

CanSat / モデルロケット向け  
Windows 用地上局アプリケーション

## 取扱説明書

Flight 9001  
種コン投入仕様一般公開版  
(Build: 2017/04/27)

バグ報告や質問などは  
Twitter: @geo\_vitya あるいは lyncs.keio@gmail.com まで

# 目次

1. Lepus とは
2. 基本的な使い方

## Lepus のバージョン付番法則について

Lepus のシステムは本体に加え、プラグインを含めた複数のプログラムで構成されるパッケージとしてリリースされます。(今回は本体のみのリリースですが。) そのため、パッケージ内の一部のプログラムのみが修正された状態でも更新版がリリースされることがあり、各プログラムのバージョン (Version x.x.x.x) とは別個にパッケージのバージョンも管理されています。

パッケージのバージョンは Flight から始まる 4 桁の連番です。基本的に一番最初の桁が Lepus 本体のバージョンと一致します。(例: Lepus Version1.x 系のとき、Flight1xxx)

例外として、試験的リリースは本体バージョンに依存せず、9000 番代の Flight 番号になります。(今回のリリースがこれに当たります)

# Lepus とは

Lepus (LYNCS Enhanced PC-based Utility for CanSat) は慶應義塾大学宇宙科学総合研究会 (LYNCS) 工学部門ソフト班の @geo\_vitya が制作した地上局ソフトウェアです。2016 年の能代宇宙イベントにて投入した地上局をベースに、信頼性および性能の向上、プラグインによる拡張対応などを行っており、LYNCS としては 2017 年の第 13 回種子島ロケットコンテストにて初めての実戦投入を行いました。

この公開バージョン (Flight 9001) は LYNCS 独自のミッション用プラグインを一旦除外し、Lepus 本体のみをひとまず体験できるものとしてリリースしています。単純にターミナルソフト (TeraTerm など) の代替として使用することができるほか、自力で目的にプラグインを開発していただければミッションにも投入できます。

LYNCS が本番で投入したグラフ生成ツールなどは順次公開する予定ですので、Twitter の @keio\_LYNCS を随時ご覧ください。

## Lepus の特徴

### ●効率的な設計・試験をサポートする設計

CanSat などの製作中にプログラムをテストする際、センサのデータなどシリアル通信内容を記録する必要がある場合でも、トラブルが発生した場合はつつい保存を忘れてしまいがちです。Lepus では標準で通信内容を保存するようにしており、意識的に保存操作をする必要がありません。

また、切断 / 再接続しても画面上に通信ログが残る (消去も可能です) ため、検証で切断操作を繰り返してもデータを容易に比較できます。

さらに、データ表示部分にはテキストエディタエンジンの Azuki を採用しており、行番号が表示されるほか、アップリンクコマンドの履歴が保持され、同一のコマンドを入力し直さずにクリックで送信できます。

### ●無線転送のロスを考慮したデータ整合性チェック

Xbee を利用する場合、距離によってデータ欠損が発生することがあります。試験やログ取得であれば欠損データを無視すればよいのですが、自動的に処理する場合は不正なデータがエラーや処理停止を引き起こすことがあります。Lepus ではそういった欠損データを検知した場合は Data Validator (データバリデータ) 機能によって自動的に削除する事が可能です。現在は簡易的な機能に過ぎませんが、将来的に強化される予定です。

※検出に対応した記法で送信する必要があります。詳細は後述します。

### ●Lepus Pipeline

Lepus は単なるデータ受信 / 送信を行うターミナルに留まるのではなく、データをやり取りするハブとして機能するように設計されています。例えば、シリアル通信で受信したデータを Lepus 経由で他のプログラム (プラグイン) に渡す、ということが容易に可能です。

これを **Lepus Pipeline** (パイプライン) と称しており、**Inbound Pipeline** (インバウンド・パイプライン、Lepus に入ってくるデータ) と **Outbound Pipeline** (アウトバウンド・パイプライン、Lepus から出ていくデータ) の 2 つに分けられます。

### ◆Inbound Pipeline

Lepus にデータを渡す Inbound は Version1.0.0.1 現在でシリアル通信のみの実装となっています。

将来的には、SSH プロトコルによる遠隔データ転送、CW (モールス信号) デコーダによる実衛星からのダウンリンクにも対応できれば、などと考えています。

### ◆Outbound Pipeline

現時点ではこちらの実装がメインになっています。

Outbound では外部プログラム (exe など) にデータを転送します。Lepus ではマルチプロセスを採用しており、Lepus 本体と Outbound は別個のプロセスとして動作します。プラグインを地上局の内部に組み込んでマルチスレッドにするよりメモリ使用量は増加しますが、万が一ミッション部分のプログラムにバグがあったとしても通信部分のプロセスだけは必ず生き続けるという堅牢性を重視する観点、プラグイン開発の制約を緩める (Lepus と同一の C# である必要がない) 観点からの仕様です。

#### ◆実施タイミング

- EveryReception モード …… 1 回の受信ごとにデータを転送します
- Connect モード …… 接続時に起動します。(未実装)
- Disconnect モード …… 切断時に起動します。(未実装)
- None …… そのプロセスを動作させません

#### ◆データ転送手段

- Argument モード …… コマンドライン引数でデータを渡します。  
最も基本的なモードです。
- IPC モード …… Inter Process Communication( プロセス間通信 ) を用いて、  
データを転送します。(未実装)
- File I/O モード …… 一時ファイルを保存して他のプログラムに読ませます。  
大容量のログを送信する場合などに使います。(未実装)

### ●LDV 記法 (Lepus Data Validation Notation)

Lepus のデータ整合性チェック機能に合わせた記法です。データの開始を [ (角括弧 , U+005B) で、データの終了を ] (角括弧閉じ , U+005D) で、データの分割を | (バーティカルバー , U+007C) で示します。実際の 1 回あたりのデータの例は以下のようになります。

**[35.554760|139.655643|53.545825|LDG]**

この記法で受信したデータを分割する .NET Framework 向けライブラリとして DspLib (Data Splitter Library) を用意していますが、各言語や環境の標準文字操作 (置換や分割) でも容易に処理可能だと思います。

# 基本的な使い方

Outbound を利用しないのであれば、TeraTerm など通常のシリアル通信ターミナルとほぼ同じ感覚で扱えます。

## Lepus の導入

Lepus は Windows Vista / 7 / 8 / 8.1 / 10 向けのソフトウェアです。

Lepus の動作には .NET Framework 4.5.2 以上が必要です。Windows Vista / 7 にはプリインストールされていないので、必要に応じて Microsoft の公式サイトから .NET Framework をインストールしてください。

### Microsoft .NET Framework 4.6.2

<https://www.microsoft.com/ja-jp/download/details.aspx?id=53344>

あるいは「.NET Framework ダウンロード」で検索

Windows の仕様上、インターネットからダウンロードされたプログラムは実行前にブロックを解除する必要があります。(下のような表示が出る場合、ブロック解除されていません。)

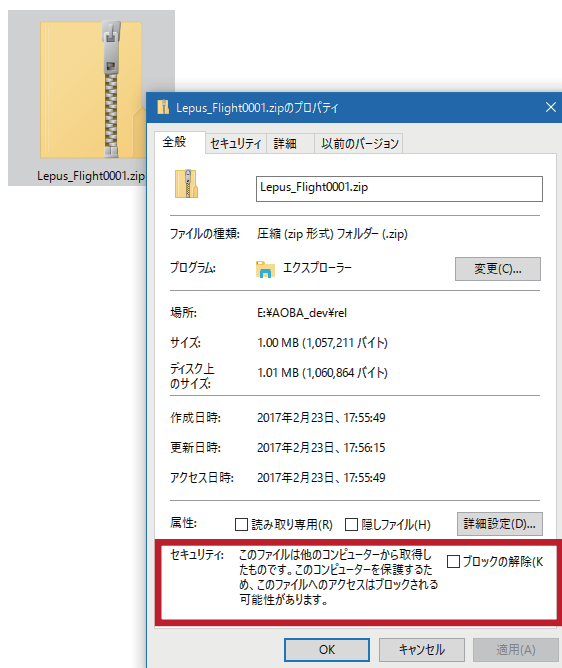
Lepus 本体 (Lepus.exe) のみに限らず、Lepus が参照しているライブラリ (DLL ファイル) も同様にブロック解除が必要となります。ZIP ファイルに対して解除操作を行えば含まれているファイルは一括でブロック解除が行われるため、Lepus の ZIP ファイルを展開する前に操作をする事をおすすめします。(既に展開してしまった場合、当該フォルダを削除して解除した上で再度展開すれば OK です。)

### Windows によって PC が保護されました

Windows SmartScreen は認識されないアプリの起動を停止しました。このアプリを実行すると、PC に問題が起こる可能性があります。

[詳細情報](#)

実行しない



### ブロック解除手順

- ①ファイルを右クリック→プロパティ
- ②「ブロックの解除」にチェック あるいは 「ブロックの解除」 ボタンをクリック
- ③OK をクリック

# Lepus の起動

Lepus の導入が完了したら、**Lepus(.exe)** を起動してください。

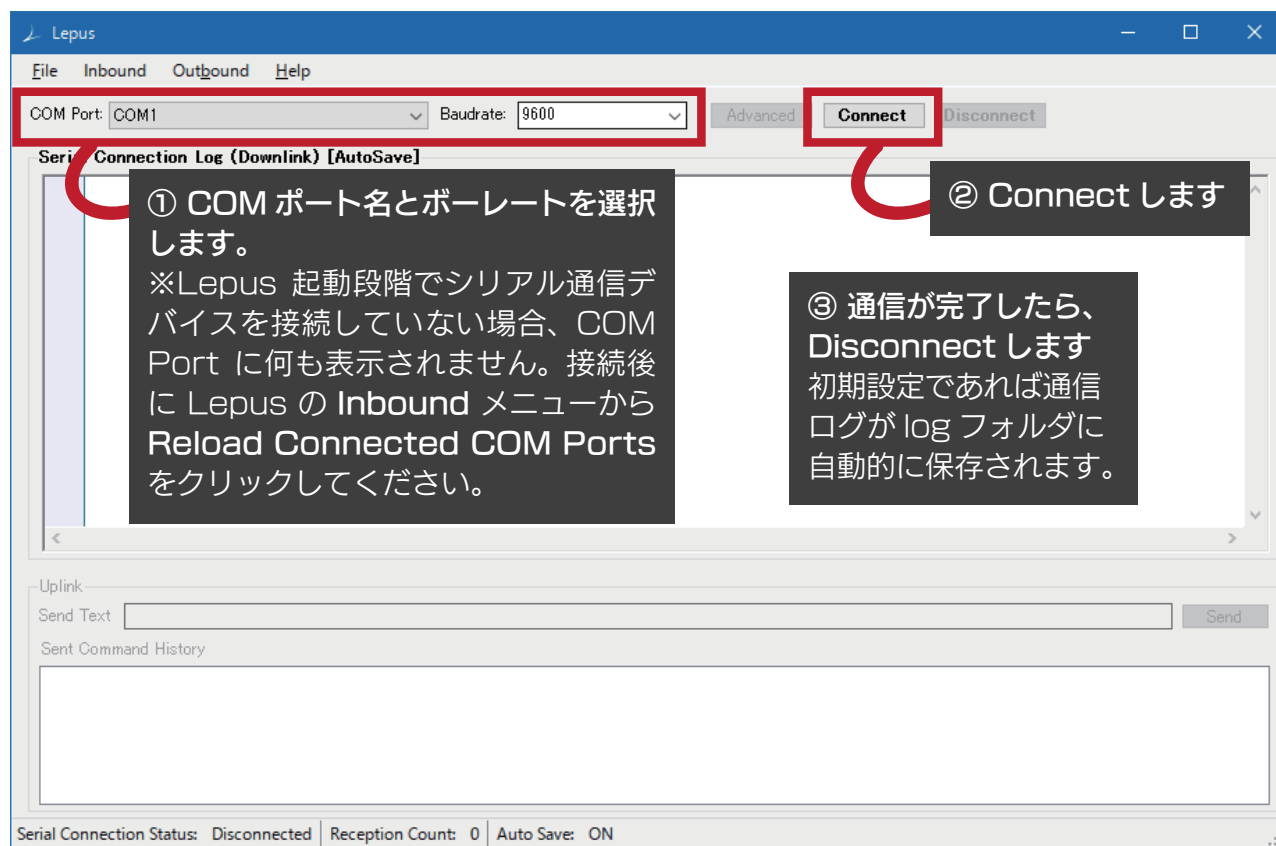
Flight9001 では画面がすべて英語となっていますが、これは仕様です。順次日本語化も行います。

(余談：ARLISS など海外環境において現地 PC で動作させる可能性があることを念頭に、多言語化を行うための設計です。.NET の開発において多言語化を行う場合、ニュートラル言語とよばれる初期設定の言語は英語が望ましいのです。現状、言語パックにあたるものが用意できていないので日本語表示がされない、といった状況です。)

## 起動後にエラーが発生する場合

LYNCS 内の検証においては一部のウィルス対策ソフトに動作が阻害されることが確認されています。(特にトレンドマイクロ社のウィルスバスター)  
必要に応じて設定を変更してください。

## シリアル通信での接続（ダウンリンク）



複数のデバイスを PC に接続している場合、ポート名とデバイス名の対照関係がわかりづらいかもしれません。その場合は Windows のデバイスマネージャからデバイス名とポート名が確認できますのでご活用ください。

# シリアル通信での接続（アップリンク）

---

※接続まではダウンリンクと同じです

現在執筆中 & 機能強化中

## 通信ログの確認

---

Lepus ではダウンリンク記録を自動保存するモードが標準で ON になっています。通信内容は Lepas と同一のフォルダにある「log」フォルダに AutoSavedDownlinkLog\_2017-02-26\_23-07-11.txt という日時を示すファイル名でテキストファイルとして保存されます。自動保存が不要なときは Outbound メニューから Downlink Log Autosave のチェックを外してください。

また、Lepas のシステムログは LepusSystemLog\_2017-02-27\_01\_36\_12.txt というファイル名で log フォルダに保存されます。動作の検証や不具合報告をする場合にご利用ください。

（現状は例外処理が甘いため log フォルダを破棄するとエラーが発生します。お手数をおかけします。なお、log フォルダ内の各種テキストファイルは必要なければ削除しても構いません。）