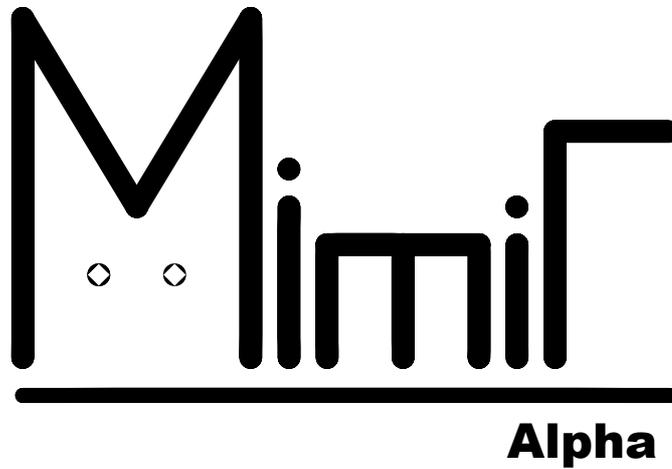


MiMic Getting started for LPCXpresso1769

2011.10.27 nyatla@nyatla.jp



本書は、MiMicを使った開発環境の導入手順書です。ゴールはWebブラウザからLPCXpresso1769のLED2を点灯させることです。手順を実施することで、MiMicシステムを使った開発環境を整える事が出来ます。

Rev	日付	
1	2011/10/29	新規作成
2		MiMicILの説明追加/画像追加
3	2011/12/11	mbed対応の追記、APIマニュアルのURL更新

目次

1 必要な物と知識、そして予算.....	3
2 MiMic.....	4
3 導入シナリオ.....	5
3.1 LPCXpresso1769 評価ボードと部品の入手.....	6
3.2 最低限のハードウェアの組み立てかた.....	7
3.3 MiMic ファームウェアの入手.....	9
3.4 ファームウェアの書き込み.....	10
3.5 MiMic のネットワーク設定.....	12
3.6 LED 点滅プログラムの実行.....	15
4 MiMic を使ったアプリケーション作り.....	16
5 MiMicIL.....	17
6 MiMic の遊び方(おまけ).....	18
7 文献.....	19
8 最新の情報について.....	19

1 必要な物と知識、そして予算

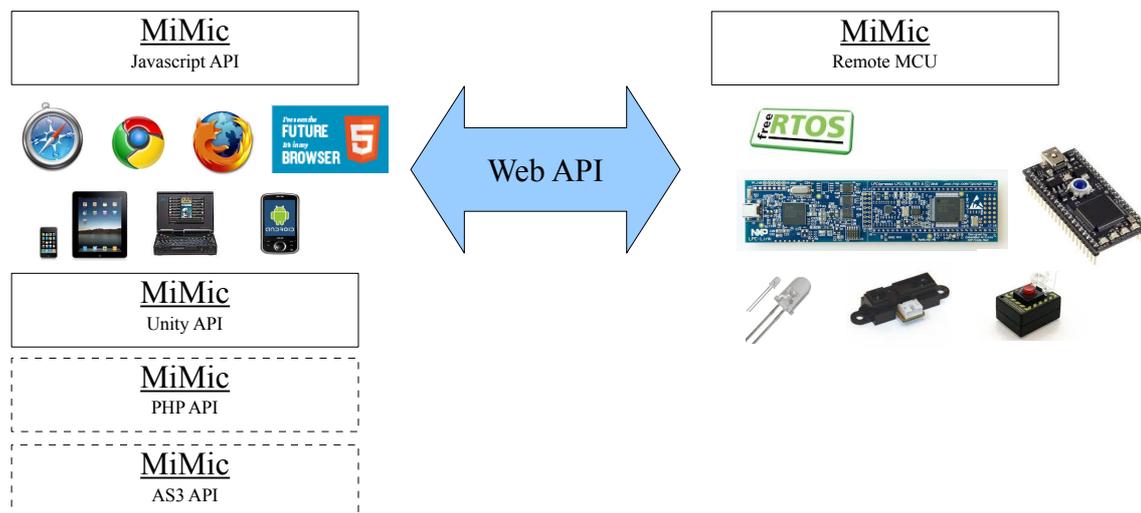
MiMic の開発環境を得るには、ソフトウェアに対する知識と、電子工作スキルが必要です。必要な知識とスキルは以下の通りです。

- 半田付けスキル
- LPCXpressoIDE の操作知識 (ファームを焼ければ OK)
- ホームルーターを設定する程度のネットワーク運用知識
- 予算(3000 円くらい)

また、MiMic を使った開発には、初歩の Javascript,HTML5 に関する知識が必要です。

2 MiMic

MiMic は、TCP/IP をインタフェースとして、WebAPI からハードウェアを制御する為のソフトウェアです。LPCXpresso1769/ mbed で動作するファームウェアとその SDK、ブラウザや他の Web プラットフォームで動作するスクリプトライブラリで構成されています。



MiMic のファームウェア(MiMicRemoteMCU)は、LPCXpresso1769/ mbed の持つ(割込み以外の)全ての機能を、ネットワークを通じてユーザインタフェースへ引き出します。アプリケーション開発者は、Web アプリケーションを開発するように、LPCXpresso1769 を使ったフィジカルコンピューティングを体験できます。

例えば、次のような Html を書いてブラウザで閲覧すると、LPCXpresso1769 の LED を制御することができます。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script src="MiMicCore.js"></script>
<script src="LPCXpresso1769.All.js"></script>
<script type="text/javascript">
function init() {
    var mcu=new LPCXpresso1769.Mcu("192.168.0.39");
    var pin=mcu.getPin(LPCXpresso1769.P0[22], "GPIO");
    pin.setOpt({dir:1, pin:{mode:1, od:0}});
    var i=0;// blink LED2 every 1 second.
    setInterval(function() {pin.setValue((i++)%2);},1000);
}
</script>
</head>
<body onload="init();">
<h1>LED flash</h1>
</body>
```

3 導入シナリオ

次の順番で説明します。

1. LPCXpresso1769 評価ボードと部品の入手
2. 最低限のハードウェアの組み立てかた
3. MiMic ファームウェアの入手
4. ファームウェアの書き込み
5. MiMic のネットワーク設定
6. LED 点滅プログラムの実行

3.1 LPCXpresso1769 評価ボードと部品の入手

MiMic を動作させる為に必要な、最低限の部品は以下の通りです。

1. LPCXpresso1769 評価ボード
2. パルストランス内蔵モジュラージャック(RJ-45)
3. タクトスイッチ (リセット用)



他に、これらを接続する為のブレッドボード※1 や、変換基板、配線などが必要です。(写真の部品は、マザーボードも含めて組み立てるときの部品です。) 部品は、秋月電子、マルツパーツ等で購入できます。

秋月電子 <http://akizukidenshi.com/catalog/default.aspx>

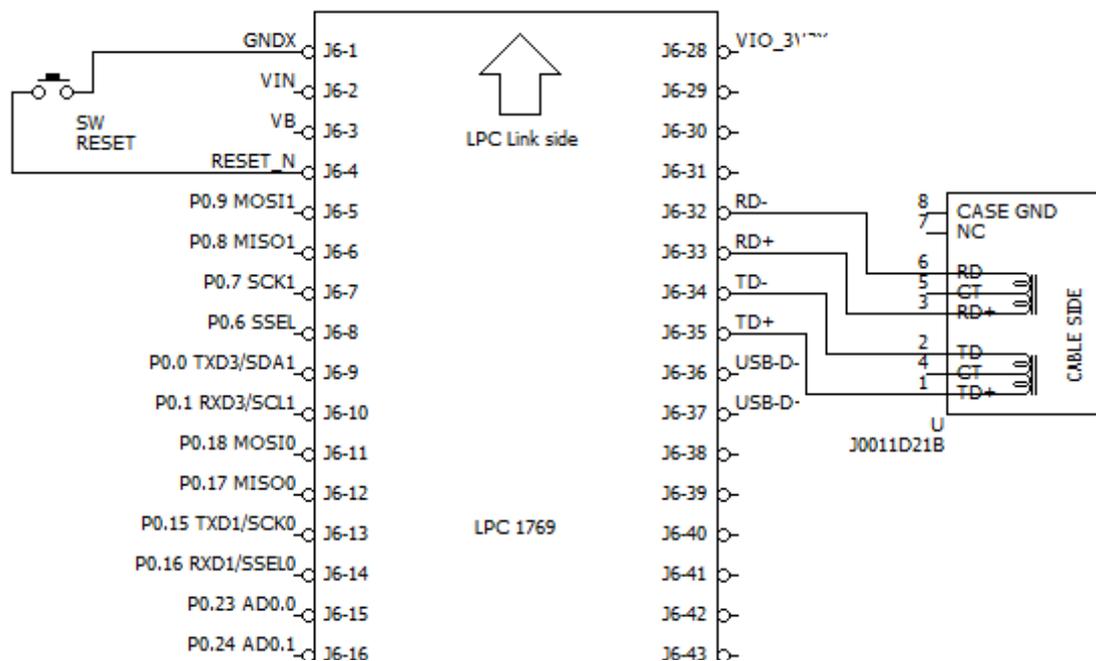
既にイーサネットポートを持つマザーボードをお持ちの方は、何も準備することなく、そのまま流用できます。

※1 ブレッドボードに実装するときには、ブレッドボードの長さに注意してください。LPCXpresso1769 評価基板は、27pin が直列に並んでいます。短いブレッドボードだと、長さが足りないことがありました。

3.2 最低限のハードウェアの組み立てかた

リセットスイッチ、LAN モジュラジャックを LPCXpresso1769 評価基板と接続します。

回路図は以下の通りです。

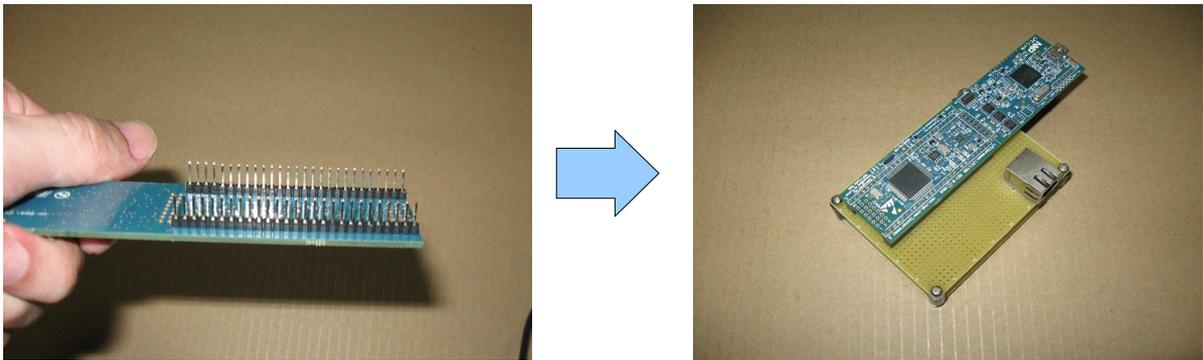


LAN コネクタのピンアサインは、モジュラジャックの規格書をよく読んでください。ブレッドボードに実装する場合は、変換基板などがあると便利です。

- Breakout Board for RJ45
<http://www.sparkfun.com/products/716>



組み立てはまず、LPCXpresso1769 評価ボードにピンを半田付けして、マザーボードや、ブレッドボードに取り付けられるようにします。



マザーボードを組み立てる場合は、組み立てて、部品を実装します。ブレッドボードの場合は、部品を装着して、回路図どおりに配線してください。

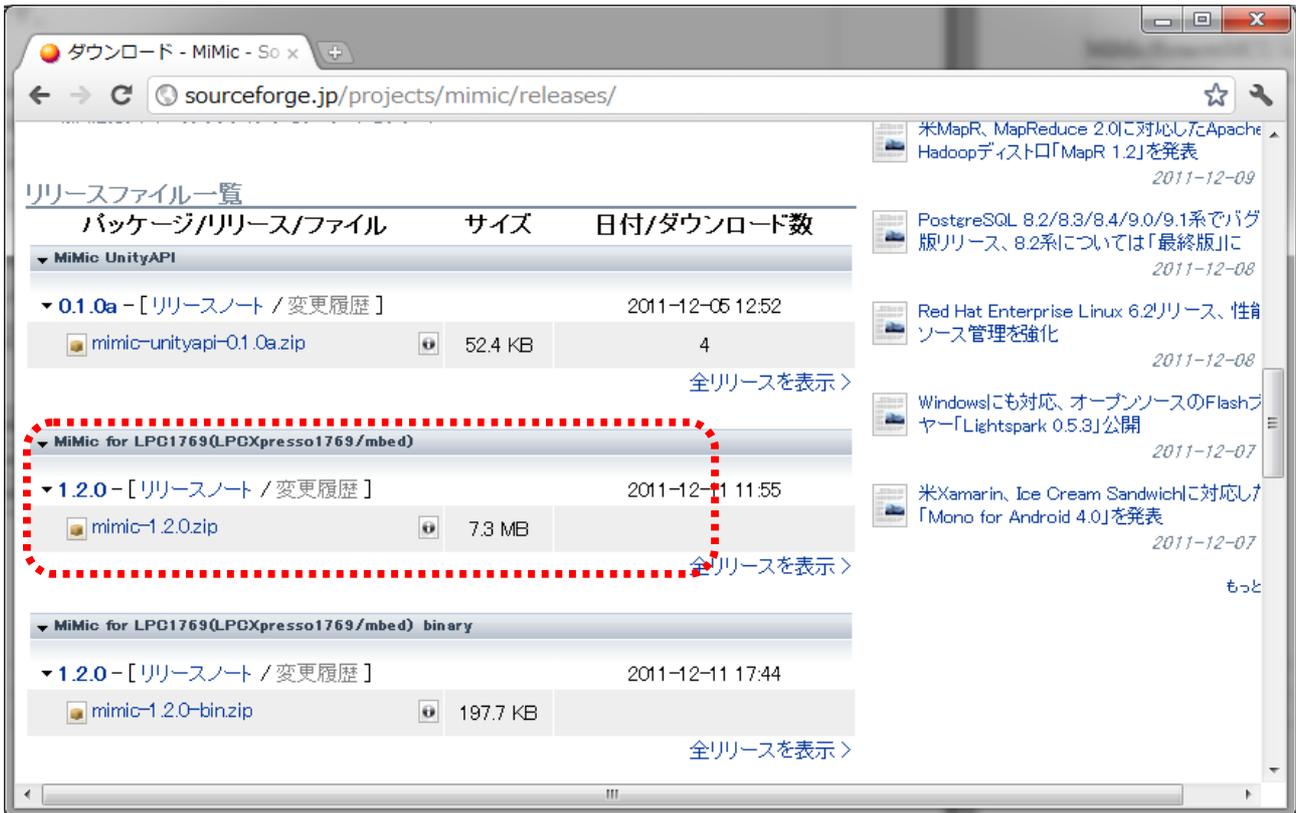
写真にあるシンプルなマザーボードでよい場合は、Web ページで詳しい作り方を紹介しています。(このマザーボードには、部品の増設がし難い、致命的な問題点があります…。)

- MiMic on NXP LPCXpresso 1769 評価ボード(ハードウェア編)
<http://nyatla.jp/mimic/wp/?p=59>

3.3 MiMic ファームウェアの入手

MiMic のファームウェアは、sourceforge からダウンロードできます。以下の URL を開いて、最新のバージョンをダウンロードしてください。

<http://sourceforge.jp/projects/mimic/releases/>



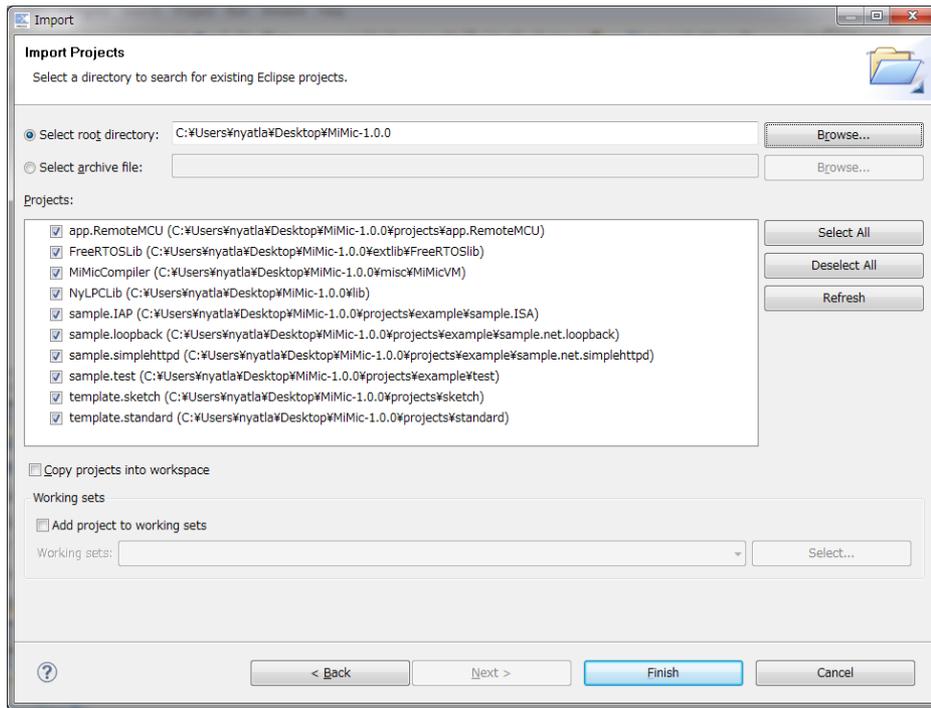
ダウンロードした zip ファイルは、適当な作業ディレクトリに展開して置いてください。

名前	更新日時
extlib	2011/10/29 0:38
lib	2011/10/29 0:38
misc	2011/10/29 0:38
projects	2011/10/29 0:38
usbser.inf	2011/08/30 16:35

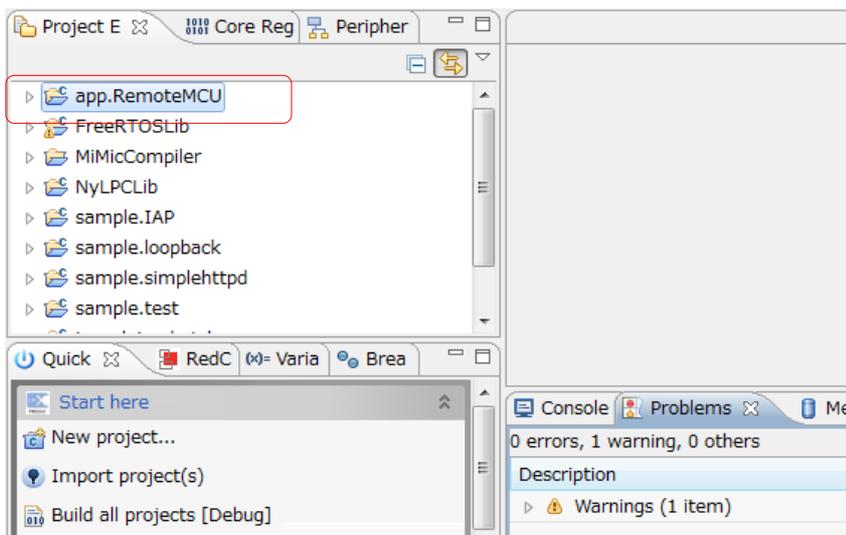
3.4 ファームウェアの書き込み

ソースコードからファームウェアをコンパイルして、LPCXpresso1769 へ書き込みます。
LPCXpresso1769 評価ボードと PC を、USB ケーブルで接続して置いてください。

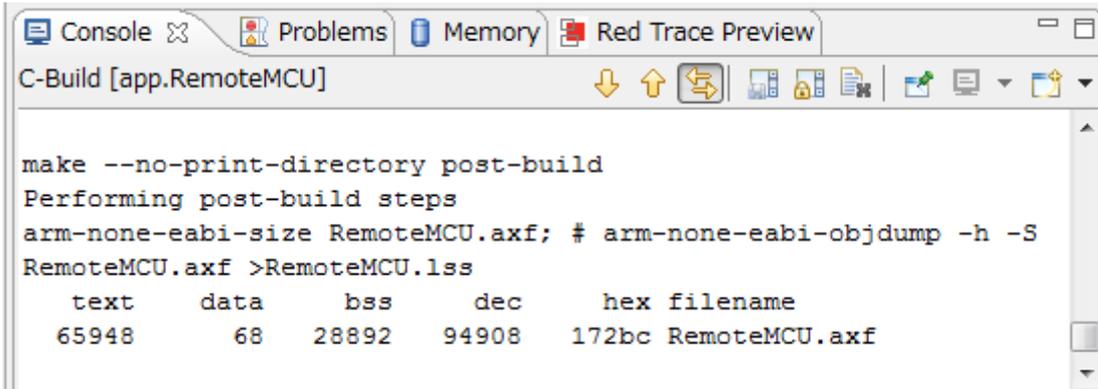
1. LPCXpressoIDE を起動し、展開したファイルのトップディレクトリを Import してください。



2. 次に、app.RemoteMCU のプロジェクトをビルドして、RemoteMCU.afx を作ります。



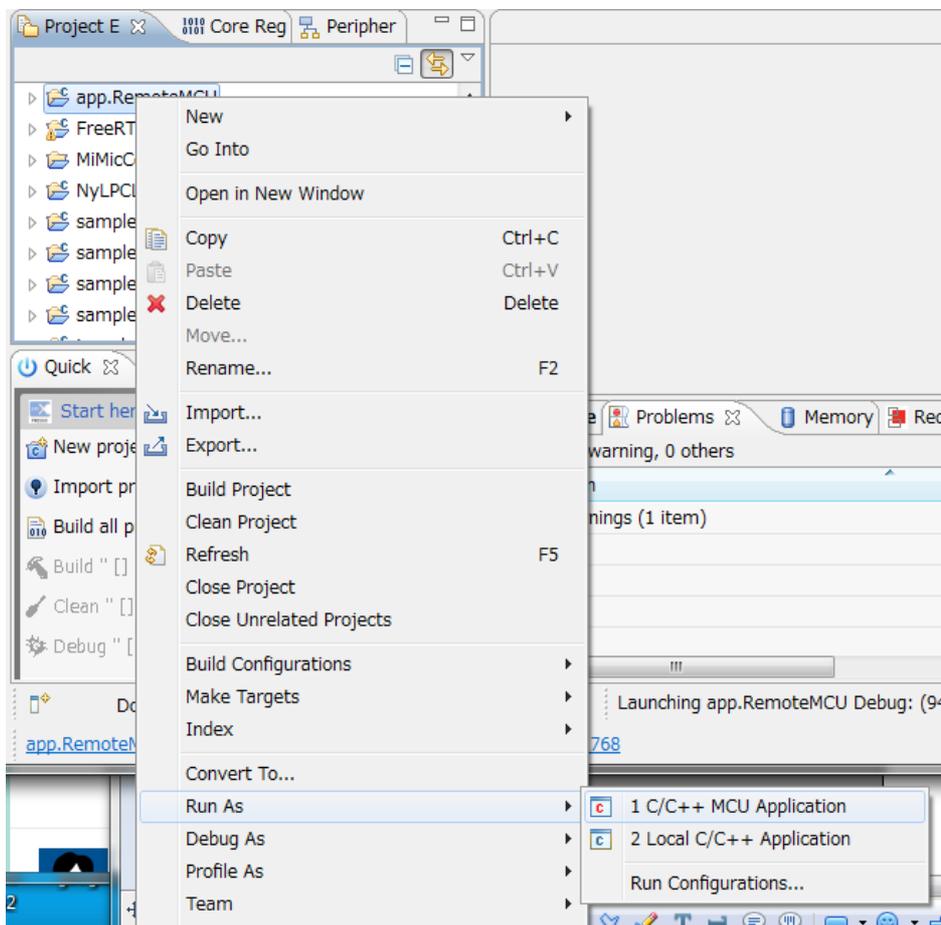
3. 問題がなければ、以下のように、ファームウェアが完成します。



```
C-Build [app.RemoteMCU]

make --no-print-directory post-build
Performing post-build steps
arm-none-eabi-size RemoteMCU.axf; # arm-none-eabi-objdump -h -S
RemoteMCU.axf >RemoteMCU.lss
  text    data    bss     dec     hex filename
 65948    68    28892   94908   172bc RemoteMCU.axf
```

4. 最後に、Run As でファームウェアを LPCXpresso に書き込んで、一度実行します。



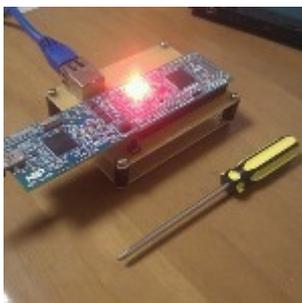
以上でファームウェアの書き込みは終了です。LPCXpressoIDE は終了してもかまいません。

LPCXpressoIDE のインストール方法、本手順の詳細な解説は、以下の URL でも説明しています。

- MiMic on NXP LPCXpresso 1769 評価ボード(開発環境編)
<http://nyatla.jp/mimic/wp/?p=77>
- MiMic のソースコードを LPCXpressoIDE へインポートする
<http://nyatla.jp/mimic/wp/?p=154>

3.5 MiMic のネットワーク設定

ファームウェアを LPCXpresso1769 に書き込んだら、イーサネットケーブルを LPCXpresso1769 に接続してください。数秒後に MiMicRemoteMCU が起動し、LAN アクセスを示す LED が点滅を始めます。



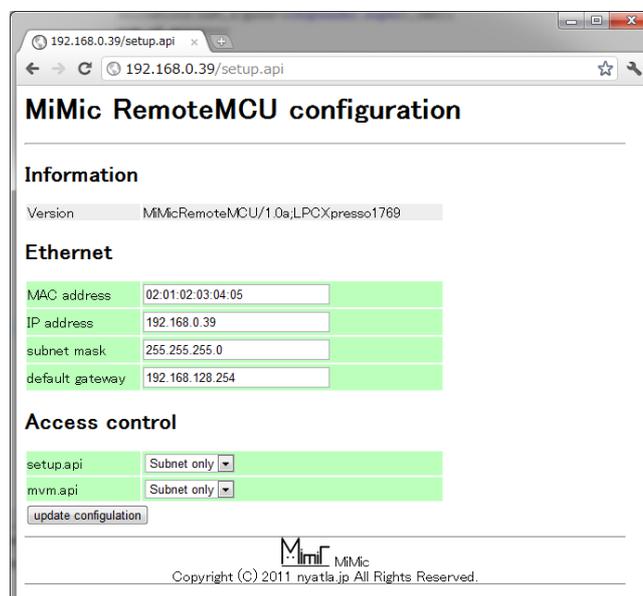
次に、MiMicRemoteMCU の設定ページにアクセスして、ネットワーク環境を設定します。ネットワーク環境の設定は、ホームルーターと同じように、Web ページから行います。

MiMicRemoteMCU の初期 IP アドレスは 192.168.0.39 です。設定用に、アクセスするコンピュータの IP アドレスを、192.168.0.1 等に設定します。

コンピュータの IP アドレスを設定したら、<http://192.168.0.39/> にアクセスしてください。以下の画面が表示されれば、MiMicRemoteMCU は絶好調に作動中です。



次に、MiMicRemoteMCU の IP アドレスを、希望する IP アドレスに変更します。Menu から、[RemoteMCU configuration](#) をクリックして、設定画面に進みます。



設定項目は以下の通りです。MiMic を 1 台だけネットワークに接続するなら、IP アドレスの設定だけで十分です。

- Information

MiMicRemoteMCU の情報を表示します。

Version	MiMicRemoteMCU のバージョン情報です。
---------	----------------------------

- Ethernet

イーサネット、TCP/IP に関する設定です。

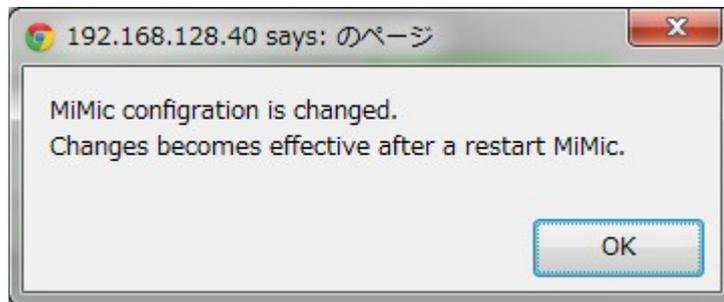
MAC address	Ethernet アドレスです。6 桁の 16 進数を区切りで指定します。複数の MiMicRemoteMCU サーバをネットワーク接続するときには、不要な LAN カードから取り外した番号などで置き換えてください。通常はそのまま大丈夫です。(?)
IP address	MiMicRemoteMCU サーバの IP アドレスです。区切りの 4 桁で指定します。
subnet mask	サブネットマスクです。
default gateway	デフォルトゲートウェイの IP アドレスです。

- Access control

アクセス許可の設定です。

setup.api	MiMic の設定画面(この画面)の公開範囲を指定します。Subnet(サブネット内のみ),All(全てのホスト)の選択です。
mvm.api	MiMicVM(操作インタフェイス)の公開範囲を指定します。

設定が完了したら、update configuration ボタンを押します。次のダイアログが表示されれば、MiMicRemoteMCU の設定更新は成功です。リセットボタンを押して LPCXpresso1769 を再起動するか、電源を入れなおすことで、設定が有効になります。



MiMicRemoteMCU には工場出荷時機能設定のスイッチ類はありませんので、万一設定内容を忘れてしまうと、ファームウェアの再書き込みを行わない限り回復できません。IP アドレスの設定は慎重に行ってください。

3.6 LED 点滅プログラムの実行

MiMicRemoteMCU が再起動したら、その IP アドレスで MiMicRemoteMCU が作動中であるか、念のため確認してください。(トップページにアクセスできれば作動中です。)

次に、ダウンロードしたパッケージの、misc/MiMicVM/api.js ディレクトリを開いてください。幾つかの HTML があると思います。メモ帳などで、led_blink.html を開いてください。

HTML の以下の部分を、MiMicRemoteMCU の IP アドレスに置き換えます。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script src="../../MiMicCore.js"></script>
<script src="../../LPCXpresso1769.All.js"></script>
<script type="text/javascript">
function init() {
    var mcu=new LPCXpresso1769.Mcu ("192.168.0.39");
    var pin=mcu.getPin(LPCXpresso1769.P0[22], "GPIO");
    pin.setOpt({dir:1, pin:{mode:1, od:0}});
    var i=0;
    setInterval(function() {pin.setValue((i++)%2);}, 1000);
}
</script>
</head>
<body onload="init();">
<h1>LED flash</h1>
</body>
```

書き換えたら保存して、Firefox,Chrome,Safari 等のブラウザで HTML を開いてください。LED が点滅を始めます。※1

これで、あなたのブラウザから、LPCXpresso1769 のほぼ全ての機能にアクセスする準備が整いました。MiMic によるフィジカルコンピューティングをお楽しみください。

※1.InternetExplorer は技術的な問題で対応し切れていません。

※2.プログラムを間違えると(特に低レベル API 操作時)、MiMicRemoteMCU は容易にフォールトします。動作しないときには、トップページにアクセスして、MiMicRemoteMCU が作動中かを確認してみてください。作動が確認できなければ、reset スイッチを押して再起動してください。

4 MiMicを使ったアプリケーション作り

MiMicを使ったアプリケーションは、ブラウザで実行する Html や Javascript で作ります。開発環境は、notepad でも、vi でも、emacs でも、好きなものを使用してください。

html には、MiMic のライブラリファイル2つを読み込まれるようにしてください。

ファイル名	機能
MiMicCore.js	MiMic javascript API の、低レベル API を定義します。 低レベル API は、MiMicRemoteMCU にアクセスするための API です。LPCXpresso1769 で動作する MiMicVM の操作手段を提供します。
LPCXpresso1769.All.js	LPCXpresso1769.*.js を全て連結したファイルです。MiMic javascript API の、高レベル API を定義します。 高レベル API は、低レベル API を使ったハードウェアドライバです。例えば、GPIO などを、簡単にコントロールすることが出来ます。

API の仕様については、ドキュメント MiMic_javascript_API_specification を参照してください。

<http://mimic.sourceforge.jp/doc/current/>

基本的には、次のように実装します。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script src="../MiMicCore.js"></script>
<script src="../LPCXpresso1769.All.js"></script>
<script type="text/javascript">
function init() {
    //LPCXpressoとの接続
    var mcu=new LPCXpresso1769.Mcu("192.168.0.39");
    //初期化処理
    var pin=mcu.getPin(LPCXpresso1769.P0[22], "GPIO");
    pin.setOpt({dir:1, pin:{mode:1, od:0}});
    var i=0;
    //メイン処理
    pin.setValue(1);
}
</script>
</head>
<body onload="init();">
<h1>LED flash</h1>
</body>
```

初期化処理までは、HTML の onload イベントで処理するのが良いでしょう。初期化が終わった後のシーケンスは自由です。例えば、ボタンの onclick イベントの中に pin.setValue(1)を書けば、ボタンを押すと LED が点灯するようになります。

5 MiMicIL

MiMicRemoteMCUとブラウザの間で取り交わされるVMバイトコードの開発には、開発ツールMiMicIL editorが便利です。これはMiMicで実行するバイトコードフラグメント(MiMicBC)を、中間言語MiMicILから生成し、試すことが出来るツールです。HTML5で作られています。MiMicIL editorは、InternetExplorer以外のブラウザで使うことが出来ます。



ネットワーク経由でLPCXpressoの現在のメモリ内容を読み出したり、書き込んだりすることが出来るので、ちょっとした動作テストなどにも便利です。例えば、メモリを読み出すには、次のようなコードを実行します。

```
MGET #0,#0x0 ;0x00000000 のメモリ内容をワークレジスタ 0 へ読み出す  
SPUT #0 ;ワークレジスタの内容をストリームへ出力する。  
EXIT ;終了  
.END ;セクション区切り
```

MiMicIL Editorは、パッケージの、misc/MiMicVM/tool/mimic_ileditor.htmlにあります。

6 MiMic の遊び方(おまけ)

MiMic は、それ単体では、何の役にも立ちません。ネットワークと接続し、ブラウザなどの操作インタフェース、制御装置と接続したときに、初めて能力を発揮します。

ネットワークに接続した MiMic には、色々な使い方があります。ブラウザを中心としたシステムの場合、ブラウザの持つマルチメディア機能、インターネットに接続された Web サービスと容易に連携することが出来ます。例えば、殺風景なマイコン制御画面を、HTML や CSS で装飾することが考えられます。LED の点灯に合わせてコンテンツを再生することで、電子デバイスを効果音で装飾することも、簡単に出来るでしょう。逆に、ブラウザで起こった出来事を、現実の電子デバイスに反映することも出来そうです。Web コンテンツに合わせて照明を調節したり、モーターを回転させる事も可能でしょう。WebAPI を通じて、オンラインコンテンツの情報をデバイスと共有することも考えられます。

MiMic を操作するプログラムは、ブラウザにダウンロードされて動作します。プログラムは、ローカルディスクでも、インターネットサーバでも、何処にあってもかまいません。

もし、貴方が自分のプログラムを共有したいなら、プログラムに MiMic の IP アドレスを設定するコードを追加して、ソースコードをそのまま Web サイトで公開してください。そのページにアクセスするだけで、同じハードウェアを持つユーザは、そのプログラムを体験できるでしょう。

MiMic はインターネットに直結することも出来ますが、残念ながら全くの無防備です。もし、貴方が不特定多数のユーザに MiMic を触らせるのであれば、MiMic とインターネットの間に、セキュリティを担保する為の代理サーバを設置すると良いでしょう。代理サーバとの MiMic との通信は HTTP ですから、PHP や JSP 等で、中継機能を実装することができます。

MiMic には、いろいろな応用例があります。代表的なケースについては、別資料で公開します。

7 文献

- MiMicVM.pdf
- MiMic_javascript_API_specification
<http://mimic.sourceforge.jp/doc/current/>
- lpcpresso.lpc1769.schematic.pdf
<http://ics.nxp.com/support/documents/microcontrollers/pdf/lpcpresso.lpc1769.schematic.pdf>
- UM10360.pdf
http://www.nxp.com/documents/user_manual/UM10360.pdf

8 最新の情報について

MiMic に関する最新の情報は、MiMic project のウェブサイト <http://nyatla.jp/mimic/wp/> で公開しています。デバイスドライバなども公開する予定ですので、チェックしてみてください。