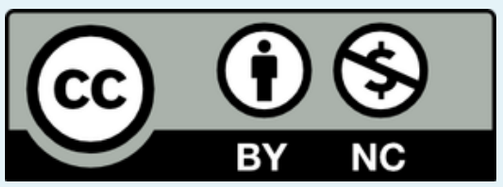
(ちょっと)高レイヤーオーディオ組み込み2020

DSP言語/プログラマブルボード/オープンハードウェア



2020/09/01 松浦知也(me@matsuuratomoya.com/matsuuratomoya.com)

プログラマブル

オーディオボード

Owl→Magus(Rebel Technology)

- STM32Fベース、C++のコードを自由に鳴らせる
    - Webベースのオーサリングツール、Gen~/Heavyエクスポートの支援
- MagusではOwl向けのパッチはそのまま使える（実質後継機）
- FreeRTOSで動いているらしい
- ファームは全機種公開されているが、ドキュメントはほぼなし

Bela

- BeagleBone Black(ちょっと高級なラズパイ)+専用拡張ボード+カーネル拡張(Xenomai)
    - ユーザーコードはC++（を経由して、FaustやPd+Heavyなど）
    - LANで繋ぐとWebサーバベースのIDEが立ち上がる
- Linuxベースだがオーディオ関連のコードはカーネルをバイパスしているのでバッファを2サンプルまで詰められる
- 無理やりユーロラック化したシリーズとか出てる（でかい）

Daisy

- チップ：STM32H750IBK6(Cortex-M7,400MHz)
- AKMの2in2out,~192kHz/24bitオーディオコーデック乗ってる
- 64MB SDRAM(wavとか載せやすい)
- stm32duino対応なのでArduino IDEで開発できる
- 専用のDSPライブラリDaisySPがある
- 抜き差し容易なガワがユーロラック/ギターペダル/スタンドアロンと揃っており賢い

Teensy

- ARMコアベースの安くて速いマイコンボード(ArduinoIDE対応)
- チップはSTMでなくNXP製、3.x:Cortex-m4　4.0:Cortex-m7
  - 4.0ならクロック600MHzで$19.95
- オーディオ用の拡張ボードや、ビジュアルプログラミングできるオーディオライブラリがあるなど音まわりの環境が充実
- インディーモジュラー系での利用実績多し (BASTL 、Ornament&Crimes)

MODDevices MOD Duo

- Linux上で動くLv2というVST類似プラグイン規格を走らせられるギターペダル
- 中身は普通のLinux on Arm A7 1GHz(パワーでゴリ押し感)
- レイテンシは5ms(多分バッファサイズ128の往復？)

Monome Norns

- ラズパイベース、LuaスクリプティングとSuperColliderベースの音響合成
- ORCAとか、アプリケーションを入れ替える感覚で中身を変えられる
- インターフェースの拡張はmonome製品同士が中心

Critter & Guitari Organelle

- Linux上でPuredataを立ち上げて操作できる
- Pd限定なのでNornsよりできることの幅は狭いが参入障壁低い
    - Heavyでなく純粋なPdなので、パソコンで作ったパッチはほぼ動かせるはず
- これも、インターフェースは標準のボタンやノブ限定
- 中身はRaspberry Pi Compute Module3+

他

- Axoloti http://www.axoloti.com/ 独自パッチャーフォーマット
- midiglue https://sigboost.audio/midiglue/ プログラマブルmidiプロセッサ 独自ビジュアル言語

余談：STM32H750を直接使うなら...

- RtoRオペアンプx2 (ゲインの選択肢が-15,-7,-3,-1,2,4,8,16)
- DACは12bitだがADCは16bit x3
  - ADCは多重化で最大36ch使える機種もある
- 12bitDACも高周波にノイズシェーピングすれば使えるはず
- というか、速度に任せてソフトウェア1bitDACを作るのもあり

DSP特化言語

DSP言語一覧

- C++エクスポートが第一目標の言語
  - **Faust**
  - **Soul**
  - **Vult**
- 副次的にC++/Cエクスポートができるもの
  - **Cycling’ 74 Max (Gen~)**
  - **Puredata+Heavy**

Faust

- 関数型DSP特化言語。文法がとにかく特殊。
    - 基本はC++トランスパイラだが、Architectureファイルと呼ばれるDSP以外のC++部分のテンプレートを作っておくことで、多様なプラットフォームにコマンド一つでエクスポートできる
- ライブラリが豊富。フィルタ/オシレータ/物理モデリング関連はアルゴリズム提案者が書いてたりするので実質リファレンス
- ハード周りのエクスポートも

Faust
最近ちょっと使いやすくなった

- 開発・コンパイルはほぼWebで完結(https://faustide.grame.fr/)
- routeプリミティブの導入
- soundfileプリミティブの導入
- 今後、サンプルレート変更を可能にするondemandプリミティブが入るとか・・・(参考:https://drive.google.com/file/d/1LkT8KviWocnzt6hqRsLlSri5uwVDWI5p/view)

Soul

- JUCEなどを作っているROLIが主導で作っているDSP言語
- 手続き型＋部分的にデータフローっぽい形でまあまあ読みやすい
- Faust同様Web上で編集できる
- まだ全部のソースがオープンになっていない（HEARTという、より低レイヤーの中間表現があるとか）
- アセットやGUIなどのマッピング情報を含めた.soulpatchというプロジェクトフォーマットがある

Vult

- Leonardo Laguna Ruizという人が副業的にやっているソフト・ハードシンセプロジェクトのための言語
- C++,JS,LuaJITなどにエクスポートできる
- これもWebで動かせる
- コンパイラはOcaml(実装が綺麗)
- 組み込み向けを意識した言語仕様

Vult
組み込み向け機能

- Floatを専用演算命令が無いプラットフォーム向けにfixed-pointとして出力することができる

- 関数をテーブルルックアップに自動変換してくれる記法がある（Faustでもできるが、ちょっと手間がかかる）

- Wavファイルをコンパイル時に固定配列として埋め込んでしまえる

Gen~

- Maxの中のGen~というDSP特化パッチフォーマット
- Maxと似た形のデータフロー型ビジュアルプログラミングも、genexprという手続き型テキストで記述も可
  - genexprの抽象化能力はそんなに高くないのでリバーブとか作ろうとすると大変、大規模な変数管理が辛い
- C++エクスポートはおまけっぽく、公式ドキュメントの解説も少ないがOwlでは積極的に活用されている
  - RebelはemscriptenWeb上でプレビューできるツールを作って運用しているらしい

Puredata+Heavy

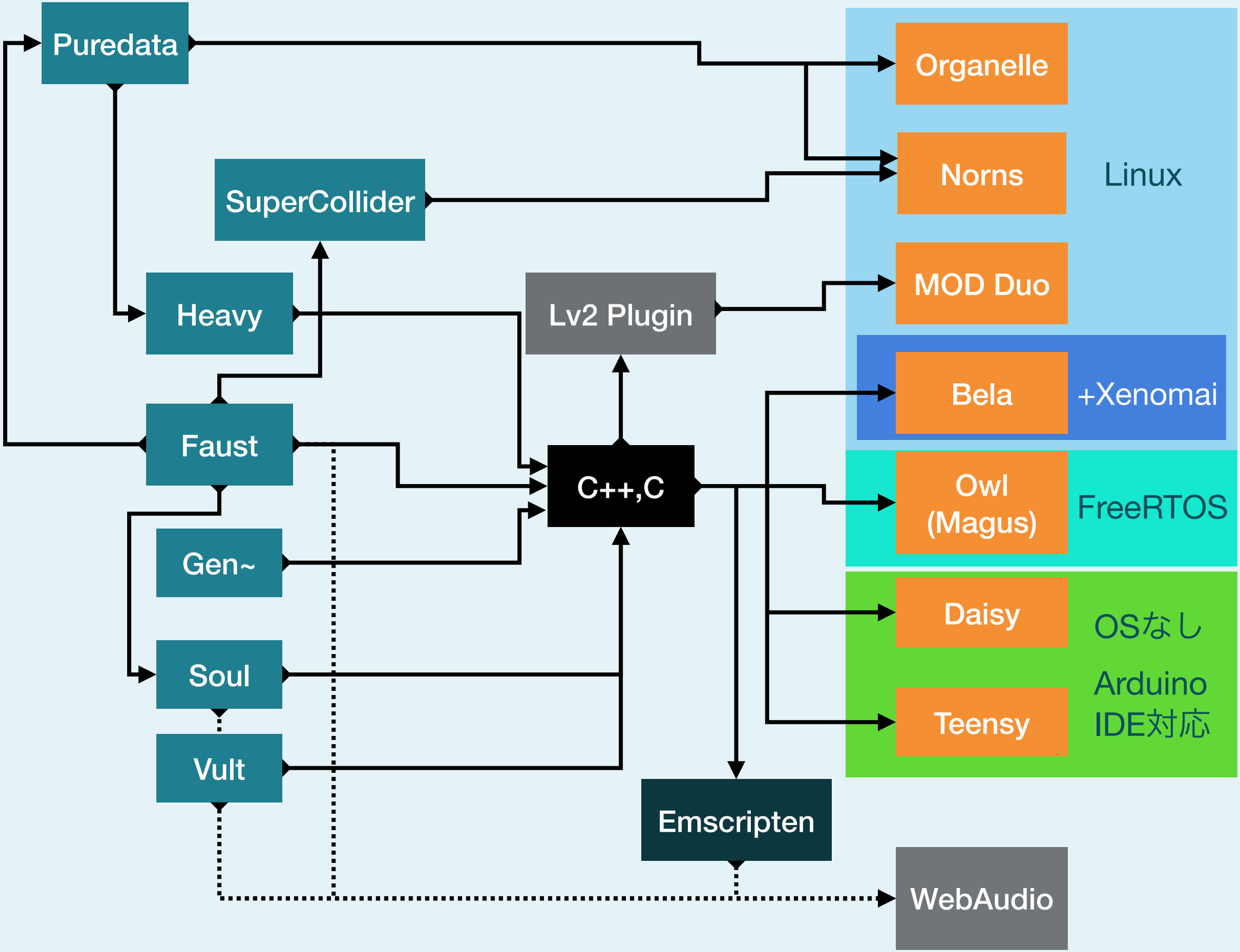
- hvcc: Pdパッチ→Cのトランスパイラ (実装はPython)

- libpdなどとは完全に異なる実装で、スケジューラとかは存在しない。Pdのパッチを無理やりGen~レベルのパッチとして解釈している感じ

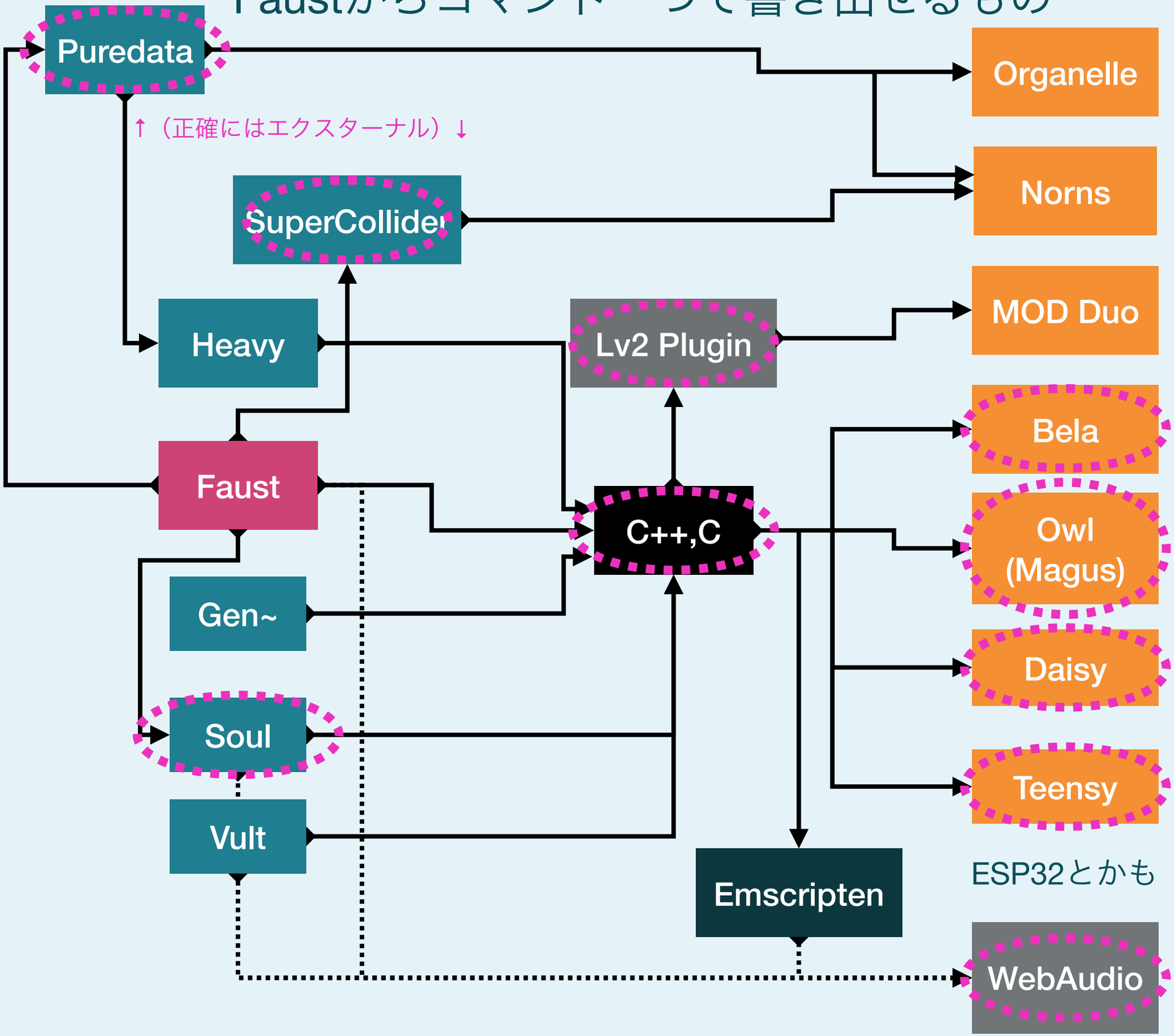
- 開発元のEnzien Audioが2年前に倒産、既に会社のWebも消滅
  - hvccはGPLとして公開されるも一切メンテされず、OSSとして開発が引き継がれる様子もなし
  - の割にPdの仕様が枯れていることもあり一応動く。BelaやOwlでは引き続き利用されている(これもRebelがWebプレビュー持ってる)

DSP言語比較

	C++ベタ	Faust	Soul	Vult	Gen~	Heavy	Pd/SC
可読性	△	×	○	○	△	○	○
抽象化能力	腕次第	◎	○	○	△	△	△
オーディオファイル	△	△	○	○	△	△	○
実行性能	腕次第	○	○	○	○	○	△
安定性	腕次第	○	○	○	○	△	○
将来性	○	○	?	△	△	×	○
WebAudio	△ (emscripten)	○	○	○	△ (Owlが独自対応)	△	×
備考		文法が独特 ライブラリ豊富	一部ソース非公開, 開発途中	組み込み利用実績あり, 個人開発OSS	C++関係の開発・ 機能追加は非活発	開発終了	OS必要


Puredata
SuperCollider
Heavy
Faust
Gen~
Soul
Vult
Lv2 Plugin
C++,C
Emscripten
WebAudio
Organelle
Norns
Linux
MOD Duo
Bela
+Xenomai
Owl
(Magus)
FreeRTOS
Daisy
OSなし
Teensy
Arduino
IDE対応

Faustからコマンド一つで書き出せるもの



DSP言語のパフォーマンスは？

- 正直、一度C++を経由するとC++コンパイラの最適化で似たり寄ったりになる(特にClang系のLLVMベースのものを使った場合)
- メモリの効率は必要なデータを一括で確保するのでDSP言語の方が多少有利
- Cortex-M4/7はDSP向けの専用命令セットがあるが、有効活用しきれずSTM用ライブラリを使ったほうが早くなる可能性はある
  - ちなみにDaisySPは専用命令ガンガン使ってるのかと思いきやSTL完結だった
- パフォーマンスのためにカリカリにチューニングしてめちゃくちゃ読みづらいC++のコードの実行性能を100、C++で読みやすさ最優先して書いたコードを50とすると、DSP言語で適当に書いても70くらいのパフォーマンスが出せるイメージ

余談：Cortex Seriesまとめ

命令セット

	M3	M4	M4F	M7
Fixed	○	○	○	○
DSP	×	○	○	○
SP Float	×	×	○	○
DP Float	×	×	×	一部

DSP言語を使うメリット

- DSPはDSP処理としてコードが確実に分離できる
  - ≒開発のプロセスがきちんと分離できる。性能の問題と表現の問題を分けて考えられる
- ドキュメンテーション、要件定義としての価値
  - 最悪言語処理系が死んでも言語仕様は残るので解読しやすい
- リングバッファとかディレイ周りのメモリ管理関係の失敗が減る
- VCVRackなどのソフトシンセ、Webなど他プラットフォームと並行して運用する可能性があるなら◎

DSP言語を使うデメリット

- 導入実績・可読性・開発の枯れ具合、将来性など正直一長一短
  - Faustは安定しているが、組み込みがメインターゲットというわけでも無い
  - Soulは将来的に組み込みをかなり意識しているが発展途上
  - Vultは小規模ではあるが利用実績あり
- ビルドプロセスが一段複雑になる
- FFTが絡んだり、部分的にサンプルレートを変えるとかが厳しい

余談: Sound Open Firmware

- LinuxFoundationが支援して、Intelの人が中心に開発中
- 地味にLinux Kernel 5.2からもう取り込まれている
- 今のところサポートはIntel系のx86CPU&TensilicaのXtensa HiFiシリーズの組み合わせ(実機で試すならMinnowBoard Turbotぐらい)
- 直接実用的になる日まではまだ遠そう（特にARM対応とか）だけど、QEMUでエミュレーションしてるのとかアプローチとして参考になる

まとめ

アーキテクチャの選択肢

- OSを使うか否か？
  - Linuxの場合、Xenomaiのようなカーネル拡張をするか？
    - ALSA System on a Chip(ASoC)やSOFなど、組み込み向けオーディオ機能は今後充実していきそう
- OS使わない場合、Arduino IDE対応or非対応？
  - なんだかんだIOの共通規格として成立しつつあるので、ボードがディスコンになっても替えを効かせやすいかも

アーキテクチャ選択の判断基準

- ユーザーにプログラム書き換えを開放するか?
  - さらに、ファーム書き換えorホットリロード
    - OSありの方がホットリロードなども含め対応しやすい
- コストの問題

オープンソースハードの実際

- 「ソースが公開されている」と「ドキュメントがきっちりまとめられている」は全然違う
  - ハードも含めたドキュメントまできちんとまとめているのは Mutable Instrumentsくらい（結果的にそれで広がってもいる）
  - せめてReadmeくらい書いてくれ（.inoファイルがドンと置かれていても困る）
- ファームウェア、回路図/レイアウトも含めた共有はまだいろいろ難しいがhttps://www.writethedocs.org/guide/とか参考になりそう

その他所感

- DaisyはDSPライブラリほかでも使い回し効きそうだし、STMベースの開発のプロトタイプとして十分使えそう
- QEMUとかでプロセッサ/ロジックレベルのエミュレータ作るとか、アナログ・メカ部分のシミュレータ用意しておくとか、あんまりやってるとこないけど本気で長続きさせたいなら有効かも