

# オペレーティングシステム

第12回(2009.07.02)

組み込み機器用OS

# この部屋に「コンピュータはいくつある？」

- 「コンピュータ」=「箱」と考えると・・・
  - 「コンピュータ」という名前がついている機械の数
- 「コンピュータ」=「CPU」と考えると・・・
  - 様々な電子機器にコンピュータが使われている
    - 腕時計
    - OHP(プロジェクタ)
    - テレビ(特に、デジタル放送用)
    - ビデオ
    - ファミコン、ポータブルゲーム機
      - (誰だ！カバンの中がゲーム機だらけの学生は！)
    - 自動車(制御UNITに一つずつ？)
    - コピー機
    - FAX
    - などなど、などなど

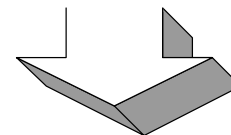
# 見えないコンピュータ

- 「人」が使うコンピュータ
  - モニター画面＋キーボード(+マウス)が必須
    - 人が結果を見て、人が操作する
- 「機械」が使うコンピュータ
  - モニターは不要
  - 操作の入力機器は、機器内部にある
    - か、または、数個のボタンの組み合わせに、様々なメニュー機能を盛り込む
      - 「操作が難しい」機械が出来上がる。
  - タッチパネルで操作させる
    - マン・マシン・インターフェース

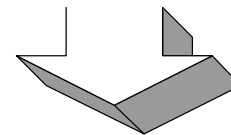
# 電子機器の制御（歴史的背景）

- アナログ信号を演算増幅器などで処理し、専用回路で制御信号を生成した。
  - 「仕様」が変わるたびに、大幅な設計変更
- A/D変換でデジタル化し、デジタル演算で制御する。
  - ロジックをシンプルに出来る。
- A/D変換し、制御はプログラムで行う。

アナログ系  
ハードウェア技術者



デジタル系  
ハードウェア技術者



ソフトウェア技術者

# 電子機器の制御（歴史的背景2）

- 小型CPUのメモリの制約
  - 主記憶2キロバイト、など必要最小限のコードしか書けない
  - ハードウェアの初期化部分など、専用周辺ICで対応
  - 毎回、制御用プログラムを「書き下ろし」する
- 制御構造の複雑化
  - より高機能を求められる
  - コードが複雑化 ⇒ 共通部分のオーバーヘッドが大きい
- 小型CPUが進化した、というよりも、「普通のパソコン」が機器の中に入り込んできた。
  - よりきめ細かな制御 ⇒ 複雑なロジックが求められる。

# 組み込み用CPUの例

- H8SX/1635仕様
  - 内部バス:32ビット
  - 電源電圧:3.0~3.6V
  - 最大動作周波数:50MHz
  - 32ビット乗算／除算器内蔵
  - H8SX/1631F(Flash 256KB, RAM 24KB)
    - A/D変換, D/A変換搭載
- 通常のCPUよりも遅い
- メモリサイズが小さい
- 低消費電力

H8SX/1635グループ ブロック図

H8SX-CPU @50MHz	ROM FLASH	RAM	クロック 発振器
INTC		BSC	
TPU0:6ch		PLL	
TPU1:6ch			
PPG0		TMR(8bitタイマ) :8ch	
PPG1			
DMAC:4ch		WDT	
ADC 10bit:4ch		DTC	
ADC 10bit:4ch			
SCI:7ch		DAC 10bit:2ch	
I2C:2ch*		I/O port	
UBC		JTAG I/F	

\*I2Cバスインターフェースは、Philips社の提唱しているI2Cバス(Inter IC Bus)インターフェース方式に準拠しており、サブセット機能を備えています。ただし、I2Cバスを制御するレジスタの構成が一部Philips社と異なりますので注意してください。

■:特徴機能

# 汎用OSと異なる環境

- ハードウェア構成が決まっている。
  - 「どんなメーカーのハードディスクでも」つなぐということはない
  - 初期状態のパターンが決まっている場合が多い。
- 初期化の時間がゆっくりとれない。
  - 電源ONしたら、早い時で1秒以内、遅くても数秒でスタンバイするようなニーズが多い。
  - 高速性を最優先したプログラムが求められる。
- ユーザの操作で個別のプロセスが起動することはない
  - 「スタート」ボタンのような、数種類の起動要求はある。
  - すべて機器側の都合で動作させるプログラムが決まってくる。
- マンマシンインターフェースは、必要最小限
  - キーボードやマウスは存在しない。
- ADC, DACなど、アナログ信号を入力とする場合が多い

# ADC, DACって何？

- A/D Converter (Analog/Digital変換)
  - アナログ信号をデジタル信号に変換する。
  - マイクロフォンの「音」をデジタルに変換する。
- D/A Converter (Digital/Analog変換)
  - デジタル信号をアナログ信号に変換する。
  - 例：音源ICのデータを、「音」に変換する。
- 温度、光（明るさ）、音などのデータを、コンピュータで扱える形に変換する回路



# 汎用OSと共通の要求事項

- OSと名がつく以上・・・
  - 開発者は、「アプリケーション」プログラムを書く感覚で使える
    - ハードウェアの細かい構成を、必要以上意識しなくて済むようにする。
      - ⇒ OSの役割(仮想化技術)
      - ハードウェアとソフトウェアの両方を知っているエンジニアが不足している。
  - 同じOSであるならば、他の機種に移植しても「そのまま」動作する。

# 「組み込みならでは」の要求

- コードは短く、処理は速く、データは小さく。
  - OSのコードは、できる限り小さくする。
    - ターンアラウンドタイムの上限が切られている処理が多い。
      - (500msとか、100msなど、高速処理が必須)
  - 高速演算を可能にするロジックを用いる。
    - 読みやすさよりも、演算の高速性
    - コンパイルされた後の、機械語(アセンブラ)レベルで、ニーモニックごとの実行サイクルを意識して作る。
  - データのサイズは、できる限り小さくする。
    - ビット単位で、必要なサイズを圧縮する。
      - 代入演算ではなく、ビットレベルのAND/OR演算を用いる。

# リアルタイムOS

- リアルタイム処理
  - 「実時間処理」 すぐに応答を返すシステム
  - リアルタイム性：
    - 完全に「遅延」が存在しないシステムは、実現不可能
  - ターンアラウンド時間が短い
    - 非常に短い時間に処理されるため、利用者にとって「ほとんどリアルタイム」に見える。
      - ⇒ これで十分
      - どの程度のターンアラウンドが要求されるか、策定する。
- リアルタイムOS
  - リアルタイム処理を行うOS

# ITRON

- 東京大学の坂村健教授に提唱された
  - 国産のリアルタイムOS
    - 携帯電話などに組み込まれた。
    - 2001年の携帯電話搭載率 100%
  - TRON: The Real-time Operating system Nucleus
  - T-Engine
    - 組み込み機器用のオープンソースのkernel
    - 情報端末の標準化を目指す。
- TRONプロジェクトの一部
  - Industrial TRON
  - BTRONやCTRONなど、他のTRONもある。

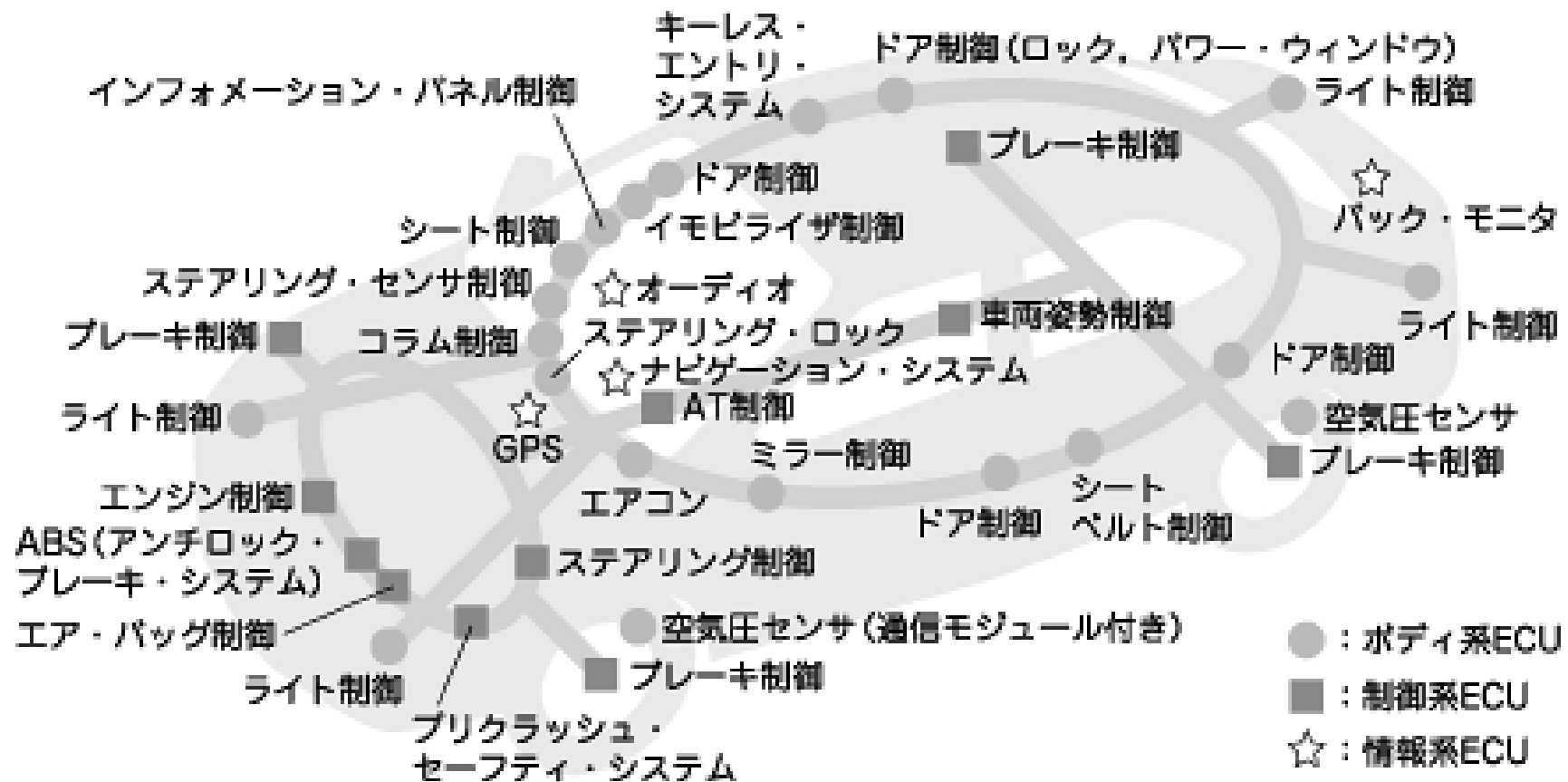
# OS搭載携帯電話

- 携帯電話にOS？
  - 携帯電話には、様々なアプリケーションが搭載
    - 電子マネー/Suicaなど、認証が必要なもの
    - 住所録など
  - 「電話」⇒話すための道具
    - ⇒ 通信端末としての役割
  - 「携帯」⇒持ち運ぶ「端末」
- OSを搭載することで、開発を効率化

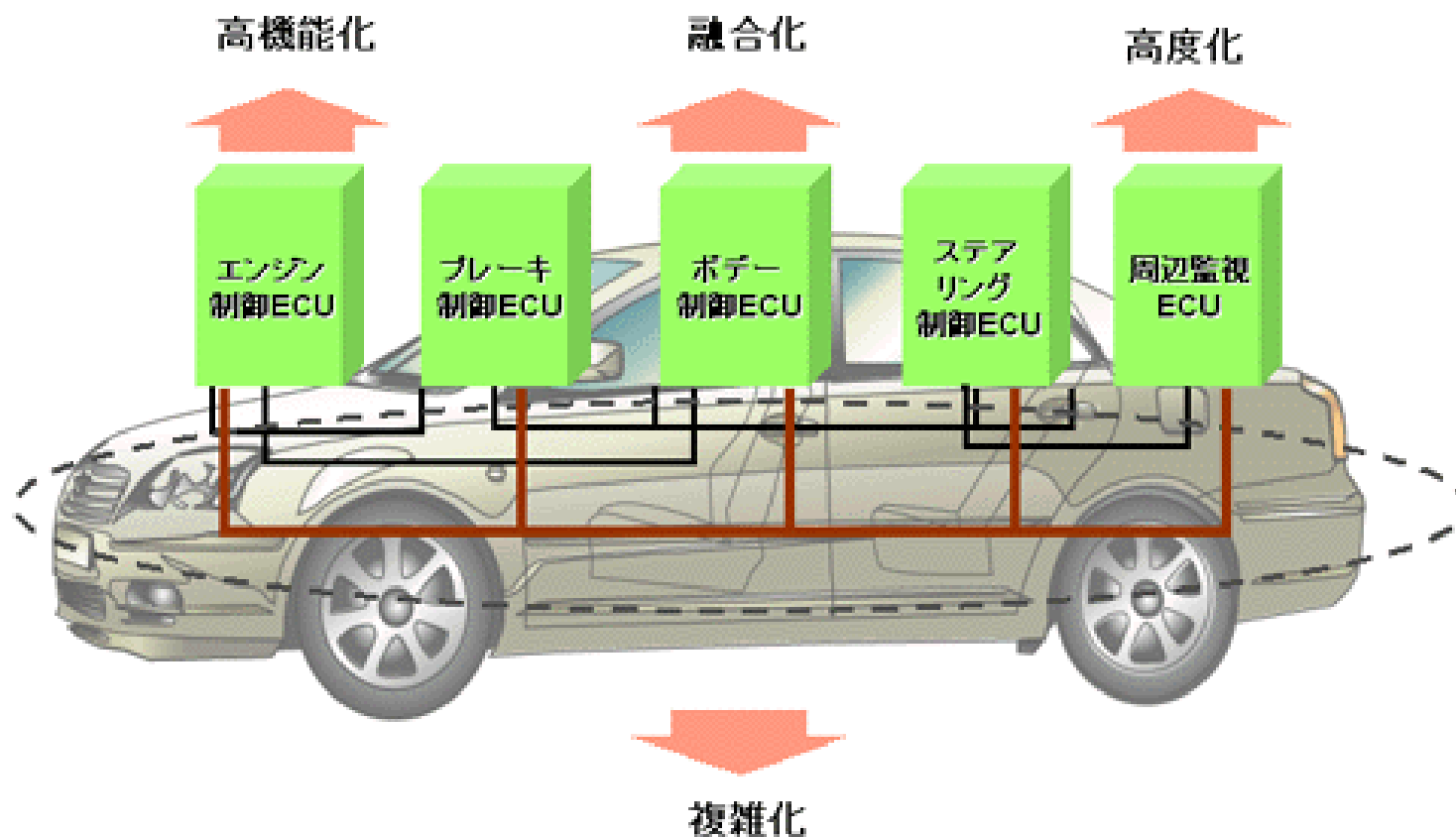
# Symbian OS

- 現在シェアを伸ばしている携帯端末用OS
  - SO906i (SONYのDocomo携帯)はSymbian OS搭載
  - 日本国内の搭載端末は83機種 (2008年6月現在)
- 仕様
  - 32bit OS
  - マイクロカーネルアーキテクチャ
  - マルチタスク、マルチスレッドに対応
  - 開発環境を安価に揃えられる
  - 内部文字コードはUnicode(UTF-8)で国際化が容易
  - キー入力、ペン入力をサポート

# 自動車と制御



# 自動車用OSの共通化 (JasPar)



経済産業省と自動車メーカー10社と共同で自動車の電子制御用OSを開発と言う事で話が進んでいます。

<https://www.jaspar.jp/guide/background.html>



# Bluetoothって何？

- 無線通信の規格
  - IEEE 802.15.1
  - 近接したデバイス（機器）とデバイスの間を2.4GHzの周波数帯を用いて電波での情報のやりとりを行う、無線通信規格およびその技術である。(Wikipedia)
  - 機器間をワイヤで接続する代わりに、電波で交信させる。

# ロボットの制御

- 組み込み制御の特徴
  - ユーザからの入力の代わりに、センサーによって測定された情報が入力になる。
  - 画面表示の代わりに、制御信号が出力になる。
    - 産業用ロボットの延長
- いわゆる「ロボット」は組み込み制御技術の延長。

# 組み込み用Linux

- <http://lss.etsy.com/new/class.php?200>
- 用途: 家電用、ロボット制御用、PDA用、携帯電話向け、Bluetooth向け
- 用途向けに、様々な分化している。

- aqua-vit          BusyBox          CELF          diceMP
- EDK              ELDS              ELITE          FineWS
- JFFS2            Linux(A&A\_Linux)   Linux(ART-Linux)   Linux(axLinux)
- Linux(BlueCat)   Linux(cclinux)   Linux(CE\_Linux)   Linux(DIAPM-RTAI)
- Linux(DIAPM-RTAI)   Linux(DSPLinux)   Linux(Embedix)   Linux(EMJ-Linux)
- Linux(Hard\_Hat\_Li...Linux(KURT)          Linux(Linux\_DA\_O/S)   Linux(miniRTL)
- Linux(Mobilinux)   Linux(MontaVista\_...Linux(PeeWeeLinux)   Linux(RedHawk)
- Linux(REDICE-Linux)   Linux(RT\_Linux)   Linux(SiliconLinux)   Linux(TASTE)
- Linux(TimeSys\_Linux)Linux(Tynux)          Linux(uLinux)          Linux(VMELinux)
- Linux(ZetaLinux)   PEG              PLDE              RED-Builder
- RED-Probe          RedBoot          Red\_Hat\_Embedded\_...sgsdk
- sikigami          USBLink          XTNDAccess\_Blue\_SDK

# 組み込み用Windows

- Microsoft Windows XP Embedded
- Microsoft Windows CE
  - Windows用のプログラム開発は、開発者人口が多い。
    - ⇒ 開発者を集めやすい。
    - テスト環境などにWindowsを使うことで、開発が容易になる。

# おまけ

## -- 携帯アプリの開発環境 --

- Javaを覚えたら携帯アプリを作ってみよう！
- <http://allabout.co.jp/internet/java/closeup/CU20070414A/>
- 携帯端末用アプリケーション
  - 小メモリ、Event-Driven、リアルタイム性
  - 組み込み系の要素を持っている。

# まとめ

- 組み込み用OSは
  - ターンアラウンドタイムが短い
  - メモリの制約が大きい
  - などの環境で使用されるOSである。
- 様々な組み込みOSが用途ごとに提唱されている。
- 携帯電話、自動車など、従来は「話すだけ」「乗るだけ」の道具が、インテリジェント化し、情報処理システムとして進化し続けている。
  - これらのプログラム開発を容易にするために、組み込み用OSが利用されている。