Teaching Embedded Software Concepts Using Android

Jogesh K. Muppala
Dept. of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clearwater Bay, Kowloon, Hong Kong
muppala@cse.ust.hk

Talk Outline

- Introduction
- Background
- The Smartphone Platform
- The Course
    - Course Topics and Structure
    - Hands-on Laboratory Exercises
    - Course Projects
- Students
    - Student background
    - Student feedback
- Conclusions

Introduction

- Embedded systems growing in importance, widespread deployment and complex with time
- Software becoming an important component of embedded systems
- Training manpower on the design and implementation of embedded software becoming more important
- Is there a “right” way to teach embedded software concepts?

Introduction

- Computer science students often stay away from embedded systems:
  - Perception as a “hardware” course
  - Missing out on participating in a dynamic field
  - How can we “correct” this misperception?
- There is indeed lot of scope for CS students to participate in this field without getting too “involved” with the hardware

The Smartphone

- A highly integrated device with rich functionality:
  - Telephone
    - Voice call, video call, SMS, voice mail
  - Personal organizer
    - Calendar, contacts
  - Internet access/communication device
    - Internet browser, email, internet tethering
  - Multimedia/Entertainment device
    - Music player, video player, camera
  - Gaming console
  - GPS/navigation device
    - Maps, navigation, direction

The Smartphone Hardware Platform

- Powerful Application CPUs
  - Currently single-core CPUs integrated in a System on a chip (SOC)
  - Trend: multi-core CPUs within a single SOC
- On-board Storage/Memory for applications and data
- User interaction provided through display, typically a touchscreen sometimes augmented with physical buttons
- Communications/Connectivity of several kinds:
  - Telephony, 3G, WiFi, Bluetooth
- Dedicated chips for display driver, touchscreen control, GPS, bluetooth, WiFi more...
- Powerful graphics capabilities, both 2D and 3D
- Sensing the environment, real-world through various sensors
  - Accelerometer, proximity, light, temperature, etc.
- One or more Camera(s)
- Power supply through a battery

The Course

- What should be covered?
  - Embedded software development
  - Real-time and embedded operating systems (RTOS)
  - Interaction with the real world
  - Memory management
  - Testing and debugging
- Striking a good balance between theory and practice

Course Topics and Structure

- Introduction
- The Smartphone Platform
  - Typical components in a smartphone
  - Embedded Systems Structure
- Embedded Software Development
  - Hosts and Targets
  - Mobile Application Development
  - The Android SDK
- Android Overview
  - Android Architecture
  - Components of an Android Application: Activities, Services, Broadcast Receivers and Content Providers
  - Component Lifecycles
- Android User Interface
  - Event-driven Programming
  - Handling UI Events in Android
  - The Model-View-Controller Framework
- Real-Time Operating Systems (RTOS)
  - Review of Operating Systems Basics
    - Processes and Threads
    - Process Synchronization and Coordination
    - Inter-process Communication
    - Memory Management
  - Processes and Thread Support in Android
  - Concurrency, synchronization and Inter-process Communication Support in Android
  - Memory Management Issues in Android
- Multimedia and Graphics Support in Android
  - Audio and Video Support
  - 2D and 3D graphics
- Data Storage Support in Android
- Interaction with the Real World:
  - Location Services and Maps in Android
  - Sensors and Sensor support in Android
  - Networking Support in Android
- Testing and Debugging Support in Android

Android and Embedded Software

- Illustration of cross-platform development
- Android architecture
  - Illustrates the typical design of a layered embedded system implementation
- Structure of an Android application
  - No “main” function
- UI Design
  - Event handling and event listeners
  - Model View Controller framework
- Android component lifecycle

Android and Embedded Software

- Timing constraints and Application not Responding (ANR)
- Interprocess communication
  - Intents
  - Service component
- Memory management and memory “leak” problems
- Android/Junit testing framework

Hands-on Laboratory Exercises

- Major goals:
  - Introduce students to various aspects of Android applicationn development
  - Preparing the students for implementing the course project
- Laboratory exercises implemented as simple full-fledged applications
  - Incremental development of several features in the application(s)

Course Project

- Team project with teams of up to two students
- Main emphasis on demonstrating the use of techniques learnt in the course
- Students felt that this experience illustrated to them that a whole new arena of small device programming was easily accessible to them and provided them with alternate avenues for future career

Course Projects

- A photo album display application that connects to online service like Facebook, downloads and displays photos.
- A Bluetooth based remote controller for a robot vehicle. This project required the design of a custom protocol for communication between the phone and the robot.
- A multi-function application targeting the university students that enables the students to access information from the university's student information system, and provide PIM features (implemented using iOS platform).
- A traveller's organizer including checklist management, and destination information management.

Course Projects

- An innovative wake-up alarm that requires users to play a number game to completion in order to turn off the alarm.
- An augmented reality game that made use of the camera, user's location and graphics.
- A musician's assistant that includes a metronome, tuner and additional music exercises.
- Personal information and finance management application.
- A public transit information and route planner application.
- A 2D cannons and shooter game implemented for an Android Tablet using *AndEngine* game development engine.

Student Background

- A good mix of students in their junior and senior year of undergraduate studies
- A good mix of students from the computer engineering and computer science stream
- Background includes courses on:
  - Computer programming including OO
  - Computer architecture and organization
  - Operating Systems
  - Practically no software engineering

Student Feedback

- *It is good to learn android platform in this course.*
- *The instructor provides a lot of examples to demonstrate how the technology can be done on an android device, like proximity alert, accelerometer. This is easy to catch up what is going on. For the theory part, multiple fields also be included, many new knowledge can be learnt or the old one can be refreshed. It is interesting to have an overview of how this different issue of different aspects comes together. And the most important thing is good is that the instructor often give a general view of a topic first, then focus on the android/an embedded system.*
- *Thorough introduction to basic concepts of embedded systems*

Conclusions

- Leveraging the smartphone platform to deliver embedded software concepts to the students
- Practical skills that the students can use in the future
- Taking advantage of this wonderful opportunity to promote the exciting field to Computer Science students
  - Still keeping them within their comfort zone