tsukumijima / libaribb25 Public

forked from HaijinW/libaribb25

Windows・Linux 共用の ARIB STD-B1 / ARIB STD-B25 ライブラリ

github.com/tsukumijima/libaribb25/releases

Apache-2.0 license

47 stars 55 forks Branches Tags Activity

Star Notifications

Code Issues Pull requests Actions Projects Security Insights

1 Branch 3 Tags Go to file Go to file Code

This branch is 183 commits ahead of `HaijinW/libaribb25:master` .

tsukumijima ドキュメントを更新 f4a1d2b · 2 months ago

.github/workflows	GitHub Actions ワークフロ...	4 months ago
aribb25	Add 'Fake' aribb25.pc for VL...	2 months ago
cmake	sudo make uninstall の実行...	last year
.editorconfig	.editorconfig の typo を修正	last year
.gitignore	CMake で b1.exe・libaribb1....	3 years ago
CMakeLists.txt	Add 'Fake' aribb25.pc for VL...	2 months ago
LICENSE	Create LICENSE	7 years ago
NOTICE	コード中の末尾の空白を削除	3 years ago
README.md	ドキュメントを更新	2 months ago
arib_std_b25.sln	Visual Studio で arib-b1-stre...	3 years ago

libaribb25

このフォークについて

散逸している libaribb25 派生のソースコードやパッチを一つのコードベースにまとめる事を目的とした、Windows・Linux 共用の ARIB STD-B1 / ARIB STD-B25 ライブラリです。
epgdatacapbon 版 版と stz2012 版 を統合し、同じコードベースから Windows 向けと Linux 向け両方の libaribb25 をビルドできるようになったほか、スカパー！プレミアムサービス (libaribb1) への対応、arib-b25-stream-test への対応、Windows 向け SIMD 実装の統合も行っています。

細心の注意を払ってコードを統合したほか、libaribb1・libaribb25 ともに動作確認を行っています。
ただし、私が C/C++ が書けず、コードの実装内容を正確に理解できているわけでもないため、100% 動作する保証はありません。自己の責任のもとでお願いします。
何か不備がありましたら Issue や Pull Request までお願いします。可能な範囲で対応します。

変更点

コードの統合

libaribb25 には主に Windows 環境で利用されている epgdatacapbon 版 と、Linux 環境で利用されている stz2012 版 の2つのフォークがあります。
このフォークでは、epgdatacapbon 版にさらに SIMD 対応を実装している HaijinW 版 をベースに、比較的更新が活発な stz2012 版 の変更内容を取り込みました。
加えて epgdatacapbon 氏が別途公開している stz2012 版ベースのフォーク での変更内容も取り込み、現時点で存在する有用な変更内容を大方取り込んだつもりです。

2つのフォークの統合にあたり、stz2012 版由来の `libarib25` という表現を `libaribb25` に統一したほか、epgdatacapbon 版由来の Linux 向け Makefile を削除し、Linux でのビルド方法を CMake に統一しています。
Windows 向けライブラリのビルドは Visual Studio で、Linux 向けライブラリのビルドは CMake で行えます。

> **Tip**
>
> 0.2.9 以降では、stz2012/recpt1 など `libarib25` というライブラリ名に依存しているソフトウェアとの互換性を保つため、インストール時に `libaribb25` から `libarib25` へのシンボリックリンクを作成するようになりました！
> 今までは `find . -type f | xargs sed -i "s/arib25/aribb25/g"  find . -type f | xargs sed -i "s/ARIB25/ARIBB25/g"` のようにソースコードを置換してからビルドしていましたが、この変更により置換作業が不要になります。

スカパー！プレミアムサービス対応の統合

さらに、パッチとしていくつか実装がある ARIB STD-B1 対応のためのコードを統合しました。

ARIB STD-B1 は、スカパー！プレミアムサービスの CAS システム（スカパーカード）の仕様が定められている標準規格です。
とはいえ、実際に ARIB STD-B1 で規定されている内容は僅かで、多くはスカパーによる非公開仕様となっています。

- libaribb25.patch (https://www.axfc.net/u/3985543)
- arib-b1-stream-test (https://www.npmjs.com/package/arib-b1-stream-test)
- B1_p2c9 (スカパーTS抜きツール詰め合わせ (https://www44.zippyshare.com/v/yWcPHjsD/file.html) に同梱)
- B25Decoder.dll to B1 diff (https://pastebin.pl/view/478245cc)

のコードを参考に、現行のコードに手作業で統合しました。
多くが非公開仕様となっている関係で、現状 EMM 処理と通電制御情報の取得は未サポートとなっています。
そのため、ARIB STD-B25 における b25.exe に相当する b1.exe のコマンドラインオプションからも、EMM の送信を行う `-m` オプションと、通信制御情報を表示する `-p` オプションを削除しています。

便宜上 libaribb1 という別ライブラリとしていますが、コードベースは同一です。
ENABLE_ARIB_STD_B1 プリプロセッサが定義された状態でビルドすると、libaribb25 ではなく libaribb1 が生成されます。
スカパー！プレミアムサービス対応の統合にともない、Windows (Visual Studio)・Linux (CMake) の両方で libaribb25 (b25.exe / libaribb25.dll) と同時に libaribb1 (b1.exe / libaribb1.dll) が生成できるよう、Visual Studio のプロジェクトファイルと CMakeList.txt を変更しています。

arib-b1-stream-test・arib-b25-stream-test の統合

arib-b25-stream-test は、標準入力から MULTI2 で暗号化された TS を受け、B-CAS カードと通信し復号した TS を標準出力に出力するプログラムです。
b25（実行ファイル）に渡す `src.m2t dst.m2t` の各引数をそれぞれ標準入出力に置き換えたものと言えば分かりやすいでしょうか。
arib-b1-stream-test は、その名の通り arib-b25-stream-test の ARIB STD-B1 対応版となっています。

Mirakurun で受信した放送波のスクランブルを解除するための decoder として多く利用されていますが、実は 5 年以上前のかなり古い stz2012 版の libarib25 がベースとなっています。
そのため、最新の stz2012 版などでは修正されている不具合が未修正のままになっているなど、いくつかの問題を抱えています。

> **Note**

> もっともほとんどのケースで問題なく動作するため、影響はさほど大きくないと思われます。

arib-b25-stream-test_for_win のコードを参考に、Linux 対応を追加して現行のコードに手作業で統合しました。
ENABLE_ARIB_STREAM_TEST プリプロセッサが定義された状態でビルドすると、b1・b25 ではなく arib-b1-stream-test・arib-b25-stream-test が生成されます。
arib-b1-stream-test・arib-b25-stream-test の統合にともない、Windows (Visual Studio)・Linux (CMake) の両方で libaribb1・libaribb25 と同時に arib-b1-stream-test・arib-b25-stream-test が生成できるよう、Visual Studio のプロジェクトファイルと CMakeList.txt を変更しています。

B25Decoder 互換インターフェイスの対応

epgdatacapbon 版由来の、libaribb25 における B25Decoder 互換インターフェイスの実装を引き継いでいます。
libaribb1.dll / libaribb25.dll をそれぞれ B1Decoder.dll / B25Decoder.dll とリネームして EDCB などの各ソフトに配置することで、オリジナルの B1Decoder.dll / B25Decoder.dll を代替することができます。

B25Decoder 互換のインターフェイスがビルドされるのは Windows (Visual Studio) でビルドした場合のみです。
Linux (CMake) でビルドした場合はビルド対象になりません。そもそも Windows API に依存しているため、Linux ではビルドに失敗すると思われます。

B25Decoder 互換のインターフェイスが実装されているのは libaribb25.cpp / libaribb25.h です。

README Apache-2.0 license

> **Note**
>
> そもそも関数名が UpperCamelCase から snake_case に変更されていることからして、本来の B25Decoder インターフェイスとの互換性はありません。
> おそらく Linux で B25Decoder のようなインターフェイスを実装した libaribb25 のラッパーとしてのコードだと思われますが、このコードが含まれていた epgdatacapbon 版でも現在は使われておらず、どのような意図で利用されていたコードなのかは不明です。本来削除しても問題はないのですが、念のため残しています。

ini ファイルでカードリーダー名を指定してスクランブル解除を行う機能の追加 (Windows のみ)

Windows 版のみ、独自に B25Decoder.dll や arib-b25-stream-test.exe などの各種 .exe / .dll と同じ名前の ini ファイルでカードリーダー名を指定してスクランブル解除を行う機能を追加しました。
主に EDCB で B-CAS カードと C-CAS カードの両方を併用したい場合に利用できると思います。

ini ファイルを配置しない場合の動作は、通常の B25(B1)Decoder.dll / b25(b1).exe / arib-b25(b1)-stream-test.exe と同じです。
具体的には、一番先に検出されたカードリーダーがスクランブル解除に利用されます。

下記の例を参考に (.exe or .dll と同じファイル名).ini 内の `Name=` プロパティにカードリーダー名を記述すると、指定されたカードリーダーに接続されている B-CAS or C-CAS or SPHD カードを使ってスクランブル解除を行います。
カードリーダー名は TVTest の TS プロセッサーで表示されているものがそのまま使えるので、コピペするのがおすすめです。

注意

- ini ファイルで指定されたカードリーダーが PC に接続されていない場合、スクランブル解除に失敗します (スクランブルされたままの TS が返される) 。
  - 無理に復号しようとしてファイルが壊れたりとかはないので、後から解除すること自体は可能なはずです。
- B25Decoder.dll と同じフォルダに winscard.dll を配置している場合は、その仮想 winscard.dll の実装で提供されている仮想カードリーダーの名前しか指定できなくなります。注意してください。
  - たとえば radi-sh 版 BonDriver の内蔵カードリーダーなら `Plex PX-x3U4 Card Reader 0` 、SoftCas なら `@OishiiSlurper` になります。
  - Windows ネイティブの winscard.dll の機能を上書きする形になるため、仮想 winscard.dll が配置されている状態では、別途物理カードリーダーが接続されていたとしても libaribb25 から認識されなくなります。
  - このため、仮想 winscard.dll を利用している環境でカードリーダー名指定機能を使う意味はありません。

利用例

- B-CAS: TVTest で `SCM Microsystems Inc. SCR33x USB Smart Card Reader 0` として認識されるカードリーダーに接続中
- C-CAS: TVTest で `SCM Microsystems Inc. SCR33x USB Smart Card Reader 1` として認識されるカードリーダーに接続中

の環境だと仮定します。

まず、EDCB (EpgDataCap_Bon.exe) と同じフォルダに B25Decoder.dll と、それをコピーした B25Decoder-CATV.dll を配置します。

```
[CardReader]
Name=SCM Microsystems Inc. SCR33x USB Smart Card Reader 0
```

次に新規作成した B25Decoder.ini に上記の内容を、

```
[CardReader]
Name=SCM Microsystems Inc. SCR33x USB Smart Card Reader 1
```

同じく新規作成した B25Decoder-CATV.ini に上記の内容を書き込んで保存します。

```
...
[SET]
FFFFFFFF=B25Decoder.dll
0004FFFF=B25Decoder.dll
0006FFFF=B25Decoder.dll
0007FFFF=B25Decoder.dll
000AFFFF=B1Decoder.dll
0001FFFF=B1Decoder.dll
0003FFFF=B1Decoder.dll
FFFEFFFF=B25Decoder-CATV.dll
FFFAFFFF=B25Decoder-CATV.dll
FFFDFFFF=B25Decoder-CATV.dll
FFF9FFFF=B25Decoder-CATV.dll
...
```

最後に、EDCB の BonCtrl.ini の [SET] 以下の記述を、上記の通りに変更して保存します。
これで B-CAS カードと C-CAS カードを同じ PC に接続した状態で、地上波と CATV (C-CAS が必要なトランスモジュレーション方式 (ISDB-C) チャンネル) を両方受信できるようになるはずです！

> **Note**
>
> BonCtrl.ini に追記した行の先頭の `FFFE`, `FFFA`, `FFFD`, `FFF9` はいずれも CATV チャンネルの network_id を示しています。
> その次の FFFF は、同じ network_id のすべてのチャンネルに対して同じデコーダーを利用することを示しています。

SIMD 実装の差異

基本的に Windows と Linux でコードベースは共通ですが、MULTI2 の復号周りのコードに関しては両者ともに大きく変更されていたため、マージは行いませんでした。
multi2.c が HaijinW 版由来のオリジナルに近い MULTI2 復号コードで、multi2.cc が stz2012 版由来の C++ で書き直された MULTI2 復号コードです。
Windows (Visual Studio) でビルドする場合は HaijinW 版由来の SIMD 実装 (ENABLE_MULTI2_SIMD) が、Linux (CMake) でビルドする場合は stz2012 版由来の SIMD 実装が利用されるように調整しています。

Windows 向け（= HaijinW 版の SIMD 実装）の b1.exe / b25.exe では、SSE2,SSE3,AVX2 のどの SIMD 拡張命令を利用するかを選択する `-i` オプションと、MULTI2 の復号のベンチマークテストを行う `-b` オプションが実装されています。
Linux 向け（= stz2012 版の SIMD 実装）の b1 / b25 では、オプションの追加はありませんが、SIMD が使用可能であれば自動的に利用するコードになっていると思われます。
なお、AVX2 を有効化するには、後述の CMake でのビルドの際に `cmake` に `-DUSE_AVX2=ON` オプションを付与する必要があるようです。

バイナリの構成

- **b1.exe / b1**
  - ARIB STD-B1 記載の処理を行うためのプログラム
  - MULTI2 で暗号化された TS を、スカパーカードと通信し復号して出力する
- **arib-b1-stream-test.exe / arib-b1-stream-test**
  - 上記のプログラムに変更を加え、標準入力から MULTI2 で暗号化された TS を受け、復号した TS を標準出力に出力するプログラム
- **libaribb1.dll / libaribb1.so**
  - MULTI2 復号処理を行うライブラリ
  - libaribb1.dll は B1Decoder.dll と互換性がある
- **b25.exe / b25**
  - ARIB STD-B25 記載の処理を行うためのプログラム
  - MULTI2 で暗号化された TS を、B-CAS カードと通信し復号して出力する
- **arib-b25-stream-test.exe / arib-b25-stream-test**
  - 上記のプログラムに変更を加え、標準入力から MULTI2 で暗号化された TS を受け、復号した TS を標準出力に出力するプログラム
- **libaribb25.dll / libaribb25.so**
  - MULTI2 復号処理を行うライブラリ
  - libaribb25.dll は B25Decoder.dll と互換性がある

ビルド方法

Releases ページにビルド済みアーカイブ (Windows: ZIP アーカイブ / Linux: Debian パッケージ) を用意しています。
ご自身でビルドを行う方は、下記の手順に従ってください。

Windows (Visual Studio)

Visual Studio 2019 で arib_std_b25.sln を開きます。

上部メニューの [Debug] を [Release] に変更し、お使いのアーキテクチャに合わせて [Win32] または [x64] のいずれかを選択します。
[ビルド] → [ソリューションのビルド] をクリックし、ビルドを実行します。

ビルドが完了すると、 `Win32/Release` または `x64/Release` 以下にバイナリが生成されています。

Ubuntu (CMake)

```
sudo apt install cmake libpcsclite1 libpcsclite-dev pkg-config
```

あらかじめ、 `cmake` ・ `libpcsclite1` ・ `libpcsclite-dev` ・ `pkg-config` のインストールが必要です。

```
cmake -B build
cd build
make
sudo make install
```

`cmake -B build` で `build/` ディレクトリに Makefile を生成してから、 `make` でビルドを実行します。

`sudo make install` でビルドした libaribb1 / libaribb25 をインストールします。
`build/` ディレクトリで `sudo make uninstall` を実行することで、インストールしたファイルをアンインストールすることができます。

このフォークに取り込んだ HaijinW 版 での変更内容は下記の通りです (原文ママ: 出典) 。

```
勉強用に復号処理を SIMD 拡張命令で実装。
既存のコードや資料などを参考に、SSE2、SSSE3、AVX2 に対応した。
初期化時には、AVX2、SSSE3、SSE2、拡張命令なしの順で利用可能なものを選択する。

ラウンド関数のあと、最後の XOR 演算はもっとよい方法があればよかったが、思いつか
なかった。
Windows環境 (x86-64) でのみ動作確認。開発環境は Visual Studio 2017 Community
(15.9.7) 。
あくまで勉強用なので、安定的な動作の保証はない。
```

以下のドキュメントは、原作者のまるも氏が書かれた元の readme.txt を、記述をそのままに Markdown 形式に書き直したものです。

ARIB STD-B25 仕様確認テストプログラムソースコード

バージョン

0.2.5

作者

茂木 和洋 (MOGI, Kazuhiro)
kazhiro@marumo.ne.jp

一次配布元

http://www.marumo.ne.jp/db2012_2.htm#13 又は

あるいは

http://www.marumo.ne.jp/junk/arib_std_b25-0.2.5.lzh

目的

ARIB STD-B25 の仕様を理解する為の、参考用の実装として公開

背景

2011 年 7 月の地上アナログ放送停波を控え、廉価な地上デジタル放送受信機の販売が待たれている

しかし、ARIB の標準文書はわざと判りにくく書いて開発費をかさませようとしているとしか思えないほどに意味不明瞭な記述になっており、このままでは低価格受信機の開発など不可能に思える

そこで、自分なりに ARIB 標準文書を読み、理解した範囲をソースコードの形にまとめて公開することにした

このコードが安価な受信機の開発の一助となることを期待する

なお、あくまでも仕様理解を目的としたものであるため、ビルド済みバイナリファイルは配布しない

実装した範囲

CA システム (B-CAS カード関連) を中心に ECM(table_id=0x82) の処理とストリーム暗号の復号処理、EMM(table_id=0x84) の処理までを実装した

EMM メッセージ (table_id=0x85) 関連は未実装となっている

プログラムの動作環境

ISO 7816 対応の IC カードリーダがインストールされた Windows PC を想定動作環境とする

ISO 7816 対応スマートカードリーダーは一般に「住基カード対応 IC カードリーダ」「e-Tax 対応 IC カードリーダ」などとして 4000 円程度で販売されているものが利用可能である

日立マクセル製の HX-520UJJ と NTT コミュニケーションズの SCR3310 で正常に動作することを確認している

ソースコードのライセンスについて

- ソースコードを利用したことによって、特許上のトラブルが発生しても茂木 和洋は責任を負わない
- ソースコードを利用したことによって、プログラムに問題が発生しても茂木 和洋は責任を負わない

上記 2 条件に同意して作成された二次的著作物に対して、茂木 和洋は原著作者に与えられる諸権利を行使しない

プログラムの構成

- arib_std_b25.h/c
  - ARIB STD-B25 記載の処理を行うためのモジュール
  - MPEG-2 TS の分離、CA システム (B-CAS カード) 機能の呼び出し、MULTI2 復号機能の呼び出し等を担当する
- ts_section_parser.h/c
  - MPEG-2 TS のセクション形式データの分割処理を担当する
- b_cas_card.h/c
  - CA システム (B-CAS カード) のリソース管理および直接の制御を担当する
- multi2.h/c
  - MULTI2 暗号の符号化と復号を担当する
- td.c
  - テストドライバ
  - PAT/PMT/ECM を含む MPEG-2 TS ファイルを読み込み、復号後のMPEG-2 TS ファイルを出力する
  - コマンドラインオプションで MULTI2 暗号のラウンド数を指定可能
    - ラウンド数を指定しない場合の初期値は 4
    - このラウンド数 4 は MULTI2 用語では 32 に相当する
    - ARIB STD-B25 では MULTI2 のラウンド数は非公開パラメータだが総当たりで実際のラウンド数は推定可能である

処理の流れ

起動時

1. アプリケーションは B_CAS_CARD モジュールのインスタンスを作成し、B_CAS_CARD モジュールに、初期化を依頼する
    i. B_CAS_CARD モジュールは WIN32 API のスマートカード関連 API を呼び出し、CA システムに接続する
    ii. B_CAS_CARD モジュールは ARIB STD-B25 記載の「初期条件設定コマンドを CA システムに発行し、システム鍵 (64 byte) 初期 CBC 状態 (8 byte) を受け取る
2. アプリケーションは ARIB_STD_B25 モジュールのインスタンスを作成し、B_CAS_CARD モジュールを ARIB_STD_B25 モジュールに登録する

データ処理時

1. アプリケーションは ARIB_STD_B25 モジュールに順次データを提供し、ARIB_STD_B25 モジュールから処理完了データを受け取ってファイルに出力していく

ARIB_STD_B25 モジュール内

1. TS パケットのユニットサイズ (188/192/204 などが一般的) が特定されていない場合 8K まで入力データをバッファしてから、ユニットサイズを特定する
    - ユニットサイズが特定できなかった場合は、エラー終了する
2. PAT が発見されていない場合、PAT が発見できるまで入力データをバッファし続ける
    - PAT が発見できずにバッファサイズが 16M を超過した場合エラー終了する
    - PAT が発見できた場合、プログラム配列を作成し PID マップ配列に登録する
3. PAT に登録されていた PMT すべてが発見されるか、どれかひとつの PMT で 2 個目のセクションが到着するまで入力データをバッファし続ける
    - 上記条件を満たさずにバッファサイズが 32M を超過した場合エラー終了する
    - PMT が到着する毎に ECM の有無を確認し、ECM が存在する場合はデクリプタを作成してプログラムに所属するストリームと PID マップ上で関連付ける
4. PMT に登録されていた ECM すべてが発見されるか、どれかひとつの ECM で 2 個目のセクションが到着するまで入力データをバッファし続ける
    - 上記条件を満たさずにバッファサイズが 32M を超過した場合エラー終了する
    - 各 ECM に対して、最初のセクションデータが到着した時点で MULTI2 モジュールのインスタンスをデクリプタ上に作成する
    - ECM セクションデータは B_CAS_CARD モジュールに提供してスクランブル鍵を受け取り、MULTI2 モジュールにシステム鍵、初期 CBC 状態、スクランブル鍵を渡し、MULTI2 復号の準備を行う
5. i. 暗号化されている TS パケットであれば、PID から対応ECM ストリームを特定し、デクリプタの MULTI2 モジュールに復号させて出力バッファに積む
    ii. 暗号化されていない TS パケットであれば、そのまま出力バッファに積む
    iii. CAT を検出した場合、EMM の PID を取得して EMM の処理準備を行う
    iv. EMM を受け取った場合、B-CAS カード ID と比較し、自分宛ての EMM であれば B-CAS カードに引き渡して処理させる

EMM 処理オプションが指定されている場合

6. ECM が更新された場合、B_CAS_CARD モジュールに処理を依頼し、出力されたスクランブル鍵を MULTI2 モジュールに登録する
7. PMT が更新された場合、ECM PID が変化していれば新たにデクリプタを作成して 4 に戻る
8. PAT が更新された場合、プログラム配列を破棄して 3 に戻る

終了時

1. 各モジュールが確保したリソースを解放する

更新履歴

- 2012, 2/13 - ver. 0.2.5
  - WOWOW でノンスクランブル <-> スクランブル切り替え後に復号が行われないことがあるバグを修正
  - http://www.marumo.ne.jp/db2012_2.htm#13 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.2.5.lzh
- 2009, 4/19 - ver. 0.2.4
  - 終端パケットが野良パケット (PMT に記載されていない PID のパケット) だった場合に、ECM が 1 つだけでも復号が行われないバグを修正
  - transport_error_indicator が立っている場合はパケット処理を行わず、そのまま素通しするように変更
  - http://www.marumo.ne.jp/db2009_4.htm#19 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.2.4.lzh
- 2008, 12/30 - ver. 0.2.3
  - CA_descriptor の解釈を行う際に CA_system_id が B-CAS カードから取得したものと一致するか確認を行うように変更
  - http://www.marumo.ne.jp/db2008_c.htm#30 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.2.3.lzh
- 2008, 11/10 - ver. 0.2.2
  - 修正ユリウス日から年月日への変換処理をより正確なものへ変更
  - TS パケットサイズの特定方法を変更
  - http://www.marumo.ne.jp/db2008_b.htm#10 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.2.2.lzh
- 2008, 4/9 - ver. 0.2.1
  - PAT 更新時に復号漏れが発生していたバグを修正 (ver. 0.2.0 でのエンバグ)
  - 野良 PID (PMT に記載されていないストリーム) が存在した場合 TS 内の ECM がひとつだけならば、その ECM で復号する形に変更
  - EMM の B-CAS カードへの送信をオプションで選択可能に変更 (-m)
  - 進捗状況の表示をオプションで選択可能に変更 (-v)
  - 通電制御情報 (EMM受信用) を表示するオプションを追加 (-p)

    - http://www.marumo.ne.jp/db2008_4.htm#9 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.2.1.lzh
- 2008, 4/6 - ver. 0.2.0
    - EMM 対応
    - 利用中の B-CAS カード ID 向けの EMM を検出した場合、EMM を B-CAS カードに渡す処理を追加
    - ECM 処理の際に未契約応答が返された場合、処理負荷軽減の為、以降、その PID の ECM を B-CAS カードで処理しないように変更
    - EMM を処理した場合は再び ECM を処理するように戻す
    - 進捗を nn.nn% の書式で標準エラー出力に表示するように変更
    - http://www.marumo.ne.jp/db2008_4.htm#6 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.2.0.lzh
- 2008, 3/31 - ver. 0.1.9
    - MULTI2 モジュールのインスタンスが未作製の状況で、MULTI2 の機能を呼び出して例外を発生させることがあったバグを修正
    - # パッチを提供してくれた方に感謝
    - http://www.marumo.ne.jp/db2008_3.htm#31 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.1.9.lzh
- 2008, 3/24 - ver. 0.1.8
    - -s オプション (NULL パケットの削除) を追加
    - -s 1 で NULL パケットを出力ファイルには保存しなくなる
    - デフォルトは -s 0 の NULL パケット保持
    - http://www.marumo.ne.jp/db2008_3.htm#24 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.1.8.lzh
- 2008, 3/17 - ver. 0.1.7
    - arib_std_b25.h に「extern "C" {」を閉じるコードがなかった問題 (C++ コードから利用する場合にコンパイルエラーを発生させる) を修正
    - TS パケットの中途でストリームが切り替わるケースで問題が発生しにくくなるように、arib_std_b25.c 内のコードを修正
    - http://www.marumo.ne.jp/db2008_3.htm#17 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.1.7.lzh
- 2008, 3/16 - ver. 0.1.6
    - PMT 更新の際、ECM 関連の状況が変更 (スクランブル - ノンスクランブルの切り替えや、ECM PID の変更等) が行われても、それが反映されていなかった問題を修正
    - http://www.marumo.ne.jp/db2008_3.htm#16 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.1.6.lzh
- 2008, 2/14
    - readme.txt (このファイル) を修正
    - ソースコードのライセンスについての記述を追加
- 2008, 2/12 - ver. 0.1.5

    - PMT の更新に伴い、どのプログラムにも所属しなくなった PID (ストリーム) でパケットが送信され続けた場合、そのパケットの復号ができなくなっていた問題を修正
    - http://www.marumo.ne.jp/db2008_2.htm#12 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.1.5.lzh
- 2008, 2/2 - ver. 0.1.4
    - ver. 0.1.3 での PMT 処理方法変更に問題があり、PMT が更新された場合、それ以降で正常な処理が行えなくなっていたバグを修正
    - B-CAS カードとの通信でエラーが発生した場合のリトライ処理が機能していなかったバグを修正
    - http://www.marumo.ne.jp/db2008_2.htm#2 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.1.4.lzh
- 2008, 2/1 - ver. 0.1.3
    - 有料放送等で未契約状態の B-CAS カードを使った際に、鍵が取得できていないにもかかわらず、間違った鍵で復号をしていた問題に対処
    - 鍵が取得できなかった ECM に関連付けられたストリームでは復号を行わず、スクランブルフラグを残したまま入力を素通しする形に変更
    - 鍵が取得できない ECM が存在する場合、終了時にチャネル番号と B-CAS カードから取得できたエラー番号を警告メッセージとして表示する形に変更
    - 暗号化されていないプログラムで例外を発生させていたバグを修正
    - http://www.marumo.ne.jp/db2008_2.htm#1 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.1.3.lzh
- 2008, 1/11 - ver. 0.1.2
    - デジタル BS 放送等で、PAT に登録されているのに、ストリーム内で PMT が一切出現しないことがある場合に対応
    - PMT 内の記述子領域 2 に CA_descriptor が存在する場合に対応するため arib_std_b25.c 内部での処理構造を変更
    - 別プログラムと同時実行するためにスマートカードの排他制御指定を変更
    - http://www.marumo.ne.jp/db2008_1.htm#11 又は http://www.marumo.ne.jp/junk/arib_std_b25-0.1.2.lzh
- 2008, 1/7 - ver. 0.1.1
    - セクション (PAT/PMT/ECM 等) が複数の TS パケットに分割されている場合に、

Releases 3

v0.2.9 Latest
on Aug 26, 2023

+ 2 releases

Packages

No packages published

Languages