Toward HW/SW Integration: Networked Embedded System Design

Chi-Sheng (Daniel) Shih
(National Taiwan University)
cshih@csie.ntu.edu.tw

Shiao-Li (Charles) Tsao
(National Chiao Tung University)

Yeh-Ching Chung
(National Tsing Hua University)

Shyh-In Hwang
(Yuan Ze University)
Background – Embedded Courses in Universities

- High-level design methodology
  - UML, Component-based design, etc.

- Software Design:
  - Dr. Muppala’s at The Hong Kong University of Science and Technology
  - Dr. Evan’s embedded software systems at UT, Austin

- Hardware Oriented:
  - SoC Architecture design, VLSI design, and FPGA-based system design.
What are missed in CS Curriculum for embedded systems engineers?

- Embedded SYSTEMS not embedded software.
- Hardware and Software have to collaborate.
- Reliability

THE GOAL OF EVERY EMBEDDED SYSTEMS ENGINEER IS TO RETIRE WITHOUT GETTING BLAMED FOR A MAJOR CATASTROPHE.
Embedded System Design Program

SoC/Embedded Systems Faculty

Embedded System Program

Networked Embedded Real-Time System Design

Software-Oriented Design Courses

Hardware-Oriented Design Course

Lab Facility
Government Force for Embedded Software Education

- In 2004, the professors from four universities submitted a curriculum enhancement proposal. It aimed at
  - bridging the gap between HW and SW designs for Computer Science seniors and first-year graduates.
  - skills for network programming for embedded systems and low-power programming.

- Embedded Software Consortium
  - Founded in March 2004.
  - Funded by Ministry of Education, Taiwan.
  - Objectives:
    - To enrich the embedded curriculum for universities and colleges,
    - To engage academia and industry in dialogue, and
    - To support the embedded software researches.
Embedded Software Program

**Fundamental Course (Sophomore/Junior)**
- Digital Systems
- Digital Circuit Lab
- Data Structure and Algorithms
- System Programming
- Introduction to Embedded Systems Lab

**Core Course (Junior/Senior)**
- Computer Architecture
- Digital System Design
- Compiler
- Operating Systems

**Advance Program (Senior/Graduate)**
- Introduction to DIP
- Distributed Systems
- I/O Devices and Device Driver Design
- Ubiquitous Computing
- Real-Time Systems
- Compiler for Embedded Systems
- Network Embedded SoC Systems
- Advanced Computer Architecture and Organization
- Advanced Operating Systems
- Embedded Systems and HW/SW Co-design
- Introduction to DIP
- Distributed Systems
- I/O Devices and Device Driver Design
- Ubiquitous Computing
- Real-Time Systems
- Compiler for Embedded Systems
- Network Embedded SoC Systems
- Advanced Computer Architecture and Organization
- Advanced Operating Systems
- Embedded Systems and HW/SW Co-design
Network Embedded SoC Systems

Course Description:

The purpose of this course is to get the students familiar with the knowledge and hands-on programming skills for networked embedded systems.

Students: Senior/Graduate students

Class Schedule:

- Lecture: 2 hours/week for 18 weeks.
- Lab: 2 hours/week for 16 weeks.
Outline for Lectures

- Introduction for Embedded Systems
- Real-Time Systems
- Embedded Programming
  - Real-Time Sensing and Control
  - Clock Jitter and Drift
  - Real-Time Scheduling
  - Rate Monotonic Scheduling
  - Priority Inversion
  - Priority Inheritance/Ceiling Protocol
- Power Management and Low Power Design of a Networked SoC System
- Networks for Embedded Systems
- Mobile Ad-Hoc Networks

Sensor to read the degree.
Fill water
Drain water

Copyright © Chi-Sheng Shih. All Rights Reserved.
Lab Modules

- Labs are conducted in a group of 2 to 3 students or individually.
- Every week, the students need to complete the pre-designed lab.
- At the end of the semester, the students complete a complete networked embedded systems, Intelligent Navigation Systems.
The Environment

- Hardware: PPRK, PCM7230, wireless card, and PC
- Target Operating Systems on pcm7230:
  - Embedded Linux
  - Windows CE.

pcm7230 carried by PPRK

Communicate through wireless

host computer
PPRK (Palm Pilot Robotic Kit)

The BrainStem controller
40 MHz RISC processor
RS-232 serial port

IR Sensor: Sharp GP2D12
Distance range between
10cm (~4") to 80cm
(~30")

Servo motor and Wheel
The target board pcm7230

- Embedded with PCM-7230S-230CE SBC
- Intel® Xscale® PXA255 400 MHz CPU
- 10.4” SVGA LCD display with touchscreen
- Integrated 4-cell pack and charger system
- 8 DI & 8 DO function bundled with selection utility
- 4-COM AMI-120 module is bundle for function expansion
Monitoring and Control

In the lab, the PPRK will be put in the provided LEGO castle, and the students will be asked to write a program to estimate the shape of the castle.

The objectives of this lab is to develop a program using the infrared sensors on PPRK (GP2D12) to collect the environmental information.
Real-Time Control

- In the lab, the students should develop a program to control the PPRK to move along the given direction and distance.
Chase with Single Thread

- The goal of this lab is to develop a program that controls the PPRK to chase a moving object.

- In the lab, TA will put a ball in front of PPRK, and the PPRK shall chase the ball for a certain amount of time.
Chase with Multiple Threads

- The goal of this lab is to develop a multithread program to control PPRK so that it can follow another PPRK controlled by TA.

- In this lab, PPRK should
  - follow the target and
  - avoid collision.

- The students will learn and hands-on
  - Multithread programming
  - Mutex
  - Resource Synchronization
Traffic Controller

- The goal of this lab is to build an intelligent traffic coordination system.
- The students will design an intelligent vehicle that communicates with traffic controller to receive the traffic control command, and cross the intersection without any accident.
Scenario in Infrastructure Mode

1. Cars start running.
2. Detect an Intersection. Stop.
3. Negotiates with the AP
4. Green car goes.
5. Green car done.
6. Red goes!
Traffic Controller using Ad Hoc Network

- To build an intelligent traffic coordination system, the PPRKs have to negotiate with each other for the right of way so that there is no car accident on the way.
- This lab works in non-infrastructure mode.
Scenario in Non-Infrastructure Mode

1. GO!
2. Detect an Intersection, Stop!
3. Negotiate with each other
4. Green GO!
5. Green Done.
6. Red Go!
Traffic Control with infrastructure and non-infrastructure mode

PPRK with pcm7230

Ad Hoc Mode

Server to coordinate this intersection
Reflections from lecturers and students

- Well-designed lectures and labs provide the students in-depth lessons for embedded real-time systems design.
- The semester-long step-by-step labs and final project provide the students the skills to design the system and the opportunity to write their own code at the end of the semester.
- Well-trained TAs are the key for the success of this course.
Acknowledgement

☐ This curriculum development was supported in part by
  ■ Ministry of Education, Taiwan and
  ■ Microsoft Research Asia