

# ITEC824

## *Mobile Programming & Embedded Systems*

### WEEK 01 INTRODUCTION

Laurent Malvert <[lmalvert@science.mq.edu.au](mailto:lmalvert@science.mq.edu.au)>

*Macquarie University, Department of Computing*

*2009.02.26*

# Outline

## 🌿 ITEC824 and You!

- 🌿 So what's the deal with that ITEC824 mess?
- 🌿 Why?
- 🌿 What?

## 🌿 Embedded Systems

- 🌿 Definition
- 🌿 Fields and Uses
- 🌿 Types

## 🌿 Mobile Programming

- 🌿 Definition
- 🌿 Differences

## 🌿 Discussion



# ITEC824 and You!

*So what's the deal with that ITEC824 mess?*

👉 2003 – Creation

👉 2004 – Taught (by Anthony Sloane)

👉 2005

👉 2006

👉 2007 – Re-Draft

👉 2008

👉 2009 – Fresh Start

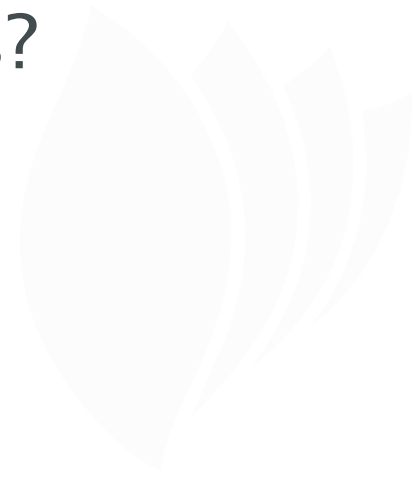


**handbook limbo  
a.k.a.  
/dev/null**

# ITEC824 and You!

*Why?*

- 👉 What are your motivations?
- 👉 What are your goals?



# ITEC824 and You!

## *What?*

- 👉 What do you already know?
- 👉 What are you interested in?
- 👉 What are you **NOT** interested in?

# Embedded Systems

# Embedded Systems

## *Definition*

### 🌿 Various Definitions

- 🌿 Specific Purposes
- 🌿 Complete / Standalone Device
- 🌿 Stateless




### 🌿 But sometimes...

- 🌿 It can be programmable
- 🌿 It is built from different sub-systems
- 🌿 It can memorize data
- 🌿 It can be state-aware



# Embedded Systems

## *Fields*

### Cross-Genres

-  Electronics
-  Robotics
-  ...

### Uses

-  Domestic Electronics
-  Machinery Control
-  Automation
-  Appliances



# Embedded Systems

## *Uses » Personal*

### Common Handhelds

- Mobile Phones
- Portable Music Players (PMP)
- Personal Digital Assistants (PDA)
- Ultra-Mobile PCs (UMPC)
- Watches!

### In-House Appliances

- Microwave / Refrigerator
- Boiling / Heating / AirCon Systems
- (Alarm) Clocks
- TV / DVD Players / Video Game Station

# Embedded Systems

*Uses » Public*

## Public Sector

- Transportation Systems
- Medical Systems
- Machine Controllers
- ...

## Industry

- Programmable Logic Controllers (PLC)
- Networking Systems
- Tracking Devices
- Automobile / Avionics
- ...

# Embedded Systems

## *Interest*

- ❖ So what do you want to do?
  - ❖ **Focused on the Hardware / System Layers?**
    - ❖ Operating System Programming
    - ❖ Low-Level Code
  - ❖ **Focused on the Applicative Layer?**
    - ❖ Mostly standard Software Engineering
    - ❖ Mobile Platforms
- ❖ Depends what you consider to be:
  - ❖ *“fun”*
  - ❖ *“profitable”*

# Embedded Systems

## *Types*

✦ Capabilities / Requirements mean...

✦ ... Impacts on:

✦ Hardware

✦ Software

# Embedded Systems

## *Types » Hardware Layer*

### Capabilities

- Human–Machine Interface
- Dependability

### Hardware Impacts

- Case & Packaging
- Connectivity / IO Channels
  - Input*                      keyboard, mouse, pad, trackball, stick, touch–screen...
  - Output*                     headless , LCD/LED, GUI
  - Transfer*                    Serial, Parallel, COM, USB, Ethernet, Wi-Fi...
- CPUs
  - Microprocessors ( $\mu\text{P}$ )
  - Microcontrollers ( $\mu\text{C}$ )

# Embedded Systems

## *Types » Software Layer*

### 🌿 Capabilities

- 🌿 Single – or Multi–task
- 🌿 Criticality / Dependability
  - 🌿 Availability
  - 🌿 Reliability
  - 🌿 Safety
  - 🌿 Security

### 🌿 Software Impacts

- 🌿 Concurrency / Scheduling / Real–Time
- 🌿 Memory and State Management



# Embedded Systems

## *Types » Simple vs. Complex Systems*

### ✦ Usually, a “Simple” System...

- ✦ Runs on a microcontroller–based platform
- ✦ Under a minimalistic task–specific OS
- ✦ With limited connectivity
- ✦ And is “headless” (and maybe “handless”)

### ✦ Usually, a “Complex” System...

- ✦ Runs on a microprocessor–based platform
- ✦ Under a full–blown embedded (RT)OS
- ✦ With maintenance and communication channels
- ✦ and has user–interactive interfaces

# Mobile Programming

# Mobile Programming

## *Definition*

- 👉 Programming for Mobile Phones?
- 👉 Programming for Mobile Devices?

# Mobile Programming

## *Differences*

### 🌿 Constraints

#### 🌿 Standard Resources

- 🌿 Time
- 🌿 Memory
- 🌿 CPU
- 🌿 Display

#### 🌿 But

- 🌿 Possibly Shared
- 🌿 Possibly Limited

### 🌿 Production / Development Lifecycle

# Mobile Programming

## *Differences » Constraints*

- Performance is a function of time, they say...
- Still... Some environmental constraints remain
  - Think of embedded medical systems
  - Then think of embedded military systems
- Think of it as developing on an old PC80386
  - Except it can be surprisingly worse than that...
  - ... even though your TI-89 beats your Intel 80386...
  - ... and your iPhone is a portable AMD K7/Athlon

# Mobile Programming

## *Differences » Standard Mobile Aspects*

### 🌿 Localization & Time

- 🌿 Geo-localization
- 🌿 Time Management
- 🌿 Internationalization

### 🌿 Data-Freedom

- 🌿 Synchronization
  - 🌿 1-way
  - 🌿 2-way
- 🌿 Import/Export

### 🌿 Interfaces

- 🌿 IO: Limited, Small, with Degraded Quality
  - 🌿 Tiny keyboards
  - 🌿 Small LCDs
  - 🌿 LEDs
  - 🌿 MIDI synthesizers



# Mobile Programming

## *Differences » Emerging Mobile Aspects*

### 🌿 Localization & Time

- 🌿 Dynamic
- 🌿 Reactive

### 🌿 Data-Freedom

- 🌿 Social Networks
- 🌿 Web 2.0
- 🌿 That sort of crap...

### 🌿 Interfaces

- 🌿 Polyphonic Audio
- 🌿 Voice
- 🌿 Touch



# Mobile Programming

## *Differences » Dev & Prod Lifecycle*

### 🌿 For Standard Systems

- 🌿 Code
- 🌿 Build
- 🌿 Test
- 🌿 (...)
- 🌿 Debug
- 🌿 Deploy



# Mobile Programming

## *Differences » Dev & Prod Lifecycle (2)*

### For Embedded Systems

#### The Old Way: For Real Hardware / System

##### *Without Native Development Environment*

- Code
- Build (cross-compilation)
- Deploy (have a coffee...)
- Test
- Debug

##### *With Native Development Environment*

- Code
- Build
- Test
- Debug
- (Deploy)

#### The Modern Way: For Emulated Hardware / Systems

- Code
- Build
- Fake-Deploy
- Test
- Debug

# Mobile Programming

## *Differences » Dev & Prod Lifecycle (3)*

### 👉 For Embedded Systems

#### 👉 The Old Way:

- 👉 Expensive
- 👉 *Without Native Development Environment*
  - 👉 Slow and difficult
  - 👉 It just \*Drives you crazy\*!
- 👉 *With Native Development Environment*
  - 👉 Slightly better

#### 👉 The Modern Way

- 👉 Cheap
- 👉 Fast / Responsive

# Discussion