ES3 Lecture 2

An overview of mobile and embedded platforms

Basic Classes

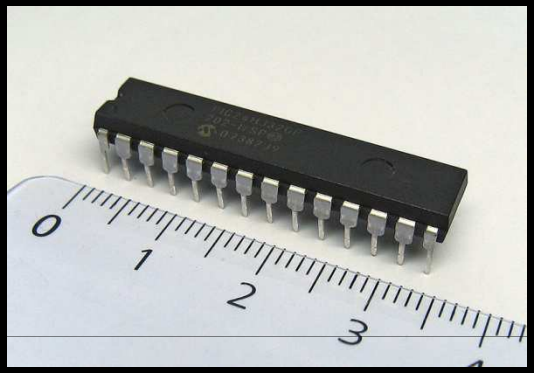
- Embedded devices:
  - These are in toasters, cars, mice, etc.
    - Usually very real-time focused (uS accuracy)
    - Very small memory, not usually high-performance
    - Cheap, small and standalone (just a chip)
    - Easy to interface with other electronics
- Mobile platforms:
  - A complete artefact for mobile use
    - Devices usually have screens, keypads, touchscreens
    - Large operating system
    - iPhones, Nokias, Palms, and so on
    - Really just a small computer with extra communications and sensing

Embedded

- Usually programmed in C or Assembly
- Expect about 128 bytes -> 64 Kbytes of RAM!
- Usually offer a large number of I/O pins
  - ADC/DAC lines
  - I2C, SPI serial interfaces for peripherals
  - RS232 and USB lines for other computers
  - "General" digital lines
- Hard real-time timers and event-driven interrupts important
  - e.g. responding to a button push "instantly"
- Often limited word lengths, virtually never anything beyond basic integer instructions
  - 8 bit or strange things like 8 bit/14 bit common
  - 24 bit and even 48 bit common for specialised DSP chips

PIC

- Extremely common range of microcontrollers
- Huge range of devices
  - 8-bit/12bit
  - 8-bit/14 bit
  - 16-bit DSP chips



- They are in *everything*
  - It's so cheap and easy to embed a microcontroller compared to building any custom logic
  - Available in all sorts of packages from hobbyist DIPs to tiny surface mount

- RISC architecture
  - Very limited instruction set -- no multiply, for example

PIC (II)

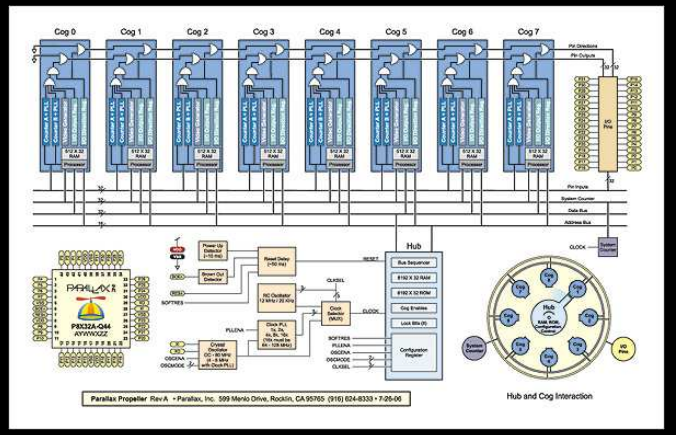
- Excellent I/O capabilities
  - Chips with ADC, DAC, USB, Serial UARTs...
- Programming often based around hardware timers
  - Throw interrupts which execute code
  - Can guarantee response within microseconds
- Programming in assembly
  - Excellent free development suite
- Commercial C compilers available, but devices are so limited that assembly is usually required
  - Less true on the DSP range

Parallax Propeller

- Interesting system: all-in-one chip with 8 parallel cores
  - Runs at 20 MIPS on each core
  - No interrupts -- you dedicate cores to I/O tasks
  - Synchronization system for using memory shared between cores
  - Cores have 512 32-bit words of core-specific memory
  - 32Kbytes of shared memory
- Fast, multi-core architecture makes it very flexible
  - Lots of protocols can be simulated in **software**
    - USB, PS/2, SPI, I2C,
  - Ultra-fast oscillators can even directly drive video hardware

Parallax Propeller (II)

- Really shows how parallel systems can work in an embedded environment
- Programmed in either assembly or it's own Spin language
  - Spin is a relatively fast interpreted language designed to make multi-core programming easier
  - Manages synchronization of resources
- Flexible and easy to interface
  - Everything is digital and standardized
  - Even ADC and DAC is done by high-frequency counters



Hub and Cog Interaction

Arduino

- Popular platform for hobbyist development
  - Single board platform -- everything comes complete
  - Based on ATMEL AVR
  - Open source (**hardware** and software!)
- Great for building quick hardware interfaces
- C-*like* language
- Simple USB programming
- Lots of hardware options
  - Motor controllers
  - I/O boards
  - Bluetooth versions

Mobile Phones / PDAs

- These devices require no extra hardware (i.e. not embedded at all!)
- Screens, audio output, cameras, touch input, keypads all common

- Usually have fairly well developed operating systems and development tools with high-level languages

- Relatively powerful processors
  - Often ARM chips in the 300-600Mhz range...
- Usually have relatively small RAM capacities, but large Flash storage available

- Most devices differ in terms of:
  - raw power (CPU power, graphics acceleration present etc.)
  - development language + API
  - input capabilities (multi-touch, accelerometers...)

Symbian

- **Nokia's** main development platform
  - Used on all series 60 phones
  - S60 phones include the N90 series and the N70 and N80 series
  - Also on other manufacturers phones (Sony Ericsson, Samsung)

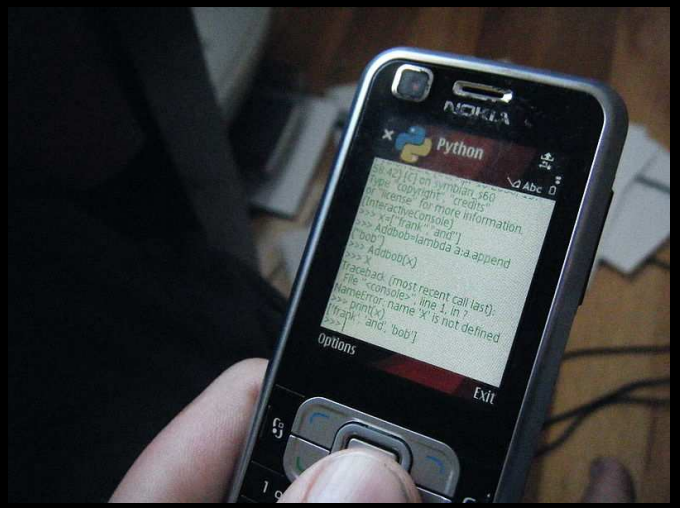
- Long history of use and widely used
  - Currently on version 9.5
  - Used in 50% of currently available smartphones

- Provides UI, OS services, a Java engine and various other functionality

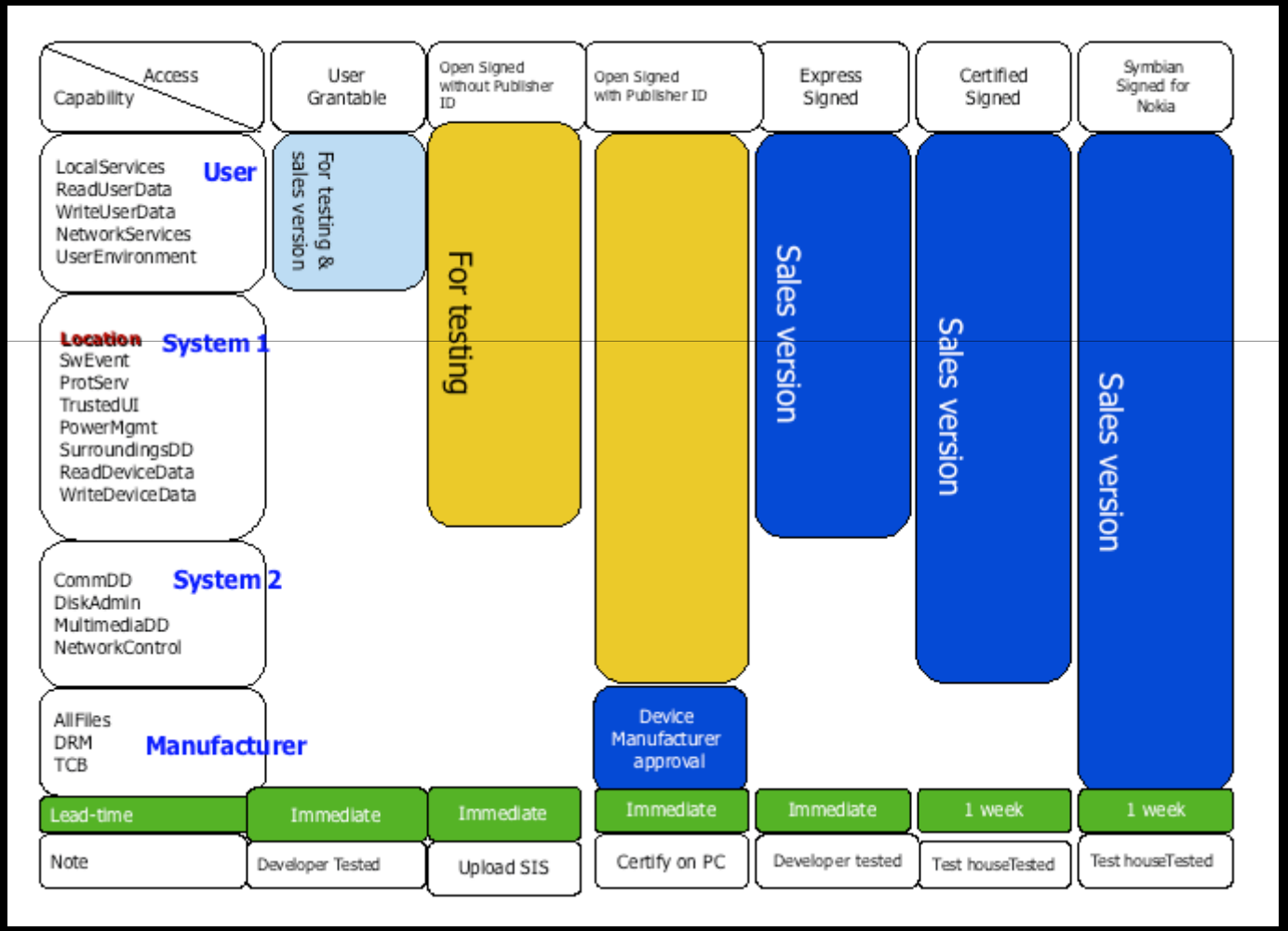
- Development in C++ with custom libraries
- Development environment very focused on memory management
  - Absolutely awful to develop for

Symbian (II)

- S60 also supports Python
  - Excellent for rapid prototyping
  - Most functions available (include OpenGLES)
- Very hard to port things
  - Symbian's C++ conventions are completely incompatible with standard code
  - OpenC provides a reasonable compatibility with standard C (but not C++)
- Platform Security since 9th ed. (2006)
  - Requires signing
  - Restrictive capability set
  - You can apply for more...
  - ...but you might not get them

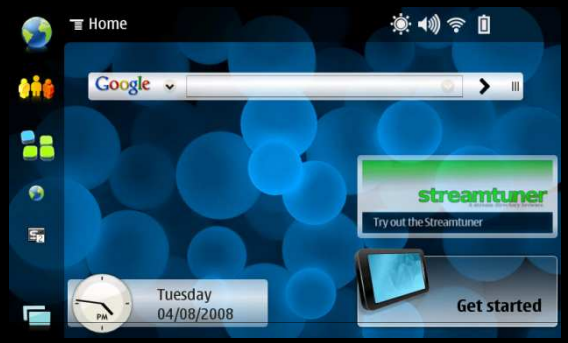


Symbian Capabilities



Maemo

- Nokia's next generation platform for high-end phones



- Linux based system
- Modified, cut-down Debian distribution
- Development in pretty much in anything you want
  - Just a standard Linux system: gcc, python, ruby, perl...
  - Easy to work with -- full shell, SSH, FTP
  - Existing Linux packages can be ported pretty easily with the standard dev. tools
- Custom drivers and UI for mobile applications
  - UI is currently GTK based ("hildon"), moving to Qt
  - Nokia is moving towards Qt as a standard UI platform
- Completely open -- no signing, no vendor approval, most of the OS is open-source
- N900 is only Maemo device with phone capability currently in production

Maemo (II)

- May well be Nokia's next main platform
  - But it's not at all clear
- Devices are only: N770 (ancient), N800, N810, N900
  - N900 is the only one with phone capabilities
  - Only supports ARM architecture
- Several versions -- now on version 5 (Fremantle) on N900
  - N900 is powerful: 600Mhz ARM, 256Mb RAM, 3D accleration
- Almost all open source
  - some parts (like power management) are closed binary modules
- Nxxx are great development platforms
  - Very few sold, little operator support...

iPhone

- Apple's phone platform
- Only a few models: 1.0, 2.0 iPod Touch, and 3.0 (3GS)

- All iPhone devices are relatively powerful (620Mhz ARM chips)
  - 256 Mb RAM, 8/16/32 Gb Flash
  - PowerVR GPU acceleration

- Multitouch interaction the "big feature"
  - Excellent OS support
  - No keypads, buttons or other controls
  - Accelerometers in all models, GPS and magnetometer in some

iPhone (II)

- Powerful GUI system with an emphasis on animation
- Development exclusively in Objective-C
  - Cocoa libraries provide UI components, basic datatypes, file handling etc
  - Apple's own XCode IDE used for development
- Application developed can be distributed on the App Store
  - Easy to get mass distribution
  - Authors get 70% instantly
- Requires Apple approval
  - You can't just distribute apps you write
  - Can be very fickle and opaque in terms what will be approved

Windows Mobile

- Originally Windows CE/PocketPC
  - Wide range of old PDA's (iPaq, Palm Treo...)
- Probably a dying platform
  - But quite a few devices use it

- Development in C++, C or C#
  - Uses Mobile .NET
  - Cutdown version of full .NET libraries
- Very wide range of devices
  - UI is very inconsistent between resistive and capacitive touchscreens
  - No multitouch support

Android

- Google's platform

- More than 18 phones running Android
- HTC most well known
  - (Nexus One, Hero, Magic, Dream)
  - Motorola Droid as well

- Java based
  - But not standard Java SDK
  - Eclipse plugin for development

- Mainly open source OS
- Has Android Market for app distribution
  - Much more open than Apple's

Android (II)

- Comparable features to iPhone/Symbian/Windows Mobile
  - Supports OpenGLES
  - Bluetooth, GSM, WiFi support built in
  - GPS, sensors, on phones that support them
- C code can be compiled as libraries and used in the Java SDK
  - Allows native access
- Vendors love to tweak Android before deployment on their devices
  - This means that there are lots of potential compatibility issues
- Applications can freely use all resources in background...
  - Great, if everything co-operates nicely

webOS

- Palm's operating system
  - For the Palm Pre range of devices only
  - All multitouch with seperate QWERTY keyboard
- Uses web technology rather than custom development
  - HTML 5, Javascript
- Basically runs webpages locally or remotely
  - Effectively a mobile web browser
- Easy to work with, but quite limited

Summary

- Lots of platforms
  - Vary in ease of development
  - Vary in ease of porting (i.e. are compatible with existing platforms)
  - Vary in development tools
- Variety of devices
  - Some are standardized (e.g. iPhone)
  - Some are much more variable (Symbian, Windows Mobile, possibly Android in the future)
- Market share is an issue
  - Symbian has 50% of the smartphone market...
  - iPhone has only 27% or so...