

# 小規模組み込みシステム向け シェル・タスク 『Natural Tiny Shell Task』

2011年度  
TOPPERS活用アイデア・アプリケーション  
開発コンテスト



# 目次

- デバッグ環境
- よくあるシェル環境の例
- なぜ入力が破綻するのか？
- Natural Tiny Shell Taskでこんなに便利に！
- 展示中のデモ・キット

小規模組み込みシステムにおける  
**デバッグ環境**

# デバッグ環境



ターゲット・システム



ホスト・コンピュータ

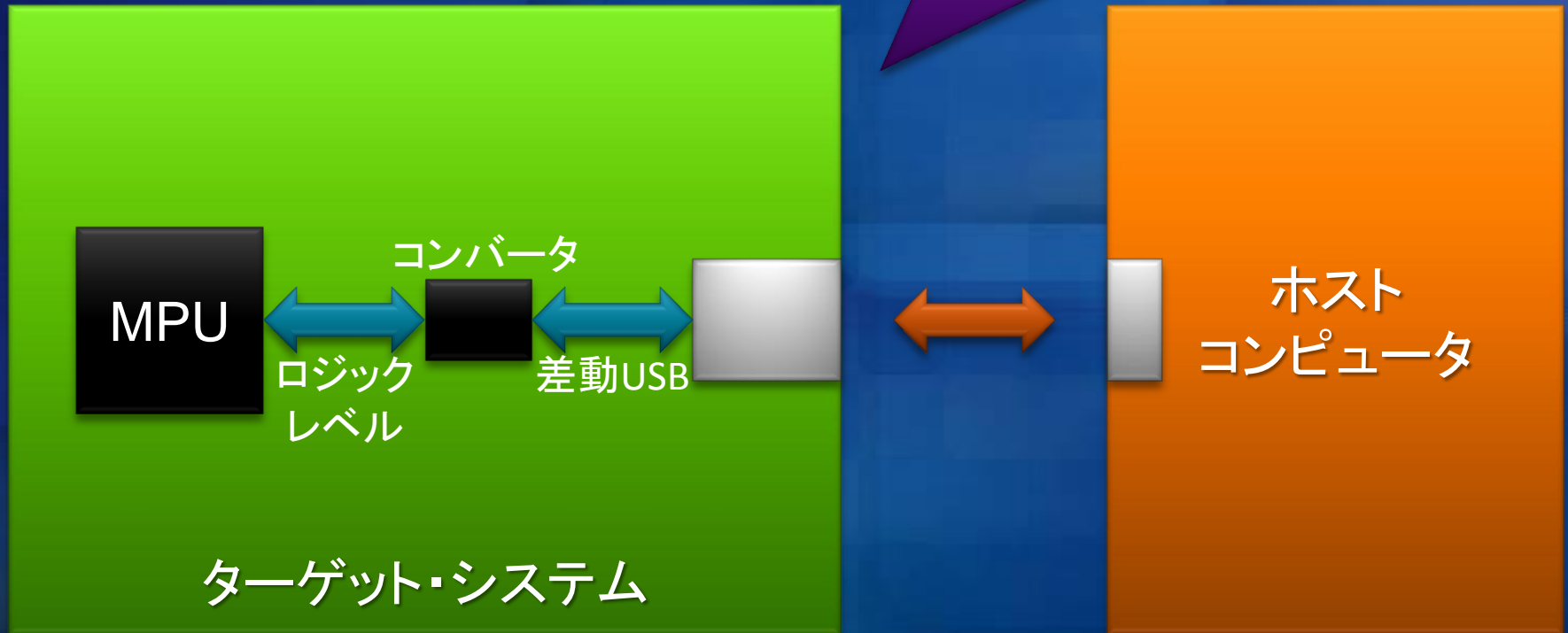


開発者

# Virtual COM Port (VCP)

～USBでシンプルに繋ぐ～

ドライバにより  
COMポートとして見える。







UART-USB変換

ターゲット・システム

# ホスト用インターフェースの実装





小規模組み込みシステムにおいて

# よくあるシェル環境の例



# ありがちな例1

～だんまり・シリアル・ログ・コンソール～

```
TOPPERS/ASP Kernel Release 1.7.0 for LPCXpresso 1768 (Sep 11 2011, 16:44:02)
Copyright (C) 2000-2003 by Embedded and Real-Time Systems Laboratory
                          Toyohashi Univ. of Technology, JAPAN
Copyright (C) 2004-2011 by Embedded and Real-Time Systems Laboratory
                          Graduate School of Information Science, Nagoya Univ., JAPAN
Copyright (C) 2010 by TOPPERS/ASP for LPC project
http://sourceforge.jp/projects/toppersasp4lpc/
```

```
System logging task is started on port 1.
```



システムが自分の状態をただひたすら表示するだけ！  
何を打っても響かない！

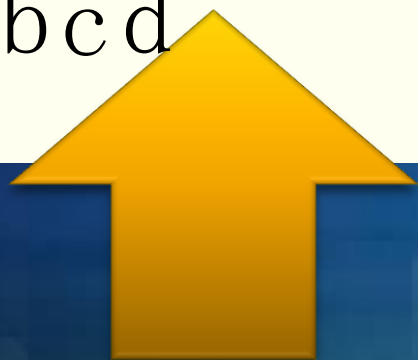
# ありがちな例2

～1文字打法で問題解決？～

```
TOPPERS/ASP Kernel Release 1.7.0 for LPCXpresso 1768 (Sep 11 2011, 16:44:02)
Copyright (C) 2000-2003 by Embedded and Real-Time Systems Laboratory
                          Toyohashi Univ. of Technology, JAPAN
Copyright (C) 2004-2011 by Embedded and Real-Time Systems Laboratory
                          Graduate School of Information Science, Nagoya Univ., JAPAN
Copyright (C) 2010 by TOPPERS/ASP for LPC project
http://sourceforge.jp/projects/toppersasp4lpc/
```

```
System logging task is started on port 1.
```

```
abcd
```



1文字で処理するから、シーケンスなんて関係ない？

# ありがちな例3

～割と動きます。たまにおかしいけど～

```
TOPPERS/ASP Kernel Release 1.7.0 for LPCXpresso 1768 (Sep 11 2011, 16:44:02)
Copyright (C) 2000-2003 by Embedded and Real-Time Systems Laboratory
                          Toyohashi Univ. of Technology, JAPAN
Copyright (C) 2004-2011 by Embedded and Real-Time Systems Laboratory
                          Graduate School of Information Science, Nagoya Univ., JAPAN
Copyright (C) 2010 by TOPPERS/ASP for LPC project
http://sourceforge.jp/projects/toppersasp4lpc/
```

```
System logging task is started on port 1.
```

```
>sys[1D#$%@! (@
Command not found
```



一部端末制御コードにのみ対応！  
対応していないコードが入ると破綻！



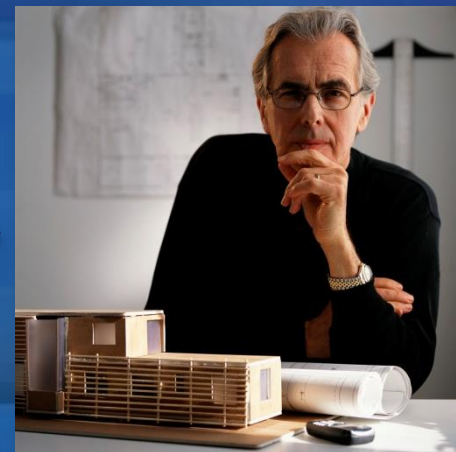
# デバッグ環境



ターゲット・システム



ホスト・コンピュータ



開発者

ターゲット・システムの色々な事象を追いながら、  
ホスト・コンピュータを介して、  
システムを操作する開発者。

入力にイチイチ気を使っていられない！



そもそも

なぜ入力が破綻するのか？

# ターミナル・エミュレータの 生い立ち



ターゲット・システム



ホスト・コンピュータ



開発者



ホスト・コンピュータ上の  
ターミナル・エミュレータの生い立ち

# VT100 (Digital Equipment Corporation社)



<http://ja.wikipedia.org/wiki/VT100>

# 端末制御コード

～メイン・フレームとターミナル間の接続～



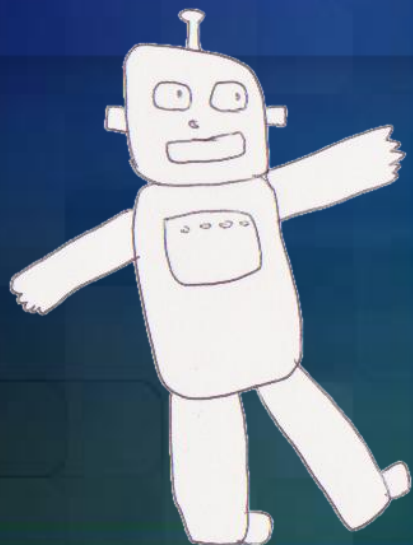
処理装置と端末間のやりとりの為の制御コード





# エスケープ・シーケンスの例

- ESC[1A : 1ラインカーソルを上へ移動。
- ESC[5D : カーソルを5文字左へ移動。



エスケープ・コードに続く  
一連の文字列が  
制御内容を示す。

# VT100ターミナル・エミュレータ

## 端末

(ホスト・コンピュータ上のソフトウェア)



既に多くの優れた  
端末エミュレータが存在。

ここ！

# 組み込みシステム

## 処理装置

(デバッグ対象ターゲット・システム)



端末エミュレータのコードを、  
正しく認識しなければ破綻。



Natural Tiny Shell Taskを使うメリット

**Natural Tiny Shell Taskで  
こんなに便利に！**

# 狙いの性能と、おまけの性能

## ● 狙いの性能

- ホスト・コンピュータから正しく文字入力可能。
- 小規模組み込みシステムで使用可能。
- 簡単に導入可能。
- 簡単に拡張可能。
- 簡単に移植可能。
- 簡単に解読可能。

## ● おまけの性能

- 履歴機能が付くと面白そう。
- TABキーで入力補間できれば面白そう。
- 色んな人に使って喜んでもらえたら嬉しい。

# 特徴

- VT100準拠で入力処理が破綻しない。
- システム固有のコマンドを簡単に追加可能。
- 履歴からの入力をサポート。
- 導入が容易。
- 小規模組み込みシステムでの使用を考慮。
  - ROM: 約9KB (実装次第で削減可能。)
  - RAM: 約1KB (入力可能最大文字数、履歴管理数に依存。)
- 移植性が高い。
  - 下層設計はOS非依存。
  - 下層設計はlibc非依存。
- 便利なコマンド・パース・ユーティリティ付き。

# 構造概略

Natural Tiny Shell Task

Natural Tiny Shell Library

シリアル通信関数

RTOS



# コマンドの追加(ntshell\_usrcmd.c)

```
static int usrcmd_help(int argc, char **argv);
static int usrcmd_info(int argc, char **argv);

/**
 * @brief コマンドテーブル。
 */
typedef struct {
    char *cmd; /**< コマンド名。 */
    char *desc; /**< コマンド説明。 */
    int (*func)(int argc, char **argv); /**< コールバック関数。 */
} cmd_table_t;

/**
 * @brief コマンドの定義。
 * @details システムで必要なコマンドの実装を追加すると良い。
 */
static const cmd_table_t cmdlist[] = {
    { "help", "Help.", usrcmd_help },
    { "info", "Information.", usrcmd_info },
    { NULL, NULL, NULL }
};
```

# コマンドの実装

```
159 /**
160  * @brief infoコマンド。
161  * @details
162  *
163  * @param argc 引数の数。
164  * @param argv 引数。
165  *
166  * @retval 0 成功。
167  * @retval !0 失敗。
168  */
169 static int usrcmd_info(int argc, char **argv)
170 {
171     if (argc != 2) {
172         static const char *msg = "info sys\ninfo ver\n";
173         serial_wri_dat(SHELL_PORTID,
174             (const char_t *)msg, ntlibc_strlen(msg) + 1);
175         return 0;
176     }
177     if (ntlibc_strcmp(argv[1], "sys") == 0) {
178         static const char *msg = "NT-Shell Example System\n";
179         serial_wri_dat(SHELL_PORTID,
180             (const char_t *)msg, ntlibc_strlen(msg) + 1);
181         return 0;
182     }
183     if (ntlibc_strcmp(argv[1], "ver") == 0) {
184         static const char *msg = "0.0.1\n";
185         serial_wri_dat(SHELL_PORTID,
186             (const char_t *)msg, ntlibc_strlen(msg) + 1);
187         return 0;
188     }
189     {
190         static const char *msg = "Unknown sub command.\n";
191         serial_wri_dat(SHELL_PORTID,
192             (const char_t *)msg, ntlibc_strlen(msg) + 1);
193     }
194     return -1;
195 }
```

# ファイル

- タスク用ファイル

- task\_ntshell.c, task\_ntshell.h, task\_ntshell.cfg
- ntshell\_usrcmd.c, ntshell\_usrcmd.h, ntshell\_config.h

- 最上位インターフェースモジュール

- ntshell.c, ntshell.h

- テキストエディタ

- text\_editor.c, text\_editor.h

- テキストヒストリ

- text\_history.c, text\_history.h

- ユティリティ

- ntlbc.c, ntlbc.h
- ntopt.c, ntopt.h

- VT100パーサ

- vtparse.c, vtparse.h, vtparse\_table.c, vtparse\_table.h

# ライセンス

- MITとTOPPERSライセンスからの任意選択可能。
  - プロプラエタリなソフトウェア開発にお使い下さい。
  - 趣味のソフトウェア開発にお使い下さい。
- もし宜しければ使っている事を教えて下さい。

もちろん無償！

TOPPERSブースにて

**展示中のデモ・キット**

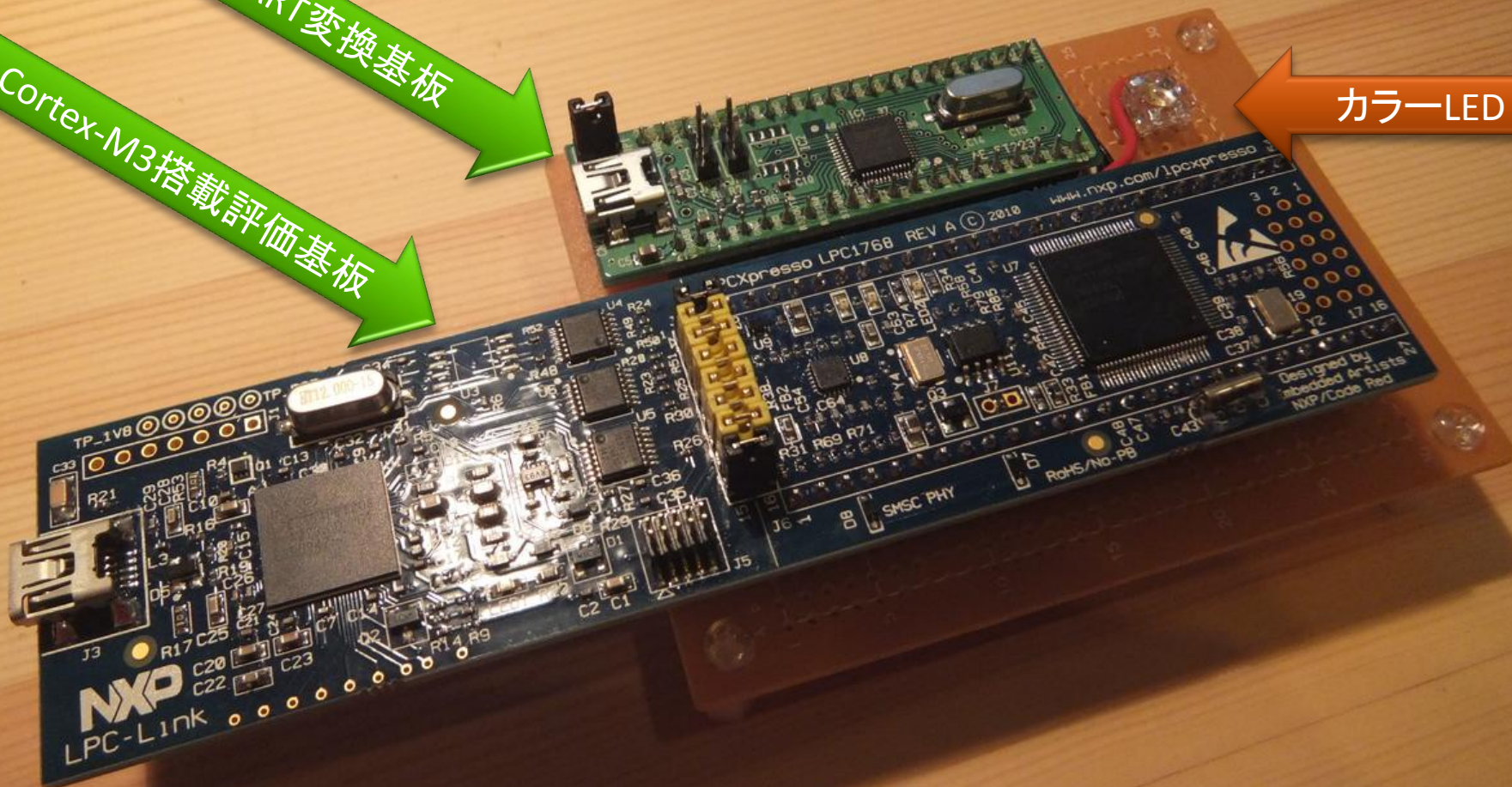


# デモ・キットの外観

USB-UART変換基板

Cortex-M3搭載評価基板

カラーLED



# デモ・キットのプログラム

- 2つのシェル・タスクが稼働。
  - task\_bugshell : 適当に実装したシェル。
  - task\_ntshell : Natural Tiny Shell Task。
- カラーLEDの点灯色をコマンドで変更可能。
  - color black, color red, color green, color blue...

2つのシェルを実際に操作して  
利便性の差を体験可能。

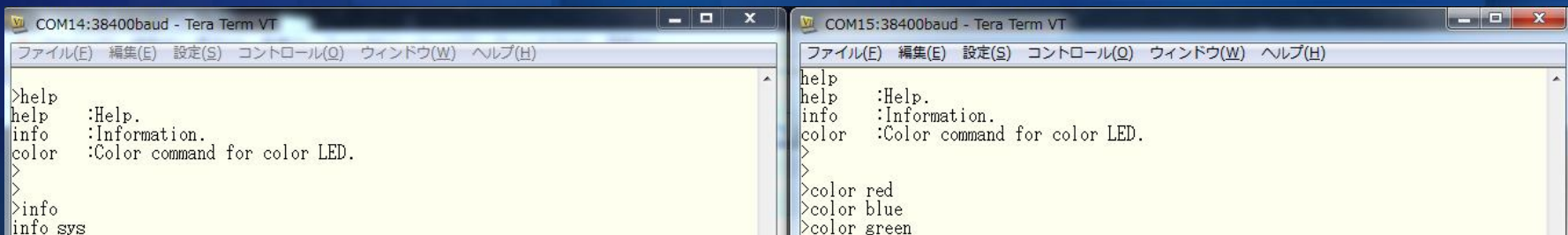
# 2つの端末を並べて比較展示

## task\_bugshell

- 文字の消去などに問題。
- 履歴機能なし。

## task\_ntshell

- 文字の消去なども可能。
- 履歴機能あり。



```
COM14:38400baud - Tera Term VT
ファイル(E) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
>help
help :Help.
info :Information.
color :Color command for color LED.
>
>
>info
info sys

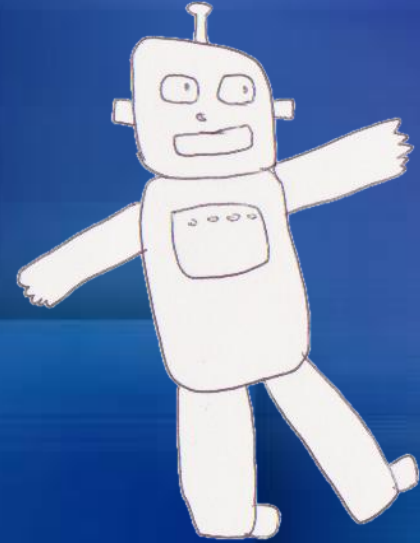
COM15:38400baud - Tera Term VT
ファイル(E) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
help
help :Help.
info :Information.
color :Color command for color LED.
>
>
>color red
>color blue
>color green
```

一見両者に違いはないものの、実際のデバッグになると・・・。

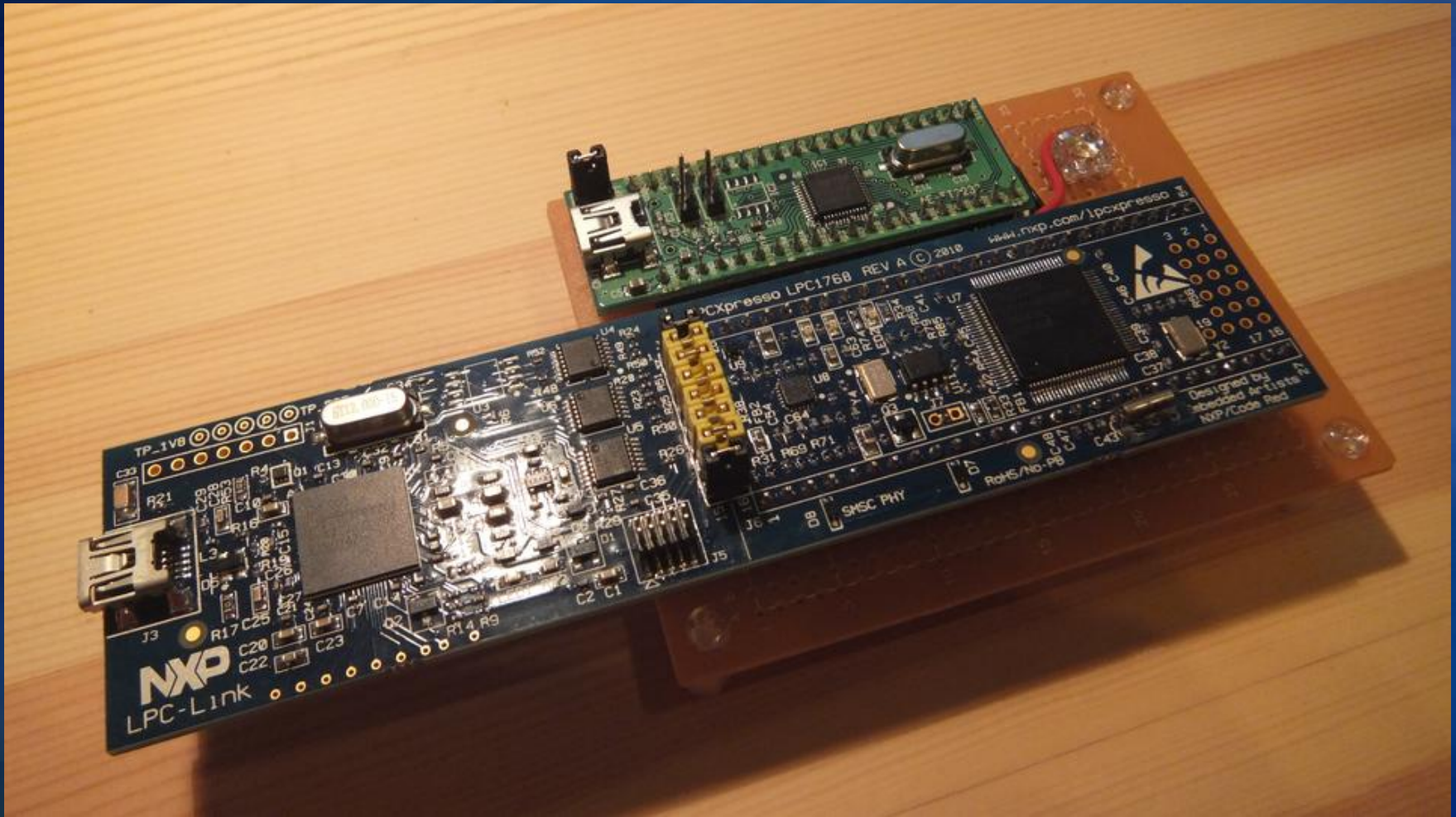


# 謝辞

- Paul Williamsさん
  - vt100.netでの情報公開。
- Joshua Habermanさん
  - vtparseのオリジナル。
- 伸幸さん
  - 初めてのユーザです。
  - DSP/BIOS上でNT-Shellを使用してくれました。
- 酔漢さん
  - NXPセミコンダクターズ社のCortex-M3プロセッサ上で動作するTOPPERS/ASPポートを公開。
  - TOPPERSの魅力を私に教えてくれました。









I2C init done.  
Codec init done.  
ADC init done.  
I2S init done.  
I2S DMA init done.  
Started I2S.

UART2  
UART3

8E6B