

MiMicRemoteMCU version 1.4 Manual

2013/05/07 MiMicProject@nyatla.jp

<http://nyatla.jp/mimic/>

wm@nyatla.jp

この文章は、MiMicRemoteMCU 1.4.18 (0b1759b8d55f)以降の導入マニュアルです。

1.0.0	2013/05/07	新規作成
1.0.1	2013/05/08	誤字修正

目次

MiMicRemoteMCU version 1.4.5.....	2
1 しくみ.....	3
2 きのう.....	4
2.1 オンチップ Web コンテンツ.....	4
2.2 ファイルサーバ機能.....	4
2.3 コンフィギュレーション機能.....	4
3 うごかす.....	5
4 つかう.....	7
4.1 Onchip contents.....	7
4.1.1 Setup.....	7
4.1.2 Local FileSystem.....	8
4.1.3 JavaScript Editor.....	9
4.1.4 MiMicIL Editor.....	10
4.1.5 Memory Dump.....	11
4.2 Internet contents.....	11
4.3 Reference.....	11
5 おうよう.....	12
5.1 ブラウザ中心のマイコンアプリケーション開発.....	12
5.2 移植.....	13
6 ありそうな質問.....	13

MiMicRemoteMCU version 1.4

最近のブラウザって便利ですよ。HTML をちょこっと書けば、絵を描いたり音楽を再生したり、OpenGL やら物理演算やらなんでも出来てしまいます。

でもコンピュータの外の事、例えばその辺にある LED を点滅させたいとかちょっとセンサの値を読み出したいとか。そういったものには途端に不器用になります。

Web ページを書くような感覚でコンピュータの外にあるデバイスを操作できれば便利だと思いませんか？

MiMicRemoteMCU をインストールした mbed(LPCXpresso1769)を使えば、コンピュータの外にあるデバイスを、ブラウザから直接扱うことができます。

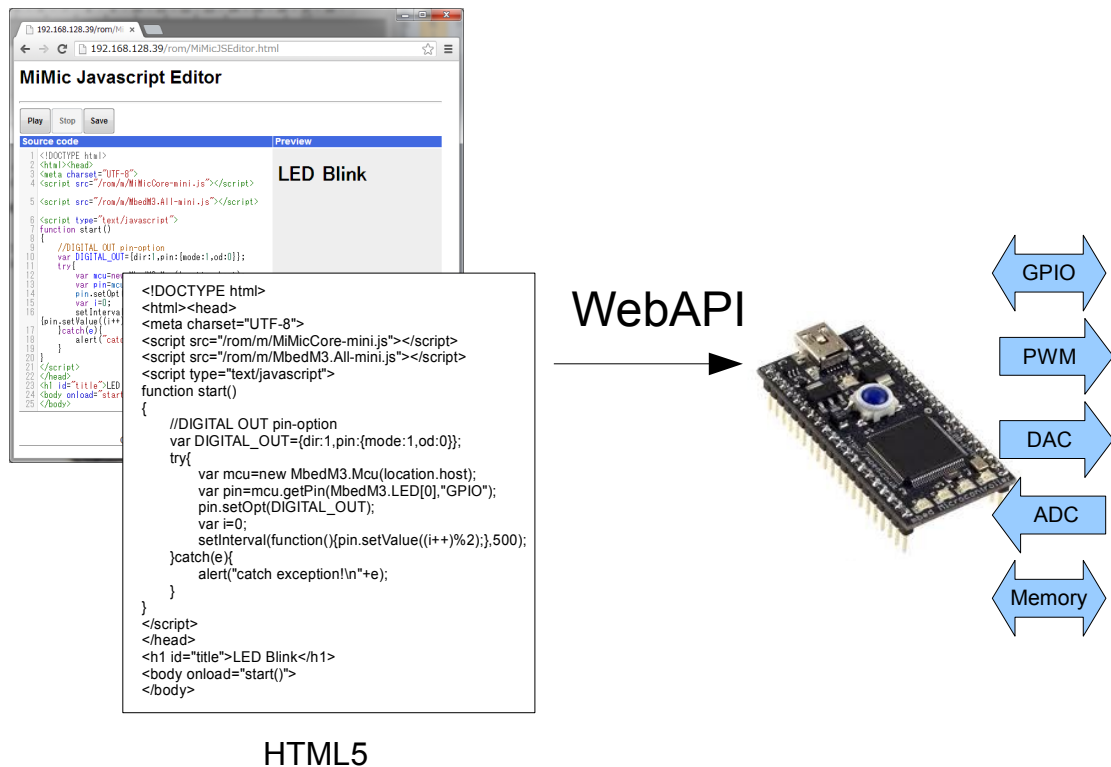


図 1. MiMicRemoteMCU

1 しくみ

MiMicRemoteMCU は mbed の MCU を管理する小さな Web サーバです。Web サーバには独自のバイトコードを解釈して MCU を操作する WebAPI があります。そこに HTTP リクエストで命令を送信することで、MCU を操作することができます。

WebAPI に送信するバイトコードはアスキーコードの羅列です。バイトコードは MCU の全てを制御できますが、良く使う機能については判りやすい JavascriptAPI が準備されています。

この2つの仕組みにより、MiMicRemoteMCU は Web アプリケーションを実装する感覚で mbed を制御することができます。

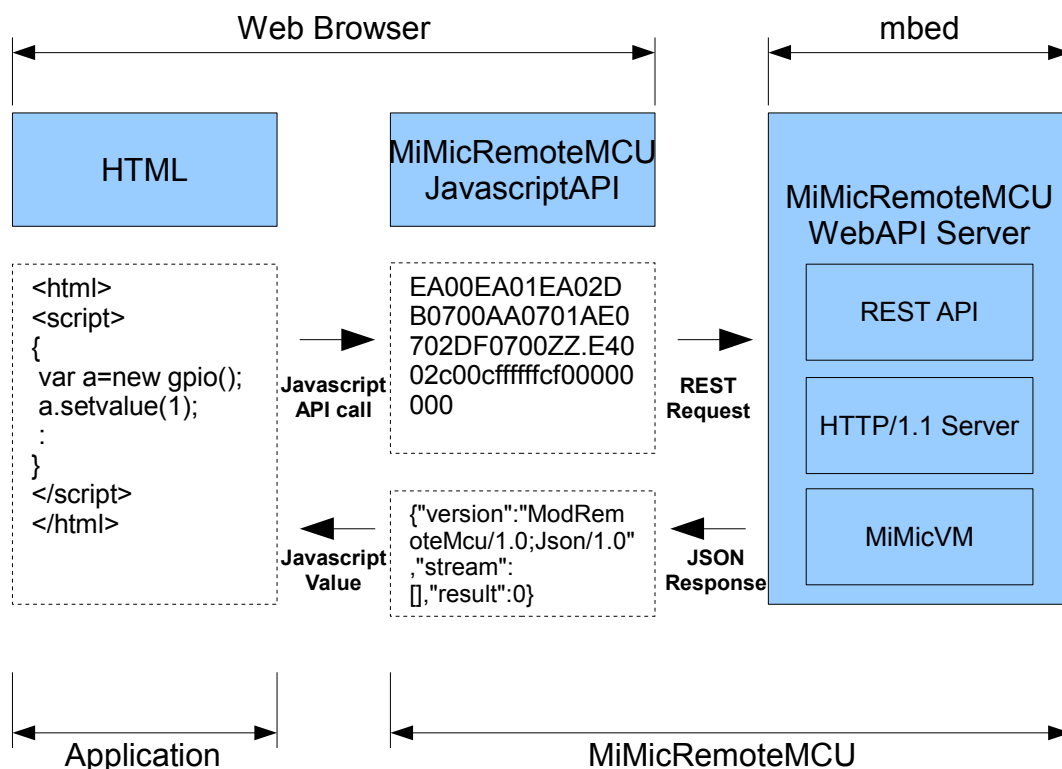
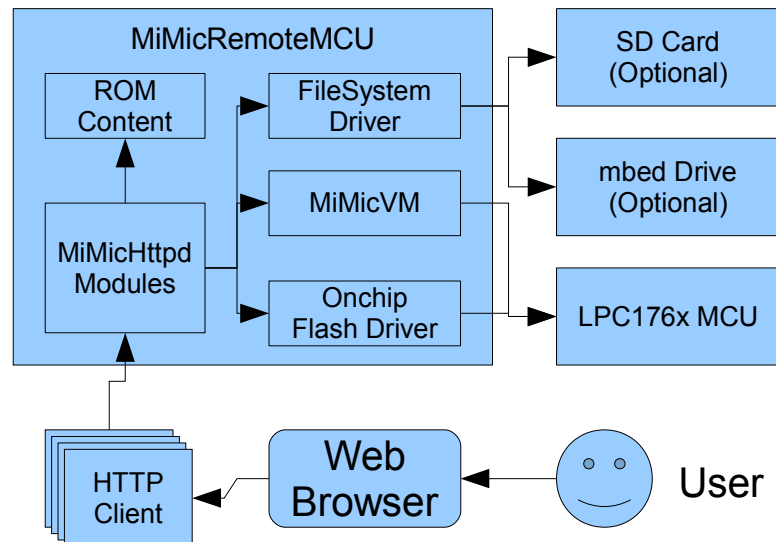


図 2 MiMicRemoteMCU の役割分担

2 きのう

MiMicRemoteMCU の機能を紹介します。対外的な機能は MiMicRemoteMCU の ROMContents に収録されており、内蔵 Web サーバを通じて Web ブラウザから提供します。この Web サーバは最大 4 接続までの接続を、平均 1.6Mbps で提供することができます。



2.1 オンチップ Web コンテンツ

オンチップの Web サービスには、MiMicIL 開発環境、コンテンツ開発環境を内蔵しています。また、MCU のメモリをブラウズするためのメモリビューアがあります。これらは全て ROMContents に収録された HTML コンテンツです。個々の機能は”つかう”のセクションで解説します。

2.2 ファイルサーバ機能

MiMicRemoteMCU は、mbed の LocalFileSystem、SD カードに格納したファイルを公開することができます。また、プログラム ROM に書き込んだファイルイメージを公開することができます。この機能は MiMicHttpd のモジュールにより提供されます。

2.3 コンフィギュレーション機能

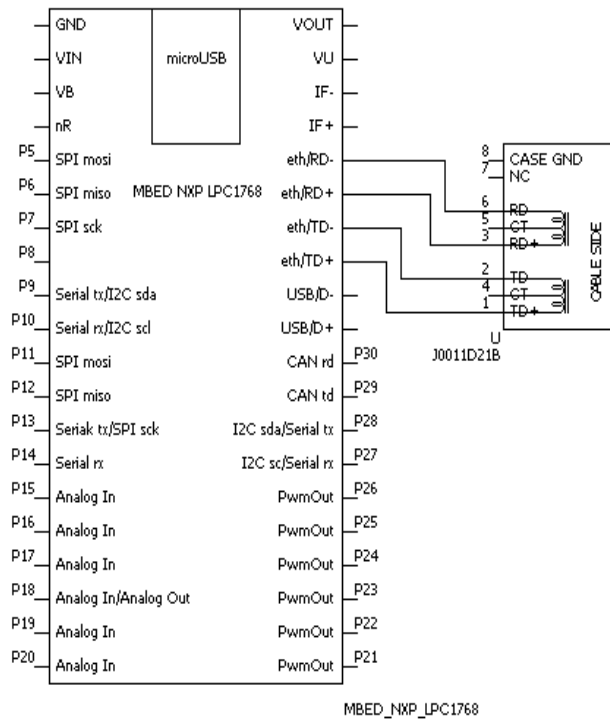
MiMicRemoteMCU は、ローカルファイルのほかにプログラムメモリにネットワーク設定を保存できます。ホームルーターの設定画面のようなものです。また、設定ファイルにより、設定を変更することも出来ます。この機能は MiMicHttpd のモジュールと OnchipFlash ドライバにより提供されます。

設定値は、初期値→オンチップ設定値→設定ファイルの順に上書きされます。例えば設定ファイルがあると、その設定で上書きされてしまいます。

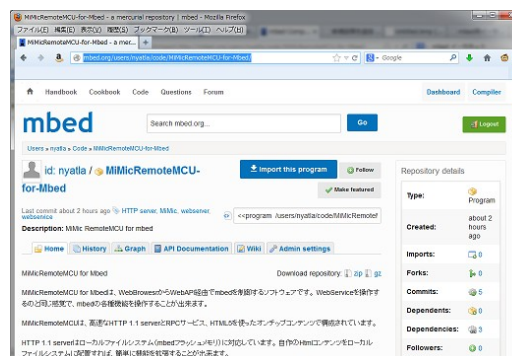
3 うごかす

MiMicRemoteMCU を動作させる手順を説明します。

1. MiMicRemoteMCU はイーサネットに接続して使います。最低限のハードウェア準備として、イーサネットコネクタをベースボードに接続する必要があります。以下のように配線をしてください。ベースボードが有る場合はそちらを利用すると便利です。



2. mbedCompiler で MiMicRemoteMCU をコンパイルして bin ファイルを得てください。
<http://mbed.org/users/nyatla/code/MiMicRemoteMCU-for-Mbed/>



3. MiMicRemoteMCU.bin を ROM へ書き込んでください。

4. ネットワーク設定ファイルを作成します。mimic.cfg の名前で mbed ドライブのルートディレクトリにおいてください。設定できる項目は ipaddr, netmask, gateway, macaddr の 4 項目です

mimic.cfg の例

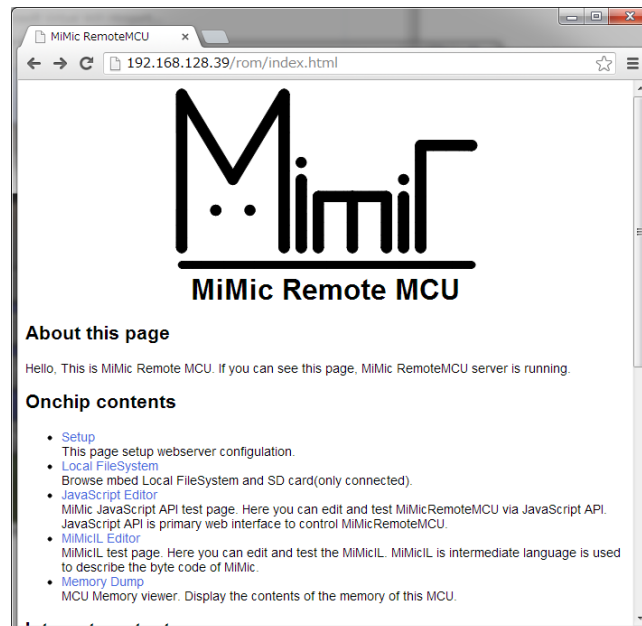
```
ipaddr=192.168.0.40
netmask=255.255.255.0
gateway=192.168.0.254
macaddr=02:01:02:03:04:05
```

各項目は省略が出来ます。省略時には、オンチップの設定値がそのまま引き継がれます。また、設定ファイルが無い場合は、オンチップの設定値が指定されます。

オンチップ設定値の初期値

```
ipaddr=192.168.0.39
netmask=255.255.255.0
gateway=192.168.0.254
macaddr=02:01:02:03:04:05
```

5. MiMicRemoteMCU にブラウザから接続してください (Internet Explorer は使わないでください。) 次の画面が閲覧できれば、MiMicRemoteMCU は起動しています。とりあえず動かしてみたい方は、“つかう”の”4.1.3. JavaScript Editor”まで読み飛ばしてください。



4 つかう

MiMicRemoteMCU のトップページには幾つかのリンクがあります。それぞれのカテゴリについて説明します。

4.1 Onchip contents

MCU のフラッシュメモリにあるコンテンツです。

4.1.1 Setup

MiMicRemoteMCU の動作設定の閲覧/設定が出来ます。オンチップコンフィギュレーションはここから操作します。ネットワーク設定ファイルがある場合は、そちらが優先して反映されます。

Current configuration	
Version	ModMiMicSetting/1.2;MiMic/1.4.5;mbed
LAN PHY	DP83848C
CDR type	undefined
MAC address	00:02:f7:f0:a4:0b
IP address	192.168.128.39
Net Mask	255.255.255.0
Default gateway	192.168.128.254

If local setting file available, current information might be different with onchip-setting. To fully enable on-chip setting, please remove the setting file "mimic.dig" in mbed local file system or SDcard file system.

Ethernet	
MAC address	00:02:f7:f0:a4:0b
IP address	192.168.0.39
Subnet mask	255.255.255.0
Default gateway	192.168.0.254

Security(unsupported)

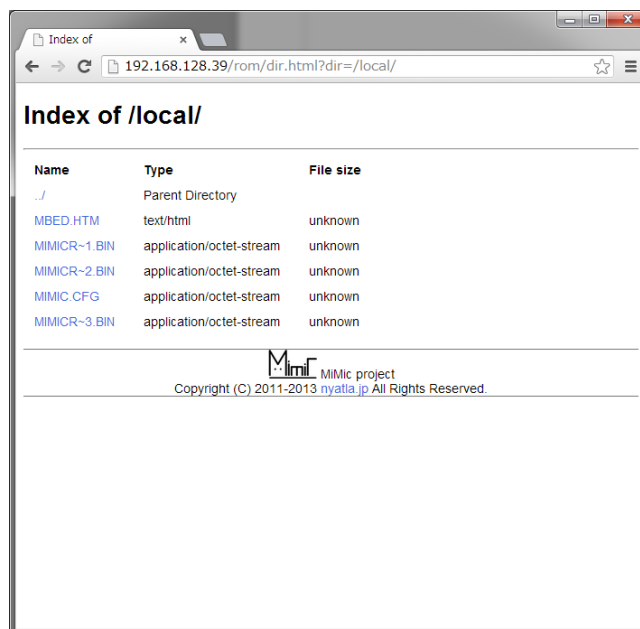
CurrentConfiguration には、現在の設定状況が表示されます。

Ethernet の項目には、現在のオンチップコンフィギュレーションの値が表示されます。この値は編集可能です。編集後に update ボタンを押して mbed をリセットすることで有効になります。

Security 項目は現在使用できません。

4.1.2 Local FileSystem

SD カード/mbed LocalFileSystem にあるファイルを閲覧できます。ハードウェアが存在しない場合には閲覧できません。

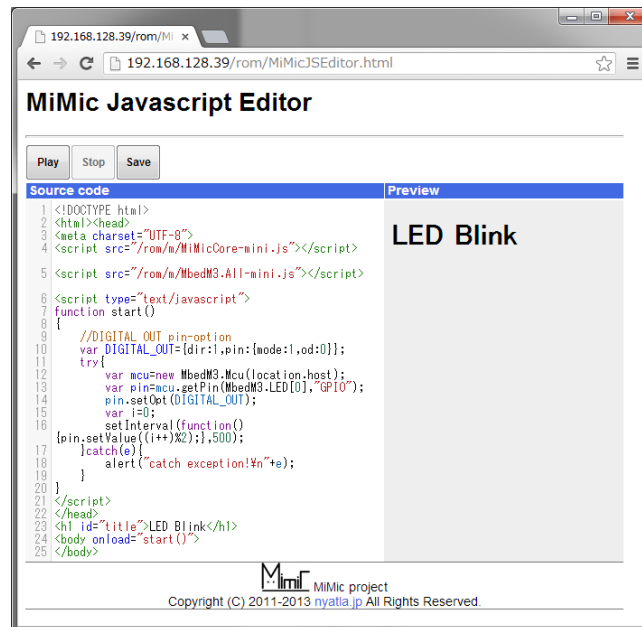


SD カードは動作中に取り外さないようにしてください。動作が不安定になることがあります。また、LocalFileSystem のサブディレクトリは閲覧できません。

ファイル名は、SD カードの場合は 256 文字、LocalFileSystem の場合は 8:3 形式の 12 文字です。日本語ファイル名はどちらの場合も使えません。(要対応)

4.1.3 JavaScript Editor

MiMicRemoteMCU の JavascriptAPI の実験コンテンツです。ブラウザから直接 mbed を制御する事が体験できます。起動直後は LED の点滅プログラムがプリセットされていますので、まずは Play ボタンを押してみてください。



SourceCode はリアルタイムに右側の Preview へ反映されます。Play ボタンを押すと、編集中のスクリプトを実行することができます。

実行中は編集が出来ませんので、Stop ボタンを押してスクリプトを停止してください。

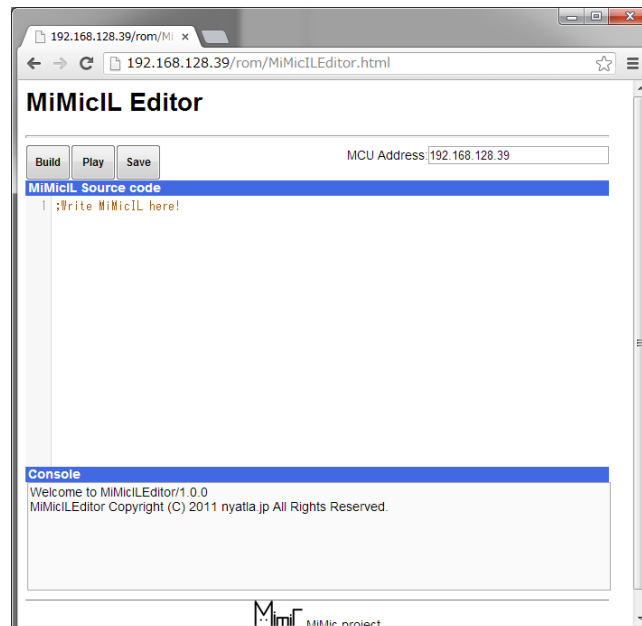
Preview は<script>タグだけは反映しません。全て無視されます。そのため、<script>タグを使用したコンテンツは正確なプレビューが出来ません。実行時には有効になります。

Save ボタンは Source code の内容を、“コンピュータの”ローカルドライブへ保存できます。mbedドライブへ保存すれば、MiMic からも閲覧が出来ます。

このエディタはプレビュー可能な HTML エディターです。MiMicRemoteMCU のコンテンツは HTML で出来ているので、一般的な Web ページのようにメモ帳や他の IDE などでも編集できます。大規模なコードを作るときには、使い慣れた編集環境を利用できます。JavaScript Editor を使わずにコンテンツを作る方法の解説は、[Twitter@nyatla](#) に催促してください。

4.1.4 MiMicIL Editor

バイトコードに相当する、MiMicIL の編集と実行が出来ます。上級者向けです。MiMicVM のニーモニックからバイトコードを生成し、実行テストを行うことが出来ます。レジスタ値による MCU の挙動変化を調べたいときに利用できます。



Build ボタンは MiMicIL からバイトコードを生成します。例えば以下のコードを Build すると、コンソールにその結果が得られます。

```
ld #0,1
ld #1,2
add #0,#1
sput #0
.end
```

```
MiMicILEditor/1.0.0;MiMicILAsm/1.1.1
2013 年 5 月 8 日 0:12:59
input text size=60
Build success.

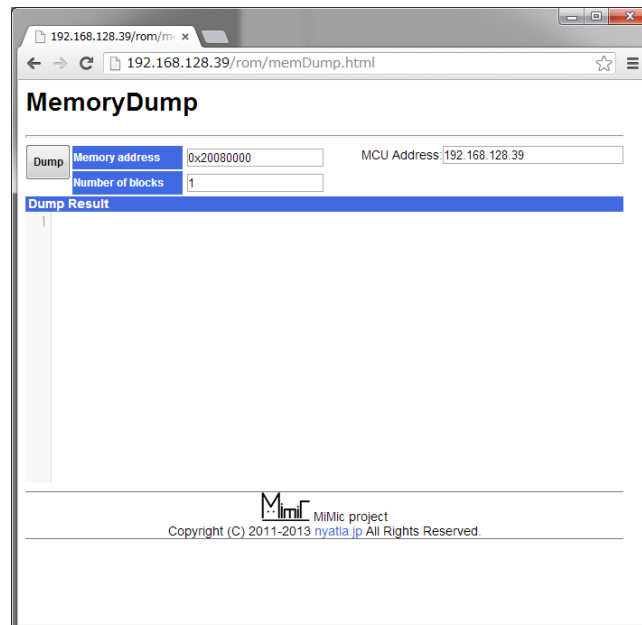
MiMicBC:
FB0000000001FB0100000002CA0001EE00.E
Finish.
```

Play ボタンを押すと、実際にバイトコードを MiMicRemoteMCU で実行することが出来ます。上記のコードを実行すると、以下の結果が得られます。

```
MiMicBC:
FB0000000001FB0100000002CA0001EE00.E
Connect to 192.168.128.39...
Request success!
Status:0
Stream:0x00000003
Disconnected from MiMicRemoteMCU.
Finish.
```

4.1.5 Memory Dump

MCU のメモリ内容を読み出して表示することが出来ます。メモリ保護機能は無いので、読み出し不能なアドレスにアクセスすると停止します。(停止した場合は mbed をリセットしてください。)



4.2 Internet contents

ソースコードやドキュメントへのリンクです。

また、オンラインで MiMic スクリプトを共有する js.doit へのリンクがあります。

4.3 Reference

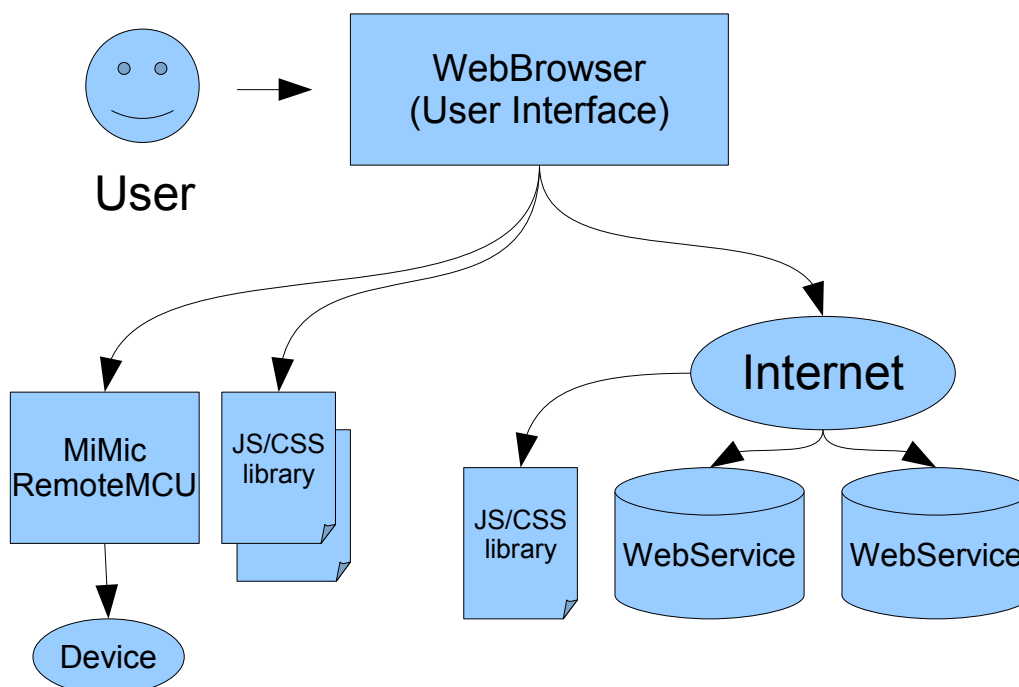
MiMic システムを構成するために使用したライブラリや開発環境へのリンクです。

5 おうよう

5.1 ブラウザ中心のマイコンアプリケーション開発

ブラウザから見た MiMicRemoteMCU は、単なる Web サーバに過ぎません。アプリケーションの中心はあくまで Web ブラウザにあります。

ブラウザ中心のシステムを設計することで、従来のマイコン単体アプリケーションでは難しかった外部 Web サービスやマルチメディアシステムとの連携を容易に実装できます。さらにブラウザからアクセスできる全ての場所にあるオンラインリソースを活用することが出来ます。



MiMic のネットワークポロジ

例えば...

1. カッコイイボタンを表示する Javascript ライブラリを使用すれば、カッコイイ制御画面を持つ操作アプリケーションが作れます。
2. センサ取得値を他の外部 Web サービスに送れば、センサを使った外部 Web サービスの操作が出来ます。
3. Web サービスから取得したデータを、MiMic に接続した外部デバイスに送ることで、外部 Web サービスからデバイスの操作が出来ます。
4. 複数台の MiMicRemoteMCU を1つの Web 画面から操作できます。

5.2 移植

MiMicRemoteMCU は WebAPI で動作する為、HTTP クライアントを搭載した機器、プラットフォームであれば、制御スクリプトのみを移植すれば問題なく動作します。

制御スクリプトは MiMic のバイトコードを合成する文字列処理関数と定義値の集合体です。移植に時間はややかかりますが、難易度は高くないと思います。

現在は実験的に Unity や PHP に移植をしています。

6 ありそうな質問

- 今までのマイコン開発環境と何が違うの？
一般的なマイコン開発環境は、完成したプログラムをマイコンに書き込み、単体で機能が完結するものが殆どです。MiMic は、外部のユーザインタフェース(ブラウザ)を中心としてシステムを実装し、複数の機器が接続された状態で機能を構成するところに違いがあります。
- mbed の RPC と何が違うの？
機能的にはほぼ同じですが、MiMic の場合はユーザインタフェースと制御手順をファームウェアに実装しないところに違いがあります。MiMic の場合、MiMicVM によって MCU のリソースが操作可能であり、コンテンツホスティングと操作 API が分離している為、ファームウェアを書き換える必要が殆ど無く、コンテンツの変更のみで機能が実現できます。
- LPCXpresso1769 でも動くの？
はい。MiMic はプラットフォームの自動認識ができるのでそのまま動きます……が、ROM サイズが大きいのので無償の LPCXpressoIDE で書きこむ方法がありません。512ドル払うか、別の書き込み方法を探す必要があります。なお、LPCXpresso は MacAddress が無いので Onchip コンフィギュレーションの値が使われます。また、LocalFileSystem を利用することが出来ません。
- 単独で動かない？
現在の MiMicRemoteMCU には、単独動作の機能がありません。かならずブラウザなどの”制御 UI”とセットで動きます。ファームウェアのソースコードを公開していますのでタスクを実装すれば単独でも動かすことはできます。
- 複数の MiMic を1つのネットワークに接続できる？
IP アドレスの設定は必要になりますが、できます。
- インターネットに直接接続しても良い？
技術的には問題ありませんが、セキュリティ的には大問題です。MiMic は MCU の全てのリソースをネットワーク越しに公開します。その気になればオンラインアップデートなども出来てしまいますので、実験目的以外では公開しないことをお勧めします。今後セキュリティを強化するモジュールの構想もありますが、もし必要でしたら相談に乗ります。
- ライセンスは？
LGPLv3 ですが、様子をみて MIT か Apache2 ライセンスに変更する予定です。変更時期は、FreeRTOS 関連の LAN ドライバモジュールを差し替え後を予定しています。
- 商用利用していい？
ライセンスさえ守ってもらえれば構いませんが、相談してくれると嬉しいなと。
- 何で作ったの？ Arduino でいいじゃん。
ブラウザは世界で最も普及した UI であり、HTTP(TCP/IP)は世界最大の通信網です。この2つだけを使ってマイコン開発が出来れば、それは素晴らしいことです。特定の開発アプリ使って C 言語で UI 実装とか、USB やシリアル接続で通信とか面倒ですよ。