組込みLinuxを使ったガジェット工作入門
～gumstixを使ったbc9 (PDA:Android端末)の製作例～

株式会社 ビート・クラフト　龍池哲也
大塚聡史

about us

- BeatCraft, Inc.
  - http://www.beatcraft.com/
  - 墨田区錦糸 (JR, 半蔵門線 錦糸町駅)
  - メディアフレームワーク

– 龍池 哲也
  - ハードウェアの設計・試作を担当

– 大塚 聡史
  - Android 移植作業を担当

本日の内容について

- 本日の内容はインターネット上に公開しています
  - 情報は随時更新・追加しています

http://labs.beatcraft.com/ja/ (日本語版)

http://labs.beatcraft.com/en/ (英語版)

本日の予定

- イントロダクション
- ハードウェア編
  - 設計のコツ・注意事項
  - デジタル回路の基礎知識
  - ベースボードの設計
  - プリント基板の製作
- Android移植編

イントロダクション

- イントロダクション
- ハードウェア編
  - 設計のコツ・注意事項
  - デジタル回路の基礎知識
  - ベースボードの設計
  - プリント基板の製作
- Android移植編

ガジェットとは?

gadget [名詞]

- ちょっとした(便利な)機械装置[仕掛け]
- 気のきいた小道具, 付属品,部品

(ジーニアス英和辞典 第4版より)

→ 気のきく、便利で携帯して使える
お気に入りの小道具を作ること

ガジェットとは?

- ガジェットの例
  - PDA (Personal Digital Assistant)
    - ChipCard, DataSlim, Palm, Clie, EM-ONE…
  - デジタルカメラ
  - 携帯電話
    - iPhone, TouchDiamond, EMONSTER,...
  - その他
    - Chumby

不満な点

- いままでのPDAの不満
  - センサなどのデバイスが拡張できない
    - メーカーが用意したものだけ
  - OSやドライバがクローズドで改良できないものが多い

→ アイディアがあっても実装できない

→ ならば、必要な機能をもったガジェットを自作してしまおう

組込みLinux

- Linuxは、組込みでも使われている
  - PDA
  - NAS
  - 携帯電話
  - デジタルカメラ
  - ルーター

  .....

  パソコン(ワークステーションやサーバ)以外でも広く使われている

作りたいガジェット

- Linuxが動作する
  - Linuxの資産の利用
  - デバイスドライバが作れる
- 携帯可能
  - 持って歩け、移動先でも手軽に使える
- タッチスクリーン付カラー液晶
- 音声の入出力
- 3軸加速度センサ、電子コンパス、GPS

などなど….

bc9の紹介

- bc9 外観

YouTube でも動画を公開中!
beatcraft, bc9 で検索

bc9のハードウェア構成

- CPUボード
  - gumstix Verdex PRO XL6P (米gumstix社)
    - PXA270 @600MHz
    - RAM 128MB/Flash ROM 32MB
- ベースボード
  - 電源、インターフェイス回路、センサ搭載
    - DC 5V電源, Li-ion電池対応
    - USB, AC97, GPS, シリアルコンソール
    - 3D加速度センサ
    - 入力ボタン, LED出力
- タッチスクリーン付カラー液晶
  - 480 x 272 @24bitカラー
  - 白色LEDバックライト
  - 抵抗膜式タッチスクリーン

bc9のソフトウェア構成

- OpenEmbedded Linux
  - device drivers
  - Applications
- Android
  - Linux Kernel
  - device drivers
  - Android Applications

etc...

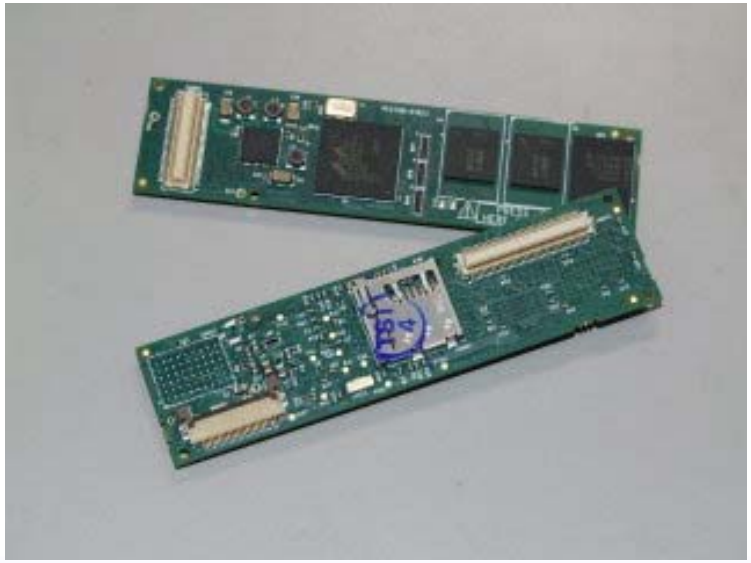
ハードウェア編

- イントロダクション
- ハードウェア編
  - 設計のコツ・注意事項
  - デジタル回路の基礎知識
  - ベースボードの設計
  - プリント基板の製作
- Android移植編

gumstix verdex pro

- 米国 gumstix社の製品
  - CPUボード、拡張ボード、アクセサリで構成
    - CPUボードと使いたい機能の拡張ボードを組みあせる
  - ハードウェア情報、ソフトウェア情報が公開されている

日本から個人輸入可能
詳しくは http://gumstix.com

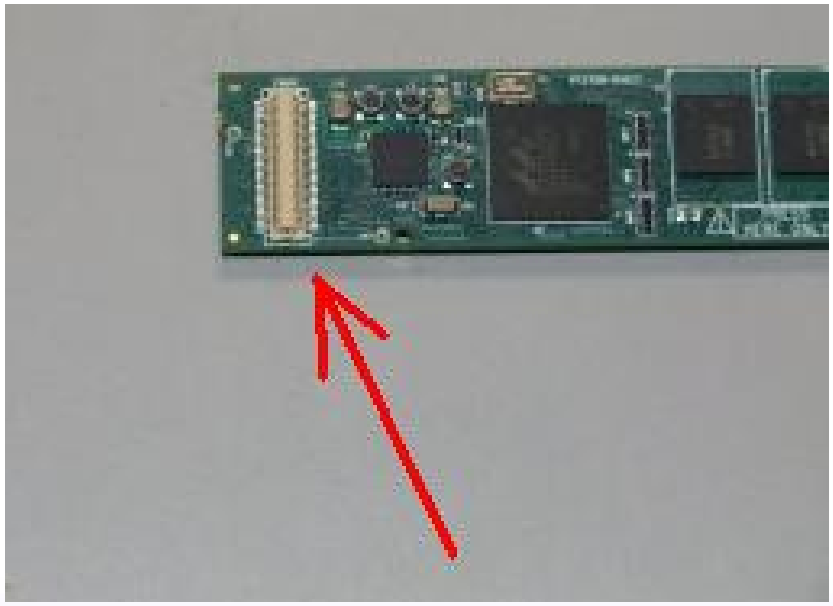


PAX270

- PDA用途の組込み用CPU
- xScale (armベース)
- 豊富な内蔵コントローラ
  - LCD (800x600まで オーバーレイ有)
  - USB (2.0 OTG対応, Fullspeed 12Mbps)
  - AC97
  - シリアル
  - MMC (SD)
  - CF (Wave LAN対応)
  - QuickCapture (C-MOSカメラ)
  - I2C, SPI, PWM,...

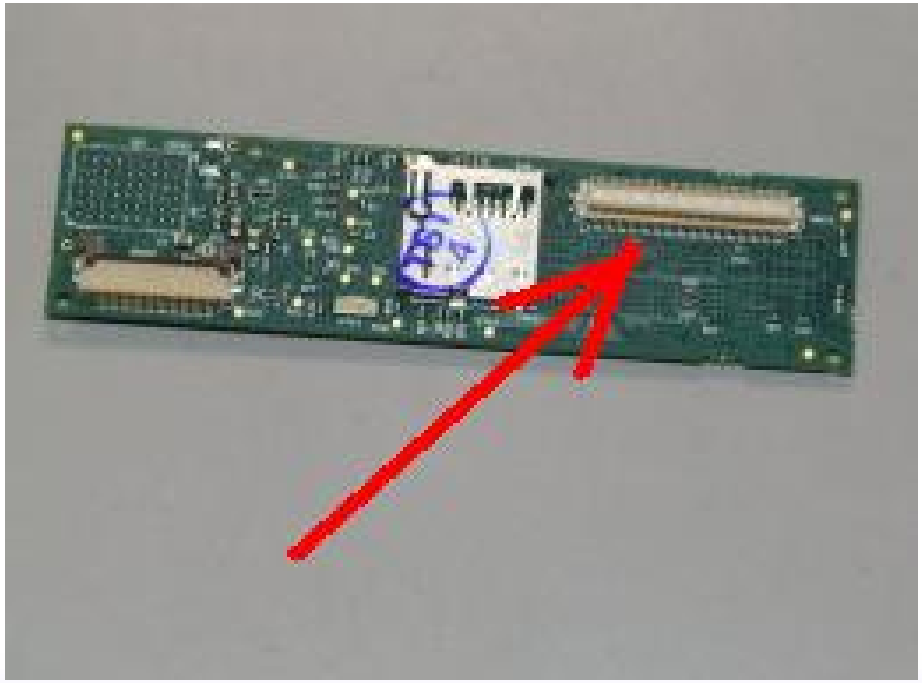
verdex pro 拡張用コネクタ1

- 汎用拡張コネクタ (Hirose 60pin)
  - 電源
  - LCDインターフェイス
  - AC97
  - シリアル通信
  - SPI, I2C
  - GPIO



verdex pro 拡張用コネクタ2

- 拡張IO (Hirose 80pin)
  - MMC (SD)
  - QuickCapture
  - CF
    - Compact Flash (Stroge device)
    - Wave LAN



ハードウェアの設計

- イントロダクション
- ハードウェア編
  - 設計のコツ・注意事項
  - デジタル回路の基礎知識
  - ベースボードの設計
  - プリント基板の製作
- Android移植編

設計のコツ・注意事項

- 実例をコピーする (著作権に注意)
  - データシートのリファレンス(サンプル)
    - 一部が省略されている場合がある
  - アプリケーションノート
- 基礎知識
  - 定石通りにすればOK (あとで紹介)

設計の注意事項1

- 現実の世界はアナログ
  - デジタル回路もアナログで構成されている
    - オームの法則とキルヒホッフの法則は必須
    - ノイズの回り込みで動作不良
      - デジタル回路は、無線回路に近い
    - 配線の銅線は抵抗を持つ
    - 配線は、コイルとコンデンサを形成
    - 部品は理想通りに動作しない
- 電圧の違いに注意
  - デジタル回路に使われる電圧は様々
    - 5V, 3.3V, 2.8V, 2.5V, 1.8V...
  - 異なる電圧は単純に接続できないことが多い

設計の注意事項2

- データシートを必ず読む
  - 互換品や後継品にも注意
  - 日本語版は ほとんどの場合が参考資料
    - 日本製でも英語の資料が正式なことが多い
  - Absolute Maximum Ratings(絶対最大定格)まで使っていいわけでない
    - recommended operating conditions(推奨動作範囲)で使用する

パーツの仕様確認と選定

- 必要な機能を満たすパーツを探す
  - パーツメーカーや代理店のサイト
    - ラインアップの中から比較検討する
    - 他社が互換品や競合品を製造していることも
    - DataSheetは、メーカーが公開しているものを読む
      - ほとんどは英語
      - 日本語は参考資料が多い (リリースされてないことも)
    - 互換品は、その範囲に注意
      - 上位互換や機能限定の互換品もある

パーツの入手性

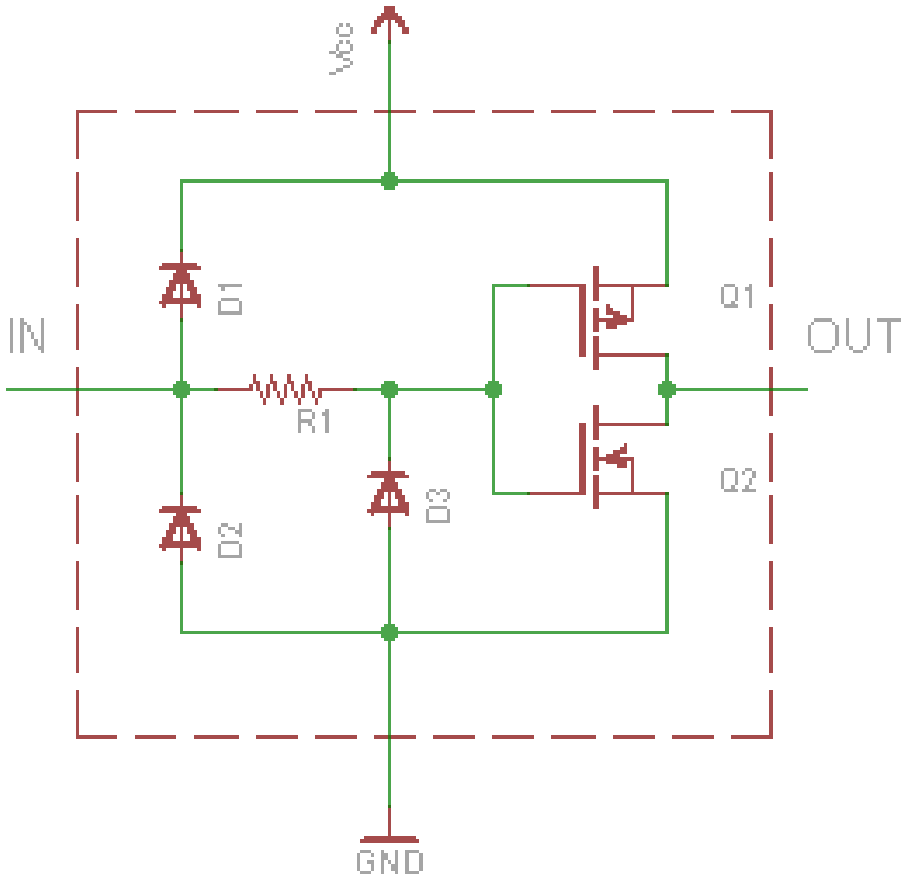
- 入手性をチェック
  - 通販サイトから入手可能か?
    - メーカーダイレクトで買えるものも多い
  - 在庫切れや出荷予定日に注意
  - 同じ機能でもパッケージが違うと困難な場合も
  - 数百やそれ以上のロット単位の場合も
  - RoHS非対応は避けた方が無難

デジタル回路の基礎知識1

- 入力端子は必ず配線する
  - オープン(未接続) は 0V (== L)ではない
  - オープンドレインには Pull-Up抵抗を繋いでHレベルにする
    - オープンな状態は不安定になり素子破損の原因
- 電源より高い電圧を入力しない
  - 入力端子から電源へ逆流して素子破損の原因
  - 一部のICでは、大丈夫な端子もある。
    - データシートで必ず確認する

C-MOSデバイスの構造

- 代表的な構造
  - 入力端子
    - オープンだと Q1, Q2 が不安定になり、大量の電流が流れる



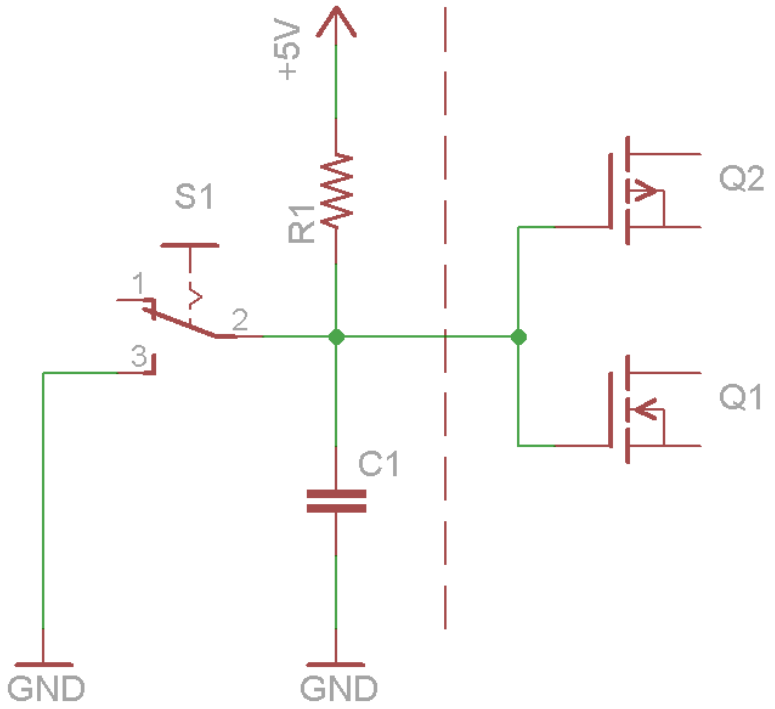
デジタル回路の基礎知識2

- スイッチの入力
  - 開いたとき: Hレベル(Pull-Up)
  - 閉じたとき: Lレベル (GNDに接続)
  - 接触するときは、H/Lが激しく変化する
    - 20msecくらい待つ (スイッチによって異なる)
  - 開いたときにどこにも繋がっていない状態はNG
    - 素子が不安定になり、破損の原因にもなる

入力スイッチの例

- 入力スイッチの例
  - 開放時
    - pull-upにより Hに固定
  - 閉塞時
    - GNDと繋いでLに固定

スイッチを閉じたときでも
pull-up抵抗によりショート
状態にならない

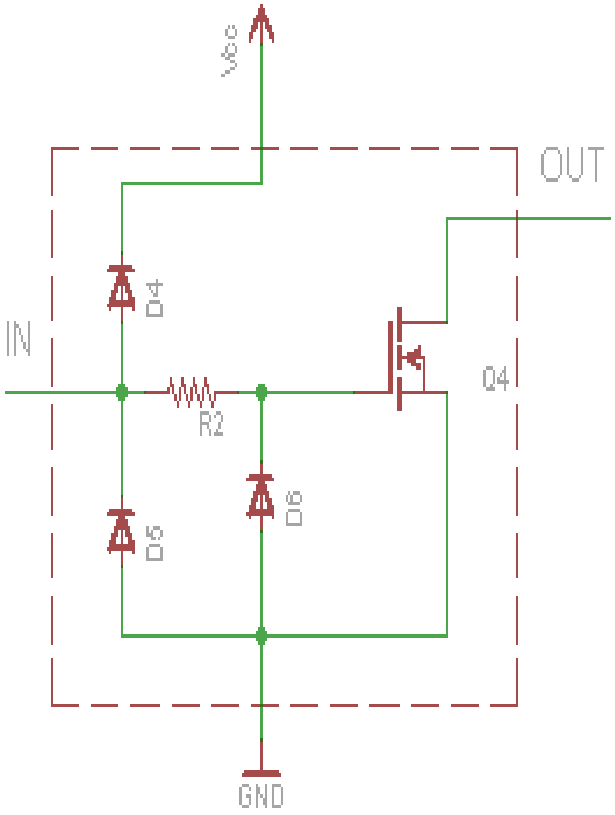


デジタル回路の基礎知識3

- 出力端子は まとめない
  - オープンドレイン出力端子を除く
- 出力端子の電流容量に注意
  - LEDなどの電流は IC全体の消費電流の一部になる
  - LEDなどを接続するときは、ドライブ用のICを介する

オープンドレイン出力

- オープンドレイン
  - FETのドレイン端子がそのまま出力されている
    - ドレインに電流を放出させることが出来る
    - ワイヤード接続
    - 電流駆動デバイスの接続
    - 異電圧デバイスの接続



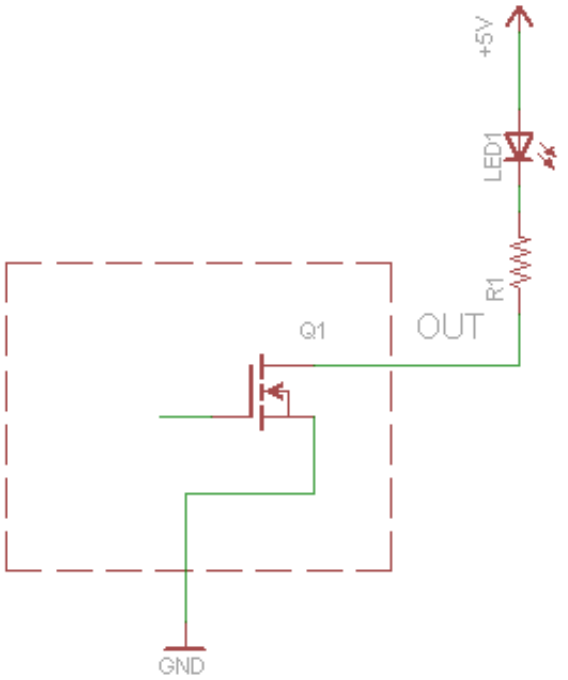
デジタル回路の基礎知識4

- LEDの点灯方法

  明るさは 電流値(平均) に比例する
  - 基本的に直接接続しない
  - ドライブ用のバッファ(オープンドレイン出力)を介する
    - 基本的に LEDを多くドライブする能力がない
  - LEDは(電流制限抵抗)を使う
    - 順方向降伏電圧(ドロップ電圧)を超えると点灯する
      - 赤色LEDなら約1.5～2V (データシートを確認)
    - 抵抗がないと大量に電流が流れ焼損する
    - 抵抗値の計算例:
      - 抵抗値 = (電源電圧 – 順方向降伏電圧) / 消費電流
      - (5V – 2V) / 0.01A = 300Ω

LED接続の一般的な例

- オープンドレイン出力端子に
  - Hで消灯、Lで点灯
    - 出力端子とLEDの電圧差を利用
    - Lの時の電圧は ほとんど0V
    - FETのON抵抗は小さい
      - LEDに流れる電流を計算するときは無視できる
    - TTLの時代は オープンコレクタ出力
    - バイポーラトランジスタから FETに置き換わっただけ



デジタル回路の基礎知識5

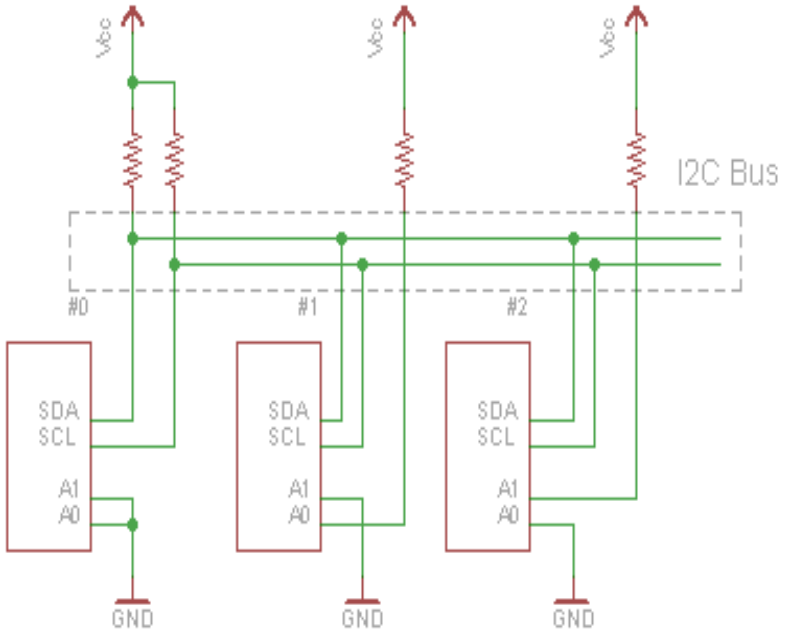
- ICの電源端子にはパスコンを配置する
  - 特にC-MOSタイプは間欠的に電流が流れる
  - パスコンは、一時的な電流不足を補う
    - まとめたり、大容量のものにしない
      - 通常は 0.1μF程度
    - ICから遠く離れては、効果がない
  - ブロック毎に 大きめのコンデンサを追加

デジタル回路の基礎知識6

- 規格に準拠する
  - I2C, SPIなど規格を参照
  - 実例はインターネット上にも豊富
  - 規格が規定している範囲を把握しておく

I2Cバスの接続方法

- 2本の信号線で通信する
  - SDA,SCLを並列につなぐ
  - Pull-Up抵抗を入れておく
  - デバイスアドレスを指定して通信
  - マスタからスレーブに向かって通信
  - どのデバイスもマスタになれる



ベースボードの設計

- 基本的な手順
  - 実装する機能の検討
  - パーツの選定 (仕様確認)
  - 回路図設計
  - プリント基板設計 (アートワーク)

実装する機能の検討

- gumstixの搭載
  - 電源供給
- タッチスクリーン付液晶パネル
  - タッチスクリーンインターフェイス
  - バックライト
- オーディオ (AC97)
- シリアル通信 (コンソール/GPS)
- 3D加速度センサ
- RTC (Real Time Clock)
- 入力スイッチ
- プログラマブルLED

gumstixの接続

- gumstix用コネクタを配置する
  - Hirose製60極コネクタ
  - コネクタから各種IOが配線されている
    - 対になるコネクタの型番が公開されている
- 電源を配線
  - DC 5V 1Aを供給

カラー液晶パネルの接続

- consoleLCD16-vx と同じ機能を実装
  - 回路図が公開されているのでコピーする
  - LCD用コネクタを配置
    - 位置に注意
    - LCDデータバス(16bit), コントロールバスを接続
    - タッチスクリーン用回路を配線
    - バックライト用回路を配線
      - 当初は使っているICが不明だったので互換回路を製作

電源回路 (Li-ion電池)

- DC 5V入力、DC5V出力
  - gumstixCPUボードと 拡張ボードに電力供給
  - Li-ion電池に対応
    - Li-ion電池に対応した専用IC (MAX8677)を採用
  - 出力は DC 5V 2Aに対応
    - Li-ion電池の出力(3～4V)を 5Vに昇圧
    - DCDCコンバータ (MAX1708)を採用
  - 電源スイッチ
    - 半導体リレーで on/off

AC97 (オーディオ)の搭載

- Audiostix2 と同じ機能を実装
  - 回路図が公開されているのでコピーする
    - 但し、少しだけ変更 (MICとLINE INを分離)
    - ミニステレオプラグを配置
      - 配置する位置に注意 位置決め用の小さい穴も

シリアルポートの搭載

- シリアルコンソール用
  - consoleLCD16-vxの FFUART用と同じ
    - 3.3V TTLレベルとRS232-Cレベルの変換
- GPS用
  - 接続するGPSの通信仕様にあわせる
    - 今回のGPSモジュールは 3.3V TTLレベル仕様

    → そのまま接続する

3D加速度センサの搭載

- gumstix社の拡張ボードにない機能
  - インターネットで候補を探し搭載する
    - I2C, SPIで制御可能なもの (デジタルIF)
  - I2Cで接続
    - I2Cの規格に沿って配線

設計のチェック

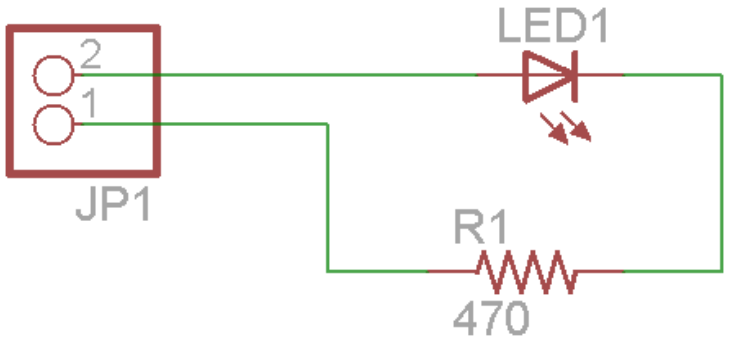
- 全体を通した、接続のチェック
  - 全てが正しく配線されているか
  - パーツの推奨動作範囲を満たしているか
  - 保護回路は充分か
    - 過電圧、コネクタの逆接続など…
  - 動作モードがある場合
    - 動作切り替えの対応

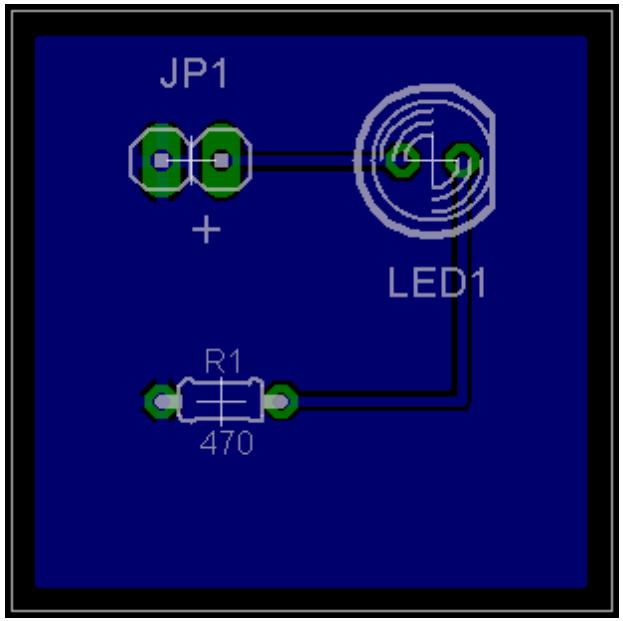
プリント基板の製作

- お断り
  - 実際のbc9を例にしていると時間が足りません
  - 今回は、非常にシンプルな例で説明します
    - 基本的な手順は同じです
  - bc9の回路図、ボートデータはWebで公開

プリント基板の製作

- 発注データの作成
  - CAMソフトウェアを使う
    - ここではEAGLEを紹介
- プリント基板の発注

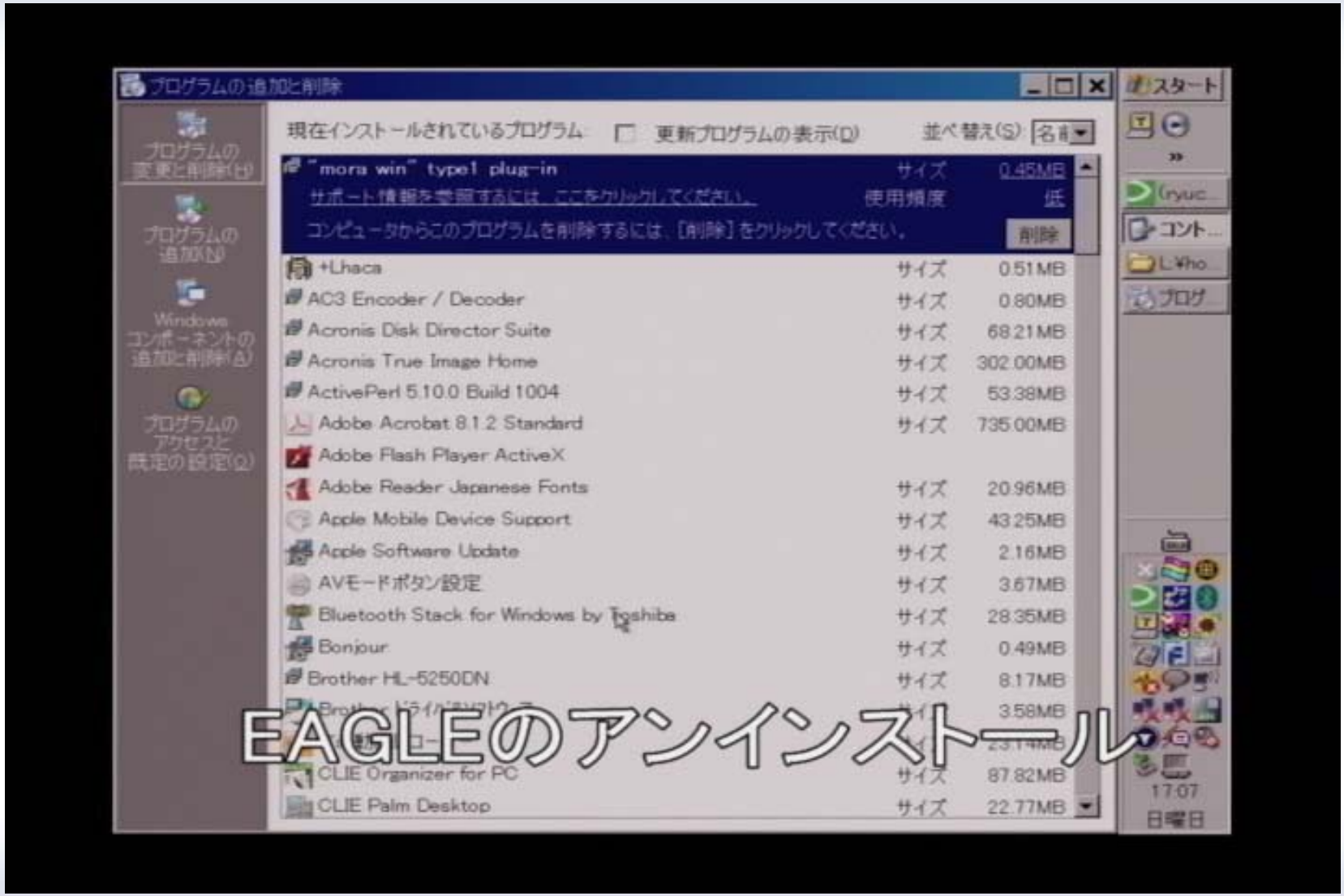




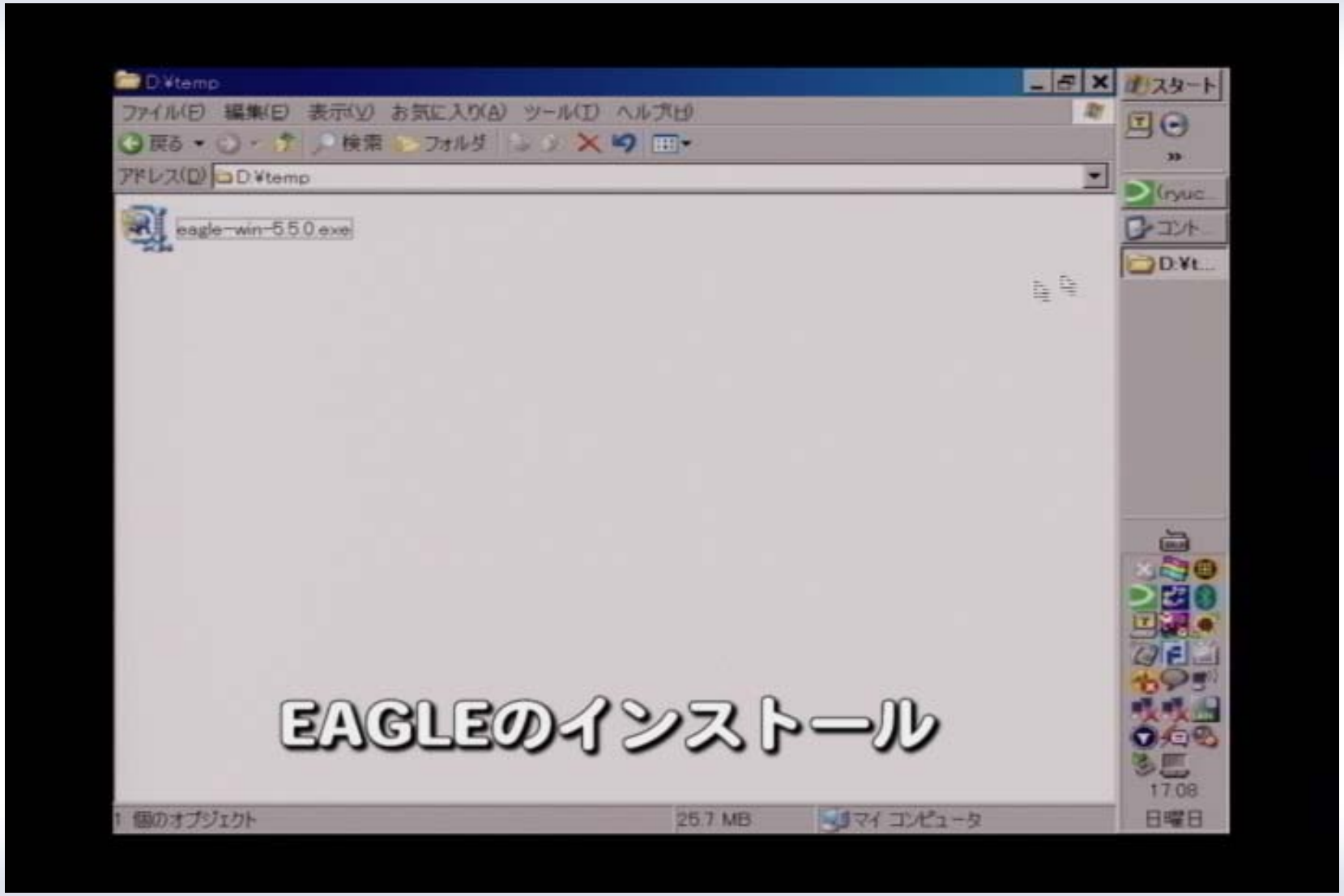
EAGLEの入手

- CadSoft社 (独)の製品
  - 最新版は 5.6
    - 連休中にリリースされたらしい
      - デモビデオは 5.5で撮影してしまいました
  - 英語版と独語版がある
  - Windows, Linux, MacOS版がある
  - 使える機能によって いくつかのEditionがある
    - 同一バイナリでライセンスファイルで切り替わる
    - フリーソフトウェア版は、機能や目的が制限
    - 日本では、Circuit Boards Service, Inc.が販売
      - http://homepage3.nifty.com/circuitboards/v2_software/EAGLE/v2_software1.html
      - 日本語訳されたマニュアルも販売されている

旧版をアンインストール



EAGLEのインストール



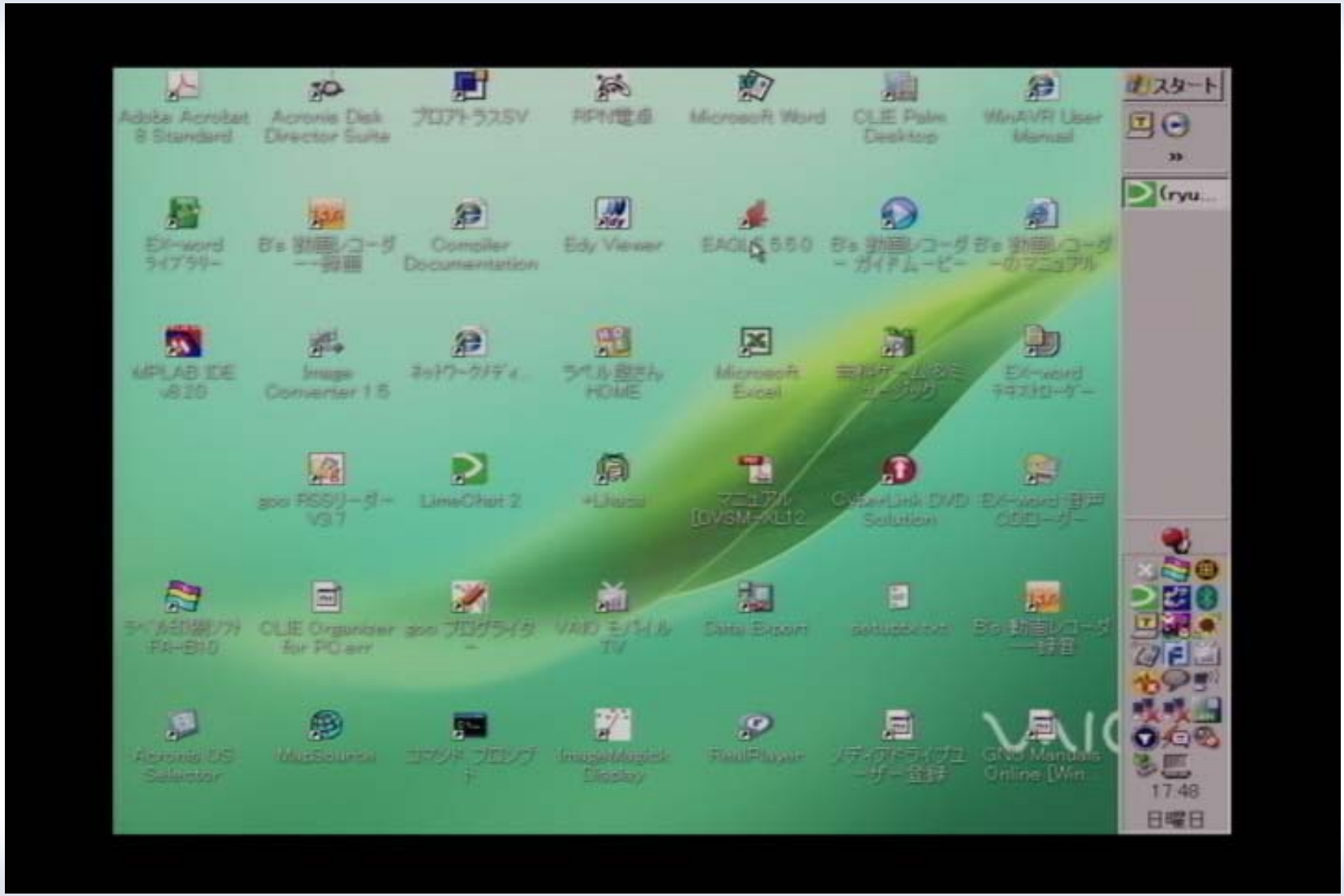
EAGLEの基本的な使い方

- EAGLEの基本的な使い方
  - 部品ライブラリの製作
  - 回路図の作成
  - アートワークの作成
  - gerberデータの出力

EAGLEライブラリ

- 部品のデータを集めたもの
  - パーツのフットプリントや端子の機能など
  - パーツのフットプリントが違うと実装できない
  - 標準で多くの部品が収録されている
    - 標準化されている抵抗、コンデンサなど
    - 古くからある、オーソドックスなICなど
- ライブラリにない部品も多い
  - ライブラリにない部品は作成する
    - 端子間隔の狭いIC類
    - コネクタ、スイッチ類
    - 新しいものや欧米での流通が少ないもの

EAGLE ライブラリ



回路図の作成

- 回路図エディタで回路図を作成
  - 正確に作成する
    - 不正確な回路図は後日の作業ミスの原因
  - ライブラリから正しいパーツを選択する
    - プリント板のフットプリントになる
    - 大きさが違えば、互換性がない

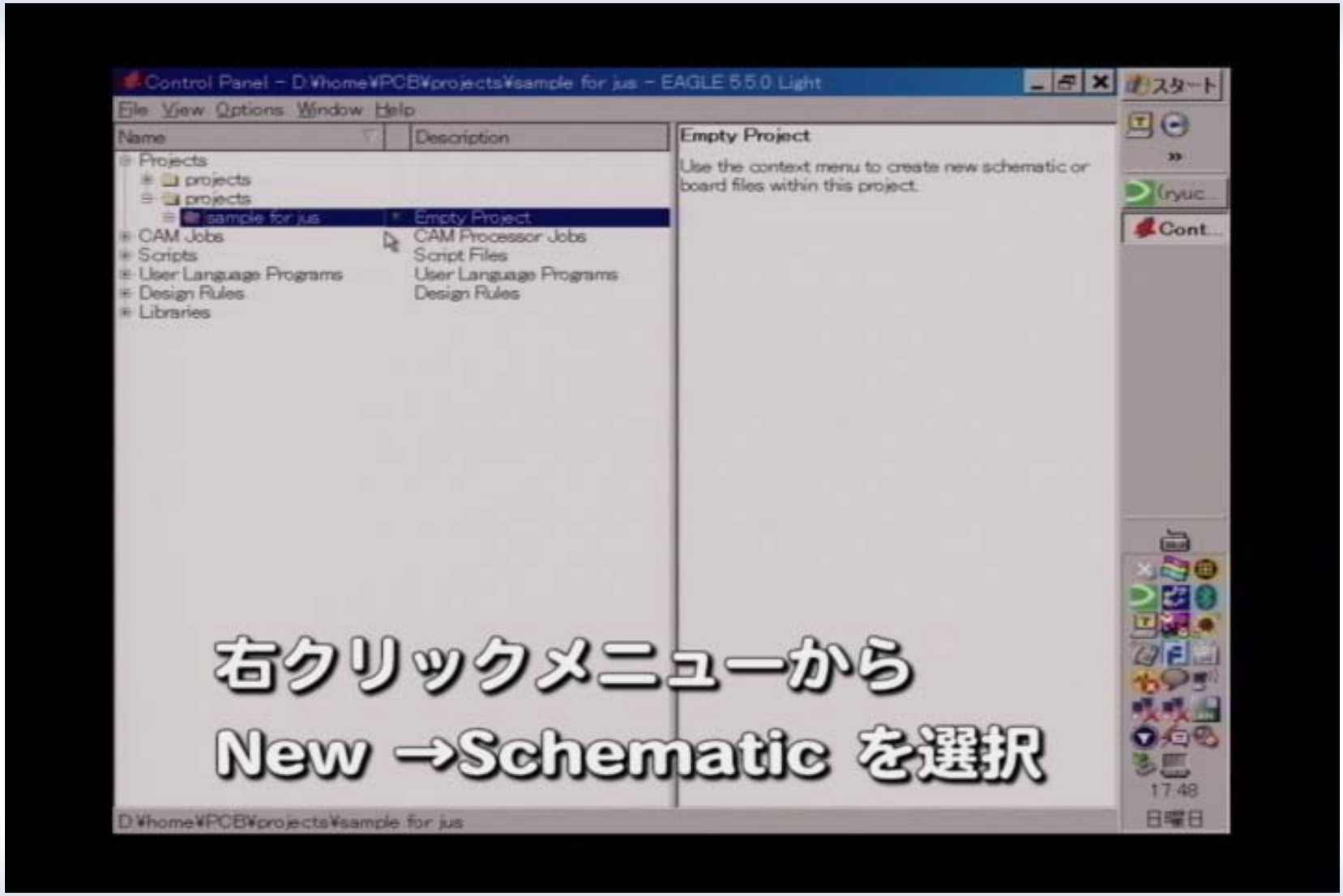
アートワーク作業

- 発注先の仕様を事前にチェックしておく
  - 最小パターン幅や最小クリアランスなど
- 部品を配置して、配線を行う
  - 大まかな位置を決める
  - ICなどの近くに配置するコンデンサに注意
  - 組立て作業時のことも考える
- オートルート機能もある
  - 部品を適切な位置には配置してくれない
  - 部品の配置を変えながら何度も試す
  - 最初から手作業で、配線するほうがいいかも
- 配線ができたら必ずチェックツールで確認
  - クリアランスやオーバーラップをチェック

プリント基板データの出力

- EAGLEネイティブ形式
  - EAGLEの形式を受付けるメーカーもある
    - .brd ファイルを渡すだけ
- 業界標準のデータ形式
  - gerberデータ RS274-X形式
    - この形式で製造できないメーカーはないらしい
    - EAGLE添付のCAMプロセッサで出力可能
    - 基板の各層毎のパターンファイル
      - これをまとめて、メーカーに渡して製造してもらう

回路図/レイアウト作成例



プリント基板の発注

- オンラインで受付けるメーカーが多い
- メーカーの指定に合わせてデータを送信
  - EAGLE or Gerber データ
- bc9用のボードの場合
  - 納期: 約1週間 (もっと早いコースもある)
    - 短納期は、金額が高くなる
  - 金額: だいたい 2万～3万円 (10～20枚)

最初は日本のメーカーがいいでしょう

安い海外メーカはなれて自信がついてから

パーツの発注

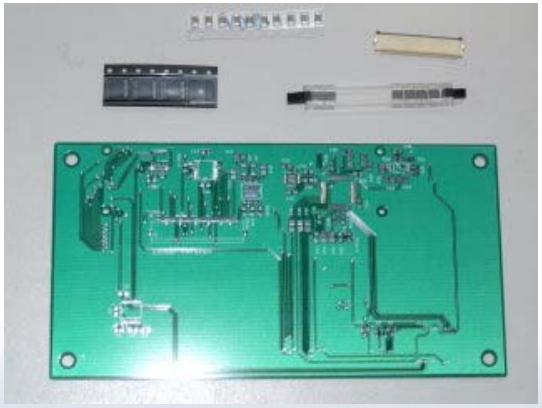
- 必要なパーツを発注して入手
  - Digi-Keyなど
    - 基本的に米国から個人輸入だが航空便で早い
    - 大量に買うと安い
      - よく使う部品は、まとめて買う
    - いくつかのオンラインサイトを使う
  - 秋葉原の店頭では、汎用品＋αしか買えない
- 入手性は既に確認済みのはず
  - パーツが足りないと機能しない
  - 設計からやり直し
  - 基板は作り直しができない

組み立て作業

- 部品が揃ったら組み立て作業
  - 組み立ての順序を検討してから作業
    - 基本的に背の低い部品から
    - ICを先に実装
    - ICの周りの抵抗やコンデンサなど
    - パーツの取り付け向き、同じ大きさの部品に注意
    - 多ピンのコネクタ類は、再生が難しい

組み立て作業

- 半田こてでひとつひとつ丁寧に…



動作チェック

- 組み立てが完了したらチェック
- 動作させる前に
  - 取り付けミス(違う部品、取り付け向き)がないか
  - 半田付け不良がないか
    - 半田の浮き、割れ
    - 見えにくい場合は、4～10倍以上のルーペで
- 異常を感じたら
  - 電源を切って 回路図なども再チェック

おまけ bc9設計・製作談1

- 最初は半田付けに苦労しました
  - 最近の電子部品のほとんどはチップ部品
  - 抵抗やコンデンサは 1.6 x 0.8mmが多い
    - 最近の機械実装だと 0.8 x 0.4mmなどもある
  - ICやソケットなどのピッチは 0.5mmが多い
    - 小さいものは 0.3mmもある
    - ピン数は ～80ピンや それ以上も...
  - リードレスタイプもある
    - QFNなどなら手半田付けは可能
    - BGAだと手半田は無理

おまけ bc9設計・製作談2

- 複雑な配線に苦労しました
  - アートワーク作業に多くの時間を割きました
  - プリント基板は 2層または 4層
    - 同じ層で配線が交差してはいけません
    - 裏や別の層を経由するにはスルーホールが必要
      - 配線密度がさがり、配線可能な面積が減る

CAMソフトのデザインルールチェックを活用

おまけ bc9設計・製作談3

- 動かないときの原因の追究
  - なぜ動かないのか、わからない
    - 半田付け? 回路設計ミス?
    - デジタル回路も繋がっていればOKではない
  - 結局はアナログな世界
    - 理想的な回路は存在しない
    - 電子が流れる気持ち、電子が受ける力

bc9 へのAndroid の移植

- 我々は bc9 の OS として Android を選択しました。
- 実機への移植を通して学んだことをお話します。

- ところでなぜ Android を選択したの？

OpenEmbedded

- OpenEmbedded

組み込み向け Linux distribution

正確には、組み込み Linux をビルドできるクロスコンパイル環境

対応プラットフォーム 165

OE/org.openembedded.stable/conf/machine/

の中の定義ファイル数より

パッケージ数 1609

OE/org.openembedded.stable/packages/

の中の定義ファイル数より

gumstix OpenEmbedded

- gumstix verdex/verdex pro は OpenEmbedded をビルドシステムに採用
- OpenEmbedded の local 設定作成機能を使って gumstix 用にカスタマイズされています。

  gumstix-oe/com.gumstix.collection/

  以下に独自の設定が全て纏められています。

NetBSD

- NetBSD 5.0

NetBSD/evbarm

Gumstix verdex support

NetBSD Current-users 26 Jan 2009 の投稿より

http://mail-index.netbsd.org/current-users/2009/01/26/msg007601.html

OpenBSD

- OpenBSD 4.5

4.5 What's New より

http://www.openbsd.org/45.html#new

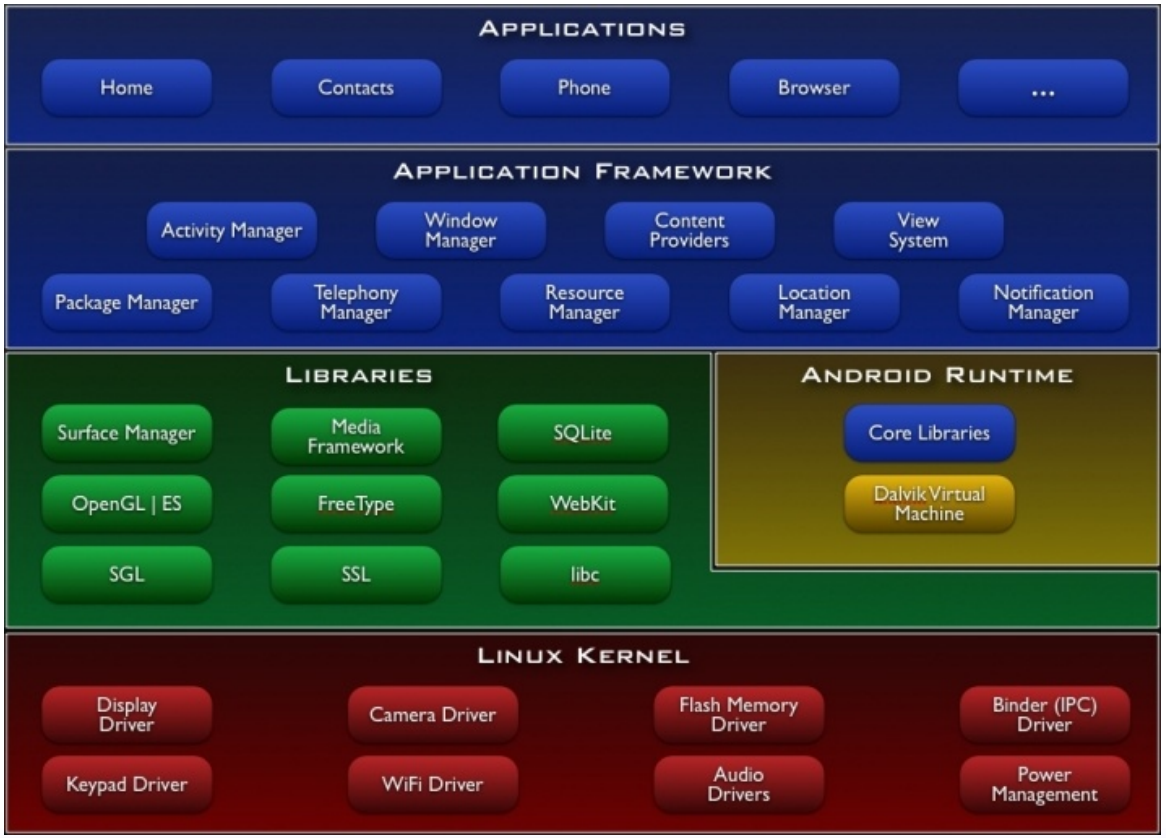
- New/extended platform
  - Initial ports to the xscale based gumstix platform and the ARM based OpenMoko

Android

- プラットフォームとして新しい。面白い。
- ARM前提で互換性がある。
- GUIが軽そう。
- Android の中の人たちとの宿縁
- open source 版(Android Open Source Project)公開後、gumstix-users ML で話題になったが3月時点で移植成功報告なし
  →やってみよう。

Android の構成

Android は以下の図のような構成になっています。



kernel version

- Kernel 2.6.25 (Release 1.0 branch)
  - http://code.google.com/p/android/download/list

    2.6.27 (master branch, cupcake branch)

    2.6.29

    git://android.git.kernel.org/kernel

Android の kernel 変更点

- Goldfish
  - SDK emulator の CPU ARM926T SoC 対応
- YAFFS2
  - NANDでmemmap するための新ファイルシステム
- Bluetooth
  - bugfix と access control 機能の追加
- Scheduler
- **Android 独自の追加新機能**
- Power Management
- その他の細かい変更

kernel への追加新機能

- binder

  IPC（プロセス間通信）のためのモジュール

  – kernel/drivers/misc/binder.c

- lowmemorykiller

  不要なプロセスをkillするモジュール

  – kernel/drivers/misc/lowmemorykiller

- logger

  DalvikVM上のuser process からも読み書きできる log モジュール

  – kernerl/drivers/misc/logger.c

- ashmem

  Anonymous Shared Memory

  – kernel/mm/ashmem.c

- Alarm

  RTCや経過時間を元にwakeup alarm 機能を提供するモジュール

  – kernel/drivers/rtc/alarm.c

移植必須のdrivers

Android の起動に不可欠なのは

- ashmem
- binder

の二つだけ。

(これがないと Android の init が segfault で起動もしません。)

最低限この二つだけ移植すれば Android Runtime は動作します。

移植必須のdriversその二

が、これだけだとうまく動かなかったとき何が起こっているのかさっぱりわかりません。

ですので、

- logger

も必ず動作するようにしましょう。

kernel の build

- kernel の Android 対応というのは要するに追加のdriver を足して build すればいいということです。
- Kconfig と Makefile に追加した driver を足して make menuconfig, make で build できれば Android対応化は完了です。
- ただし若干の注意点あり

注意点

- bc9 は OpenEmbedded による gumstix 対応独自patch の関係でkernel 2.6.24 を使用していて logger がそのままでは build が通らないため、修正を加えました。（kobjの変更によるもの）
- Android の userland には insmod や modprobe がありませんので、後で純 Android 環境にしたい場合 driver をモジュールにしてはいけません。

kernelのAndroid対応後

- Android の独自機能を追加しても linux kernel そのものなので、この kernelで全く問題なく linux userland を起動することができます。
- Android の system library は携帯電話向けにごくごく限定されたものしか入れられていないのでとても不便。

  いったん linux として起動してから chroot で Android を起動すると問題点を探りやすいです。
- userland を Android のものだけにしてしまうのは chroot で起動できてからでOK。

Android userland

- ここではkernel 以外の部分をまとめて userland と呼んでいます。
- Android Open Source Project の git repositry から取得したものを手順通りに make して得られた ARM 版のuserland 一式は

  mydroid/out/target/product/generic/

  の中にあります。Android SDK の emulator や G1 Phone/dev phone の中身と同じものです。

build されたもの

```
ls mydroid/out/target/product/generic/
 android-info.txt
 clean_step.mk
 data/
 obj/
 ramdisk.img
 root/
 symbols/
 system/
 system.img
 userdata.img
```

Android の /

```
ls
  mydroid/out/target/product/generic/root
  /
 data/
 default.prop
 dev/
  init
 init.goldfish.rc
 init.rc
 proc/
 sbin/
 sys/
 system/
```

chroot で起動してみる

- android-root/ というようなディレクトリを作ってその中に Android の / を配置したら起動してみます。

  # chroot android-root /init

- うまくいくとandroid_の文字が出たあとブートアニメーションが始まり、少し経ってから Android の UI が表示されます。

- ただ、なかなか一発では起動しませんでした…

strace

strace –f –ff –tt –s 200 chroot android-root /init

Android の init の処理のどこで止まっているのか調べましょう。

-f forkしたプロセス全部
-ff 各プロセスを全部ファイルに書き出す
-tt microsecond で実行時間を記録
-s 200 出力するstringサイズを200に

logcat

- kernel に追加した logger が正しく動いていると、logcat コマンドが使えます。
- Android Runtimeの起動過程を全て追うことができます。
- Android の各プロセスは正しく起動しているようなのに画面に何も表示されないといった場合に有効です。

Android Debug Bridge

- Android Debug Bridge (adb)

http://developer.android.com/guide/developing/tools/adb.html

- network が使える状態になっていると、Android の起動時に /sbin/adbd が service として実行されます。(initd.rc を参照)

- Network さえ使えれば android SDK の tools/adb で実機に接続できます。

adb logcat

- adb shell でAndroidをコマンドラインから操作できます。
- adb push でファイルを送り込んだりadb pullで取り出したりできます。
- adb install で apk をインストールできます。
- adb logcat でリモートから logcat を実行できます。制限の少ないリモートのクライアントで log を取ってあとは地道に

  debug debug.

tool の追加

- Android の /system/bin/ には busybox のような toolbox という multicall binary がありますが使いにくいです。sh も補完がきかず作業が面倒です。
- そこで、自分で busybox やbash を build して送り込んでしまうと便利です。

サイズに目を瞑ってstatic link のbinaryにするか、Android.mk を書いて dynamic link のbinary を作るかはお任せします。

Make Tokyo Meeting 3

- Make Tokyo Meeting 3に参加します

- 場所: デジタルハリウッド 八王子製作スタジオ
- 日時: 2009年 5月23日(土)～24日(日)

- http://www.oreilly.co.jp/mtm/03/