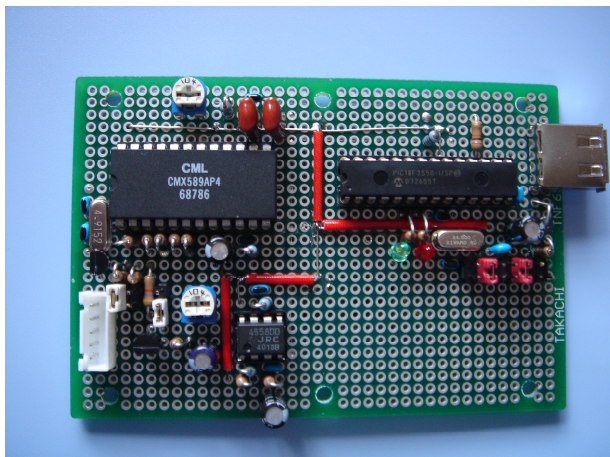


D-STAR デジピータ・インターフェース

安田 聖 7M3TJZ

D-STAR の DV モードの基礎

DV モードの各種無線機、アダプター、インターフェース等を制作する場合、この手順を知らなければ制作することができない。この手順については、JARL から公表されている「アマチュア無線のデジタル化技術の標準方式」に書かれていますが、この情報だけでは不十分であり、完全な互換性のある無線機等は制作できない。このため、これまで DV モードの機器の自作の障害となってきた。本記事では、DV モード機器を製作するのに最低限必要な情報について、解説し、その応用例として WiRES、IRLP、や EchoLink のノードに相当するインターフェースの製作の方法について解説する。



1. 音声パケットとフレーム

DV モードでは、3600bps の音声と 1200bps の

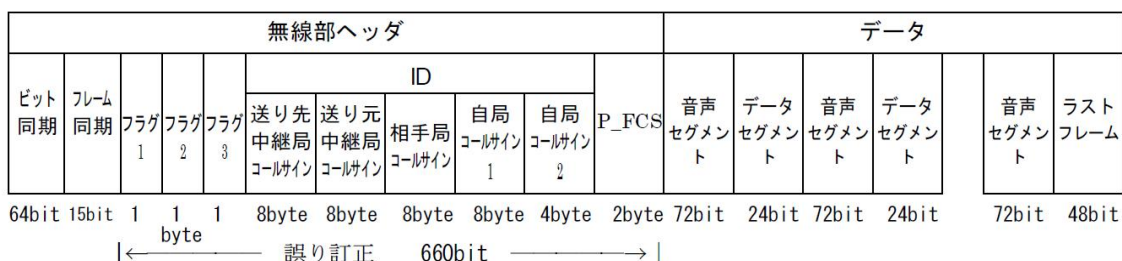
データを、4800bps のデジタル信号として送受信しています。PTT が押されてから PTT が離されるまでの間に送信されるデジタル信号の一連の情報を音声パケットと呼び、下記のような構成（フレーム）になっています。この音声パケットを現在市販されている DV モード対応の無線機では、同期信号方式で 4800bps の GMSK (BT は、0.5) として FM 変調する方式が採用されています。この音声パケットの構成は、下記の図のようになっています。この図からもわかるように、音声パケットは、大きく分けて無線部ヘッダーとデータ部（音声と簡易データを含めてデータと呼ぶ）から構成されています。

このパケットは、PTT が押されますと、無線部ヘッダーが送信され、これに引き続き音声セグメント、データセグメント（簡易データ部分）が交互に送信されます。また、PTT が離された場合、送信すべき音声セグメントもしきはデータセグメントが無くなった時点で、ラストフレームが送信されます。これが基本です。

ただし、D-STAR では、同期信号方式が採用されているため、無線部ヘッダーには、本来必要な情報の前に、ビット同期、フレーム同期を取るための情報が付加されています。

2. 無線部ヘッダー

PTT が押された直後に中継先等の経路情報やレピータ等への動作を指定する情報が送信されます。この情報は、PTT を押された直後に 1 回送信されるだけです。この情報を取りこぼすと、レピータ等では中継できないこととなります。ただし、シンプレックスの場合は、再同期（再同期の項目を参照してください）機能が働き、コールサイン



の情報が表示されなくても音声による交信は正常にできることがあります。以下に、各フィールドの説明を示します。

2. 1 ビット同期

変調方式が GMSK の場合、64 ビットを標準として 10101010.... (16 進数で aa aa aa aa ..) を送信します。受信の場合は、復調して得たビット列の中に、このビット列をパターンを検出すると、ビットのタイミングを取ると共に次のフレーム同期に備えます。

2. 2 フレーム同期

先のビット同期がとれただけでは、情報の初めのビットを知ることができません。このため、送信側ではフレーム同期信号と呼ばれる特殊なビットパターンをビット同期信号に引き続き送信します。D-STAR では、フレーム同期信号として 111011001010000 の 15 ビットのビット列が使用されています。このビット列を検出しますと、この後のビットからが情報 (図のフラグ 1 以後の情報) と言うこととなります。

2. 3 フラグ 1-3

DD と DV の区別、レピータの使用の有無等の情報をセットします。またレピータからの応答情報もこのエリアが使用されます。現在使用されているのは、フラグ 1 だけで、フラグ 2, 3 は、0 がセットされています。下記にフラグ 1 の割り当て表を示しておきます。

上 位 bit					
Bit	7	6	5	4	3
1	データ	中継	割込あり	制御	緊急通信
0	音 声	直接	割込なし	データ	通常通信

下 位 bit				
2	1	0	機 能	備 考
1	1	1	レピータ局制御	レピータ局の制御モード
1	1	0	自動応答	自動応答時に使用
1	0	1	(未使用)	(未使用)
1	0	0	再送	再送要求フラグ
0	1	1	ACK	ACK フラグ
0	1	0	応答ナシ	応答ナシ通知フラグ
0	0	1	中継不可	中継不可通知フラグ
0	0	0	NULL	NULL

2. 4 ID

自局、相手局、レピータ 1、レピータ 2 のコールサインを設定します。この情報を使用して、中継先等の経路を決定します。

2. 5 P_FCS

フラグ 1 から自局コールサイン 2 までの情報に対するエラーチェック用の CRC です。計算方法は、「無線部ヘッダーの処理の詳細」の「16 ビットの CRC の生成」を見てください。

これらの情報 (フラグ 1 から P_FCS まで) は、そのまま送信されるのではなく、パルス性の雑音に対してエラー訂正、及び検出ができるようにするため、各種の処理を行い、本来 41 バイトの情報を約倍の 82.5 バイト (660 ビット) に拡張 (冗長化) して送受信しています。この各種処理の内容は、

- b) 畳み込み処理
- c) インターリーブ処理
- d) スクランブル処理

です。詳細については、「無線部ヘッダーの処理の詳細」の項目を参照してください。

3. データ (音声と簡易データ)

DV モードでは、音声を 8KHz、16 ビットでサンプリング²し、このサンプリングした値を、DVSI

¹ フレーム同期信号が 15 ビットですので、ビット同期の最後の 0 をフレーム同期の最初に加えて、0111011001010000 の 16 ビット (16 進数で 76 50) とすると処理がし易くなります。ただしこの場合は、ビット同期信号を 01010101 (16 進数で 55) として処理する必要があります。さらに、4 ビットシフトして、処理すると、データ部分がバイトの切れ目となり、後の処理がしやすくなります。

² A/D、D/A 変換で、リニアコーデックと呼ばれています。

社の AMBE2020³を使用して 2400bps に圧縮し (コーデックと呼ぶ)、さらに AMBE2020 で 1200bps のエラー訂正符号 (FEC⁴) を付加して、3600bps にします。この音声信号に 1200bps の簡易データ⁵を加えて 4800bps として、9 バイト (72 ビットの音声) と 3 バイト (24 ビットの簡易データ) を交互に送信します。ラストフレームだけ、24 ビットでなく、48 ビットを送信します。(ラストフレームの項を参照してください。)



4. ラストフレーム

PTT が離されますと、送信すべき AMBE2020 からの情報もしくは簡易データの情報が無くなると、データセグメントの最初から、32 ビットの 10101010.. の繰り返しに引き続き、0001001101011110 のビットパターンを送信します。(16 進数で、aa aa aa aa 13 5e)

以上が、DV モードの基本ですが、これ以外に下記の再同期信号が定期的に挿入されます。

³ AMBE2020 によるコーデックは、その手順が公開されていませんので、DVSI 社の AMBE2020 もしくは AMBE2000 を使用しないかぎり互換性のある無線機等は実現できません。

⁴ AMBE2020 の FEC コード生成方式は、BLOCK CODE が使用されています。

⁵ このセグメントを使用して、再同期信号を周期的に挿入しているため、実際には 5% 程度、実質転送速度は遅くなります。また現在市販されています ICOM 社の D-STAR 対応リグでは、このフィールドを利用してメッセージ等の情報を転送しているため、更に遅くなり実質転送速度は約 960bps 程度とされています。

5. 再同期信号

基本的には、上記した信号を送信もしくは受信すれば良いのですが、交信中に QSB や QRM で同期が外れることがあるため、定期的に再同期のための信号をデータセグメントを利用して挿入しています。再同期信号は、最初のデータセグメントに挿入した後、21 個目毎に挿入されます。このビットパターンは、変調方式が GMSK の場合、101010101011010001101000 (16 進数で、aa b4 68) の 24 ビットを使用しています。

6. 音声パッケージの例

下記に IC-U1 で送信されたビット列の例を 16 進数で示します。音声セグメントがバイトの切れ目になるように調整してあるので、aa aa aa aa でなく 55 55 55 55... がビット同期信号となります。7 65 0 がフレーム同期です。1 cb 3e... から ... 59 26 09 までが無線部ヘッダーです。以後、9 バイトの音声、3 バイトのデータの繰り返しです。データ部分の値が、aa b4 68 が再同期信号です。また、aa aa aa aa 13 5e がラストフレームです。⁶

```

44 ff 6f e7 7e a3 87 f7 6f ff bf c0 PTT ON
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 3f f0 3f ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff d5 55 55 ビット同期
55 55 55 55 55 55 55 55 55 55 55 55 フレーム同期
65 01 cb 3e f2 20 23 d9 51 50 53 3d 無線部ヘッダー
53 8d d1 48 d7 ae 08 6d 86 d9 e6 09
b4 4d 9d 54 e5 3c 83 3f 4e ff ac 37

```

⁶ 表示されているビット列の例は、インターフェースを通して受信したもので、データ部分がバイトの切れ目になるようにフレーム同期信号を検出した時点でインターフェースのプログラムで調整するようになっています。

71 a5 43 45 91 bc 0f 4e f0 5e fc 35
71 e7 76 9e 6f bb 87 12 92 7b 9f 60
37 40 bf 18 ad 57 57 cd 4e 09 8c a9
38 40 b5 62 0e f5 2e ed 27 59 26 09
35 36 18 dd f6 19 0a 0e b7 aa b4 68 音声+データ
5d f3 5c d2 c5 4f e5 29 87 a4 f2 c9
75 f5 6d c2 49 8c f5 a4 a5 0e d0 5b
51 30 2d c6 45 4e e1 e6 a7 a4 b8 6b
73 d1 58 8e a4 fd 47 6d ff cc d8 cd
7f 21 74 19 4e 0e d1 27 d2 a4 f6 eb
5f 63 15 03 d2 0b 64 44 cf 9c b8 6b
51 f7 55 58 81 74 4d b9 8b a4 30 e3
01 f4 40 c0 07 fc 5b 84 4e 0a f6 0b
11 b7 70 d0 85 af 73 04 3f a4 78 0b
71 34 5c da 41 dc dd a1 97 84 30 43
61 17 0c 58 b3 f0 a2 48 af a4 f6 cd
7f 92 7d c8 0e b6 35 cf bd e2 40 05
47 57 4c c4 8e d7 09 86 8d a4 d8 9b
4b 17 3d c8 c9 87 1d cd de 54 f6 0b
5d f7 7c dc 94 d8 68 88 db a4 f6 cd
55 b2 6d de 8a 2d d1 a4 37 0a f6 53
69 35 4c d0 5e 68 4c 8a 89 84 e8 af
55 71 2c c6 07 0c e1 2f a7 68 94 af
f9 14 65 48 9e 63 15 c1 b3 68 94 af
75 32 34 48 07 e6 71 b3 99 68 94 af
e5 16 44 50 5f 83 35 42 b1 aa b4 68 再同期
55 97 55 d2 b0 99 12 9f b3 a4 f2 c9
f9 96 25 40 96 60 01 82 a3 0e d0 5b
6b f3 49 c8 1d 1b b3 25 bd a4 b8 6b
cf c0 6c 97 d8 51 29 02 37 cc d8 cd
f5 31 70 c5 ed 94 f7 55 8b a4 f6 eb
51 97 71 c5 7e c3 bb ea 81 9c b8 6b
7d 71 35 40 83 16 65 fe 88 a4 30 e3
f9 13 04 4c 16 f1 3d 0e d1 0a f6 0b
e7 14 45 ca 75 51 3f 63 ed a4 78 0b
cf 97 1d 4c 7b ab bf 35 81 84 30 43
21 38 59 53 74 0f dc 74 15 a4 f6 cd
05 b9 35 12 ba 15 42 f4 e3 e2 40 05
19 3d 66 a3 50 cd 06 4f aa a4 d8 9b
85 f1 15 93 f1 a0 d7 a2 db 54 f6 0b

f9 74 5c 4a b5 f2 6f 96 8b a4 f6 cd
51 d3 5c 48 b9 31 ba c5 cf 0a f6 53
79 f1 7d c2 cf 4c ed e9 a6 84 e8 af
4f 42 70 0d dc c1 6b cc c2 68 94 af
f5 c3 50 9b 1f 7d 80 ce 5d 68 94 af
e9 a5 09 99 7e 0c fe 50 54 68 94 af
85 fb 15 09 7b 55 2d 10 69 aa b4 68 再同期
03 fb 7e 4f f7 ae 0f d5 c3 a4 f2 c9
途中省略
eb 77 0d d4 d5 a1 59 38 d7 a4 b8 6b
0d 59 ad 4e 55 3b 55 b7 93 cc d8 cd
2d 5b df 42 97 d8 51 32 d1 aa aa aa ラストフレーム
aa 13 5e

このビット列を送信、受信ができればDVモードの対応機が出来ることとなります。

7. 無線部ヘッダーの処理の詳細

a) 16ビットのCRCの生成

処理方法は、CCITT-CRC方式の生成多項式

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

が採用されていますが、Web等で公開されている上位ビットから処理する方式でなく、下位ビットから処理する方法が採用されています。また、初期値は、16進数でFFFFです。最終結果は、補数を取った後、上位バイトと下位バイトを入れ替えています。

Cでプログラムされたサンプルプログラムを

<http://d-star.dyndns.org/program/crc.c>

に置いてありますので参考にしてください。

b) 畳み込み処理

41バイトの情報に冗長化を持たすために、情報1ビットを2ビットに拡張してエラー訂正に対応しています。この方法として、畳み込み符号化が使用されています。畳み込み符号化をする場合の条件は、

符号化率 1/2

拘束長 3

ハングオーバービット 4

生成多項式

$$G_1(D) = D + D_1 + D_2$$

$$G_2(D) = D + D_2$$

です。つまり、変換するビットは、それより前の2ビットの情報を加味して2ビットに変換されます。なお、 D_1 、 D_2 の初期値は、0です。また、畳み込み処理をされたビット列を元に戻すときの処理のために、最後に2ビットの0を加えたビット列に対して畳み込み処理をします。つまり、41バイト(328ビット)に2ビットを加えた330ビットに対して畳み込み符号化を行い、660ビットにします。なお、41バイトに関しては、各々バイトの下位バイトから処理を行います。Cでプログラムされたサンプルプログラムを

<http://d-star.dyndns.org/program/convolution.c>に置いてありますので参考にしてください。また、畳み込み符号化された情報を元に戻す方法としては、Viterbiの方法が使用されますが、このCのプログラム例も

<http://d-star.dyndns.org/program/viterbi.c>に置いてありますので参考にしてください。

c) インターリーブ

畳み込み符号化で冗長化されたビット列を、そのまま連続して送信するのではなくパルス性の雑音に対応するため、ビットを24ビット毎にばらして送信します。最初のビットはそのまま(1ビット目として以後、説明します)、次のビットは、25ビットに、3ビット目は49ビット目と言うように送信します。このため、660ビットを一度並び替えた後、送信する必要があります。この処理を、テーブルを使用して実行するCのサンプルプログラムを

http://d-star.dyndns.org/program/interleave_gen.c

に置いてありますので参考にしてください。受信したデータを元に戻すCのプログラムは、

http://d-star.dyndns.org/program/interleave_decode.c

に置いてあります。

d) スクランブル

スクランブル処理は、同じ記号が続き変調信号が偏ることによるエラーを避けるため行われる処理です。この処理には、下記の式が使用されます。初期値は、16進数でFFです。

$$S(x) = x^7 + x^4 + 1$$

この値は、ビットの位置で固定されていますので事前に計算をしておき、テーブルの形で処理が可能です。

<http://d-star.dyndns.org/program/scrambletable.c>

にCプログラムで使用できるテーブルが置いてありますので、使用してください。使用方法は、送信する順、もしくは受信した順に上位ビットから下位ビットにビットをセットした後、各々のバイトの位置に相当するテーブルの値と排他的論理和(XOR)を取ります。送受信とも同じテーブルの排他的論理和で処理できます。なお、簡易データフィールドに関しては、データセグメントが始まる毎に、このテーブルの最初から3バイトを各々のバイトと排他的論理和を取った後、上位ビットと下位ビットを入れ替えます。(送信順序が下位ビットから送り出しているためです。)ただし、再同期信号とラストフレームには、この処理を行いません。

無線部ヘッダーの生成から復調までの一連のプログラムを例も

<http://d-star.dyndns.org/program/sample02.c>の置いてありますので、参考にしてください。

これらの処理ができれば、無線部ヘッダーの送受信はできることとなります。同期信号からバイトの切れ目を検出し、後は、データ部分(音声+簡易データ)の情報を、無線部ヘッダーに引き続き、AMBE2020から送られてくるデータを72ビット単位で、また簡易データの24ビット(この情報には、スクランブル処理を行います)単位

で、交互に送信、もしくは受信したデータを72ビットと24ビットに切り分け、AMBE2020と簡易データ端子に渡せば、受信ができることになります。

8. 再同期信号の処理

送信時には、最初のデータセグメントに101010101011010001101000を入れ送信し、その後データセグメントの21個目毎に同じビットパターンを入れます。受信時には、このビットパターンを検出した場合は、ビット合わせ、そして音声セグメントの位置合わせを行います。

以上がDVモードで機器が備えていなければならない機能です。今回は、このうちAMBE2020を除いた部分をPICを使用して実現したインターフェースの製作とその応用例としてエコーサーバーを作成しました。この詳細については、次の制作記事とインターフェースの使用方法の記事を見てください。