

ソフトウェア TNC の KISS ポートと AGW ポート

Fumio ASAI/JA3TDW, JARL and AMSAT-NA member

1. KISS プロトコル

AX.25 パケット通信用の標準的なターミナルノードコントローラ (TAPR TNC-2 クローン) は① COMMAND MODE, ② CONVERSE MODE, ③ TRANSPARENT MODE, ④ KISS MODE という 4 つの動作モードをもつ。このうち KISS MODE で動作する TNC (KISS TNC) とコンピュータとの間でやり取りされるデータを KISS (Keep It Simple, Stupid) データと呼ぶ。KISS データは Phil Karn/KA9Q 氏が提案した KISS プロトコル (KISS フレームフォーマット) [1] に従って構成される。元々 KISS プロトコルはコンピュータが TNC を制御して人の手を介さずにパケット通信を行うために考案されたが、以下の 3. で述べる利点があるため現在ではアマチュア衛星や超小型人工衛星 (CubeSat) のミッションデータやテレメトリデータを取得・保存・解析するために利用されている。そのため DireWolf[2] や SoundModem[3] のようなソフトウェア TNC には KISS TNC の機能 (KISS protocol over TCP socket) が実装されている。Mobitex のような AX.25 以外のプロトコルに対応させた SoundModem にもこの機能が実装されている。

2. KISS TNC と KISS フレーム (AX.25 プロトコルの場合)

(1) KISS TNC

- ① TNC はモデム機能、PAD 機能、プロトコル処理、KISS 置換/逆置換 (表 2) を行う
 - ・モデム機能：アナログ信号とデジタル信号の変換 (変調と復調)
 - ・PAD (Packet Assembler Disassembler) 機能：パケットデータとシリアルデータの変換
 - ・プロトコル処理：NRZI 符号化、フラグ、ビット詰め、FCS などに関連する処理
- ② TNC と PC の間で送受信されるデータは KISS フレームフォーマットに従う

(2) KISS フレーム

- ① KISS フレームにはデリミタ (FEND) 呼ばれるバイトデータ (0xC0) が付加されている (図 2)
- ② TNC を制御するために COMMAND と呼ばれるバイトデータが付加されている (図 2 と表 1)
- ③ AX.25 フレームのフラグと FCS は含まれず、ビット詰めされていない (図 1 と図 2)
- ④ 透過性を保証するため KISS 置換 (KISS substitution) と呼ばれるコード変換が行われている (表 2)

FLAG	ADDRESS	CNTL	PID	DATA	FCS	FLAG
------	---------	------	-----	------	-----	------

図 1 AX.25 フレーム

FEND (0xC0)	COMMAND (0x00 etc)	ADDRESS	CNTL	PID	DATA	FEND (0xC0)
-----------------------	-----------------------	---------	------	-----	------	-----------------------

図 2 KISS フレーム

表 1 KISS フレーム COMMAND

0x00 : データ	0x01 : TXDELAY 設定	0x02 : PERSISTENCE 設定	0x03 : SLOTTIME 設定
0x04 : TXTAIL 設定	0x05 : DUPLEX 設定	0x06 : オプション	0xFF : KISS MODE 終了

表 2 KISS 置換/逆置換 (文献 4 より引用)

KISS 置換
① Add a C0 00 to the beginning of each frame and a C0 to the end.
② Every time you have a C0 in your data, substitute a DB DC.
③ Every time you have a DB in your data, substitute a DB DD.
KISS 逆置換
① Each data frame will start with C0 00 and end with C0.
② Every time you see the bytes DB DC you should delete them and put in a C0.
③ Every time you see the bytes DB DD you should delete them and put in a DB.

ソフト TNC に実装された KISS ポートはアプリケーション(クライアントソフト)用の TCP/IP ソケット通信ポートで、両者で送受信される KISS フレームはプロトコルスタックでカプセル化されているが、通信用のモジュールやライブラリでシームレスに処理されるため意識する必要はない。

(3) AX.25 以外の KISS フレーム

AX.25 プロトコル以外の KISS フレームは元データ(図 1 の DATA)に KISS 置換を施したものになる。KISS フレームの一例を図 3 に示す。これは CubeSat の GreeCube が送信するテレメトリデータ(正確にはテレメトリデータに識別用データなどが付加されたデータ)の 16 進ダンプである。表 2 上 ①の処理によりフレームの先端には 0xC0,0x00 が、フレームの終端には 0xC0 がそれぞれ付加され、元データに含まれていた 0xC0 が表 2 上②の処理により 0xDB,0xDC に置換されている(赤線枠)。

```

*****
2023-05-21T08:10:44.604Z
Packet: #48
Packet Size: 105 bytes

0000: C0 00 82 92 31 00 76 1A 01 34 64 69 D2 01 00 9C
0010: 0C DB DC 0A 14 00 DD 01 51 03 0E 0E 0E 0E 0F 0F
0020: 00 76 00 1F 20 6A 01 16 0D 21 95 F6 FF 95 11 0D
0030: 01 F1 00 9D 5A 00 33 00 14 00 BA 00 2B FB 92 11
0040: 7C 24 C7 0D BB FE FE 00 FE 00 0A 00 0B 00 00 00
0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060: 00 00 00 00 00 00 83 01 C0

...1.v..4di.....Q.....v..j...!.....
.Z.3.....+...|$......
...
*****

```

図 3 KISS フレームの例

3. KISS フレームの利点と元データの復元

KISS フレームには CR(0x0D)や LF(0x0A)、タイムスタンプなど、TNC を CONVERSE MODE で使用すると付加される元データとは無関係のデータが含まれないので、特にバイナリ形式のミッションデータやテレメトリデータの解析が容易になる。また、デリミタ(FEND:0xC0)により 1 フレームごとのデータが容易に識別・抽出できるという利点もある。

ソフト TNC に実装されている KISS ポートから送出される KISS フレームは元データに表 2 下の処理が施されているので、元データを復元するには KISS ポートから取得したフレームのデータに対して、まず FEND(0xC0)と COMMAND(0x00 etc)を除去し、次に表 2 下の処理の②と③を施す必要がある。図 4 に Python で記述した処理コードの例を示す。

```

kssData = so0.recv(bufferSize)
lenKssData = len(kssData)
dbFlag = 0
frameData = bytearray()
for i in range(2, lenKssData-1):
    v = struct.unpack('B', kssData[i])[0]
    if v == 0xDB:
        dbFlag = 1
        lenKssData = lenKssData - 1
    elif v == 0xDC:
        if dbFlag == 1:
            v = 0xC0
            dbFlag = 0
    elif v == 0xDD:
        if dbFlag == 1:
            v = 0xDB
            dbFlag = 0
    if dbFlag == 0:
        frameData.append(v)

```

図 4 元データの復元処理

4. AGW ポートと AGW API

(1) AGW ポート

ソフトウェア TNC には KISS-ポートとともに AGW ポート(AGWPE Server Port)が実装されている。AGW ポートは KISS ポートと同様に TCP/IP ソケット通信ポートであるが AX.25 プロトコル専用で、パケット通信用ソフト TNC の元祖とも言える AGW Packet Engine (AGWPE) [5]に初めて実装された。TCP ポート番号の初期値は 8000 で、これは後発のソフト TNC にも引き継がれている。

(2) AGW API

AGWPE をはじめとするソフト TNC は UISS のようなアプリケーションのリクエストに応じるために AGW API (AGW Application Program Interface) [6]を提供している。AGWPE のみ DDE ベースの API もサポートしているが、後発のソフト TNC に実装されている API は TCP/IP ベースのみである。そのため以下の解説では TCP/IP ベースの AGW API に限定する。

5. AGW フレームと AGW ヘッダ

(1)AGW フレーム

ソフト TNC とアプリケーションの間で送受信されるデータは AGW フレームフォーマットに従う。AGW フレームは AGW ヘッダ部と必要に応じてヘッダに付加されるデータ部で構成される。

(2)AGW ヘッダ

AGW ヘッダは AGW API のアクセスに必要な情報を含む 36 バイト固定長のバイト列である。AGW ヘッダの構成を図 5 に示す。各フィールドの役割の意味や解説は文献 6 に委ねるが、最も重要なフィールドは DataKind で、これによりリクエストやレスポンスの意味(内容)が識別される。

Field	AGWPE Port	Reserved	DataKind	Reserved	PID	Reserved	CallFrom	CallTo	DataLen	User
byte	1	3	1	1	1	1	10	10	4	4

図 5 AGW ヘッダの構成

(3)AX.25UI フレームと AGW フレーム

AX.25 パケット通信には接続型モードと接続レス型モードがあり、データの伝送にはそれぞれ I フレームと UI フレームが使用される。AGW フレームはどちらのモードにも対応しているが、ここでは CubeSat のコマンド/データの送受信やデジピーター衛星のアクセスなどに利用される UI フレームに関連する 3 種類の AGW フレームの概要を解説する。

①'m'フレーム

アプリケーションがソフト TNC にモニター(受信)した UI フレームの送信をリクエストするフレームで、DataKind は 0x6D である。'm'フレームの 16 進ダンプを図 6 に示す(0x を省略、以下同様)。

AGWPE Port	Reserved	DataKind	Reserved	PID	Reserved	CallFrom
00	00 00 00	6D	00	00	00	00 00 00 00 00 00 00 00 00 00 00
CallTo			DataLen	User		
00 00 00 00 00 00 00 00 00 00 00			00 00 00 00	00 00 00 00		

図 6 'm'フレームのヘッダ部

②'M'フレームと'V'フレーム

アプリケーションが生成した UI フレームの送信をソフト TNC にリクエストするフレームで、DataKind は'M'フレームが 0x4D、'V'フレームが 0x56 である。これらのフレームは AGW ヘッダ部とデータ部で構成され、デジピーターを指定する場合は'V'フレーム、指定しない場合は'M'フレームを使用する。指定するデジピーターの局数とコールサインはデータ部に格納する。一例として送信元 JA3TDW がデジピーター RS0ISS 経由で宛先 CQ にデータ test を送信する場合の'V'フレームの 16 進ダンプを図 7 と図 8 に示す。データ部のバイト数が 15 なので DataLen=0x0F となる。

AGWPE Port	Reserved	DataKind	Reserved	PID	Reserved	CallFrom
00	00 00 00	56	00	00	00	4A 41 33 54 44 57 00 00 00 00 00
CallTo			DataLen	User		
43 51 00 00 00 00 00 00 00 00 00			0F 00 00 00	00 00 00 00		

図 7 'V'フレームのヘッダ部

デジピーター局数	デジピーターコールサイン	送信データ
01	52 53 30 49 53 53 00 00 00 00 00	74 65 73 74

図 8 'V'フレームのデータ部

③'U'フレーム

ソフト TNC がモニターした UI フレームをアプリケーションに送信するフレームである。DataKind は 0x55 である。アプリケーションがこのフレームを受信するには事前に'm'フレームを送信しておかなければならない(送信は 1 回でよい)。'U'フレームの構成を図 9 に示す。AX.25 ヘッダ部とデータ部は 0x0D で区切られ、データ部の後に 0x0D,0x00 が付加される。

AGW ヘッダ部	AX.25 ヘッダ部	0D	データ部	0D 00
----------	------------	----	------	-------

図 9 'U'フレームの構成

②で例示した'V'フレームに対応する'U'フレームの AGW ヘッダ部、AX.25 ヘッダ部、データ部を図 10～図 12 にそれぞれ示す。AX.25 ヘッダ部は'U'フレームを特徴付けるヘッダ部で、テキストデータ(ASCII キャラクタ)のみで構成される。そのため図 11 では 16 進ダンプではなくテキスト表示している。'V'フレームと異なり、デジピーターの局数とコールサインはデータ部に格納されない。AGW ヘッダ部を除くフレームデータの総バイト数は 62+1+4+2=69 なので DaraLen=0x45 となる。

AGWPE Port	Reserved	DataKind	Reserved	PID	Reserved	CallFrom
00	00 00 00	55	00	00	00	4A 41 33 54 44 57 00 00 00 00
CallTo			DataLen	User		
43 51 00 00 00 00 00 00 00 00			45 00 00 00	00 00 00 00		

図 10 'U'フレームの AGW ヘッダ部

```
1:Fm JA3TDW To CQ Via RS0ISS <UI pid=F0 Len=4 P=0 >[10:00:26]
```

図 11 'U'フレームの AX.25 ヘッダ部

```
送信データ
-----
74 65 73 74
```

図 12 'U'フレームのデータ部

6. まとめ

デジピーター衛星をアクセスするためソフト TNC と組み合わせて使用するアプリケーションは既に数多く開発・提供されているので新規開発の必要性は乏しい。しかしこれらを含め、既存のアプリケーションで CubeSat のコマンド/データの送受信を行うのは難しい。バイナリデータの送受信機能を実装したアプリケーションを新規に開発するには KISS ポートと AGW ポートのどちらのポートを利用するか2つの選択肢がある。どちらも任意のビット列に対する透過性をもつが、AGW フレームは AX.25 プロトコルだけをサポートしており、フレームフォーマットが複雑である。そのためプロトコルに依存せずフォーマットが単純な KISS ポート(フレーム)を利用するほうがよいと思われる。AGW ポートの利用が適している I フレームを使ったパケット通信が下火になる一方、IL2P[7]のようにエラー訂正機能が実装されたプロトコルが主流になりつつあることを踏まえると、KISS ポートの理解と活用は今後ますます重要になるとと思われる。

* 参考資料

- [1] Phil Karn/KA9Q, The KISS TNC: A simple Host-to-TNC communications protocol : <http://www.ka9q.net/papers/kiss.html>
- [2] DireWolf : <https://github.com/wb2osz/direwolf>
- [3] SoundModem : <http://uz7.ho.ua/packetradio.htm>
- [4] John A. Hansen/W2FS, TNC-X: An Expandable Microcontroller-Based Terminal Node Controller : <https://tnc-x.com/dcc3.doc>
- [5] George Rossopoulos/SV2AGW : <https://www.sv2agw.com/downloads/>
- [6] Ing. Pedro E. Colla/LU7DID and George Rossopoulos/SV2AGW, AGWPE TCP/IP API Tutorial : <https://www.on7lds.net/42/sites/default/files/AGWPEAPI.HTM>
- [7] Nino Carrillo/KK4HEJ, Improved Layer-2 Protocol : <https://tarpn.net/t/il2p/il2p-specification-draft-v0-5.pdf>