

「CW Skimmer 風」 アプリケーション

「帯域全体を監視し、複数の信号を同時に捕まえて解読する」\*\*という、まさに CW Skimmer (Alex Shovkoplyas, VE3NEA 氏開発の有名なソフトウェア) のコンセプトを Web ブラウザ上で再現したものです。

この「CW Skimmer 風」アプリケーションの核心となる仕組みと、実用的に使うためのセットアップ方法について少し詳しく解説します。

## 1. 「普通のデコーダー」と「CW Skimmer 風」の違い

従来の Web CW デコーダー（最初に提示されたもの）と、今回作成した Skimmer 風デコーダーの決定的な違いは\*\*「耳の数」\*\*です。

普通のデコーダー:

ラジオの VFO（同調ノブ）を回して、1 つの信号をフィルターの中心（例: 700Hz）に合わせる必要があります。

のように、特定の狭い範囲しか見ていません。

今回の CW Skimmer 風:

0Hz~3000Hz の全帯域を常に監視します (FFT 解析)。

「あ、600Hz で音がした」「1250Hz でも音がした」と自動で判断し、その数だけデコーダーを動的に生成します。

これにより、パイルアップ（多数の局が同時に呼んでいる状態）でも、複数のコールサインをリストアップできます。

## 2. コードの核心部分（どうやって実現しているか）

このプログラムの「脳」にあたるのは `manageDecoders()` という関数です。ここで以下のような処理を毎秒 60 回行っています。

山を探す (Peak Detection):

FFT で得られたグラフ（周波数スペクトル）を見て、しきい値 (Threshold) を超えて突き出ている「山」をすべて探します。

追跡する (Tracking):

見つけた山が、既に監視中の信号なら「継続」と判断します。

もし新しい周波数の山なら、新しいデコーダー (ChannelDecoder クラス) を\*\*即座にインスタンス化 (生成) \*\*して割り当てます。

消去する (Garbage Collection):

数秒間信号が途絶えたデコーダーは自動的に削除し、メモリを節約します。

### 3. SDR と連携させる (推奨セットアップ)

このアプリを最大限に楽しむには、PC のマイクでスピーカーの音を拾うのではなく、SDR (Software Defined Radio) の音を直接ブラウザに入力するのがベストです。

以下の手順で「仮想オーディオケーブル」を使用すると、クリアな信号を入力できます。

仮想ケーブルの導入:

Windows なら VB-CABLE (無料) や Virtual Audio Cable をインストールします。

Mac なら BlackHole などを使用します。

SDR ソフトの設定 (HDSDR, SDR#, SDRUno など):

SDR ソフトの音声出力先 (Output) を「CABLE Input (VB-Audio Virtual Cable)」に設定します。

フィルタ帯域を広め (3kHz~5kHz) に設定します。

ブラウザの設定:

PC のサウンド設定で、規定の「入力デバイス」を「CABLE Output」にするか、ブラウザがマイク入力を求めてきた際に「CABLE Output」を選択します。

こうすることで、SDR で受信したクリアな CW 信号がデジタルのままブラウザに届き、本家の CW Skimmer のようにウォーターフォールを見ながらコールサインが次々とデコードされる様子を楽しめます。

#### 4. 今後の拡張アイデア

もしさらに改良を加えるなら、以下のような機能が考えられます（プログラミングの学習テーマとしても最適です）。

クリックして同調:

ウォーターフォール上の信号をクリックすると、その周波数だけを強調してスピーカーから再生する（オーディオフィルタ機能）。

RBN (Reverse Beacon Network) 風の投稿:

「CQ」や「DE」の後のコールサインを検出したら、時刻と周波数とともに別のリストに記録する、あるいはサーバーに送信する。

AudioWorklet の使用:

現在はメインスレッドで処理していますが、AudioWorklet を使うと、バックグラウンドでより高度な信号処理（狭帯域フィルタリングなど）が可能になり、デコード精度が向上します。

ぜひ、SDR と組み合わせて「Web 版 CW Skimmer」として遊んでみてください