

学生による超小型人工衛星プロジェクトのための衛星地上局ソフトウェアの開発Ⅱ

浅井 文男

Development of Satellite Ground Station Softwares for Student CubeSat Project II

Fumio ASAI

学生による超小型人工衛星プロジェクトではプロジェクトサポーターによるデータの受信と提供が衛星の運用や実験に極めて重要な役割を果たしている。本研究では2006年度より科学研究費補助金の支援を得て、プロジェクトサポーターのために最適設計された衛星地上局ソフトウェアを開発・提供している。2010年度は2009年度までの研究で開発した KissTerm と NetworkDecoder の統合に向けた改良と機能強化を行い、実用性を向上させた。具体的には、KissTerm では受信データの受け渡し方法をファイルから UDP/IP 通信に変更することで、ネットワークに接続されたりリモート PC によるテレメトリデータの準リアルタイム解読を可能にした。NetworkDecoder ではサーバソフトを Kiss Mode TNC に対応させ、通信プロトコルを TCP/IP に加えて UDP/IP も選択できるようにした。これにより NetworkDecoder のサーバソフトと KissTerm のプラグインソフト(デコーダ)を組み合わせ使用することができるようになり、ユーザビリティが向上した。さらに TCP/IP 通信を選択した場合はクライアント接続制限機能も実装し、サーバソフトのセキュリティ改善を図った。今後の課題は実践的な評価である。

1. KissTerm の問題点

2009年度までに開発した KissTerm は2008年度までに開発した KissDecoder と NetworkDecoder の両方の利点を併せ持つソフトウェアである。すなわち、Kiss Mode TNC に対応させることで、衛星データの完全な自動受信・解読・保存を実現し、かつ、解読機能をプラグインソフト(以下、デコーダと表記)に記述することで複数の衛星への対応も実現したり。しかし、以下のような課題が残された。

(1) 衛星データを準リアルタイムに解読できない

KissTerm のメインソフトが受信した衛星データはメインソフトの終了時にファイルに書き出され、メインソフトによって起動されたデコーダが書き出しファイルを読み込んで解読するので、衛星データの解読結果を準リアルタイムで確認することができない。

(2) デコーダの作成が容易でない

新たに開発された衛星が送信する衛星データを解読するためには、解読情報に基づいて作成したデコーダが必要になる。KissTerm で採用されたプラグインソフト形式は、NetworkDecoder に実装したプラグインファイル

方式よりも衛星データの解読情報が容易に記述できる利点がある。しかし、デコーダを作成するには開発環境(Visual Studio 2008 C#)とプロジェクトファイルが必要になり、テキストエディタのみで作成・対応できるプラグインファイル方式と比較すると迅速性や利便性に欠ける。

2. KissTerm の改良

2.1 準リアルタイム解読機能の実装

衛星データの準リアルタイム解読を実現する方法として NetworkDecoder で採用したクライアント・サーバ方式の利用が挙げられる。すなわち、メインソフトとデコーダにそれぞれネットワーク(TCP/IP)通信機能を実装し、メインソフトをクライアント、デコーダをサーバに対応させて、メインソフトが TNC から受信したシリアル(COMポート)データをソケットストリームに変換してデコーダに送れば、デコーダで準リアルタイムに解読結果を確認することができる。しかし、TCP/IP 通信ではクライアントとサーバの接続・切断処理が必要であり、また、メインソフトがクライアントでデコーダがサーバという役割分担は不適切である。

そこで本研究では TCP/IP 通信の代わりに UDP/IP 通信を使用する。UDP/IP 通信では TCP/IP 通信のように 100% の信頼性は保証されないが、衛星データは誤り訂正のための伝送制御を行わない AX.25 プロトコルの UI フレームを使用して送信されるので大きな問題とはならない。

.Net Framework を利用して UDP/IP 通信を実装するには UDP/IP 通信専用の UDPCClient クラスを使用することもできるが、本研究では NetworkDecoder との統合を考慮して、Socket 通信用の Socket クラスを使用する。改良後の KissTerm の動作を表すフローチャートを図 1 に示す。

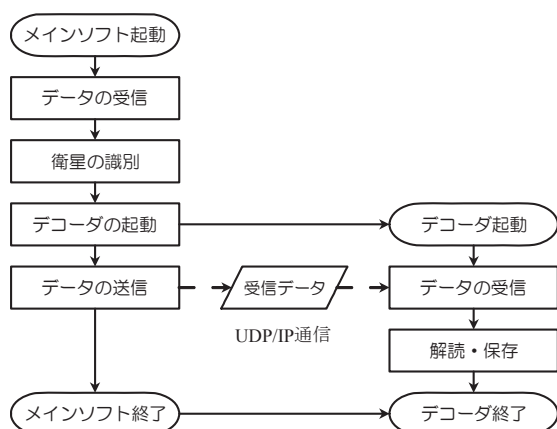


図 1 KissTerm の動作フローチャート

衛星が可視範囲に入るとメインソフトが起動し、衛星データを受信すると、受信データに含まれるアドレスデータから送信元衛星を識別し、対応するデコーダを起動する。受信データは UDP/IP 通信でメインソフトからデコーダに送信され、デコーダはテレメトリデータを解読・表示・保存する。衛星が可視範囲から出るとメインソフトはデコーダを終了させて、自らも終了する。これら一連の動作により、複数の衛星のテレメトリデータと解読結果を自動的にファイル保存・蓄積することができる。

2.2 オンラインアップデート機能の実装

本研究では Windows アプリケーションソフトウェアに実装されるようになったオンラインアップデート機能と同様の機能を KissTerm のメインソフトに実装する。すなわち、浅井研究室で運用する FTP サーバに新しいデコーダを登録すれば、自動的にそのデコーダ情報が KissTerm のユーザーに通知され、ダウンロードすればインストールも行われるという機能を実装する。これによりプラグインソフト方式の欠点が解消され、

KissTerm のユーザーは容易かつ迅速に新規開発衛星のテレメトリデータを解読できるようになる。

オンラインアップデート機能に使用されるプロトコルとして FTP と HTTP があるが、本研究ではアクセス制限を柔軟に設定できる FTP を採用する。.Net Framework を利用して FTP を実装するには WebClient クラスと WebRequest クラスがあるが、ファイル情報の取得もできる WebRequest クラスを採用する。KissTerm のメインソフトに実装したデコーダのオンラインアップデート機能の基本動作を図 2 に示す。

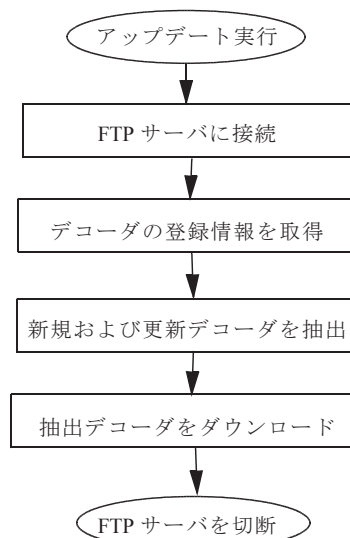


図 2 アップデートの動作フローチャート

オンラインアップデートを実行するとメインソフトは FTP サーバに接続し、サーバに登録されているデコーダ情報を取得し、ローカルディスクにファイル保存する。次に今回取得したデコーダ情報と、前回取得したデコーダ情報を比較し、サーバに新規登録されたデコーダと、タイムスタンプが更新されたデコーダをダウンロードする。ダウンロードが終了するとサーバから切断する。デコーダはレジストリ登録が不要な実行プログラムなのでインストール作業は不要である。

3. NetworkDecoder の問題点

2009 年度までに開発した NetworkDecoder はクライアント・サーバ型のソフトウェアなので、ネットワーク上の複数のコンピュータで同時に準リアルタイムに衛星データが解読でき、クライアントソフトに解読情報を記述したプラグインファイルを追加するだけで新規開発衛星のテレメトリデータも解読できるという特徴をもつ。しかし、以下のげるような課題が残された。

(1) サーバソフトが Kiss Mode に対応していない

KisstTermはTNCをKiss Modeに設定して使用するが、NetworkDecoderはTNCをConverse Mode(Terminal Mode)のままです使用するので互換性がない。バイナリデータの受信や自動的なデータ受信は衛星地上局用ソフトウェアには不可欠の機能なので、NetworkDecoderもKiss Modeに対応させる必要がある。そうすればNetworkDecoderのサーバソフトとKisstTermのデコーダを組み合わせて使用したり、NetworkDecoderとKisstTermを一つのソフトウェアに統合することも可能になる。

(2) サーバソフトにセキュリティ機能がない

NetworkDecoderはネットワークで接続された複数のクライアントPCで準リアルタイムに衛星データを受信することができるようにするため、サーバは任意のクライアントPCからTCP/IP接続できる仕様になっている。不正アクセスやリソースの浪費を防ぐためには許可したクライアントPCからの接続要求のみを受け入れる制限機能を実装する必要がある。しかし、NetworkDecoderのサーバソフトが使用している.Net FrameworkのTCPListenerクラスは接続制限機能をもたない。そこで、Socket通信のSocketクラスを利用する。これにより、SocketPermissionクラスが使用できるようになり、サー

バソフトにクライアント接続制限機能が実装できる。

(3) サーバソフトにUDP/IP通信機能がない

KisstTermとNetworkDecoderを併用したり、統合するためには通信プロトコルを統一しておくことが望ましい。しかし、TCP/IPとUDP/IPのどちらが適切かは衛星地上局ソフトウェアの使用目的や使用環境によって異なる。そこで、ユーザーが必要に応じて通信プロトコルを選択できるように、NetworkDecoderのサーバソフトにUDP/IP通信機能も実装する。Socketクラスを利用すればTCP/IP通信にクライアント接続制限機能を実装するとともに、UDP/IP通信機能も追加実装することは容易である。

4.NetworkDecoderの改良

4.1 Kiss Mode TNC への対応

NetworkDecoderのサーバソフトに実装されているシリアルデータ受信イベント処理関数はCOMポートから受信したシリアルデータをそのままクライアントソフトに送信している。TNCをKiss Modeにした場合、COMポートから受信したシリアルデータは図3に示すKiss変換処理によりKissフレームデータになるので、シリアルデータ受信イベント関数にKissフレームデータ解読処理(図4に示すKiss逆変換処理)を組み込めばよい。

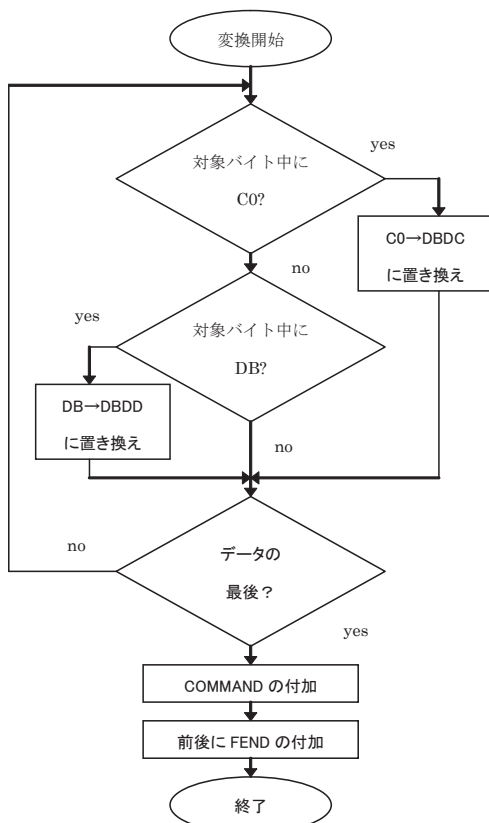


図3 Kiss 変換処理

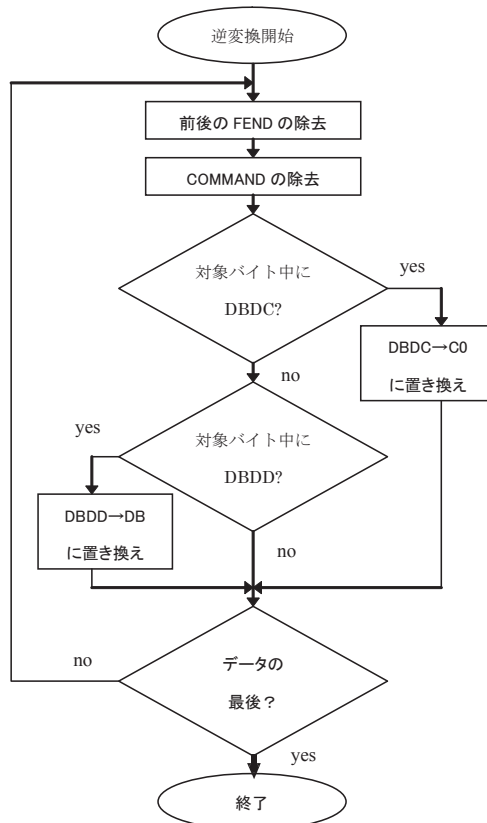


図4 Kiss 逆変換処理

4.2 セキュリティ機能の実装

まず最初に NetworkDecoder のサーバソフトに使用されている TCPListen クラスを Socket クラスに変更した。次に SocketPermission クラスを使用して、IP アドレスに基づくクライアント接続制限機能を実装した。具体的にはサーバに接続要求してくるクライアントの IP アドレスを調べ、図5に示す処理手順で以下の3種類のクライアント接続制限が選択できるようにした。

- (1) ローカルホスト (IP アドレス : 127.0.0.1) のみ接続可
- (2) ローカルホストとアクセスリスト (ホワイトリスト) に IP アドレスが記述されているホストのみ接続可
- (3) すべてのホストが接続可

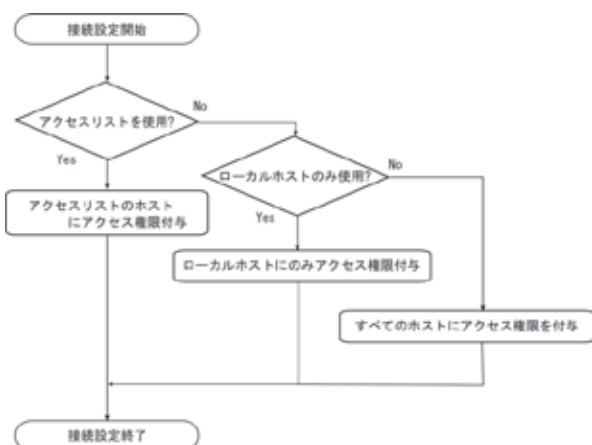


図5 接続制限機能の動作フローチャート

4.3 UDP/IP 通信機能の実装

NetworkDecoder のサーバソフトに使用されている TCPListen クラスを Socket クラスに変更したので、Socket のインスタンスを生成するとき、SocketType を Stream から Dgram へ、また、ProtocolType を Tcp から Udp に変更し、接続要求受け入れ処理と接続確立処理を削除することで、TCP/IP 通信から UDP/IP 通信に切り替えることができる。具体的にはサーバソフトの設定メニューに通信プロトコル選択タブを作成して、図6に示すように通信プロトコルの切り替えられるようにした。

5. 動作検証実験

実装した各機能の動作検証を図7に示す実験装置を構成して行った。衛星データには浅井研究室に設置した地上局設備で受信・録音し、Windows Media ファイルに保存しておいた東京大学 CubeSat/XI-V と日本大学 CubeSat/SEEDS II のパケットデータを使用した。実験の結果、KissTerm のメインソフトとデコーダの連係動作、NetworkDecoder のサーバソフトとデコーダの連係

動作、デコーダアップデート機能、クライアント接続制限機能のいずれについても設計通りに動作することを確認した。

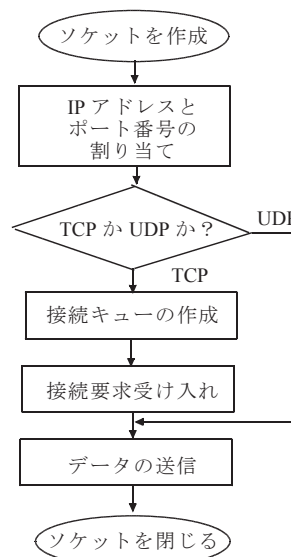


図6 TCP/UDP 通信の動作フローチャート

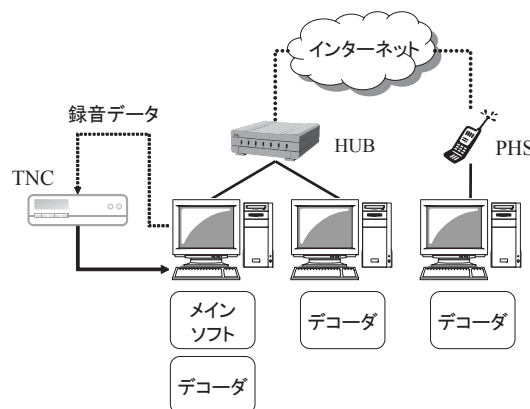


図7 動作検証実験装置の構成

謝辞

本研究は平成18年度科学研究費補助金(課題番号16500570)、平成19-20年度科学研究費補助金(課題番号19500772)および平成21-22年度科学研究費補助金(課題番号21500854)の支援を受けた。また、ソフトウェアの開発には上島佳佑、大上一貴、阪口紘生、山本彩織の各氏の協力を得た。これらの支援と協力に感謝します。

参考文献

1) 浅井文男, 学生による超小型人工衛星プロジェクトのための衛星地上局ソフトウェアの開発, 奈良工業高等専門学校研究紀要, Vol.46, pp.19-24, (2011).