D-STAR　DV 附加装置制御プログラムの作成

Development of DV adapter control program

若鳥陸夫†
Rikuo (Rick) Wakatori
†(株)アールエルエル研究所
Rick Logical Laboratory, inc.

キーワード：アマチュア無線，D-STAR，AP132 **応用，通信制御**

1　はじめに

このプログラムは，アマチュア無線のデジタル化実験プロジェクトの一つである（社）日本アマチュア無線連盟（JARL**）が提唱した** D-STAR**（由来：**Digital Smart Technologies for Amateur Radio**）仕様の自作無線機の支援ソフトウエアである。**

DV **附加装置と呼ぶ金物**[1 **及び** 2]**を自作することによって，普遍的なアナログ無線機をデジタル化し,計算機からその** DV **附加装置及び無線機を同時に制御する機能を提供する。**

なお，ここでの DV **とは，**Digital Voice **の略記号であって,** D-STAR **方式は，音声電圧波形を** GMSK **変調し，それに** TCP/IP **風な頭情報を付加したデータ流れで構成されている。**Internet **との整合性もよく，レピータ間はインターネットで接続され，最大で四つのレピータを串刺しに使い，電波の届き難い遠方との交信を実現できる。**

2　利用者要件

2.1　金物の構成

金物の全体構成は，図１のとおり，アマチュア無線機（例：Yaesu　FT-817**）に** DV **附加装置を** 2 **本の信号線(規格：**ANSI/EIA-232-E**)によって計算機に接続する。**

無線機の通信パラメタは 9600bps，8 **ビット，パリテイなし，**1 **スタートビット，**2 **ストップビットとし，**DV **附加装置の通信パラメタは** 9600bps，8 **ビット，パリテイなし，**1 **スタートビット，**1 **ストップビットとする。計算機の非同期通信出入り口は，他と共用させない。**

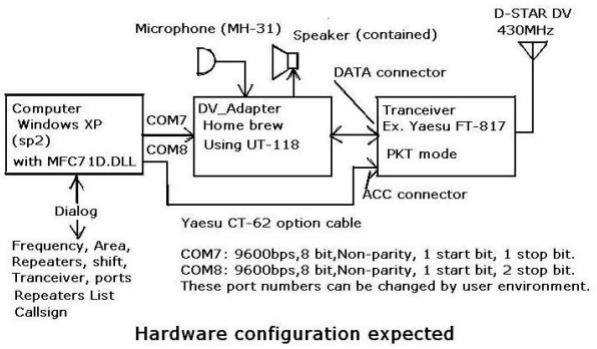

図１　金物の構成

2.2　対話入出力する要素

利用者が対話入出力する要素は，次のとおり，自局呼出符号，自局付加情報，呼出地域，自レピータ候補，相手レピータ候補，転送メッセージ，DV **出入り口番号，無線機型番，無線機接続口番号を記述又は選択する。**

2.2.1　自局呼出符号（8 文字以下）

自局呼出符号は, 日本国総務省から指定された識別符号とJARL に登録した機器番号とから英数字 8 文字で構成する 例 :7L1RLL D)。

このうち，7L1 **は前置詞（**prefix**），**RLL **は後置詞（**suffix**),** D **は** D-STAR **独特な** 1 **英字による機器識**

別子（A～Z 又は空白）である。
この前置詞は，国際通信連合（ITU）によって，国又は離島に割り当てられている。

2.2.2　自局付加情報(4 文字以下)

任意の付加情報を英数字 4 文字まで記入できる。この付加情報は，多くの場合，相手局画面にも表示される。

2.2.3　呼出地域（7 文字以下）

日本国内の場合，1～0 の 10 地域に区分されている。ここでは単向通信：Simplex，関東：JP1，東海：JP2，近畿：JP3，日本国の中国：JP4，四国：JP5，九州・沖縄：JP6，東北：JP7，北海道：JP8，北陸：JP9，信越：JP0 とする。ここは，グアム島：KH2，ケープタウン：ZS5 など国際識別子でもよい。ただし，後述するレピータ表の地域識別子と文字列が一致する必要がある。

2.2.4　自局レピータ候補

地域識別で利用可能な DV モードレピータの候補を表示し，利用者はその中の一つを指定する。地域識別を“ Simplex ”(単向通信)と指定した場合，周波数候補例を表示する。

この候補は，1 局当たり 54 文字以下の規定様式の要素とし，最大 256 局までとする。

2.2.5　相手局レピータ候補(256 局以下)

相手局のレピータ候補は，“ Simplex ”の場合を除き，その国又は地域で登録されたすべての局を表示し，利用者がその内の 1 局を選択する。

この候補は，地域識別子（7 文字以下），レピータの名前及び略称，周波数，複向通信の周波数偏移周波数及び正負，呼出符号（8 文字以下）並びに通信出入り局の呼出符号（8 文字以下）によって構成する。

2.2.6　DV 附加装置出入り口番号(1 文字)

DV 附加装置の出入り口は，COM1～COM9 までの 9 種類を 1～9 までの番号によって指定する。

この番号は，計算機の構成を変えない限り一定になるので，前回，成功した利用の番号を記憶し，次回の利用のときに提示する。

2.2.7　無線機型番(15 文字以下)

予めプログラムし試験した利用可能無線機型番のうちから任意の一つを選択する。この無線機型番は，多くの場合，前回利用成功した型番を記憶させておき，プログラム始動時に表示する。

2.2.8　無線機出入り口(1 文字以下)

無線機を接続した出入り口（COM1～COM9）のうちから一つを数字によって指定する。この出入り口は，多くの場合，固定利用されるので，前回利用した数字を次回の起動時に表示する。

2.2.9　送信データ(20 文字以下)

無線機から送出する文字データとして 20 文字以下を記述して，ＤＶ附加装置に送る。

2.3　表示情報

利用の進展にしたがって，次の事項を表示する。

2.3.1 交信局及びレピータ

相手呼出符号，自呼出符号，最寄レピータ，インターネットへの出入り局呼出符号を受信した場合，表示する。

2.3.2 自局無線機周波数

自局が選択したレピータの周波数を kHz 単位で表示する。

2.3.2 レピータ偏移周波数

複向通信の場合，レピータへの上り周波数とレピータからの下り周波数の差が局ごとに決められて候補レピータ表に記載されているので，そのレピータを選択すると表示する。

2.4　制御ボタン

2.4.1.START/STOP

設定項目の選択又は入力が終了したら押すと，その設定データが DV 附加装置及び無線機に送られる。

2.4.2. TX_msg

メッセージを投入終わった合図で，データは DV 附加装置経由で無線機に送られ，送信される。

2.4.3. EXIT

プログラムを終了させる。

3.実装

Windows API32 関数［例えば，文献 5］を多用して，MS_VisualC++.net[文献 4]の対話型プログラムで実装した。図 2 に制御画面を示す。

主要な制御は，目的指向プログラム風にメッセージ駆動ではあるが，ANSI/EIA 規格 232-E 版の規格に適合する通信界面を制御する部分及びファイル取り扱い部分の記述流儀は，C++というより C 言語風になってしまった。

原始プログラムは，約 1000 行弱であった。

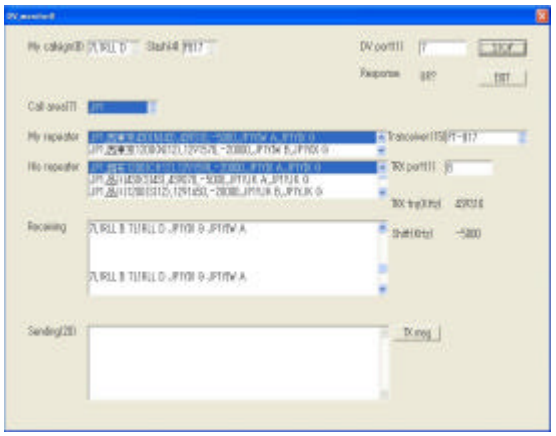

図２　制御画面

4.おわりに

4.1.期待される効果

このプログラムによって，金物による制御部を作らなくても，手持ちの Windows XP(SP2)に MFC71D.DLL を載せ，この目的プログラムを走らせることによって，DV 附加装置と無線機との周波数を自動的に合わせることができる。このプログラムによって，DV 附加装置と無線機とが連動させ易く，操作が簡便になったと言える。

4.2. 機能の不足点

この原始プログラムは機能的な課題として一般呼出しに応答する場合，その局の呼出符号を捕捉して，自彼間の経路を自動設定する機能がない。

プログラムとして受信力に一時記憶装置がないことによる処理時間の限界がある。

4.3. 安定性

実行安定性は，環境によって不安定になる要素が潜んでいる懸念がある。

4.4. 書き方の改良

書き方として class の定義がない，C++を使いながら実体は C 言語風であって，模範プログラムとして教育に利用できないなどの欠点がある。

4.5. 他のプラットフォームへの移植

利用者を拡大するには，他の言語処理系，例えば, Borland C++への移植, 操作系 Linux の GNU C++への移植などの課題もある。これらは，一人では対処しがたいので，原始プログラムを開示し[3]，それらの処理系に習熟した人達によって書き換え，向上されることを願っている。

参考文献

[1] 安田，自作 D-STAR DV モードアダプター，JARL News，2007 年秋号，pp18～pp22，（社）日本アマチュア無線連盟

[2] 若鳥，YaesuFT-817 を D-STAR 機に改造，URL ＝ http://www1.u-netsurf.ne.jp/~7l1rll/radio.html, 初版：2007 年 8 月

[3] 若鳥, D-STAR DV Adapter 制御プログラム，URL ＝ http://www1.u-netsurf.ne.jp/~7l1rll/radio.html, 初版：2007 年 9 月

[4] Microsoft，Visual C++.net SDK，2003

[5] 土井，那須，上田著，Win32API 完全マスタ，CQ 出版社，2006 年 8 月，第 8 版

（参考）原始プログラムの主要部

ボタン操作によって送られたメッセージの解読処理部分を抜粋して示す。ここらは，Windows API32 **関数の応用だから，**C++**と** Windows API32 **の双方の知識を必要する。**

```
case WM_COMMAND:
switch( LOWORD( wparam ) ){
        case ID_0: // "START" or "STOP"
                if( sw ) {  // START -> STOP
                        ComEnd();
                        if ( TRXsw ) TRXend();
                        SetDlgItemText( hwndDlg, ID_0, "START" );
                }
                else {     // STOP => START
                        ComBegin( hwndDlg );
                        TRXbegin( hwndDlg );
                        SetDlgItemText( hwndDlg1, ID_0, "STOP" );
                        send_DV_TRX_selections( h1 );
                }
                return TRUE;
        case ID_1: // "EXIT"
        case IDCANCEL:
                        ComEnd();
                        TRXend();
                        EndDialog( hwndDlg, FALSE );
                        return TRUE;
        case ID_C: // callsign
                        i = SendDlgItemMessage( hwndDlg1, ID_C, EM_GETSEL, 0, 0);
                        s = LOWORD( i );  // start potsition
                        d = HIWORD( i );  // end position + 1
                        if ( s < d ){
                                GetDlgItemText( hwndDlg1, ID_C, s1, 10 );
                                j = 0;
                                for ( i = s; i < d; i++ ){
                                        char c  = s1[i];
                                        if ( ( c != 0x0d ) && ( c != 0x0a ) ){
                                                cbuf2[j++] = c;
                                        }
                                }
                                cbuf2[j] = NULL;
                                strcpy( myCallsign, cbuf );// = &cbuf2[0];
                                myCallsign[8] = NULL;
                        }   return TRUE;
```