

組込み RTOS 向けアプリケーション開発支援ツール
TLV (トレース ログ ヴィジュアライザー)
フェーズ 4
アプリログ拡張外部仕様書

2009 年 6 月 2 日

改訂履歴

版番	日付	更新内容	更新者
1.0	09/6/2	新規作成	柳澤大祐

目次

1	はじめに	3
1.1	本書の目的	3
1.2	本書の適用範囲	3
1.3	用語の定義/略語の説明	3
1.4	概要	3
2	概要説明	4
2.1	アプリログ拡張を実施する理由	4
2.2	アプリログ拡張で変更を実施する対象	4
3	詳細説明	4
3.1	文字列可視化	4
3.2	ユーザ定義状態可視化	6
3.3	ユーザ定義状態可視化（端点付き）	6

1 はじめに

1.1 本書の目的

本書の目的は、文部科学省先導的 IT スペシャリスト育成推進プログラム「OJL による最先端技術適応能力を持つ IT 人材育成拠点の形成」プロジェクトにおける、OJL 科目ソフトウェア工学実践研究の研究テーマである「組込み RTOS 向けアプリケーション開発支援ツールの開発」に対して、その開発するソフトウェアに対する要求を記述することである。

本書は特に、フェーズ 4 におけるアプリログ拡張の外部仕様に関する記述を行う。

1.2 本書の適用範囲

本書は、組込み MPRTOS 向けアプリケーション開発支援ツールの開発プロジェクト（以下本プロジェクト）のフェーズ 4 におけるアプリログ拡張の外部仕様に関する記述を行う。

1.3 用語の定義/略語の説明

表 1 用語定義

用語・略語	定義・説明
TLV	Trace Log Visualizer
MPRTOS	マルチプロセッサ対応リアルタイムオペレーティングシステム
トレースログファイル	RTOS のトレースログ機能を用いて出力したトレースログや、シミュレータなどが出力するトレースログをファイルにしたもの
標準形式トレースログファイル	本ソフトウェアが扱うことの出来る形式をもつトレースログファイル。各種トレースログファイルは、この共通形式トレースログファイルに変換することにより本ソフトウェアで扱うことが出来るようになる。
変換ルール	トレースログファイルを標準形式トレースログファイルに変換する際に用いられるルール。
可視化ルール	標準形式トレースログファイルを可視化する際に用いられるルール。
TLV ファイル	本ソフトウェアが中間形式として用いるファイル。前述の標準形式トレースログファイルは、この TLV ファイルの一部である。

1.4 概要

本書では、組込み MPRTOS 向けアプリケーション開発支援ツールのソフトウェアの仕様を記述する。本書は特に、フェーズ 4 におけるアプリログ拡張の外部仕様に関する記述を行う。

2 概要説明

2.1 アプリログ拡張を実施する理由

フェーズ 4 での要求として、ユーザアプリケーションにおいて出力された syslog を可視化したいというものがある。アプリケーションのデバッグには、一般的に printf デバッグと呼ばれるような手法が有効である。これを uITRON 仕様で実現するものが syslog である。これを TLV で表示させることができるようにすることが本拡張の目的である。

2.2 アプリログ拡張で変更を実施する対象

拡張対象は、変換ルール、可視化ルール、リソースヘッダ、リソースファイル（ユーザによる拡張）である。拡張対象に TLV 本体が含まれていない理由は、本格的な実装の前段階として、短納期でプロトタイプを実装するためである。

3 詳細説明

本拡張では、以下の 3 種類の可視化方法の入力、出力をそれぞれ定義する。

- 文字列可視化
- ユーザ定義状態可視化
- ユーザ定義状態可視化（端点付き）

3.1 文字列可視化

文字列の可視化には、タスクに関連するものとそうでないものがある。タスクに関連する文字列出力は、TLV のタスク可視化行にぶら下げて可視化することで現在のタスク状態表示などの上に重ねて表示することとする。タスクに関連しない文字列出力は、それ専用の行にて表示することとする。

3.1.1 入力

斜体で表示されている箇所は、アプリケーション開発者が望む値に置換する場所であることを示している。

ログ書式 TLV へ入力するログの書式を定義する。

タスクに関連しない文字列出力のログ書式

[*time*] applog *str*.

time 時刻 [0-9a-zA-Z]+

str 出力文字列 [^\.]*

—— タスクに関連する文字列出力のログ書式 ——

[*time*] task *tid* applog *str*.

time 時刻 [0-9a-zA-Z]+

tid タスク ID [0-9]+

str 出力文字列 [^\.\.]*

リソースファイル書式 以下の記述は、タスクとは関係ない文字列出力を行う際に、ユーザがリソースファイルに追加するものである。

—— タスクに関連しない文字列出力リソースファイル書式 ——

```
"rowname":{
  "Type":"ApplogStr",
  "Attributes":{
    "str":""
  }
}
```

rowname 文字列可視化行の名前 [^"]+

API また、アプリケーション開発者の利便のために、先ほど示した形式でログを出力する C 言語の関数を用意する。

—— タスクに関連しない文字列出力関数 ——

int applogstr(const char* str)

str 可視化したい文字列——



—— タスクに関連する文字列出力関数 ——

int applogstrtsk(const char* str, const int tid)

str 可視化したい文字列——

tid タスク ID

3.1.2 出力

以下では、文字列の可視化方法を定義する。

タスクに関連しない文字列可視化 まず、タスクに関連しない文字列出力の可視化を示す。

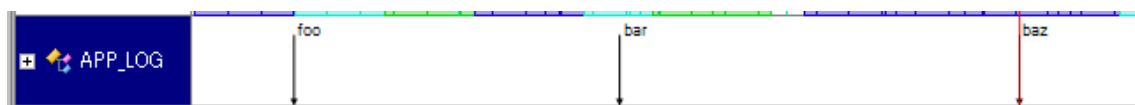


図 1 タスクに関連しない文字列の可視化

[9379610] applog foo.

このようなログが出力された場合、図 1 のように可視化を行う。

タスクに関連する文字列可視化 次に、タスクに関連する文字列の可視化を示す。



図 2 タスクに関連する文字列の可視化



[10379610] task 1 syslog F00.

このようなログが出力された場合、図 2 のように可視化を行う。タスクの状態やシステムコールと重複して見づらい場合は、ユーザが不要な可視化項目を可視化しないようにすることで見やすいよう設定することとする。

3.2 ユーザ定義状態可視化

3.2.1 入力

3.2.2 出力

3.3 ユーザ定義状態可視化（端点付き）

3.3.1 入力

3.3.2 出力

参考文献