
Androidでサクサク プロトタイプ開発

日本Androidの会 幹事

渡辺知男

Agenda

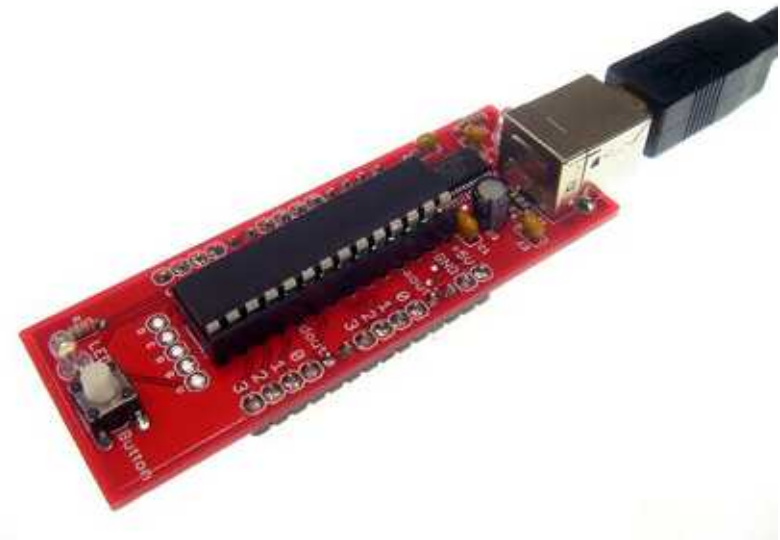
- プロトタイプ開発とは
 - Gainerの簡単な紹介
 - Gainerの接続の仕方
 - どんな製品プロトタイプができるのか？
 - AndroidでGainerのメリット
-

プロトタイプ開発とは

- 新製品を量産に移す前などに試験用として作られ、製品の設計に起因する問題やその他の不具合を発見することができる、具体的な修正の検討に入ることができる。(Wikipedia)
- 製品の試作には膨大な費用が掛かる。でも機能試作はしたい・・・
 - どんな感じで動くのか？
 - パフォーマンス的にどうよ？
 - PCを使って試作はできるが・・・でもPCでしょ？って言われてしまう...orz

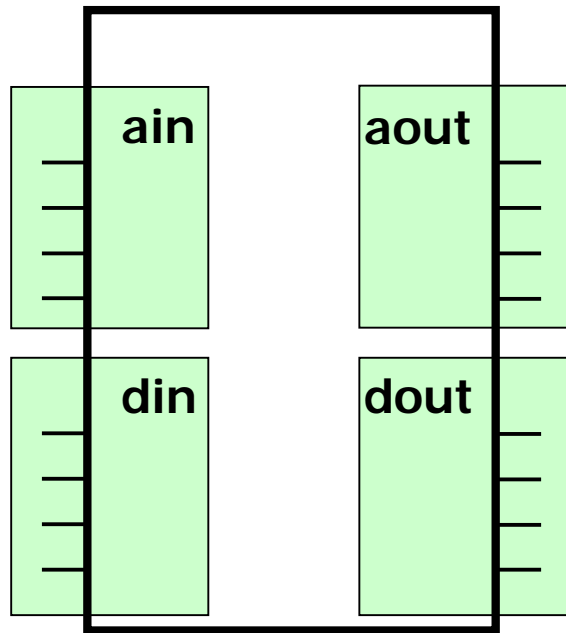
Gainerの簡単な紹介

- 16本の入出力ポートをコンフィグレーションで選択可能
- USBシリアル通信でコマンド制御可能
- 機能モジュール
 - デジタルIn/Out
 - アナログIn/Out
- 公式サポート言語
 - Flash
 - Max/MSP
 - Processing

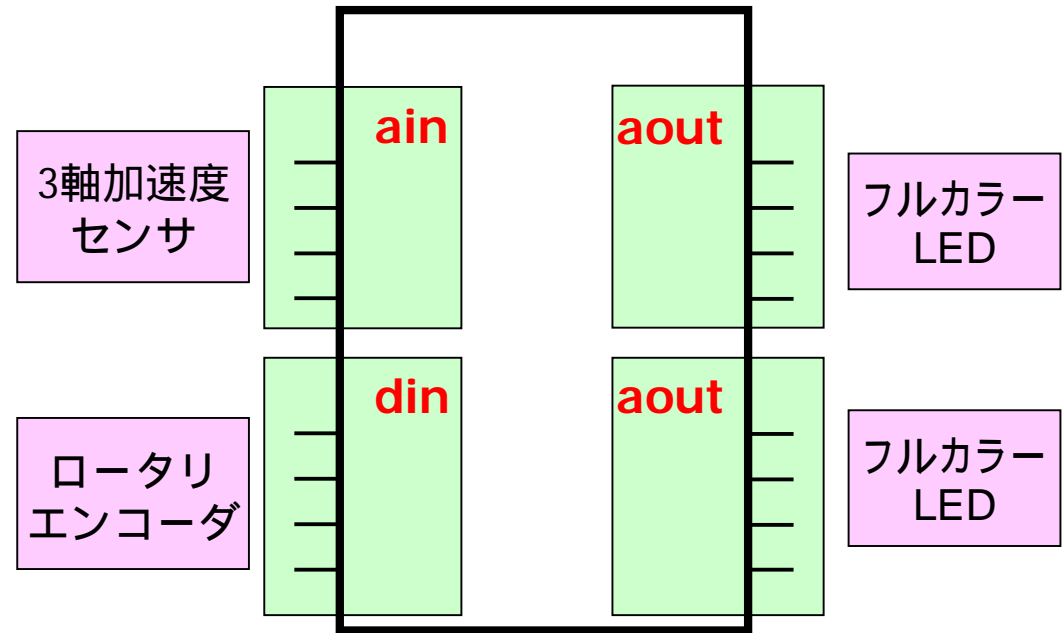


詳しくは <http://gainer.cc/>

Gainerの端子とモード

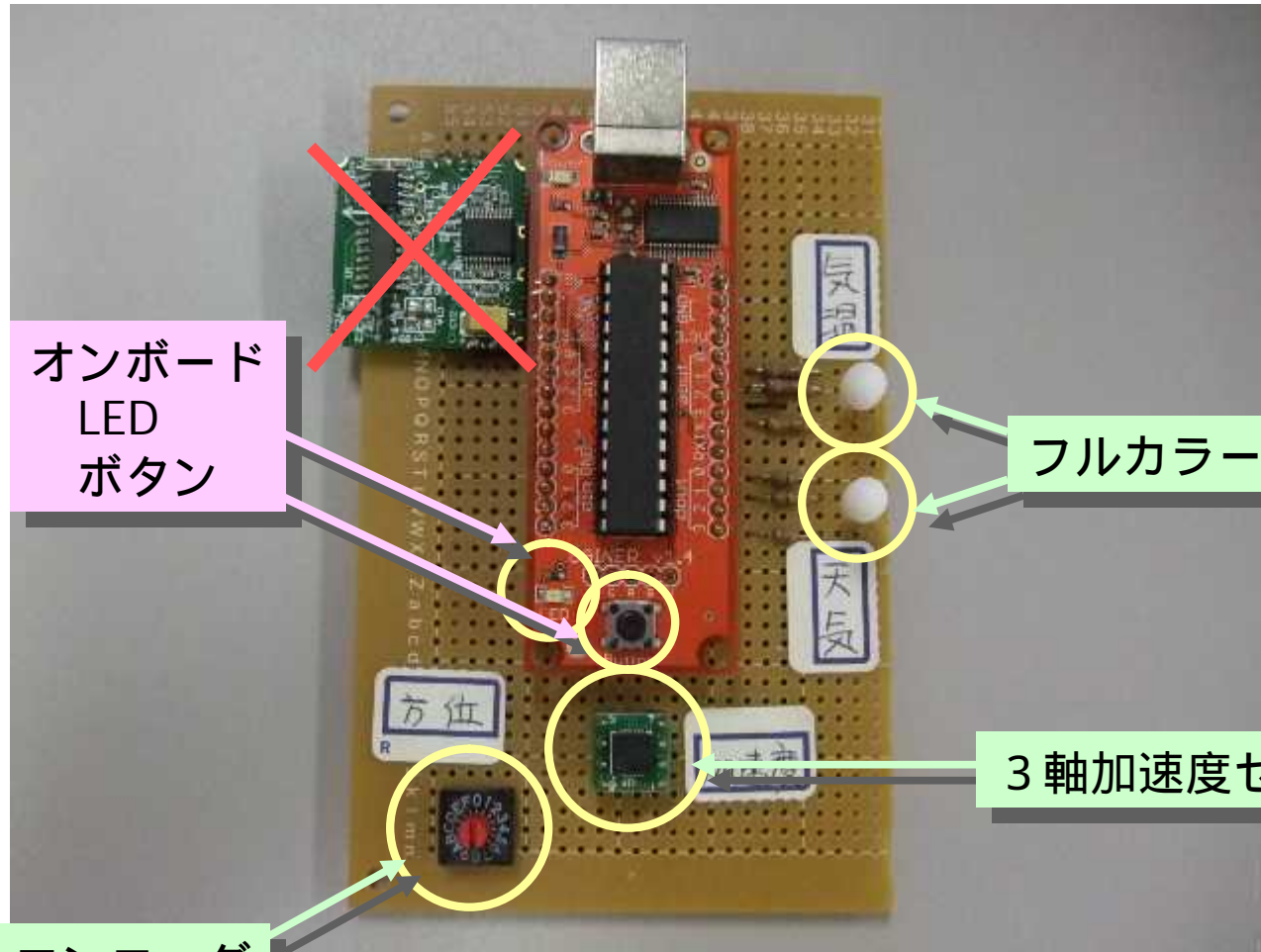


4つのブロックを
アナログ入力 / 出力
デジタル入力 / 出力
に設定が可能
MODE1からMODE7まで



今回のデモの例では・・・
「MODE3」に設定

作成したGainer基板



オンボード
LED
ボタン

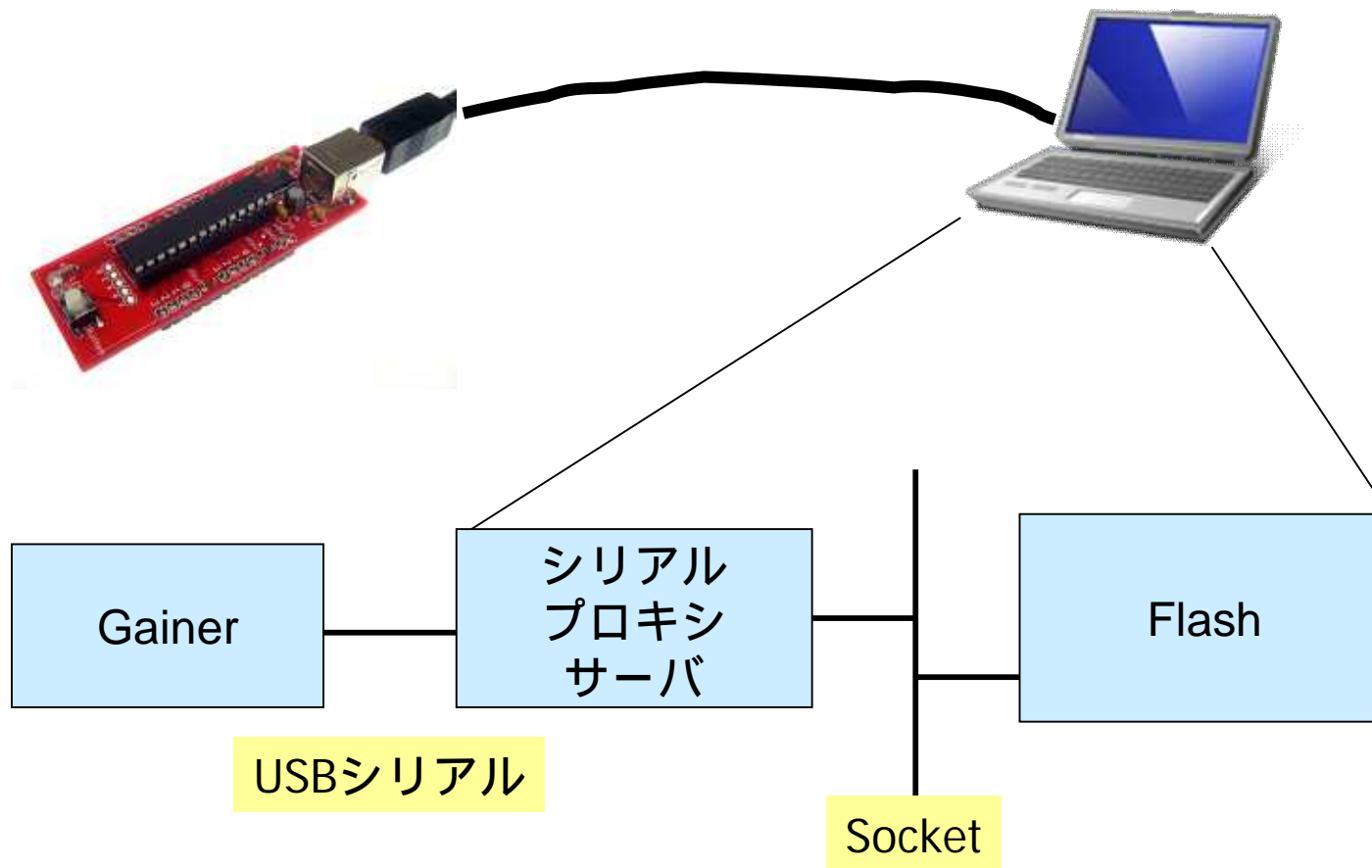
フルカラーLED

3軸加速度センサ

ロータリーエンコーダ

Gainerの接続の仕方

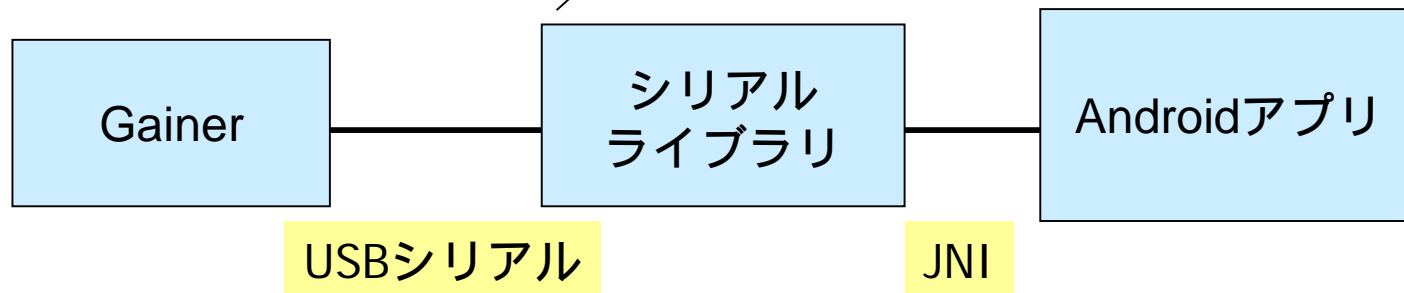
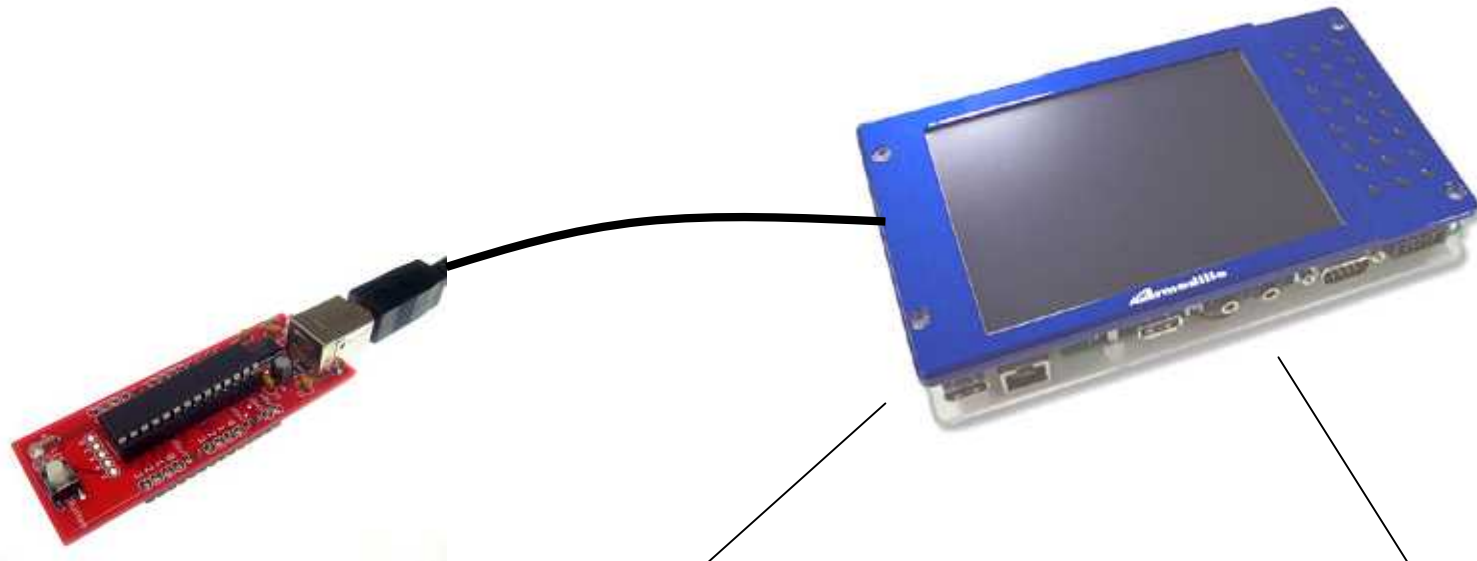
まずはPCと普通に接続する場合



Flightner (2008/8月)

- adamrocker氏がAndroid勉強会で発表
 - PCエミュレータ上でのフライトシミュレータ
 - <http://open-mobile.jp/mobilelabo/android-applications-on-real-machines/gainer-with-android-emulator>
 - これは...実機で出来ると面白いかも
 - どうしてもPC経由じゃなくて、実機と直接繋ぎたいぞっ！
 - とりあえず実機でやるにはネイティブライブラリが必要っぽいし、JNIを使うことでできるっぽい
-

GainerとArmadillo500FXの接続



実装構成



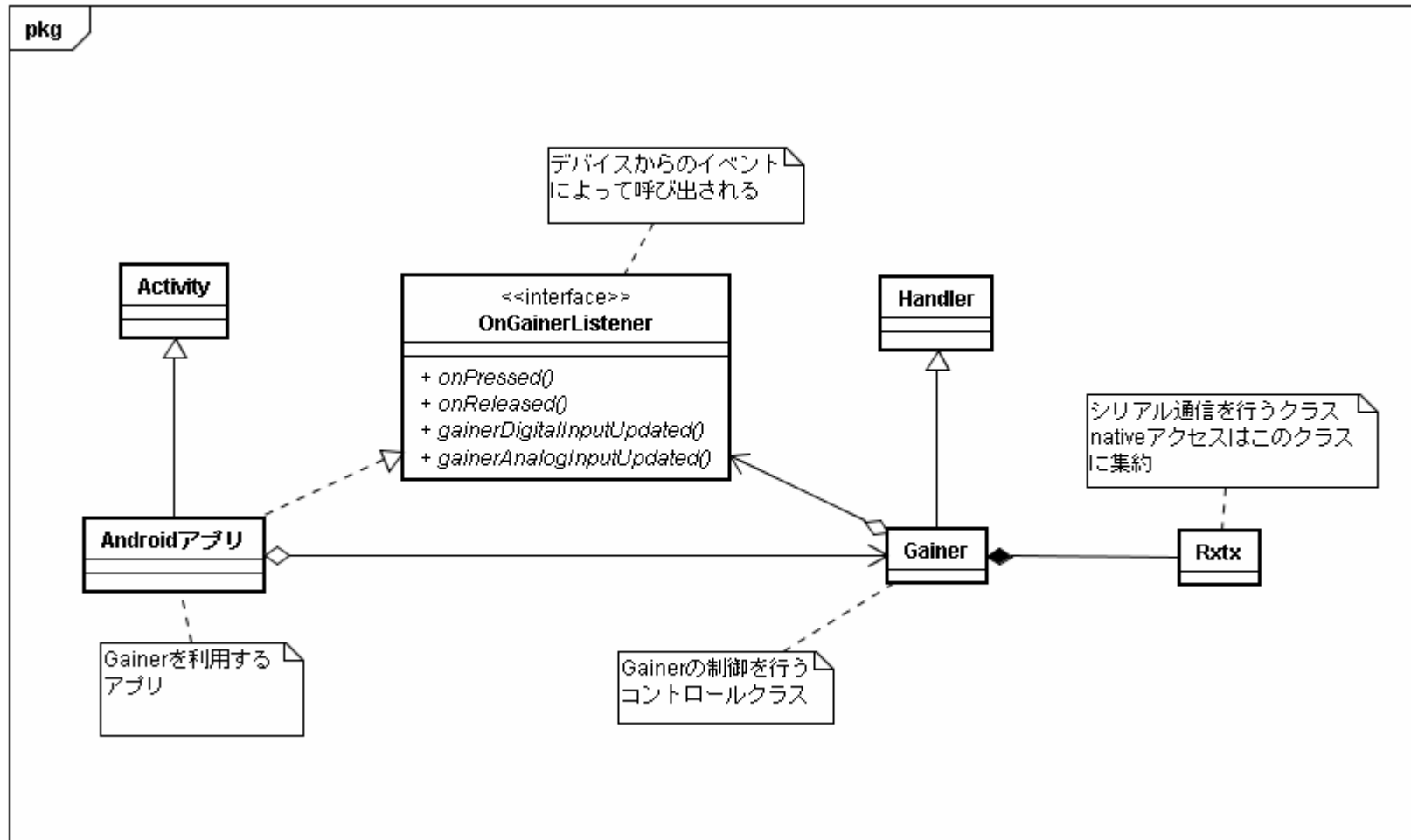
JNIライブラリの作り方

- AndroidでJNIを使う方法（参考資料）を参照
 - デバッグ用Tips
 - LOGD,LOG_IFマクロを使う
 - printf()は仮想コンソールに出力してしまう
 - #include <cutils/log.h> (system/core/include/cutils/log.h)
 - Android.mkのLOCAL_SHARED_LIBRARIES
 - liblogを指定
 - ./adb logcatでログを見られる
 - マルチスレッドの場合はライブラリが別途必要
 - JavaVMのポインタはスレッド毎に必要
 - Android.mkのLOCAL_SHARED_LIBRARIES
 - libdvmを指定
-

センサー情報の取得

- センサー情報の取得はリスナーを使用する
 - OnGainerListener()
 - interfaceクラスを定義して使用
 - Handlerクラスの利用
 - 描画はメインスレッドのみ行えるため、Handlerクラスを継承したクラスからhandleMessage()内でOnGainerListenerクラスのメソッドをコールする
-

Androidアプリ構成図



Gainerのレシピ

■ 用意するもの

- Gainerモジュール 1個
- 3軸加速度センサ 1個 (x,y,z : Analog)
- フルカラーLED 2個 (RGB : 256階調)
- ロータリーエンコーダ 1個 (4bit : Digital)

- パーツをいちいち揃えるのが面倒だったので、Gainerスターターキットを購入
- フルカラーLEDは別途追加で秋月で入手
- ブレッドボードは使わず、基板半田付け(笑)

Demo

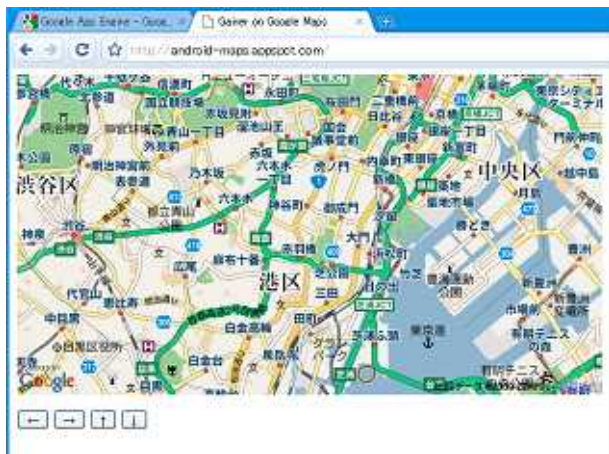
- サンプルアプリ
 - Gainer上のLEDの「点灯」「消灯」
 - Gainer上のボタンの「押す」「離す」を検知
 - フルカラーLEDの制御
 - Analog Out
 - 3軸加速度センサの入力を表示
 - Analog In
 - ロータリーエンコーダの入力を表示
 - Digital In

もちろんアイデアしだいで、いろんなセンサを接続して遊べます

ネットワークとの連携

- <http://android-maps.appspot.com/>
 - Google App Engineで作成した普通のウェブページ
- Androidからアプリ経由でアクセスすると・・・
 - ウェブページ内のJavaScriptを呼び出すことによって、ウェブページの入出力制御ができる
 - ページのソースを見てみると・・・
 - `function callJS(y_axis, x_axis)`がAndroidから呼び出す関数
 - 「初めてのAndroid」のパクリソースなのでわかりやすい(笑)
 - Androidへの入出力のインターフェースをGainerで制御することで、フィジカルな入力とネットワークが連携できる

どのように動作しているのか？



Google Maps APIの
JavaScript関数でセンター
位置を動かす



左右・前後の傾きを
取得し、WebView上の
JavaScriptを呼び出す

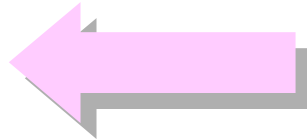
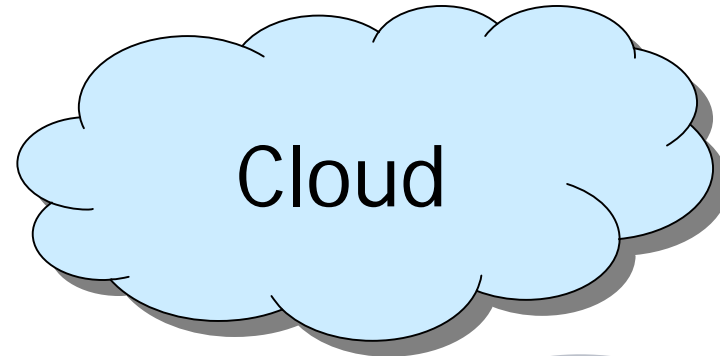


3軸加速度の
値を入力

どんな製品プロトタイプができるのか？



起きると今日の天気や
気温がわかる



天気予報



電車運行情報

状況に応じて目覚ましの
時間を変えてくれる

クラウド対応目覚まし

AndroidでGainerのメリット

- プロトタイプ開発がサクサクできる
 - 簡単に各種センサーを接続して試作できる
 - 強力なCPUのPCではなく、ARMベースの組み込み製品なので、製品化の距離が近くなる
 - PCでデモするよりも「それっぽく見える!」
 - これ結構重要
 - クラウドとのマッシュアップが試作可能
 - Gainerを通してクラウドと会話ができる
 - Android = 組み込みクラウド端末
 - Gainer = フィジカル入力
-

参考文献・参考資料

- throw Life – adamrocker氏ブログ
 - <http://www.adamrocker.com/blog/>
 - オープンモバイル・コネクションズ2008でライトニングトークしてきました
 - <http://www.adamrocker.com/blog/?s=%A4%A2%A4%CA%A4%BF%A4%CE>
 - シリアルプロキシ版Gainerライブラリ
 - <http://code.google.com/p/gainer-lib-android/>
- ブリリアントサービス・ブログ
 - <http://d.hatena.ne.jp/bs-android/>
 - AndroidでJNIを使う方法
 - <http://d.hatena.ne.jp/bs-android/20090324/1237864333>
- 「初めてのAndroid」
 - <http://www.amazon.co.jp/初めてのAndroid-Ed-Burnette/dp/4873114098>
 - 出版社：オライリージャパン
- 「+GAINER」
 - http://www.amazon.co.jp/GAINER_PHYSICAL-COMPUTING-GAINER-GainerBook-Labo/dp/4861672015/ref=sr_1_3?ie=UTF8&s=books&qid=1245581038&sr=8-3
 - 出版社：オーム社
- JNI：Java Native Interfaceプログラミング
 - http://www.amazon.co.jp/JNI-Native-Interfaceプログラミング_C-コードを用いたJavaアプリケーション開発-books/dp/4894710803/ref=sr_1_1?ie=UTF8&s=books&qid=1245581130&sr=1-1
 - 出版社：ピアソン・エデュケーション