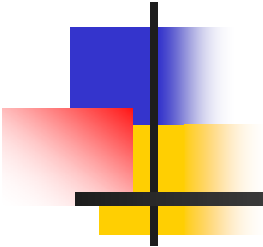


「トラ技付属78K/USBマイコン基板 によるI/O制御入門」

～ JavaScript・携帯電話/PHSによる78K基板操作から
汎用USB-JTAGアダプタの製作まで～



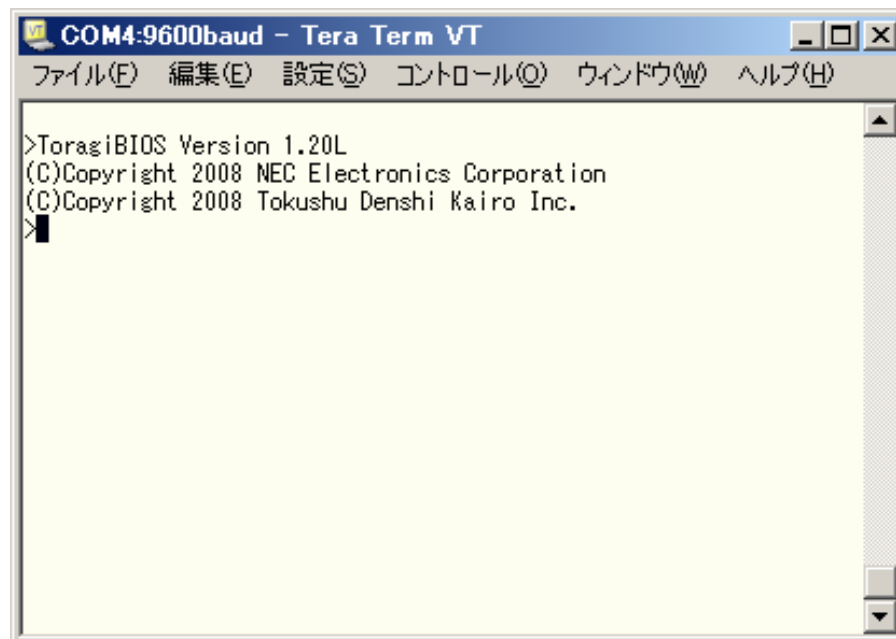
特殊電子回路株式会社 内藤竜治

技術士(電気電子部門)

naitou@tokudenkairo.co.jp

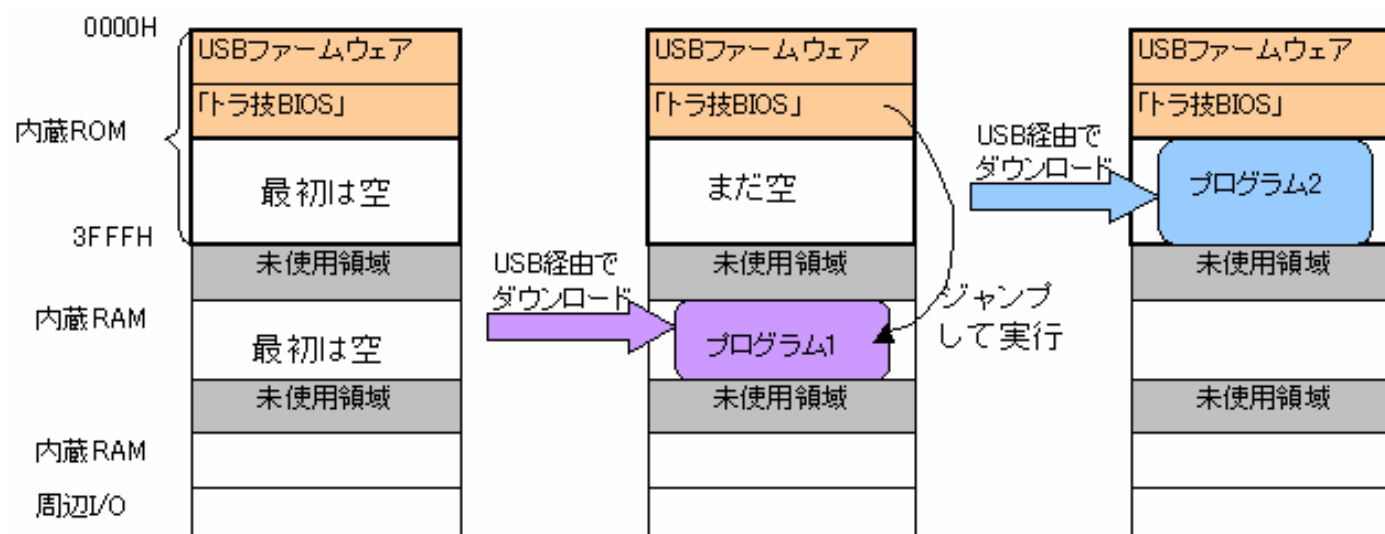
トラ技78Kマイコンの紹介

- 半田付け不要ですぐに使える
- 汎用的なターミナルソフトから操作可能
- 仮想USBシリアル
- バグなし



CPUとメモリマップ

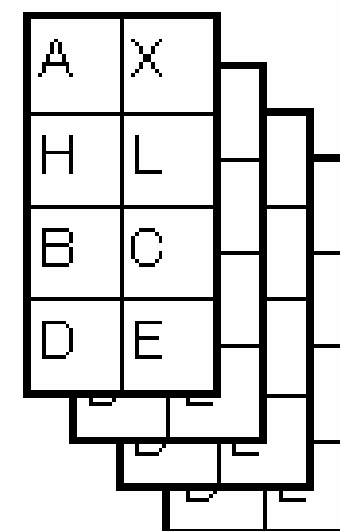
■ ノスタルジックな構成



16bitメモリ空間,8kバイトのROM,3kバイトのRAM

“AXHLBCDE”の8個の
8bitレジスタが4バンク

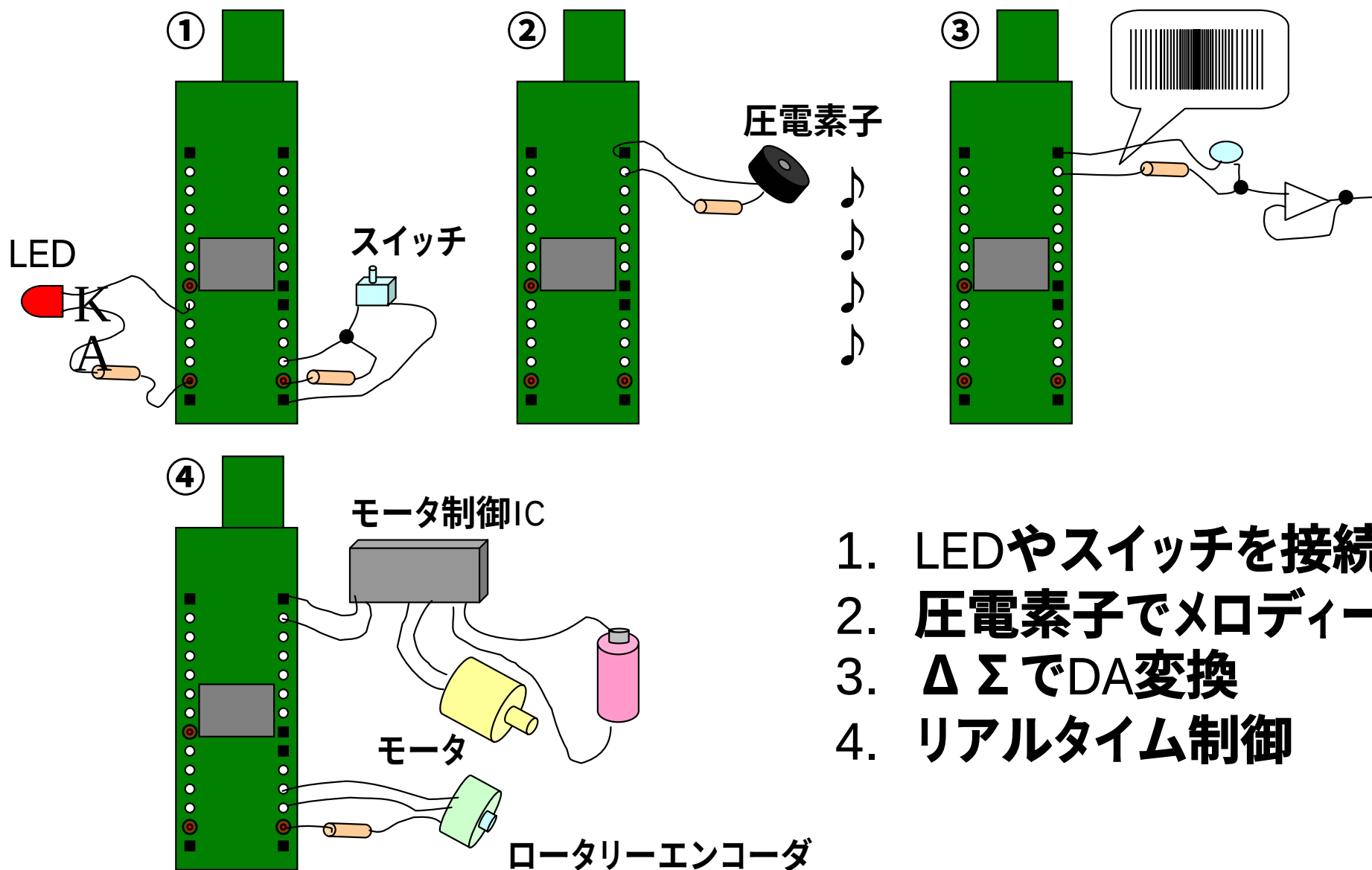
8 bit
↔



Z80で遊んだ頃の記憶をもう一度



記事でつくったアプリケーション



1. LEDやスイッチを接続
2. 圧電素子でメロディー
3. $\Delta \Sigma$ でDA変換
4. リアルタイム制御



本当に作りたかったのは..

トラ技7月号ではLEDの点滅やI/Oの読みとり方法、倒立振子の製作などの例を紹介しました。

しかし、本当に必要とされているアプリケーションは、このマイコン基板を使って**CPLD**や**FPGA**や**各種マイコン**に**USB**から書き込むための**万能ライター**ではないでしょうか？

本講演では、78Kマイコンをさらに活用するための様々な**応用事例**を**デモを交えて**紹介します。

Webブラウザで使いたい

■ なぜ、いまさらWebブラウザ？

- 凝ったGUIが簡単に作れる
- 開発環境が無料 (JavaScript)
- 操作アプリをインストール
してもらわなくてよい
- ネットと連携ができる
- ブラウザ上で78Kマイコンの
ファームウェアを書き換えられる
- 携帯電話、PDAからでも使える

動作テスト

COM4: NEC Electronics USB to RS232C Converter Virtual COM Port (COM4)

コマンド: アドレス 長さ

コマンド: アドレス データ

コマンド: アドレス

通信

```
F010 40 E4 F4 E5 80 A0 01 FF 2F FD A8 A9 20 C1 A5 29 @...../... ..)
F020 20 41 11 8D 42 CC 64 B3 AC DD 59 D9 64 8E 50 8D A.mB.d...Y.dnPm
F030 63 6A D0 F5 53 AF 66 77 D5 FC 5E 4B 80 AF E5 6F cJ...S.fw...K...o
F040 09 CE E8 FE 56 37 38 E9 AC FE 59 12 13 27 09 29 ....V78...Y...')
>dump F000 50
F000 00 FF 5E CA 10 BA E0 FF 57 FF 2A D3 32 B0 13 36 ..^.....W.*.2..6
F010 40 E4 F4 E5 80 A0 01 FF 2F FD A8 A9 20 C1 A5 29 @...../... ..)
F020 20 41 11 8D 42 CC 64 B3 AC DD 59 D9 64 8E 50 8D A.mB.d...Y.dnPm
F030 63 6A D0 F5 53 AF 66 77 D5 FC 5E 4B 80 AF E5 6F cJ...S.fw...K...o
F040 09 CE E8 FE 56 37 38 E9 AC FE 59 12 13 27 09 29 ....V78...Y...')
>load
Drop a HEX file.
complete.
>
>jmp F000
```

ユーザプログラムを実行します
かえるの歌を演奏します。P01端子に圧電サウンドを接続してください。
xキーを押してください

Hexファイル

```
:010080007F00
:01008100007E
:01008400007B
:03F000009BA8F0DC
:10F004002082A982A682E982CC89CC82F089899166
:10F014007482B582DC82B78142503031925B8E714A
:10F0240082C988B3936483548345833835F82F056
:10F0340090DA91B182B582C482AD82BE82B382A2DB
:10F0440081420A0078834C815B82F0899F82B58279
:0CF05400C482AD82BE82B382A2000A001A
:10F06000B7D696C6806168BD039B70F0009B62F0C6
:10F07000B6AFB7B1B1891CD6A10097BE01AE0149A8
```

つまり、作るのも、使うのも、お手軽

JavaScriptの問題点

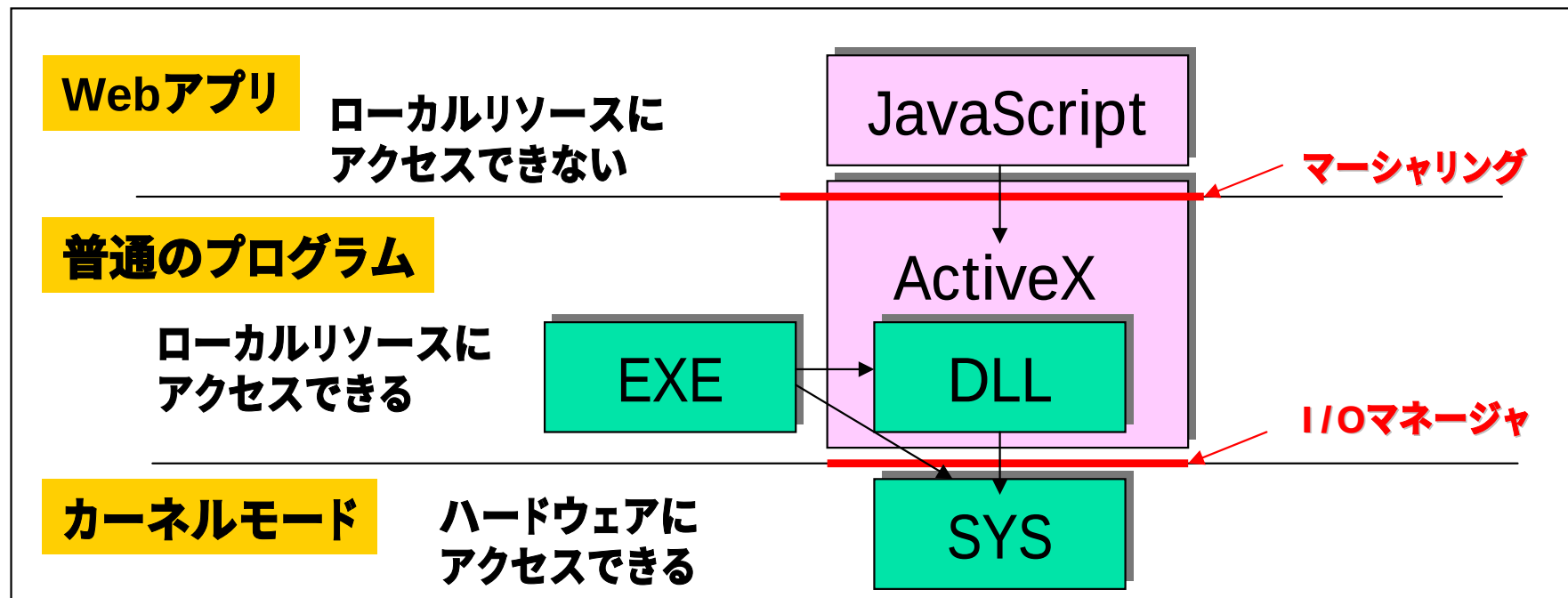
- JavaScriptは、基本的にはブラウザの中だけで動くもの
- ローカル資源（ディスク、USB、シリアルポートへのアクセスは、基本的にはできない



そこで、ActiveX

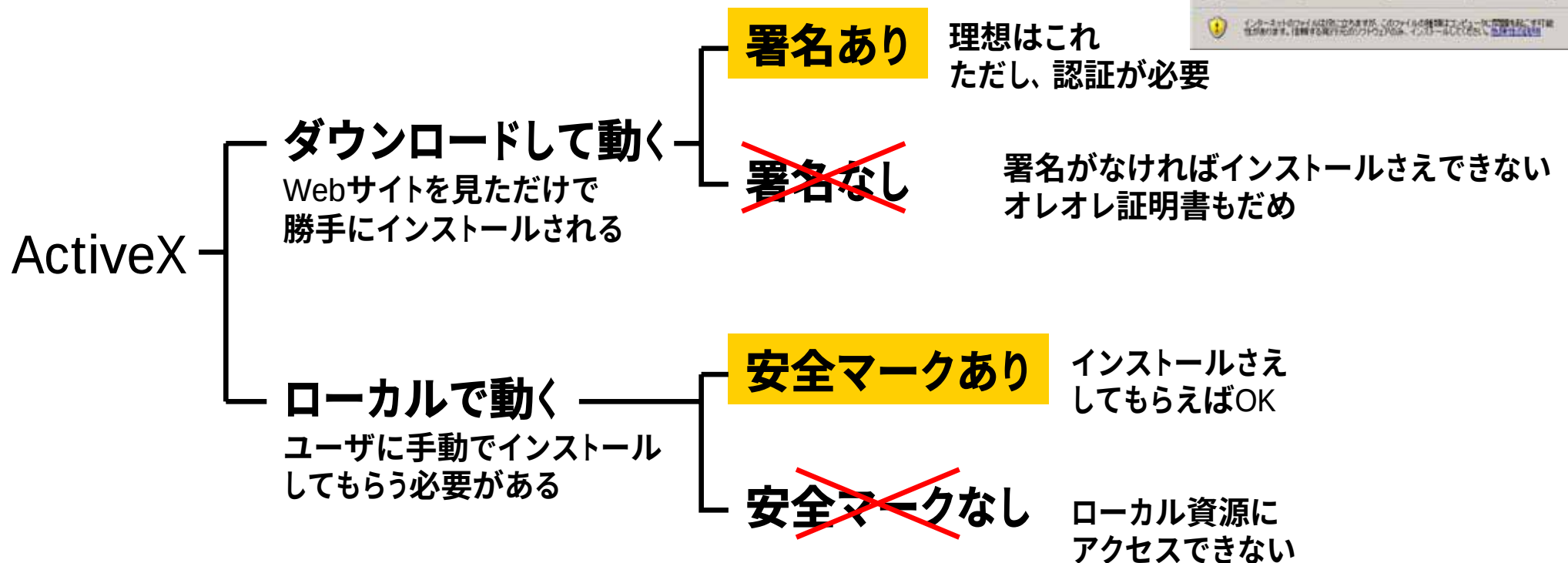
■ ActiveXとは

- DLLの上に一枚クラスをかぶせたもの
- **正体はただのDLL**なので何でもやりたい放題
- いつでもどこでも (JavaScriptでもVBでもC++でもC#でも) どこからでも (ローカルでもネットでも) 呼び出せる



ActiveXの認証と安全マーク

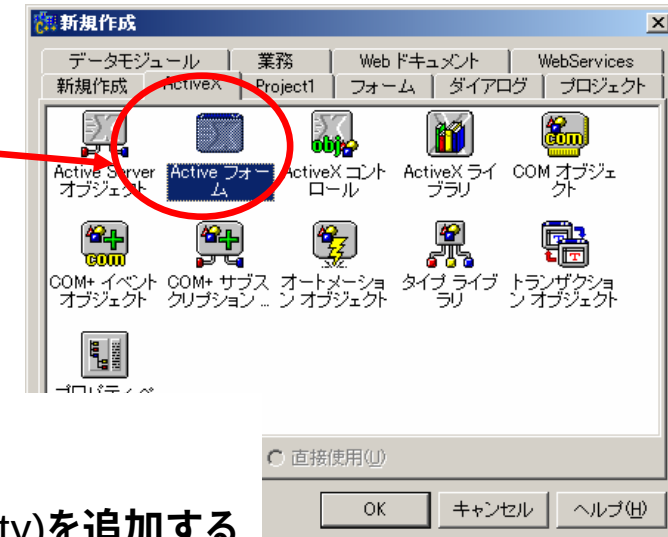
- 証明書と署名が面倒 (年間約5～10万円＋登記謄本)
- オレオレ証明書ではWinXPはごまかせない
- 安全マークをつけて、ローカルにインストールすればよい



ActiveXの作り方とインストール方法

① BorlandC++Builderで簡単に作れる

ActiveXフォームの新規作成を選択

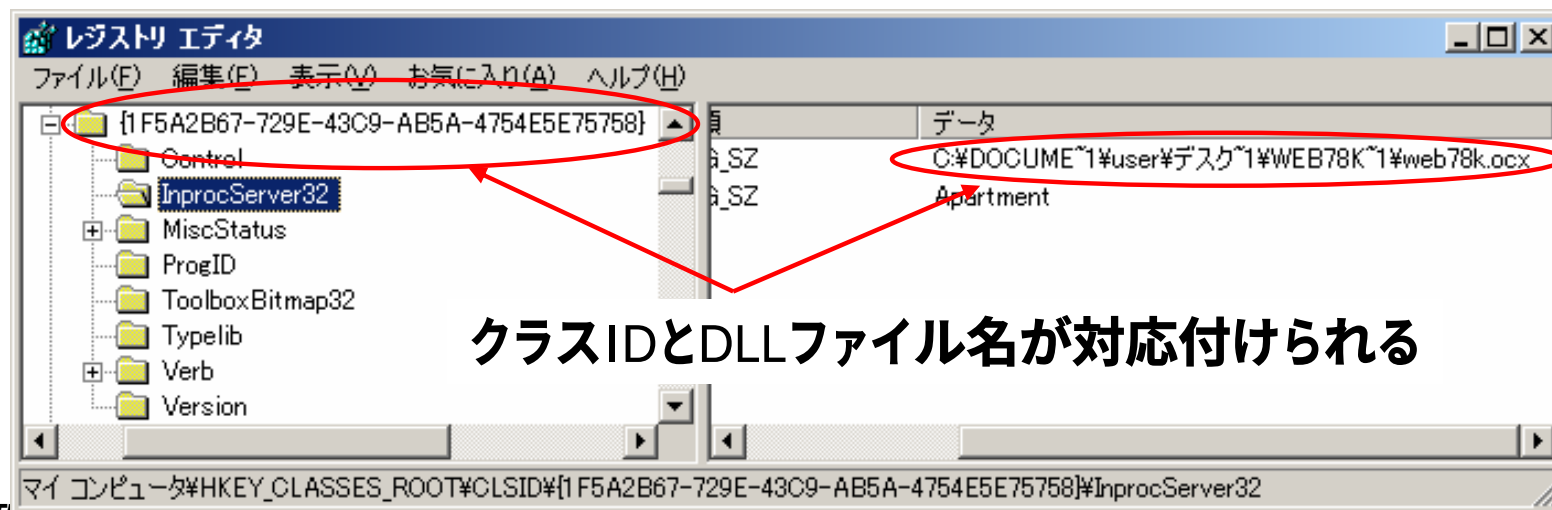


② 安全マークをつける

1. 作ったクラスがIObjectSafetyImplを継承するようにする
2. BEGIN_COM_MAP(TxxxImpl)に
VCL_CONTROL_COM_INTERFACE_ENTRIES(IObjectSafety)を追加する

③ ローカルにインストールする `regsvr32 xxxxx.ocx`を実行

④ レジストリに登録される



クラスIDとDLLファイル名が対応付けられる

ActiveXの使い方

■ HTMLにJavaScriptを貼り付けるで良い

```
<HTML> . . .  
<BODY>  
<OBJECT id=web78k codeBase=web78k.ocx height=25 width=81 align=left  
classid=clsid:1F5A2B67-729E-43C9-AB5A-4754E5E75758></OBJECT>
```

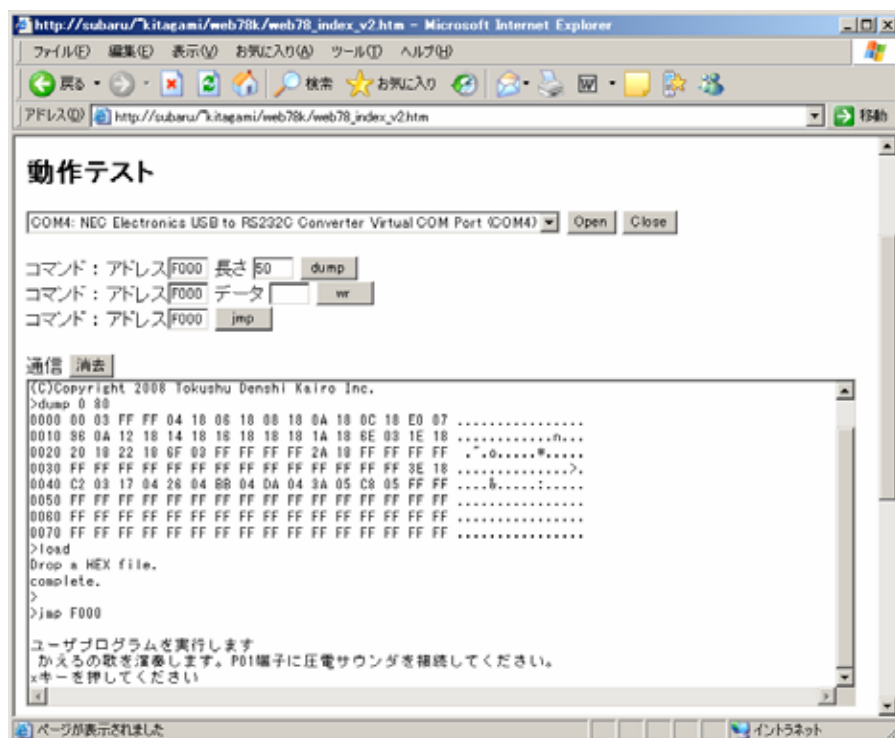
前ページでレジストリに登録されたDLLが呼び出される

```
web78k.OpenPort(1);  
web78k.WriteData("こんにちは");  
var rxtext = document.getElementById("rx") ;  
rxtext += web78k.ReadData();  
web78k.ClosePort(1);
```

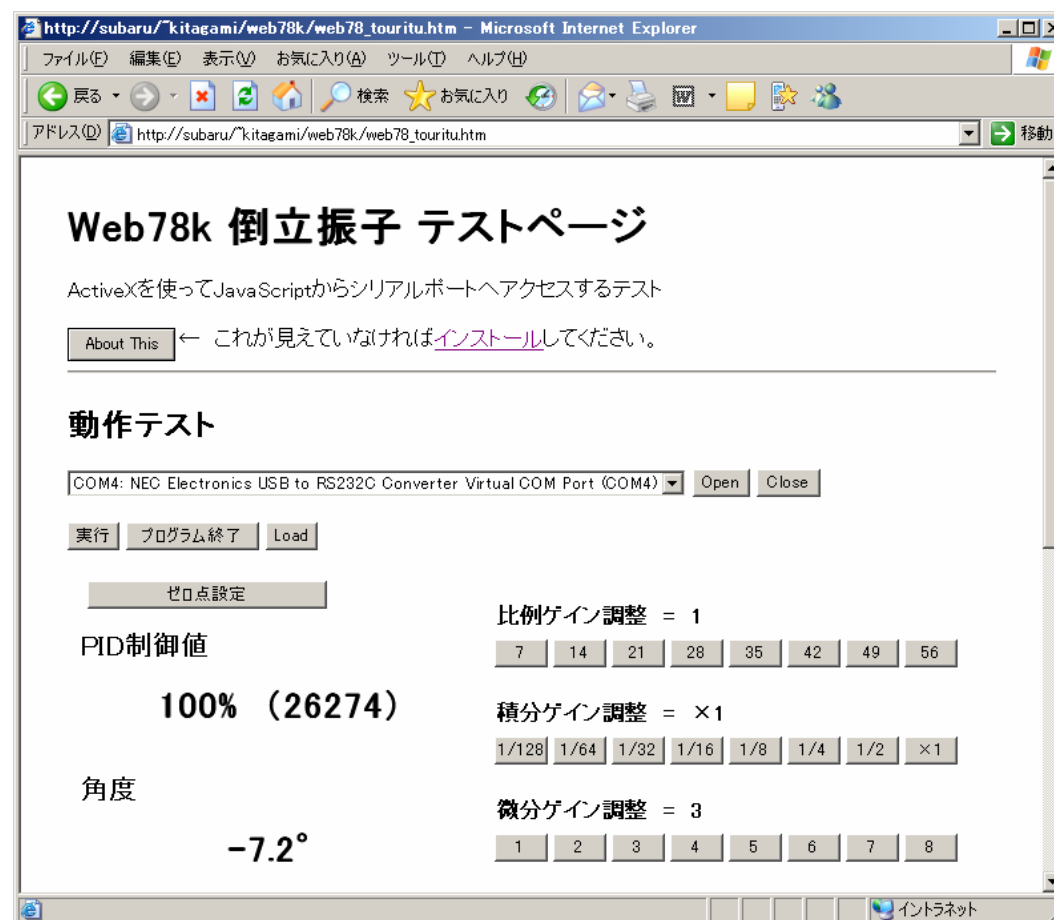
JavaScriptからこんなに簡単に78Kマイコンを操作できる

デモンストレーション

WebTerm



Web倒立振子



PDA/携帯電話から操作したい

- お手軽マイコンを使うのに、ノートパソコンなんて持ち歩きたくない。外出先で、ささっとつなげたい



※ WS003SHとはつながりません
※ 画面は合成です

Windows Mobileとは

- Windows Mobileとは
 - 携帯電話やPHSに搭載されているWindows
 - Windows CEから機能を削ったバージョン
 - アプリやデバドラは、Windows CE用に開発すればOK
- CEのデバイスドライバは・・・
 - 実は、ただのDLL。WDMより簡単。
 - カーネルモードとユーザモードがある
 - ユーザモードのデバイスドライバではメッセージボックスなどが出せる



WILLCOMのWebサイトより

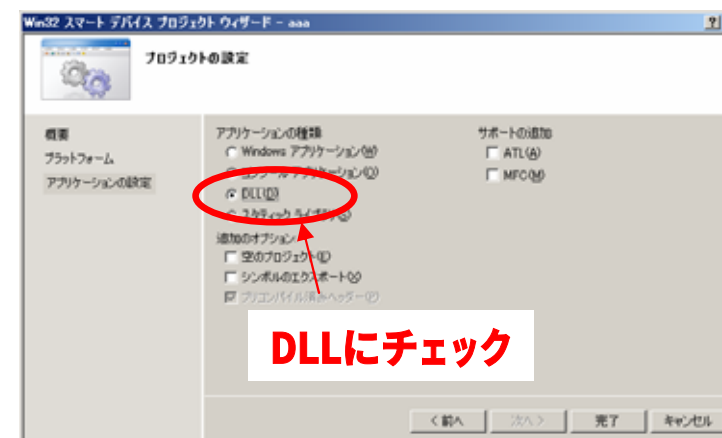
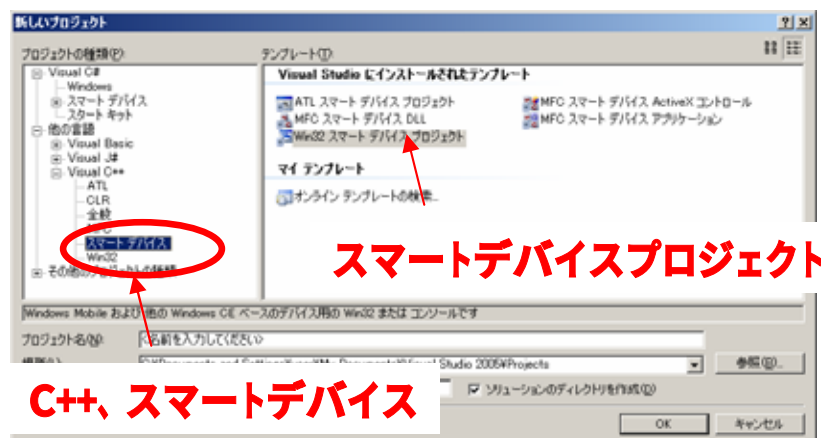
参考資料 →
08/07,08/08号



Mobile用デバイスドライバの作りかた

■ Microsoft VisualStudio 2005で開発可能

- 「Windows Mobile 5.0 Pocket PC SDK」と「Windows Mobile 5.0 Emulator Images for Pocket PC」を追加でインストールする
- [VisualC++]→[スマートデバイス]→[Win32スマートデバイスプロジェクト]で作成する
- プラットフォームビルダーは不要



■ 必要な関数をエクスポートする

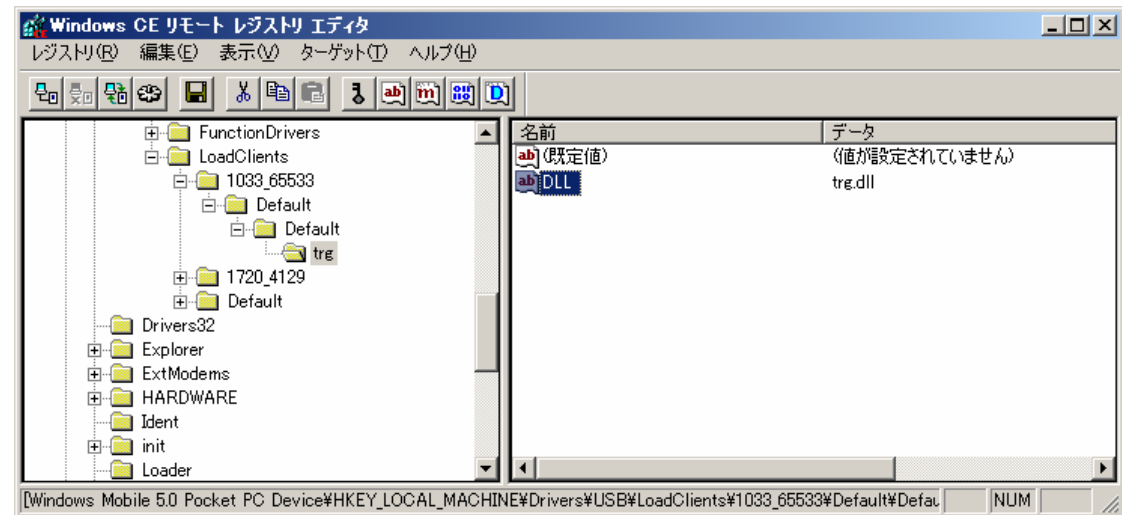
- TRG_Init(), TRG_Write (), TRG_Read ()など10種類の関数をエクスポートする
- USBの場合は、USBInstallDriver ()、USBDeviceAttach ()など追加で3個必要

■ IssueVendorControlやIssueBulkControlといった関数でUSBを低レベル操作する

78Kマイコンが認識されるしくみ

1. USBに何か挿入される
2. システムがVID(0x0409)、PID(0xffffd)を取得
3. レジストリの[HKEY_LOCAL_MACHINE¥Drivers¥USB¥LoadClients¥1033_65533¥Default¥Default¥trg]キーが検索される。

4. そこに書かれているDLLが呼び出される。
DLLは¥Windowsフォルダにあらかじめ入れておくこと。

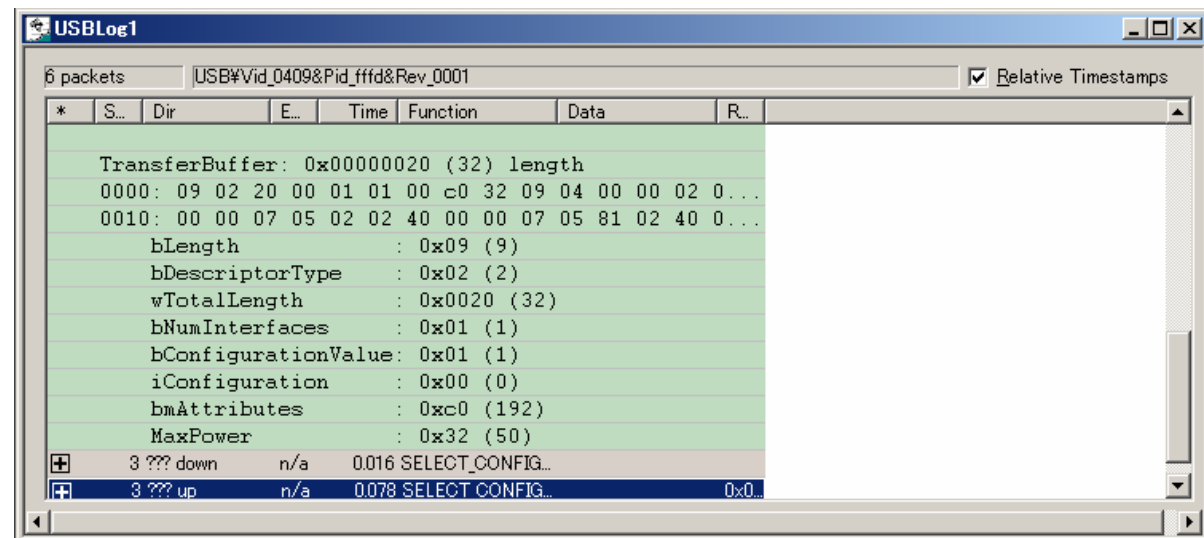


5. デバイスマネージャがUSBDeviceAttach関数をコールする

- ① [HKEY_LOCAL_MACHINE¥Drivers¥USB¥ClientDrivers¥Toragi78k]にDLLを登録する
- ② ActivateDevice関数をコールすると、レジストリのHKEY_LOCAL_MACHINE¥Drivers¥Active¥数値]に登録され、CreateFileでデバイスドライバを呼び出せるようになる。

78Kマイコンの通信プロトコルの解析

- トラ技78Kマイコンの通信プロトコルはわからない
 - フリーソフトのSnoopyProを活用して解析



- ・オープンするには、ベンダリクエスト “03 01” を発行すればよい
- ・Endpoint1にBulkOut、Endpoint2からBulkInで通信可能
- ・受信リクエストを出すタイミングが、正直、よくわからない

PHSからMobile78Kへ接続



一応、通信できているらしい

本格的な開発はまだまだこれからです

万能書き込み器を作りたい

- **トラ技78Kマイコンの欠点は
「BIOS領域の書き換えにRS232Cが必要なこと」**



RS232Cでのライターは自作可能だが、RS232Cがない

USBマイコンならば、USBから書き換えられないの？

2台つないで書き換えよう

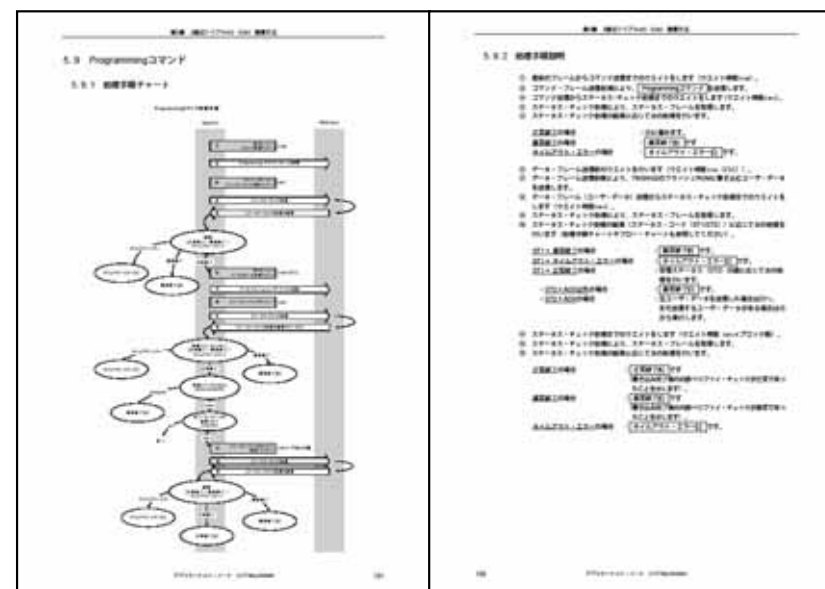
78Kマイコンに書き込むには

- RS232CとCSIのアルゴリズムがある
- CSIでの書き込みアルゴリズムはNECから公開されている！

<http://www.necel.com/micro/ja/promotion/kx2/flashprog.html>

上記ページは78K0/Kx2用だが、
そのまま78F0730にも適用できる

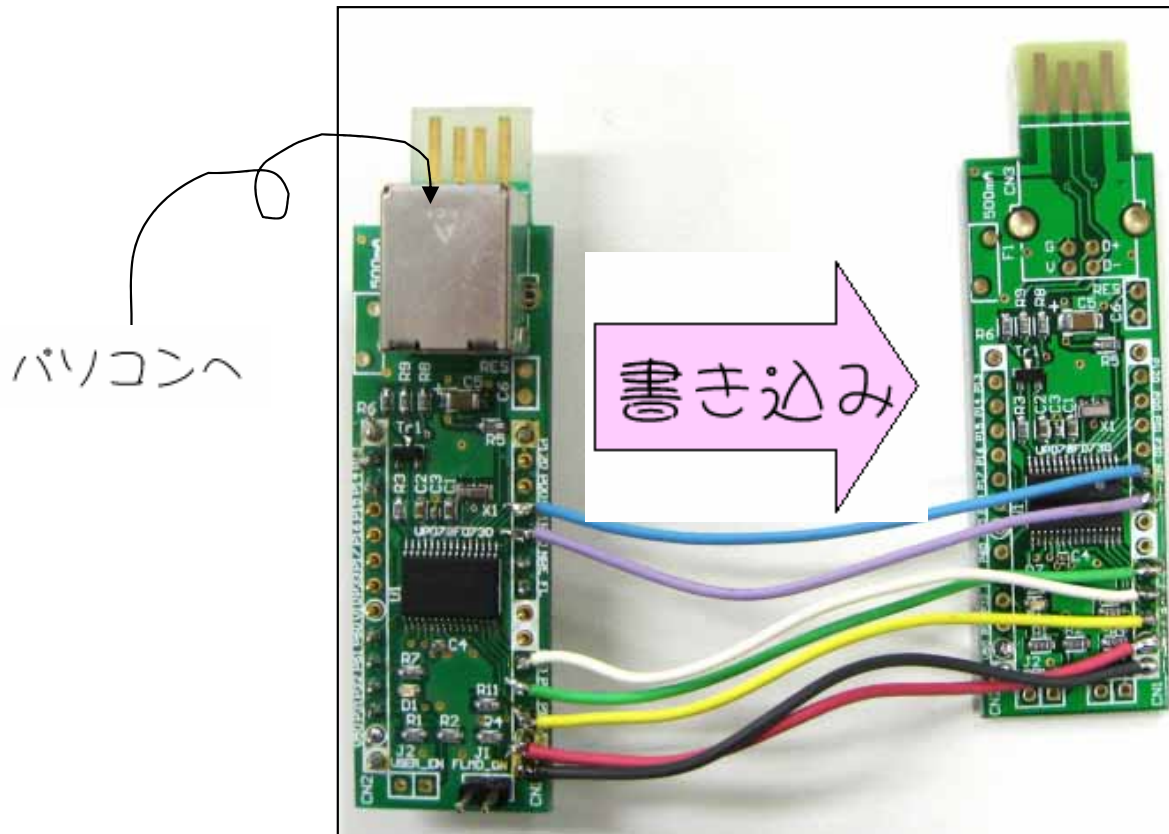
78KマイコンのI/Oポートから
右の資料に記された波形を出せば
書き込みできるはず



資料番号 U17739JJ3V0AN00

78K基板による78K基板のための78K基板ライター

■ 完全脱レガシーを実現



USBマイコンはUSBで書き込もう。
もう、レガシーRS232Cや
USB-RS232Cは
本当に本当に不要

(付録基板を2枚使います)

いろいろなマイコンに書き込むアダプタが簡単に作れるはずです

書き込みソフトウェアも作ったが・・・

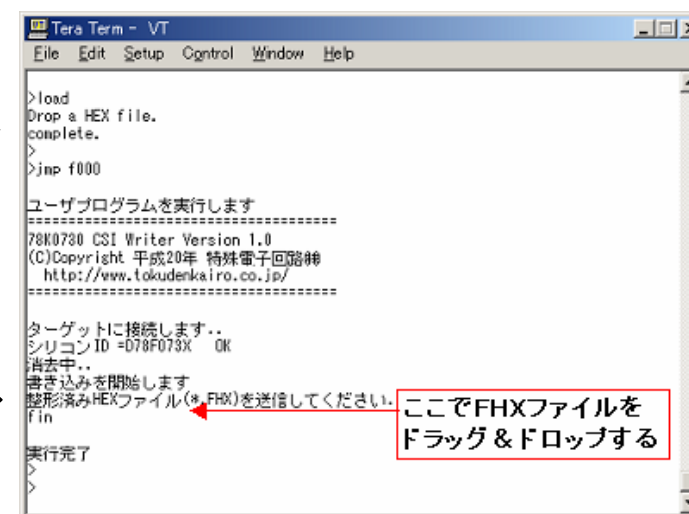
PM+が出すHEXファイルはでこぼこしている

```

1: 010080007F00↓
2: 01008100007E↓
3: 01008400007B↓
4: 03F000009B4EF72D↓
5: 10F00400007020C0C54000000003FFF66696E0A22↓
6: 10F014000057522045525B005D0A0090AE8C604E↓
7: 10F024008DCF82DD484558837483408343838B2886↓
8: ...↓
9: 10F0F400312E300A00284329436F7079726967689A↓
10: 10F10400742095BD90AC3230944E2093C18EEA9316↓
11: 10F11400648E7189F19848878A0A00202068747↓
12: 10F12400703A2F2F777772E746F6B7564656E↓
13: 10F134006169726F2E636F2E6A702F0A003D3D3D28↓
14: 10F144003D3D3D3D3D3D3D3D3D3D3D3D3D3D3D3D↓
15: 10F154003D3D3D3D3D3D3D3D3D3D3D3D3D3D3D3D↓
16: 06F164003D3D3D3D3D3D3D3D3D3D3D3D3D3D3D3D↓
17: 10F18C00B7B1891↓
18: 10F19C000630AE0↓
19: ↓
269: @2180: 4D00AD228730AE01D480BE0130978572AE0630AE07D480BE0730BE06856142AD↓
270: @21A0: CE120100FA03120000B0B6AFB7B1B1891CD6AE0230AE03A2043182BE0330BE02↓
271: @21C0: A10097BE01AE014903BD038749029D0B8730AE0180BE013097FAEAB0B0B6AFB7↓
272: @21E0: 710B21711A21712B210A012A01710B20711B200B001B00109E079AAC211A0010↓
273: @2200: D0079AAC210A001010279AAC21A600664D089D131B00100A009AAC211A00100A↓
274: @2220: 009AAC2146FAE810E8FD9AAC21B6AF1321FF1320FFAFB7B1B1891CD6A100BE↓
275: @2240: 02F001BE01A100BE03AE034D089D48AE026101BE02AE015DFEBE01AE04317F08↓
276: @2260: AE016D04BE01FA06AE015DFEBE01AE01F201A1025E01AD04A101FA02A1006902↓
277: @2280: BE02AE046101BE04AE016101BE010A01AE0341BE03FAB21064009AAC21AE02A3↓
278: @22A0: 0032B0B0B0B6AFB7B1B1891CD6A10097A0019A362287190897AE08A000309A36↓
279: @22C0: 22A100BE01AE0149089D26A0000902302903D4857087611897AE01A000090230↓
280: @22E0: 2903D485A000309A3622AE0141BE01FAD487A000309A36221003009A3622B0B↓
281: @2300: B6AFB7B1B1891CD6A10097A0029A362287190897AE08A000309A3622A100BE01↓
282: @2320: AE0149089D26A0000902302903D4857087611897AE01A0000902302903D485A↓
283: @2340: 00309A3622AE0141BE01FAD487A000309A3622AE0A4D00AD081003009A3622FA↓

```

① 専用のアプリで整形



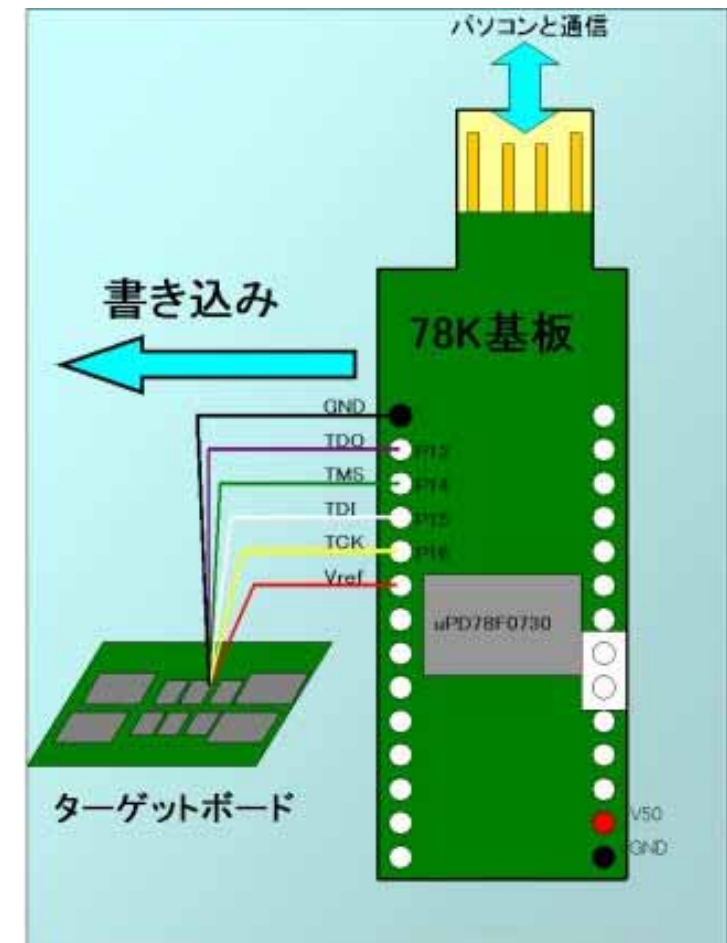
② 書き換えツールをRAMにダウンロードして起動

③ 目的のHEXファイルをダウンロード

公開したが・・・誰も使っていない！

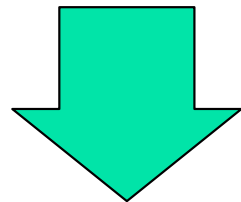
USB-JTAGアダプタの製作

- JTAGとは
 - 4本線のシリアル通信
 - TCK、TDI、TMS、TDO
 - SPIと似ているが、データ長が可変
 - FPGAやCPUに書き込める
 - CPUのデバッグ(ICE)ができる
- ハードウェアはとても簡単
 - 4個のI/OポートをJTAG用にする
 - USBからの指令で
I/Oポートを上げ下げする



フリーのUSB-JTAGの難しいところ

- 汎用の（フリーの）USB-JTAGが難しかった理由
 - JTAGデバイスの書き込みアルゴリズムがみなちがう。
 - メーカー間、デバイス間で全く互換性はない
 - JTAG書き込みアルゴリズムは基本的に非公開
 - 各社から次々と新しいデバイスが登場し、対応させるのが大変



**万能のJTAG書き込み
ツールをフリーで公開します**



DWM2007年7月号付録基板
Spartan3E (XC3S250E)

78Kマイコンで作ったUSB-JTAG
(FPGAへのJTAG書き込みの実験中。65秒で完了。)

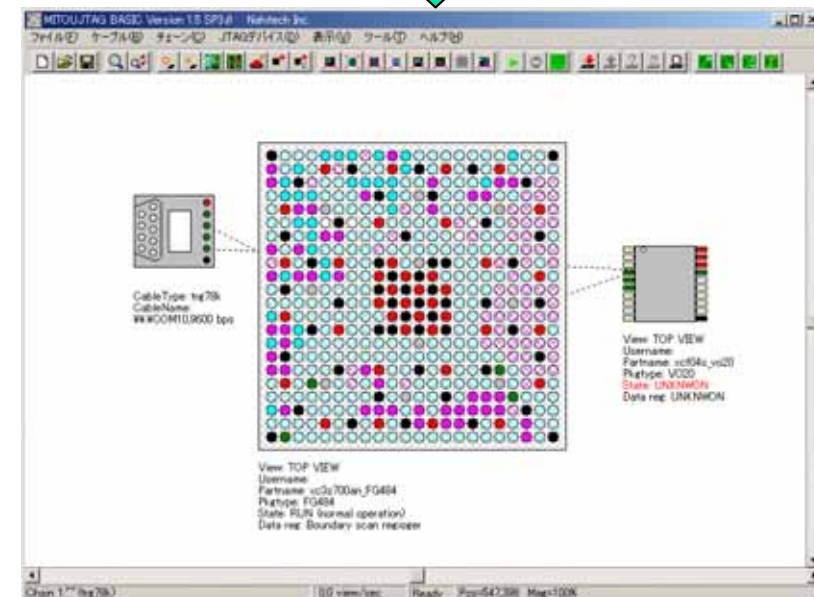
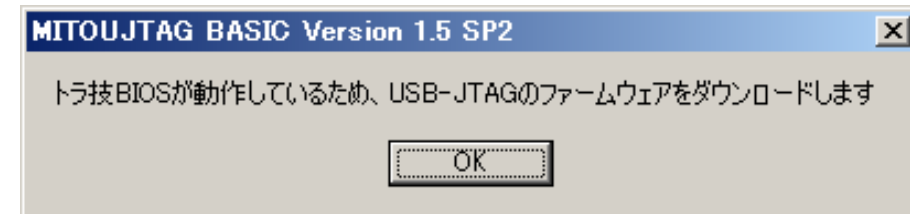
自社製ソフトに組み込んでみる

- MAX2付録基板や、Spartan3E付録基板に書き込みができた
- バウンダリスキャンもできた

```

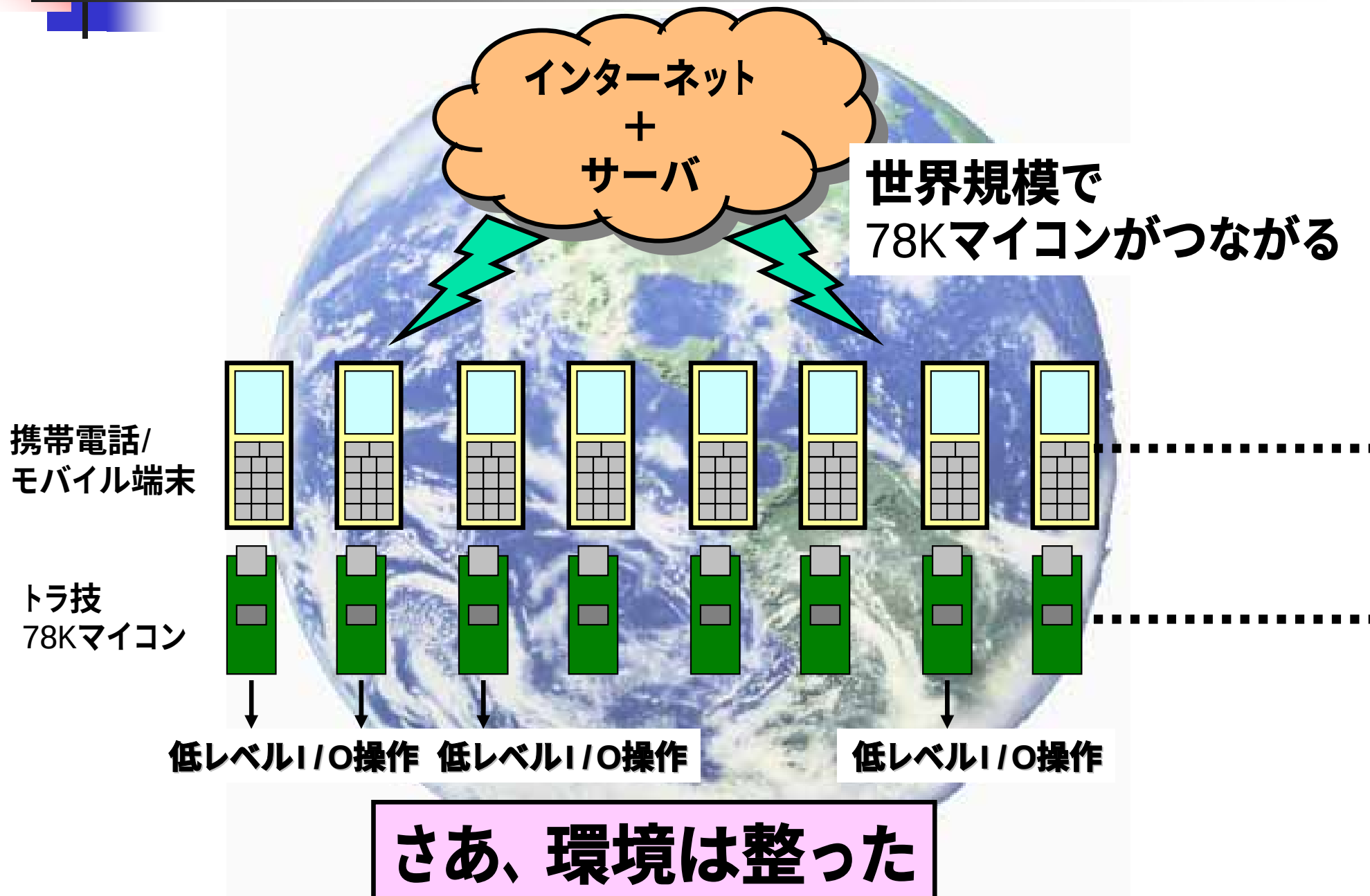
C:\%mdev%\libjtag\src\cable%78k-rs32c>mjisp -cable trg78k -bypass any -auto boarddemo_routed.bit
MITOUJTAG ISP V1.5 (C)Copyright 2008 TokushuDenshiKairo Inc.
TRG78K-JTAG cable open successfully.
自動検出を実行しています
1番目:xc3s700an_FG484が検出され、デバイス名を自動設定しました
2番目:xc3s700an_FG484が検出され、デバイス名を自動設定しました
Device position 1. Command 'bypass'. Partname 'xc3s700an_FG484'.
-> File '(null)'
Device position 2. Command 'auto'. Partname 'xc3s700an_FG484'.
-> File 'boarddemo_routed.bit'
note: Erase start
.....
note: Design name:boarddemo_routed.ncd Part:3s700afg484 2007/03/29 13:04:31
note: Write start
.....
note: Design name:boarddemo_routed.ncd Part:3s700afg484 2007/03/29 13:04:31
note: Verify start
.....
Elapsed time 474845 ms
C:\%mdev%\libjtag\src\cable%78k-rs32c>
  
```

コマンドライン版書き込みソフト



バウンダリスキャンソフト

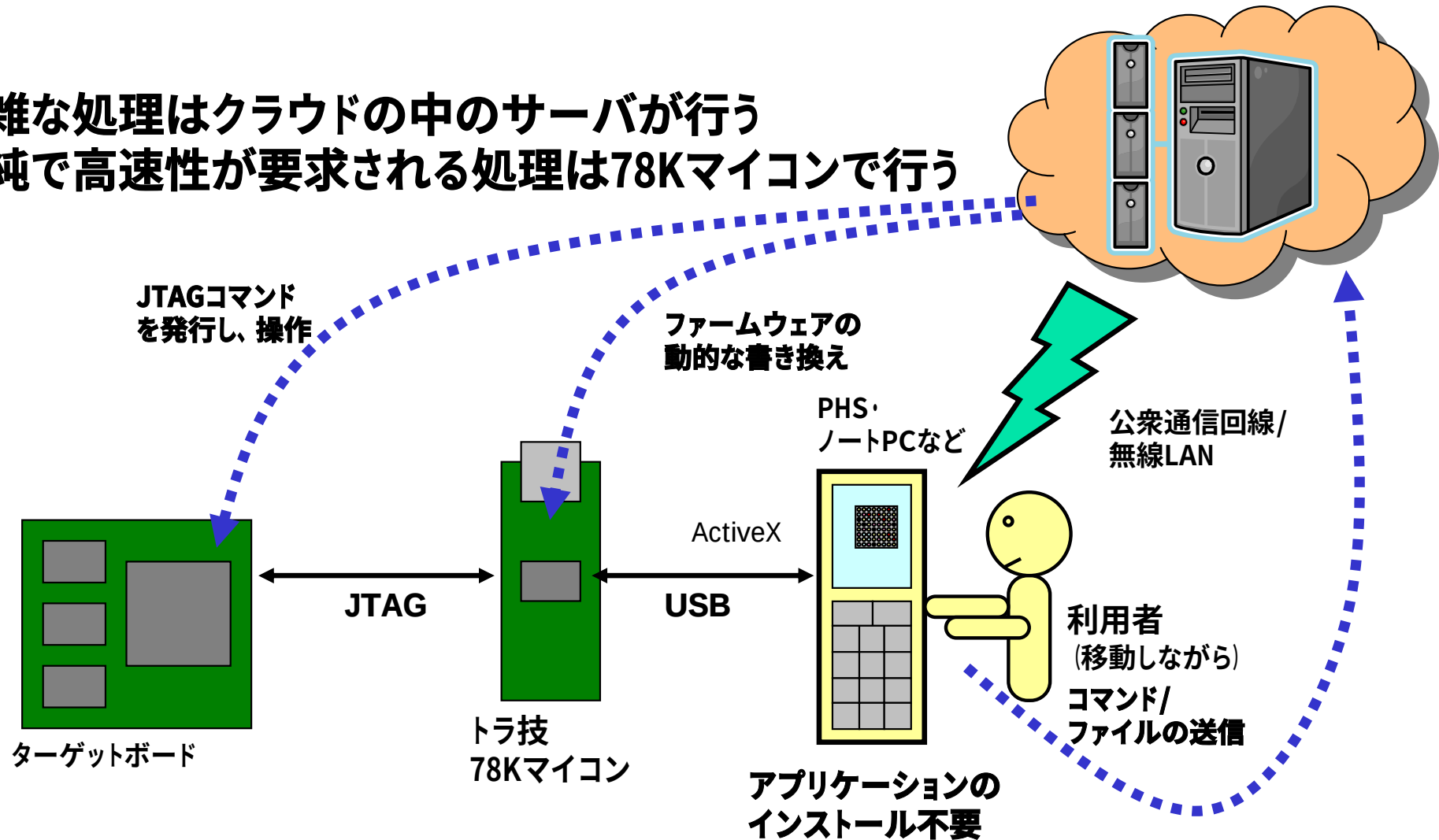
分散組み込みコンピューティングの夢





トラ技78K + Ajaxアプリケーションの例

- ・複雑な処理はクラウドの中のサーバが行う
- ・単純で高速性が要求される処理は78Kマイコンで行う



Ajax版JTAG書き込みツールを無償でWeb配布します

プログラムの入手方法



講演で使用したソフトウェアを無償で配布します

- ・シリアルポートアクセス用ActiveX
- ・各種JavaScript
- ・WindowsCE用ドライバ、DLL、サンプルアプリ
- ・USB-JTAGのクライアント側
- ・78Kによる78Kマイコン書き込み器
- ・78KマイコンとAjaxの連携サンプルなど

「トラ技78K基板を使おう」のページ
<http://www.nahitech.com/nahitafu/trg78k/index.html>
へアクセスして、秘密のフレーズに「ET見たよ」と記入し、
「ダウンロード」のボタンを押してください。

まとめ

- **トラ技78Kマイコンと、ActiveXとJavaScript、Ajaxを連携させることで、Z80で遊んだ頃の楽しかったマイコンが世界規模でつながる。**
- **78K基板の些細な諸問題（BIOS領域の書き込みにレガシーI/Oが必要、など）は解決済みである。**
- **独立行政法人情報処理推進機構のブース「E-27」にて展示とデモを行っております。どうぞお越してください。**
 - 紹介したローカル版JTAGツール (MITOUJTAG) の体験版も配布しています。
 - 78K-USB-JTAG対応パッチ、コマンドライン版書き込みツールもあります。