が無ければ、定義ファイルの存在するフォルダにあるものとしてオープンしようとします。

もし、別のフォルダにあるJPEGファイルを印刷する場合は、フルパスで指定してください。

ファイル名の文字列には、!マークから始まる各項目用変数が使用できます。

以下の例では、左から10ミリ、上から48ミリの位置を左上端として、JPEG写真を幅64ミリの大きさで印刷します。

写真の高さは自動的に調整されます。

```
【例1】#Jpg 100, 480, 640, 0, "Photo1.jpg"
【例2】#Mov $$A = "Photo2"
       #Jpg 100, 480, 640, 0, "!$$A.jpg"
```

※最近のデジカメは、画素数がとても大きいので、ペイントなどを使って画像サイズを小さく調整してみてください。
縦、横ともに1024ピクセル以下がいいみたいです。

※画像がうまく印刷されない場合は、画像サイズを小さくしてみるとか、プリンタドライバを最新のものに変えるなど調整してください。

## ビットマップを印刷(B)

**#Bmp**

ビットマップ(*.bmp)ファイルを読み込み、印刷します。概要は **#Jpg**と同じです。

## 挿入文書の設定(I)

**#Text**

QSLカードに長い文章を印刷したいときに利用してください。

この下の行から、**#End** までの間の文字列を印字します。

行間隔の指定がなければ、フォントの大きさに応じて自動的に改行されます。

文字列は、 **""**でくくる必要はありません。 座標や行間隔の数値は 0.1ミリ単位です。

文字列には、!マークから始まる各項目用変数や、!$$Aなどの変数が使用できます。

**#End**

挿入文書の終わりを示します。

```
#Text <横座標>, <縦座標>, <行間隔>
ここで長い文章を印刷します。
アワードの紹介や自己紹介などの長い文章を書くと良いでしょう。
作者は、年賀状を書くときに使うつもりです。(^_^;)
#End
```

## 印字位置補正(H)

**#SetXY**

印字位置を全体的に補正します。定義ファイルの上の方で一度だけ指定してください。

定義ファイルとQSLカード印刷ウインドウの両方の補正値を足した値が、実際の補正値となります。

定義ファイルを新たに作成したら、編集 → 試し印刷により #SetXYを調整し、その後の微妙なズレは 印刷ウインドウ側の印字位置補正で指定してやるといいでしょう。

#SetXY <横方向>, <縦方向>

## 矩形領域の塗りつぶし

**#FillBox**

指定した矩形領域を塗りつぶします。

#FillBox Left, Top, Width, Height, Color

Left

左からの開始位置を0.1ミリ単位で指定する。

Top

上からの開始位置を0.1ミリ単位で指定する。

Width

矩形領域の幅を0.1ミリ単位で指定する。

Height

矩形領域の高さを0.1ミリ単位で指定する。

Color

塗りつぶす色の指定。省略した場合は黒。10進数または16進数で指定します。

## 円や楕円領域の塗りつぶし

**#EllipsF**

指定した円や楕円の領域を塗りつぶします。

#EllipsF Left, Top, Width, Height, Color

Left

左からの開始位置を0.1ミリ単位で指定する。

Top

上からの開始位置を0.1ミリ単位で指定する。

Width

矩形領域の幅を0.1ミリ単位で指定する。

Height

矩形領域の高さを0.1ミリ単位で指定する。ゼロを指定した場合はWidthと同じ長さ（真円）

Color

塗りつぶす色の指定。省略した場合は黒。10進数または16進数で指定します。

## 文字列や数値を一時変数に代入

**#Mov**

変数に文字列や数値を代入します。

**$$A**～ **$$T** が文字列専用の変数です。代入する文字列には、他の変数も含まれます。

また、**#Jpg**命令，**#Bmp**命令，**#Load**命令のファイル名や、**#Text**命令の文字列にも変数使えます。

**$$U**～ **$$Z** が数値専用の変数です。代入した数値は、主に印刷する座標指定で使います。

【例】

```
#Mov $$A="to radio !cp"
? Potbl
#Mov $$A="!$$A　運用地：!QT"
#Mov $$X = -50
#Mov $$Y = 1000
#Mov $$Y = $$Y+300
#Print $$X+100, $$Y, "【!$$A】"
```

※マイナスを使用するときは、数字や変数との間に空白を入れないでください。（500 -　$$W　とか　$$Y -　200 では動作しない）

※加減のみ演算ができます。　【例】　$$Z=$$Z+1

## ループ命令

**#DoLoop ～ #EndLoop**

#EndLoopに続く数値変数がゼロになるまで繰り返し処理をします。

無限ループ防止のため、ループが1,000回を超えるとエラーになります。

#DoLoopと#EndLoopをネスト（ループの中にループ）した記述はできません。

【例】

```
#Mov $$Y=500
#Mov $$U=5
#DoLoop
#Mov $$A=$$U　　；数値を文字列に変換
#Print 100, $$Y, "!$$A !cp"
#Mov $$Y=$$Y+100　；１センチ下へ
#Mov $$U=$$U-1
#EndLoop $$U　；$$Uがゼロになったらループ脱出
```

## 別の定義ファイルを読み込む

**#Load**

定義ファイル実行中に、別の定義ファイルを実行します。別の定義ファイルの実行が完了したら、元の定義ファイルの次の行に戻ります。

別の定義ファイルから、さらに別の定義ファイルの実行はできません。

別の定義ファイル中で、**#Yoko　#ExitX　#Load**命令 は使えません。

【例】　　#Load "sonota.qsl"

条件分岐するだけの定義ファイルを作り、その中から条件に応じ、いろいろな定義ファイルを実行するようにしてみると便利かもしれません。

ファイル名の文字列には、!マークから始まる各項目用変数が使用できます。

## プリンタ設定ファイルを読み込む

**#PrnLoad**

プリンターを設定し、名前を付けて保存により、あらかじめ保存しておいたプリンタ設定ファイルを読み込みます。

ハガキ印刷.dev、Ａ４印刷.dev、ハガキ両面印刷.devなどのように、ポップアップメニューから名前を付けて保存しておく必要があります。

【例】　#PrnLoad "QslA4印刷.dev"

この命令は、定義ファイルの上の方で一度だけ指定してください。

この機能により、定義ファイルを切り替えるだけで他の用紙、他のプリンタにて印刷することができます。

## Rig/Antの指定

**#Rig#n**

Rig/Antの設定ウインドウの行番号を直接指定し、自局のリグ・アンテナ等の設定が読み込まれます。

次のQSOデータを読み込むまで、または定義ファイル中に次の**#Rig#**命令が現れるまで有効です。

Remarks１・２欄に書き込んだRig#??の指定よりも優先されます。

【例】

```
? Freq 3              ; ７ＭＨｚで、かつ・・・
? Data13 "%/1"     ; Remarks１に自局の移動運用地が記載されていれば、
#Rig#6                ; ６行目のリグ・アンテナを読み込む。
```

## JST⟷UTC変換

**#Jst2Utc**

交信時刻がJSTで記録されている場合、UTCに変換します。９時間遅れの年月日時分となります。

交信年月日や交信時刻を印字する行よりも上の行でこの命令を記述してください。

なお、入力ウインドウ のDXにチェックが入っていてJSTの場合は、この命令が無くてもUTCに変換されます。

**#Utc2Jst**

交信時刻がUTCで記録されている場合、JSTに変換します。**#Jst2Utc**と逆です。

交信年月日や交信時刻を印字する行よりも上の行でこの命令を記述してください。

入力ウインドウ のDXのチェックに関係なく、UTCであればJSTに変換されます。

## ADIFファイル出力

**#ADIF**
**#ADIF**命令をQSL印刷定義ファイルの上の方に記述することにより、QSLカードの印刷情報をADIFファイルに出力します。
付属の定義ファイルを参考にしてください。

**#ADIF**命令を指定した場合、**#Print**以外の印刷系の命令や印刷イメージは機能しません。
また、**連続印刷**にチェックを入れなければ機能しません。フォントの指定もできません。

#ADIF命令では、次のようにADIFファイルをフルパスで指定することができます。
【例】 #Adif="C:¥Hamlog¥Adif.adi"

以下の例１の様にフォルダ指定が無い場合は、HAMLOG本体と同じフォルダに**AdifOut.adi**という名前で出力されます。

【定義の例１】

```
#ADIF        ; HAMLOG本体と同じフォルダに出力
#Jst2Utc     ; 時刻のJSTは、すべてUTCに変換して出力
#Print 0,0,"<CALL:!Af>!cp<QSO_DATE:!Af>!DY!DM!DD<TIME_ON:!Af>!TH!TM<RST_SENT:!Af>!HR<RST_RCVD:!Af>!MR"
#Print 0,0,"<EOR>"
```

【上記定義の出力結果の例】

```
<CALL:8>JG1MOU/1<QSO_DATE:8>20181208<TIME_ON:4>0816<RST_SENT:2>59<RST_RCVD:2>59
<EOR>
```

**!Af** は、!Afの右側にある**>**と**<**に挟まれた文字列の長さに置き換わります。
**<CALL:!Af>!cp** → **<CALL:6>JG1MOU**
**<CALL:!Af>!cp** → **<CALL:10>JG1MOU/JD1**
その他、QSL印刷定義の変数や条件命令が使えます。

【定義の例２】

```
#ADIF="D:¥!nW.adi"  ; 2019-04-13.adiといった形式のファイル名
#Jst2Utc     ; 時刻のJSTは、すべてUTCに変換して出力
#Print 0,0,"<CALL:!Af>!cp<QSO_DATE:!Af>!DY!DM!DD<TIME_ON:!Af>!TH!TM<RST_SENT:!Af>!HR<RST_RCVD:!Af>!MR"
#Print 0,0,"<FREQ:!Af>!FR<MODE:!Af>!MD<BAND:!Af>!Bt<BAND_RX:!Af>!Br<QSL_SENT:1>Y<GRIDSQUARE:!Af>!GL"
#Print 0,0,"<EOR>"
```

【上記定義の出力結果の例】

```
<CALL:6>JG1MOU<QSO_DATE:8>20190128<TIME_ON:4>1810<RST_SENT:2>59<RST_RCVD:2>59
<FREQ:1>7<MODE:3>SSB<BAND:3>40m<QSL_SENT:1>Y<GRIDSQUARE:4>PM96
<EOR>
```