# 第三回 音声周波数帯信号のバンドパスフィルタの製作

JH3RGD 葭谷安正

#### ■はじめに

皆さんこんにちは。 この記事が掲載される頃にはオリンピックが開催されている頃だと思います。暑い時期ですが、健康に気をつけ、オリンピック放送をそして無線ライフを楽しみましょう。今回もあち こち脱線しながらすすめていきます。

今回は音声周波数帯信号のバンドパスフィルタを作ります。無線機のスピーカ端子から音声信号を取り 出し、その信号をバンドパスフィルタに通して特定帯域の信号のみを取り出そうという試みです。「も のづくりやろう!」と書いていますのでコイルやコンデンサのような部品を使ってハードウェア的に作る 事を想像されたかもしれません。今回の製作は、OS に Windows 10 を搭載したパソコン上で動く GNU Radio(「グヌーラジオ」と読むそうです)というアプリケーションソフトを使い、音声周波数帯の デジタル信号処理を行います。このため、ソフトウェアの説明やデジタル信号処理の説明も少しする必 要がありますので、今回、次回(多分次々回も)と記載させていただきます。ソフトウェアで実現すると いうことでプログラミングが必要ではないかと考えられるかと思いますが、既存のブロックを使って処 理をおこなうためだけならばプログラミングは必要ありません。マウスを使って、ブロックを並べ、パ ラメータを設定することで基本的な操作が完結します。

私はIC-7300でモールスやFT8で細々とオンエアし、楽しいアマチュア無線ライフを送っています。 最近の無線機の表示部は、ウォーターフォールが実装され帯域内でオンエアする局がすぐにわかり非常 に便利ですが、旧式無線機の「ダイヤルを回しながらオンエア局を探す」というのも捨てがたい味があ ります。このような古いリグをさわりたくなり、ヤフオクで入手しました。機種はTS-120Vでした。 マイクがついていなかったので、入手した無線機もモールス用に使用しました。古い無線機はTR-1200とIC-71 以外電波を出したことがありませんでしたので、TS-120Vの同調には最初戸惑いま した。何かというと、受信信号のビート音が700Hzになったとき自分の送信周波数と相手の送信周波 数とが一致するので、受信音の周波数をできるだけ700Hzになるようにダイヤルを回す必要があるこ とです。CQ局に対し何回か送信しましたところ全く応答がありません。出力が10W という非力なた めだろうと思っていたのですがそれだけではないようで、一日頑張ってみても状況は変わりませんでし た。そこで自分が700Hz だと思っている信号は本当に700Hz なのかを確認するためスマートフォン に周波数分析(FFT)ソフトをインストールして確認しました。その結果、私が700Hz と思っていた周 波数は全く違っていました。このスマホのアプリで受信音の周波数を700Hz 近辺になるように受信 し、こちらからコンタクトを試みたところ通信ができました。「俺の耳の周波数感覚はこんなに悪いの か!、俺は音痴なんでしょうか?」。XYL 曰く、「YES」。 それはともかく、入手した TS-12OV の受信ビート周波数が 700Hz に簡単にあわせることのほかに、 もうーつやらなければならないことがありました。それは、入手した無線機には CW フィルタが入って おりませんでした。このため CQ 局の近隣周波数で別の局が CQ なり交信しているとその信号音が非常 によく聞こえます。私のようなヘボオペレーターにとって複数信号が同時に聞こえてくるのは非常にや りにくい状況で、気になって仕方がありませんでした。このため CW フィルタ、具体的にはクリスタル フィルタですが、これを入手しようとしました。が、クリスタルフィルタの価格が入手したリグの金額 と同じぐらいで取引されていることがわかり、手が出ませんでした。クリスタルフィルタは中間周波数 における帯域を制限することで選択度を改善していますが、音声出力段階でスピーカにバンドパスフィ ルタなどをくっつけてフィルタをかければいい、またはヤフオクで入手と考えました。フィルタ機能を もつスピーカがオークションサイトに結構ありましたが、その中で気になったのが共振スピーカでした。 このスピーカの自作はちょっと躊躇しましたので、購入しました。なかなか味のあるスピーカです。(写 真 1)



写真1 共振スピーカとその特性

信号が浮き上がってくるように聞こ えてきて、秋の夜長に DX 局のモー ルスを聴くのが絵になるような物で した(あくまでも私の感想です)。し かしやはりフィルタが気になってし かたがありません。オペアンプで作 成された谷岡電子製のエレクトロニ クスキット「アクティブ・バンドパ ス・フィルタ モデル BPF-0257」 の基板を日本橋の千石電商さんで買 ってきて、試したりしてみました。



写真2 谷岡電子製「アクティブ・バンドパス・フィルタ モデル BPF-0257」

同時期に私のコンピュータが動かなくなったので新しくしました。それまでは 32 ビット CPU に OS は Windows 7 が載っているノートパソコンでしたが、64 ビット CPU に Windows 10 が載ったコ ンピュータになりました。そこで新しいコンピュータに GNU Radio という信号処理ソフトをいれるこ とにしました。GNU Radio は Linux 版、Mac 版、Windows 版があり、Windows 版は OS が 64 ビ ット版でなければインストールできませんでした。以前 LinuxOS をインストールしたノートパソコン にもインストールしていたソフトですので新しいパソコンでは Windows 上に GNU Radio をインス トールしてまた遊べるぞとばかりにインストールした次第です。この GNU Radio を使ってソフト的に バンドパスフィルタを作成していきます。

#### ■GNU Radio について

さて GNU Radio ですがこのソフトはソフトウェア無線の信号処理系です。GNU Project の一部とし てオープンソースのソフトウェアとして開発されたもので、無償で利用することができます。GNU Radio の歴史的経緯やライセンス関係については GNU Radio のホームページを見ていただくことに して、ここではこの GNU Radio を使うことでどんなことができるのかを見ていきたいと思います。

IC-7300 などの近頃の無線機をお持ちの方は SDR(Software Defined Radio) という用語をご存じと 思いますが、GNU Radio は、パソコンに搭載されたソフトウェアによって信号処理を行う SDR ツー ルキットと言うことができます。具体的には、無線機の製作で必要となる変調や復調等の回路の機能を ソフトウェアで実現したものです。ソフトウェアですので言語が関係しています。元々GNU Radio で は、C 言語や Python 言語の知識や無線の知識を知っている技術者がプログラミングして SDR を構成 する必要がありました。しかし GNU Radio のツールとして GNU Radio Companion(以下 GRC と 略称します)が作成されました。この GRC を使うと、GRC 上で

- マウスでブロック図を配置し、
- (2) ブロック図をダブルクリック してパラメータを設定し、
- (3) 信号の流れにしたがって結線し、
- (4) 変換や実行のボタンを押す

という操作でプログラムレスなシ ステム構築ができます。図 1 に GRC の起動画面を示します。

					▼ Core	
Options Varia	ble				<b>▼</b> A	udio
Output Language: Python Value: 3	p_rate 32k					Alaw Audio Decod
Generate Options: QT GUI						Audio Sink
						Audio Source
						CVSD Audio Deco
						CVSD Audio Enco
						CVSD Decoder
						CVSD Encoder
						g711 Alaw Audio
						g721 Audio Deco
						g721 Audio Enco
						g723_24 Audio D
						a723 24 Audio Fr
						g
						g723_40 Audio D
						g723_40 Audio D g723_40 Audio Er
						g723_40 Audio D g723_40 Audio Er ulaw Audio Deco
						g723_40 Audio De g723_40 Audio De ulaw Audio Deco ulaw Audio Enco
					► B	g723_40 Audio De g723_40 Audio De ulaw Audio Deco ulaw Audio Enco oolean Operators
< < Welcome to GNU Radio Companion					► B	g723_40 Audio De g723_40 Audio De ulaw Audio Deco ulaw Audio Enco oolean Operators yte Operators
<<< Welcome to GNU Radio Companion /3.8.2.0-57-gd71cd177 >>>	ld	Value			▶ B ▶ B	g723_40 Audio Di g723_40 Audio Di g723_40 Audio Erco ulaw Audio Enco oolean Operators yte Operators hannel Models
< < < Welcome to GNU Radio Companion /3.8.2.0-57-gd71cd177 >> >	ld Imports	Value		+	► B ► B ► C ► C	g723_40 Audio Dr g723_40 Audio Er ulaw Audio Deco ulaw Audio Enco oolean Operators yte Operators hannel Models hannelizers
<<< Welcome to GNU Radio Companion .3.8.2.0-75-gd71cd177 >>> Slock paths: C4PProgram	ld Imports ▼ Variables	Value		+	► B ► B ► C ► C	g723_40 Audio Dr g723_40 Audio Er ulaw Audio Deco ulaw Audio Enco oolean Operators yte Operators hannel Models hannelizers oding
<<< Welcome to GNU Radio Companion /3.8.2.0-57-gd71cd177 >>> 3lock paths: C4Program liegtK0NRadio-3.8Kshare¥gnuradio¥grc •	ld Imports ▼ Variables samp_rate	Value 32000		+ + ×	<ul> <li>B</li> <li>B</li> <li>C</li> <li>C</li> <li>C</li> <li>C</li> <li>C</li> </ul>	g723_40 Audio Di g723_40 Audio Di g723_40 Audio Deco ulaw Audio Deco ulaw Audio Enco oolean Operators hannel Models hannelizers oding ontrol Port
<< < Welcome to GNU Radio Companion /3.8.2.0-57.0471cd177 >>> 30cck paths: C4Program [JesKGNURadio-3.84shareVgnuradio¥grc - felocks	Id Imports • Variables samp_rate	Value 32000		+ + ×	<ul> <li>B</li> <li>C</li> <li>C</li> <li>C</li> <li>C</li> <li>C</li> <li>C</li> <li>C</li> <li>D</li> </ul>	g723_40 Audio D g723_40 Audio Er ulaw Audio Enco oolean Operators yte Operators hannel Models hannelizers oding ontrol Port ebug Tools
< < < Welcome to GNU Radio Companion 3.8.2.0-57-gd71cd1177 >> > Slock paths: C4Program Ties¥GNURadio-3.8¥share¥gnuradio¥grc - Hocks C4Program	ld Imports ▼ Variables samp_rate	Value 32000		+ + ×	<ul> <li>B</li> <li>C</li> <li>C</li> <li>C</li> <li>C</li> <li>C</li> <li>C</li> <li>D</li> <li>D</li> </ul>	g723_40 Audio D g723_40 Audio D ulaw Audio Enco oolean Operators hannel Models hannelizers oding ontrol Port ebug Tools epercated
<<< Welcome to GNU Radio Companion .3.8.2.0-57-gd71cd177>>> 3lock paths: C4Program ile#VGNURadio-3.8¥share¥gnuradio¥grc - ħlocks C4Program iles#GNURadio-3.8¥share¥gnuradio¥grc - ħlocks	Id Imports • Variables samp_rate	Value 32000		+ + ×	<ul> <li>B</li> <li>C</li> <li>C</li> <li>C</li> <li>C</li> <li>C</li> <li>D</li> <li>D</li> <li>D</li> </ul>	g723_40 Audio D g723_40 Audio D ulaw Audio Deco ulaw Audio Deco ulaw Audio Eco ulaw Audio Eco ul
<<< Welcome to GNU Radio Companion /3.8.2.0-57-gd71cd177 >>> Slock paths: C+¥Program filestRNUHRadio-3.8¥share¥gnuradio¥grc - fblocks C+¥Program filestRNUHRadio-3.8¥share¥gnuradio¥grc - fblocks codims: "D-¥gnt¥radio amp07171747.rsr."	ld Imports Variables samp_rate	Value 32000		+ + ×	<ul> <li>B</li> <li>C</li> <li>C</li> <li>C</li> <li>C</li> <li>D</li> <li>D</li> <li>D</li> <li>E</li> </ul>	g723_40 Audio D g723_40 Audio D ulaw Audio Deco ulaw Audio Deco ulaw Audio Eco olean Operators yte Operators yte Operators hannel Models hannelizers oding ontrol Port ebug Tools epercated igital Television qualizers
<<< Welcome to GNU Radio Companion 38.2.0-57-0471cd177 >>> 30.8.ck paths: C4Pfrogram illes¥GNURadio-3.8¥share¥gnuradio¥grc - fibocks C4Pfrogram illes¥GNURadio-3.8¥share¥gnuradio¥grc - fibocks .oading: "D¥gnr¥radio_amp07171747.grc" >> Done	ld Imports ▼ Variables samp_rate	Value 32000		+ + X	<ul> <li>B</li> <li>C</li> <li>C</li></ul>	g723_40 Audio D g723_40 Audio D ulaw Audio Enco oolean Operators yte Operators hannel Models hannelzers oding ontrol Port ebug Tools eprecated igital Television qualizers ror coding
<< < Welcome to GNU Radio Companion v3.8.2.0-57-gd71cd177 >>> 3lock paths: C4Program ile4¥GNURBadio-3.8¥share¥gnuradio¥grc - fblocks C4Program ile4¥GNURadio-3.8¥share¥gnuradio¥grc - fblocks .oading: "D+gn4¥radio_amp07171747.grc" >>> Done	ld Imports Variables samp_rate	Value 32000		+ + ×	B   B   C   C   C   C   C   C   C   C   C   C	g723_40 Audio D g723_40 Audio D ulaw Audio Deco ulaw Audio Deco ulaw Audio Deco oolean Operators yte Operators yte Operators dding ontrol Port ebug Tools epercated igital Television qualizers ror Coding lie Operators
<<< Welcome to GNU Radio Companion /3.8.2.0-57-gd71cd177 >>> Slock paths: C4Program ile#VGNURadio-3.8¥share¥gnuradio¥grc - fblocks C4Program ile#SinURadio-3.8¥share¥gnuradio¥grc - fblocks .oading: "D:¥gnt¥radio_amp07171747.grc" >>> Done .oading: "D:¥gnt¥amp.grc"	Id Imports Variables samp_rate	Value 32000		+ + ×	B   B   C   C   C   C   C   C   C   C   C   C	g723_40 Audio D g723_40 Audio D ulaw Audio Deco ulaw Audio Deco ulaw Audio Deco olean Operators yte Operators yte Operators hannelizers oding ontrol Port ebug Tools teprecated igital Television qualizers roro Coding ile Operators
<< Welcome to GNU Radio Companion 382.0-57.0471cd177 >>> Block paths: C¥Program IlesKGNURadio-3.8¥share¥gnuradio¥grc - Kblocks C¥Program IlesKGNURadio-3.8¥share¥gnuradio¥grc - kblocks .cading: "D.¥gnr¥radio_amp07171747.grc" >> Done Cading: "D.¥gnr¥amp.grc" >> Done	ld Imports • Variables samp_rate	Value 32000		+ + ×	B   B   C   C   C   C   C   C   C   C   C   C	g723_40 Audio D g723_40 Audio D g723_40 Audio Er ulaw Audio Erco oolean Operators tyte Operators hannel Models hannelizers oding ontrol Port ebug Tools teprecated igital Television qualizers ror Coding tie Operators ilters ourier Analysis



図 2 Audio Source ブロックの配置

図1の右端のほうに"Audio Source"などの文字がみえますが、これは各種処理のブロックです。この文字の上に移動してドラッグすると図2のように"Audio Source"のブロックが画面上に配置されます。これでサウンドボードから信号をパソコンに取り込む準備ができました。今度はコンピュータからサウンドボードに出力します。右の"Audio Sink"の文字上にマウスポインタを移動してドラッグしてきます。あとは"Audio Source"の出力端子と"Audio Sink"の入力端子をクリックすると図3のように結線されます。Audio Source"の詳細情報をパラメータとして入力する必要がありますが省略します。それらのパラメータを入力したと仮定し、あとは実行ボタン



図3 "Audio Source" と" Audio Sink "の結線

このように、簡単なシステムならば短期間で作ることができるようになりました。なんとなく状況を つかんでいただけましたでしょうか。。

### ■GNU Radio のインストーラ

ソフトは下記 URL からダウンロードしてください。 http://www.gcndevelopment.com/gnuradio/index.htm

サイズが 467MB あります。筆者の環境ではダウンロードに 10 分程度かかりました。今回は、 Windows 10 を使用しています。Linux 版もあります。Windows 版には 32 ビット版の OS に対応 したソフトはありません。

GNU Radio でフロー図を作成すると Python のコードが出力されます。Python の処理系を別途イン ストールする必要はありません。Python には Version2(2.7 系)と Version3(3.9 系)があります。今 回は Version3 系の v3.8.2.0/v2.0 をダウンロードしました。ダウンロード後にダブルクリックする とインストールが始まります。デフォルトのインストール先は C:¥Proram Files¥GNU Radio-3.8 で す。インストールのために 1.6GB ほどの領域が必要です。また、大きなシステムのデジタル信号処理 を行おうとするとかなりマシンパワーが必要になるかもしれません。私は CPU に大きく負荷をかける ほどの処理を行わせていませんが、それでも何回も GRC がフリーズしたことがあります。フロー図を 作成したらこまめにセーブしてください。(フロー図の作成方法は次回説明します)

# ■システム例

どんなものができるのかは、現物を見ていただくのが一番早いかもしれません。最終目標はバンドパス フィルタの作成ですが、今回は例をみていただくだけでフロー図の作成はしません。GNU Radio で作 成した簡単なシステムをみていただいてイメージを持っていただければと思います。なお、GNU Radio で実際に高周波信号を入出力するためには、SDR フロントエンドと呼ばれる GNU Radio が対応して いるハードウェアを準備する必要があります。私が使ったことのある SDR フロントエンドと呼べる装 置は、写真 3 の RTL-SDR くらいですのであまり難しい装置の開発についてはわかりません。また、 今回の記事ではこのようなフロントエンドは全く必要ありません。



写真3 RTL-SDRの例(広帯域受信機として使えます。数千円程度で入手可能)

フロントエンドが不要なオーディオの入出力処理に焦点を絞った処理のフロー図を以下に 2 例示して います。図中で様々な数値が見えていますがあまり気にせずにみてください。

# 例1 オーディオ信号の入出力

図3で示した例を少し拡張します。せっかくパソコンの中を信号が通っていくのですから、オシロス コープのように時間に応じてどのように波形が変化するのかを見たいと思います。これを実現するた めにブロックを追加します。 "QT GUI Time Sink"というブロックです。実行すると図4のように なります。



図4 Audio 信号の波形観測ブロック

図5が上記図4のシステムを実行したものです。



図5 Audio 信号の波形観測結果

"Multiply Const"と書かれたブロックが見えますが、これは掛け算を意味しています。1以上の掛け 算を実施すると信号が増幅されます。1未満の掛け算だと信号が減衰されます。アンプがこれだけで実 現されるわけです。上部に見える"Vol"と書かれたスライドバーで掛け算する数がかわりますので、 これで音量がかわるのでボリュームとして使えるわけです。 少し脱線してみます。増幅回路として使用してみます。試しに電子ブロックでゲルマニウムラジオを作成し、その出力をサウンドボードに繋いでみました。私と同年代のラジオ少年だった OM の方には電子ブロックは懐かしい機器ではないかと思います。アンテナはワイヤーアンテナにつなぎました。放送局が数局重なって聞こえてくるので選択度はよくありません。クリスタルイヤホンでは何とか聞こえる音量です。クリスタルイヤホンを抜き、その位置にコンデンサを一つ接続し、その出力をオーディオケーブルでパソコンに入力しました。実際の音をお聞かせすることができませんので残念ですが、大きな(割れた、複数の局が同時に聞こえている)音で鳴っていました。ちなみに、ゲルマニウムラジオの検波出力をわざわざ GNU Radio 経由でスピーカに通さなくとも、増幅回路を持つスピーカにつなげば同じように音は大きくなります。あたりまえですが。



写真 4 ゲルマニウムラジオ出力のアンプとして使用する (写真中の赤線を"Audio Source"に接続しました)

例2

オーディオ信号をデジタル信号に変換し、ローパスフィルタやバンドパスフィルタを通して信号を変換します。



図6 ローパスフィルタやバンドパスフィルタによる信号抽出

GNU Radio にはローパスフィルタやバンドパスフィルタのブロックが準備されています。したがって、 あとは使いたい周波数やサンプルレートなどのパラメータを設定すればデジタル信号処理が実現でき ます。また図のように複数の Audio 入力(Audio Source)からの信号を別々に処理し、最後にその結果 をそれぞれ右チャンネルと左チャンネルから出力するというようなこともハードウェア的に回路を付 け替えたりしなくともできます。

いかがでしょう。面白い道具ですね。このソフトウェアにもデジタル信号処理でつきもののある現象 があります。デジタル信号処理では大量のデータを記憶して一括で処理することもありますので、こ のような処理では時間遅延が発生します。上記バンドパスフィルタを作成したのですが、受信信号は フィルタが効いて聴きやすくなりましたが、電鍵を叩くと無線機から自分が叩いた電鍵の音がわずか ですが遅れて聞こえてきます。打鍵して遅れて音が出ますのでタイミングが狂ってしまいます。遅れ が出るのはデジタル信号処理の特性上仕方がありませんが、とても打てません。そのようなことも頭 の片隅に入れて次回にご期待ください。