



KYMAは、オブジェクト指向言語 SmallTalk によって開発された Digital Audio Workstation です。

SmallTalk は Java 等と同様に Virtual Machine を介して各種OS上で動作するため、Mac/PC 間のマルチプラットフォームを実現しています。 KYMA はアプリケーション単体では機能せず、PCに IEEE1394 ケーブルで接続したDSPユニット (Pacaranaと呼ばれるblackbox) で音響処理を行います。

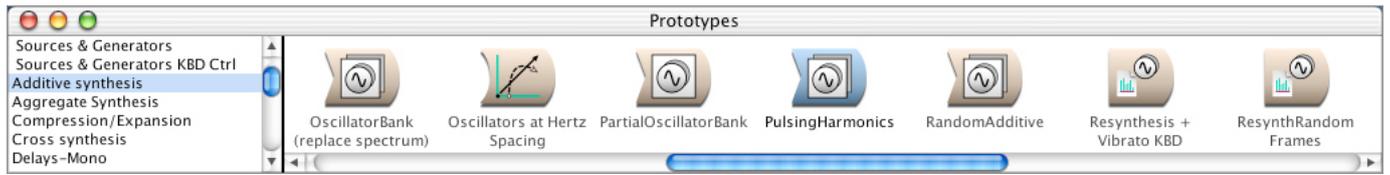


システム構成は、プログラムを走らせるPCと前述した外部接続のDSPユニット、及び USB/IEEE1394 ケーブルでDSPユニットとチェインされたAD/DAを組合せたものになります。入出力数はAD/DAの仕様に準拠しますが、扱える音声チャンネルは最大で8chまでとなっています。

アプリケーションを起動すると、開発者である Calra 女史の声「KYMA」と共にワークスペースとブラウザ、タイムライン、プロトタイプなど複数のウィンドウが表示されます。基本的なプログラムの作成は、Untitled と表示されているワークスペース Sound File Window で行います。

KYMA ではプログラムのことを Sound と呼称しますが、これが通常の「音」という意味と混同することがあるので注意が必要です。 KYMA には予めプロトタイプと呼ばれる Sound の雛形が用意されていて、ユーザーはその中から必要な部品 (object=Sound) を選択して音響システムを組み立てていきます。 Sound は独特なデザインアイコンで表現されていて、これは Symbolic Sound Co. という社名の由来になっています。

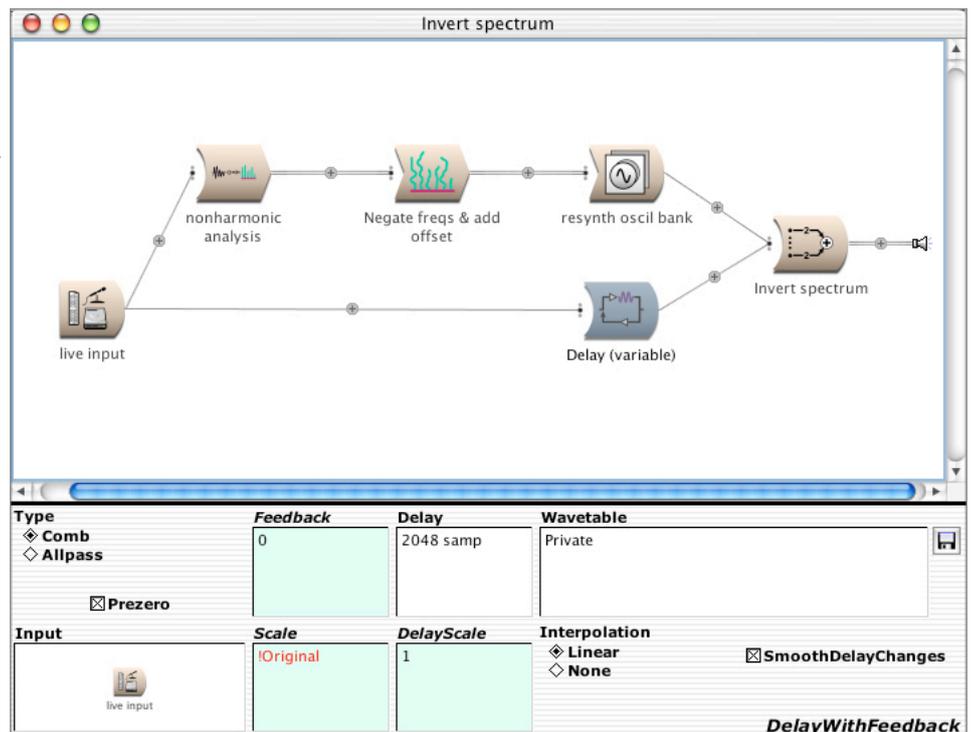
Prototype window には Delay/Filter/Vocoder 等の、普段慣れ親しんだ「エフェクター」から算術演算素子まで、多岐に渡る Sound が用意されています。



Sound の多くは即単体で利用が可能で、直感的に回路を構築することができます。Sound 単体の機能を実感したい場合は、Prototype Window 中のアイコンをクリックすると対象の Sound が立ち上がり、実際の用法を試すことができます。

編集は、Prototype Window 上にある Sound を Sound Editor Window に Drug&Drop して行います。任意の Sound を Drug&Drop してその Sound を展開すると、Sound と出力側に線で結ばれた小さなスピーカが現れます。

Sound を新たに付け加える場合は挿入位置の直前にある Sound アイコンの右側のエッジに、新たな Sound を Drug&Drop します。Sound 同士を重ねた場合は、古い Sound が新たに待ち込まれた Sound に置換されるので、注意が必要です。最終的に出来る Sound 群のトータルなイメージは Tree 状になります。



Sound Editor Window 上で選択した Sound をダブルクリックすると、その Sound の Parameter Window が表示されます。KYMA には Virtual Control Surface という機能があり、パソコンの画面上に表示されるスイッチやフェーダーを使ってパラメーターをコントロールすることができます。

ブルーに着色されたパラメータはリアルタイム操作が可能で、これらのリアルタイム・コントロール系パラメータを Hot Parameter と呼びます。(Parameter Window 内では **!attack** の様に、! と赤字で表示されます)

パラメーターには MIDI 信号をアサインすることも可能です。パラメーターを選択して Tab キーを押しながら任意の ControlChange 信号を入力するだけでアサインが完了します。

Parameter Window に任意の Sound を Copy & Paste することで、その Sound をコントロール用のパラメーターとしてアサインすることもできます。

KYMAで扱うデータの幅は通常は -1~1, 0~1 の2種類で、.(ピリオド)の前には、必ず0(ゼロ)を付ける決まりがあります。また、計算の順番が通常の演算における法則と異なり、かけ算やわり算の優先はなく、式の左から順番に計算を行います。

$$1 + 2 * 5 = 15$$

もし、通常の計算の決まりに従った結果を求める場合は

$$1 + (2 * 5) = 11$$

のように表記する必要があります。

この他、KYMAはScriptによるSoundのデザイン・制御が可能です。Script読み取り専用Soundをルートに配置することによって、それ以前に位置するSound群をSmallTalkによって記述されたスクリプトがコントロールします。単にパラメータを操作するだけでなく、Sound自体の配置・生成(フィルターアレイなど)が可能です。

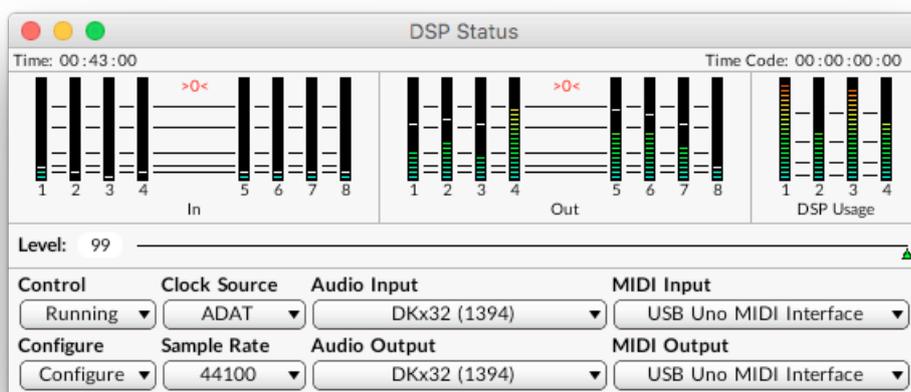
Sound(プログラム)の実行にあたって、まずPC内部でDSP言語にコンパイルした後、データ群(MidiやSample fileを含む)をDSPユニットにUp loadします。その際に自動的に使用するDSPが割り振られます。

KYMAは発音前に必ずコンパイルが必要なシステムなので、モジュラータイプのアナログシンセのようにリアルタイムで接続の変更を行うことは出来ません。



従ってプログラムチェンジのような感覚で複数のSoundをライブで使用する場合は、Soundを並列に配置することになります。

DSPの使用状況はDSP Status Windowに表示されます。ここを監視することで、複雑なSoundを製作したときに発生しがちなエラーをトレースすることができます。

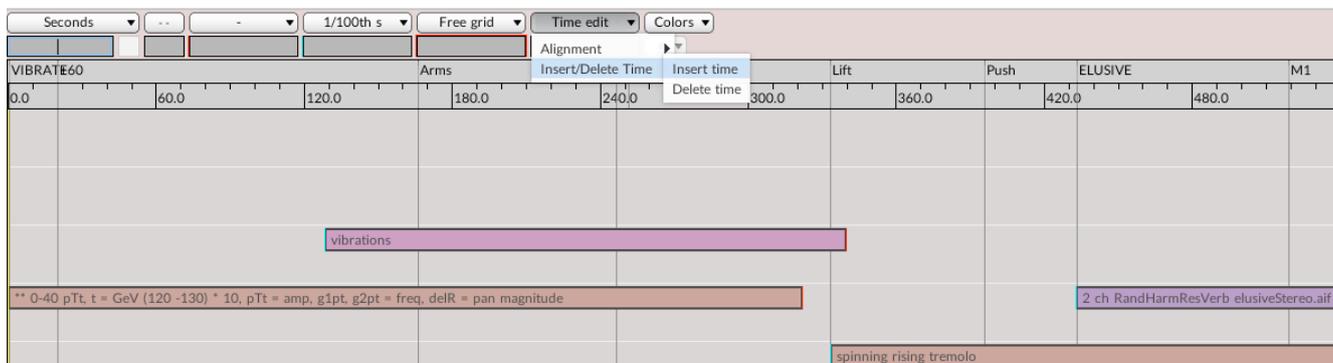


時間軸でSoundを展開するTimeline(後述)を使用することで、このような並列化を行う必要がなくなる場合があります。

前述したように、コンパイル時に処理を行うDSPが自動的に割り振られますが、まれに処理の分配が効率よく行われずにエラーが表示されることがあります。

この問題を解決するためのソリューションとして、Soundを時間軸で展開できるTimelineというモードを使用する選択肢があります。

Timelineは、予め編集されたSound群をMusical Sequencerのように時間軸に沿ってトラックごとに配置できる機能で、コンパイルを行う優先順位を調整して問題を解決できるかもしれません。



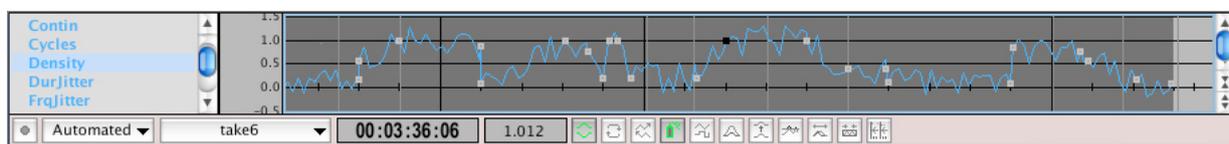
GUIはSequencerと似たような構造ですが、再生される対象が、MidiデータやAudioFileではなくSoundというオブジェクトそのものである点が特徴です。

時間軸に配置されたSound(オブジェクト)は規定したタイミングで生成・消滅するため、DSPの消費を抑制することができます。

Timelineではトラック毎のオーディオ信号を16chまでのデータBusに送ることが可能です。共通のコントローラー信号などのパラメーターをまとめて、異なるオブジェクト間でのリレーションを行うことも出来ます。オブジェクトを結びつける強力なリレーション機能は、KYMAの持つアドヴァンテージの一つといえるでしょう。

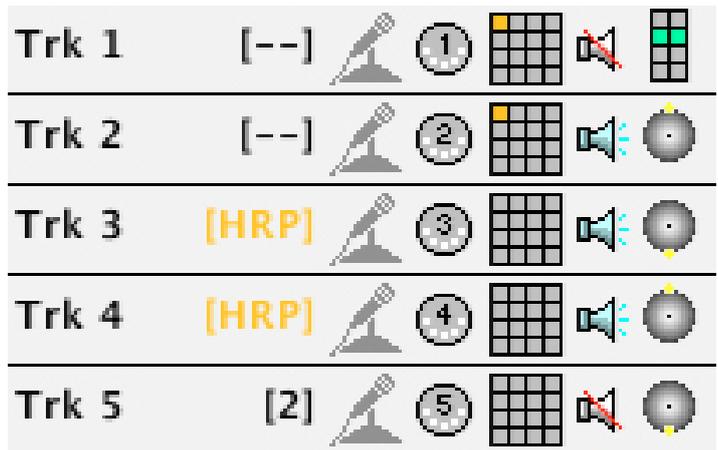
ここからは、筆者が主に使用しているTimelineの運用について具体例を挙げていきます。

Sound上に設定されたパラメーターはTimeline Control Editorにより、時間軸に沿ってグラフィカルに設定ポイントの追加・編集を行います。ローカルに分散したコントローラーを束ねるPromoteTo-Master機能は、それらを一括で操作するマスターコントロールを生成します。コントロールデータは、GUI以外にMIDI/VCS入力からリアルタイムで操作を行うことが可能です。



ここで注意しなければならないことは、Timelineでは自動的にパラメーターのポイント記録したTakeを生成することです。短時間のTimeline設定では問題はないのですが、長尺のTimelineの場合、このTakeで記録されるポイント数が膨大なデータ量になってしまい、Timelineファイル自体のサイズが簡単に10Mbを越えてしまいます。これは、コンパイル時間に影響するポイントなので、ファイルのセーブ時には新たなTakeの書き込みに注意する必要があります。

左図右端のパラメーターでは各トラックの音声出力形態を、スピーカーのディスティネーション設定、及び Pan コントロールの何れかに選択が可能となっています。Pan コントロールを選択した場合は、Preference で設定した SpeakerPlacement 設定に準じ、音声出力の位置が決定されます。つまり、Timeline 上では出力スピーカーの個数や位置関係をあまり気にせずにシステムを設計出来るのです。多チャンネル出力は、SoundEditor 上でも設定は可能ですが、この場合、プログラミング完了後に、スピーカーの個数を変更することは出来ません。

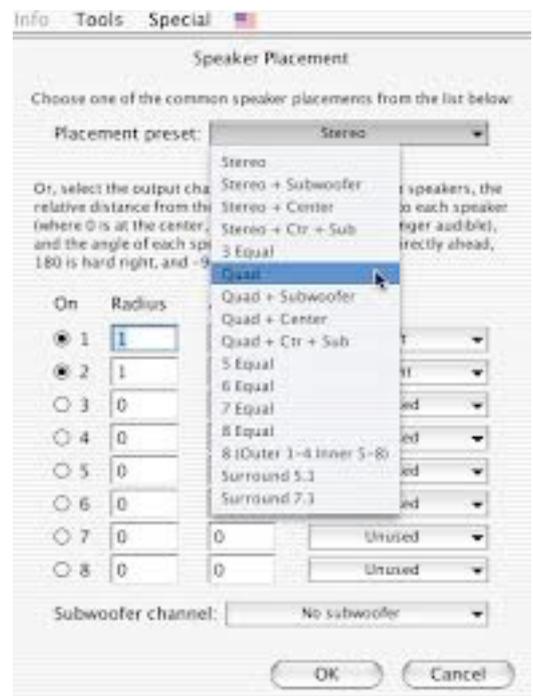


スピーカーの配置は、Preference → Speaker Placement を選択して行います。スピーカーは最大8chまでのマネージメントが可能です。展開するスピーカーの規模は Speaker Placement の設定に依存するため、大元となるプログラムを改変すること無く容易に変更が行うことが出来ます。

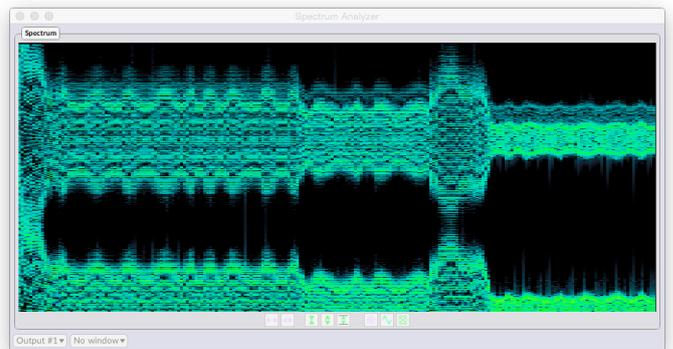
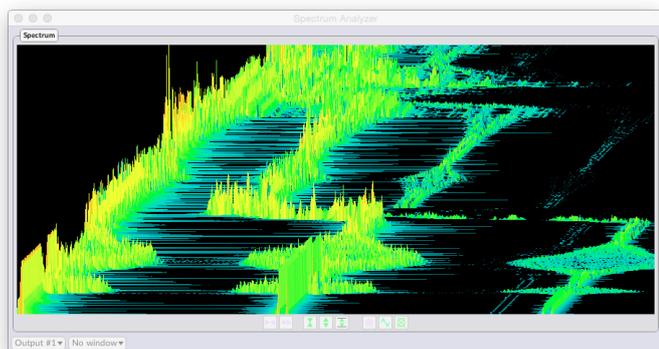
Timeline 上の定位を決定する値は、-1.0 ~ 1.0 で360度に対応します。コントロール信号にLFOをアサインして、360度の自動パンニングを実現することが出来ます。

Timeline を使用しない場合は、マルチチャンネル出力に対応した専用のSoundが用意されていますが、出力CHの設定を変更する場合に少々煩雑な作業が必要となります。

Timeline の欠点は、統合された音声信号が最適化される過程で少々劣化してしまうところにあります。単体で運用している分には気になりませんが、Sound Editor と比較すると、明らかに音質が劣化してしまうようです。



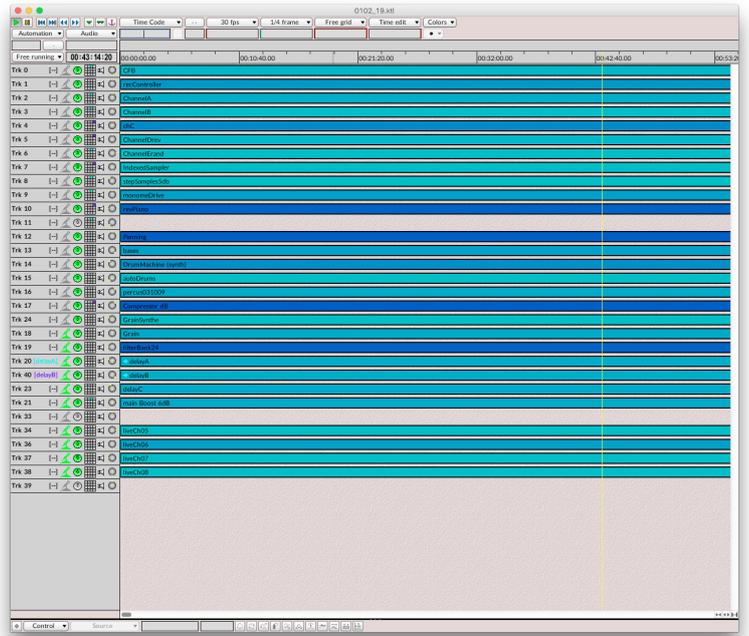
KYMA には多機能なツールが実装されています。24bit DigitalAudioRecorder やSpectrum File Analyzer / Editor、Osilloscope など専用のツールを使って音声ファイルの分析・加工を行うことが出来ます。



ここに、実際に運用している Timeline の例を示します。

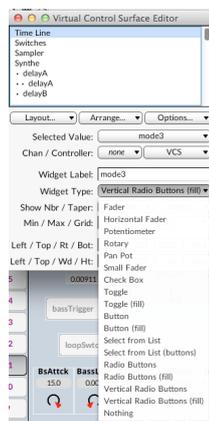
これは、ライブ演奏を想定して Realtime 性を追求した構成で、Sound の生成と消滅が可能な Timeline 上で敢えてそれを行わずに、オブジェクトを並列化しています。

Sound の並列化によって負荷が増大した結果、DSP の処理分配に問題が生じることがあります。この Timeline ではエラーを回避するために Sound を開始するタイミングを 1ms ずつズラす手当を行っています。Sound をコンパイルする優先順位に、明快な解は存在せず、カットアンドトライを繰り返して、エラーの出ない状況を探る必要があります。



Sound を多数配置した Timeline をコントロールする VCS の構成は、この規模になります。

VCS を構成するパーツは **Fader** 等の名称によって自動的にその形態が決定されますが、Editor によって形態やサイズの変更を行うことが可能です。



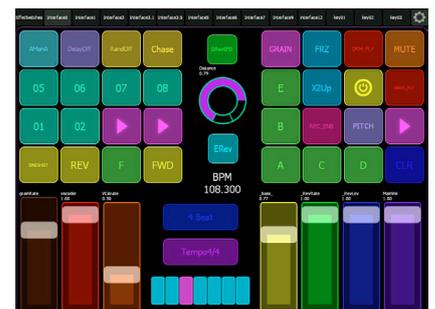
KYMA のヘルパーアプリケーションには、SymbolicSoundCo. が提供する iOS アプリや、サードパーティーが供給する KymaConnect 等があります。

KymaConnect は、PC 上に midi コントローラーのためのゲートウェイを開設できる大変便利なアプリケーションです。



iPad で使用できるアプリケーションには Liine が提供する Lemur という製品があります。VCS の状態を反映するために自動的に配置が決まってしまう KYMA のアプリとは違い、インターフェイスを自らデザインする必要があります。

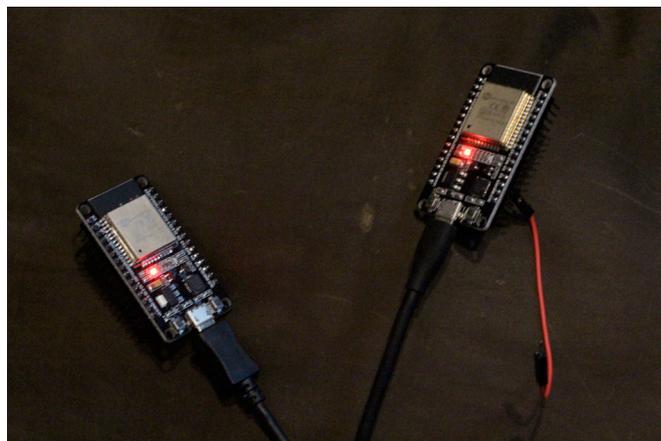
この他、Wifi から midi のゲートウェイに接続する方法もあります。



凡例としてmidi送受信専用のローカルサーバーを設置して、Wifi 経由でmidiを送受信する方法を紹介します。

midiコントローラーは ESP32 というデバイス使って自製しています。

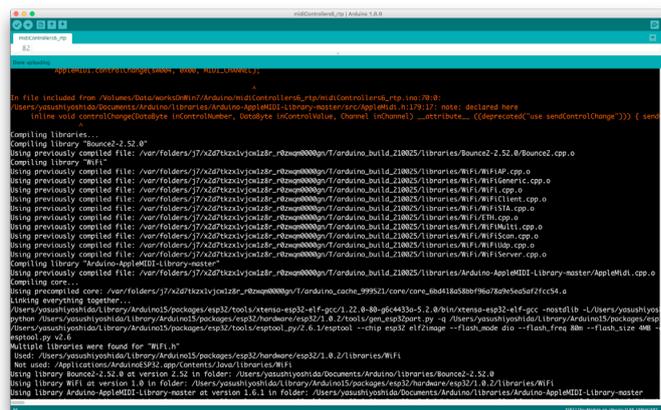
このチップは2.4GHZの Wifi が実装されている IoT デバイスで、非常に安価なのが特徴です。送受信を行うデータ・フォーマットには RTPmidi を採用しています。



プログラムの開発は Arduino で行います。

Arduino に ESP32 を導入するのと、製作例を探すのに若干の時間がかかりましたが、比較的容易にアイデアを具現化することが出来ました。

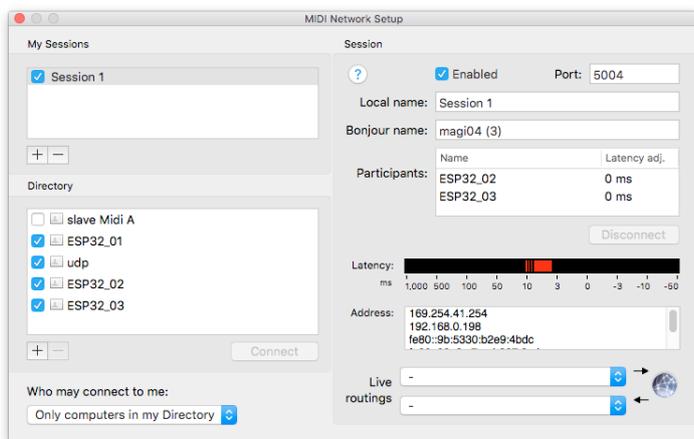
今回製作したデバイスではADコンバーターを使用して、フットコントローラーからの出力をmidiに変換しています。



Mac 側では WiFi 通信リンクを設定します。

実験では、同時に3つのデバイスからの受信を行うことが出来ましたが、通信バッファのオーバーフローに注意する必要があります。

ローカルデバイス側では、事前にデータの過剰な出力を抑えるための仕掛けが必要ですが、これは反応速度とのトレードオフになります。



詳しい製作例をこちら→

<http://audiohologram.sblo.jp/article/186587293.html>

にアップしてありますので興味のある方は参考にしてください。

KYMA は非常にフレキシブルな運用が可能な製品で、このように自作が介在できる余地があることも魅力の一つと言えるでしょう。

