# CS4101 Introduction to Embedded Systems

# Course Overview

Prof. Chung-Ta King
Department of Computer Science
National Tsing Hua University, Taiwan

# Consider the Evolution of Watches


# How about Refrigerators?

# Picture Frames


# Cameras



# Glasses

# What is the trend?

# Physical Things Augmented with Computing/Communication

**OR**

# Computing/Communication "Embedded" into Physical Things

# So, What Is Embedded System?

**A computer, pretending not to be a computer**



(Stephen A. Edwards)

# What Is Embedded System?

- "An embedded system is an application that contains at least one programmable computer ... and which is used by individuals who are unaware that the system is computer-based."

  -- Michael J. Pont, Embedded C

  - **Programmable computers require programs**
    → embedded software



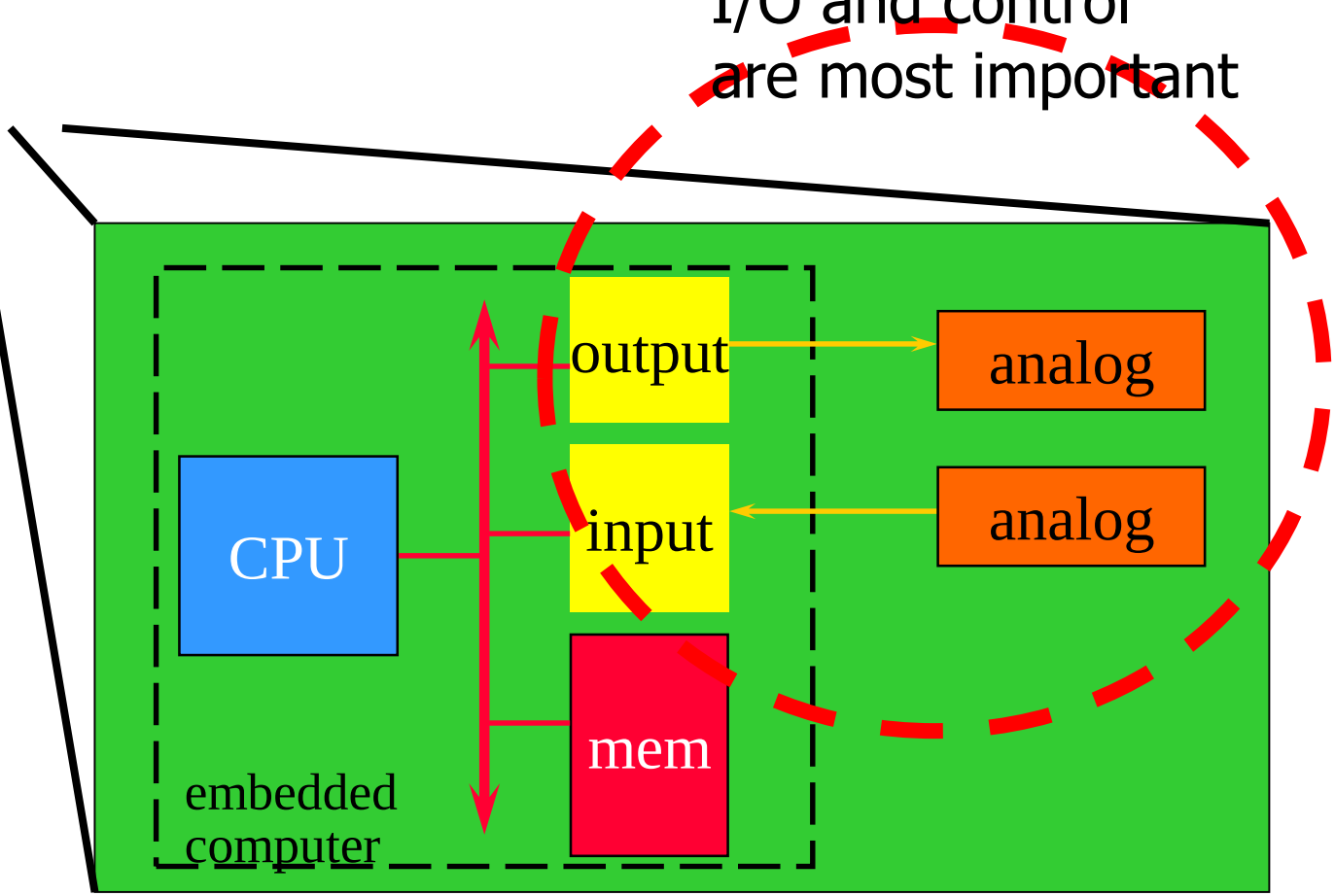

# What Is an Embedded System?

- Information processing systems embedded into a larger product [Peter Marwedel]
  - Main reason for buying is **not** information processing
- Any device that includes a programmable processor but is not itself a *general-purpose computer*

➔*Application-specific*: take advantage of application characteristics to optimize the design:

  - Do not need all general-purpose bells and whistles

# Same Basics Inside



# Why Embedded Systems?

- After all, we can still make phone calls with

or


- Why embed a computer into a phone?

# Embedded vs Pure Hardware

- Many electronic products are implemented in pure hardware (ASICs, boards)
  - Lack of flexibility (changing standards, system revisions, bug fixes, extra functionalities)
  - Costly for specialized application-specific integrated circuits (ASICs) (M$ range, technology-dependent)

- Trend towards implementation in software running on embedded processors (or possibly FPGAs)

(Peter Marwedel)

# Trends towards Software

# Some Concepts to Clarify

- Embedded systems refer to not only small devices or gadgets





- But also large, complex systems requiring strict reliability, real-time responses




# Some Concepts to Clarify

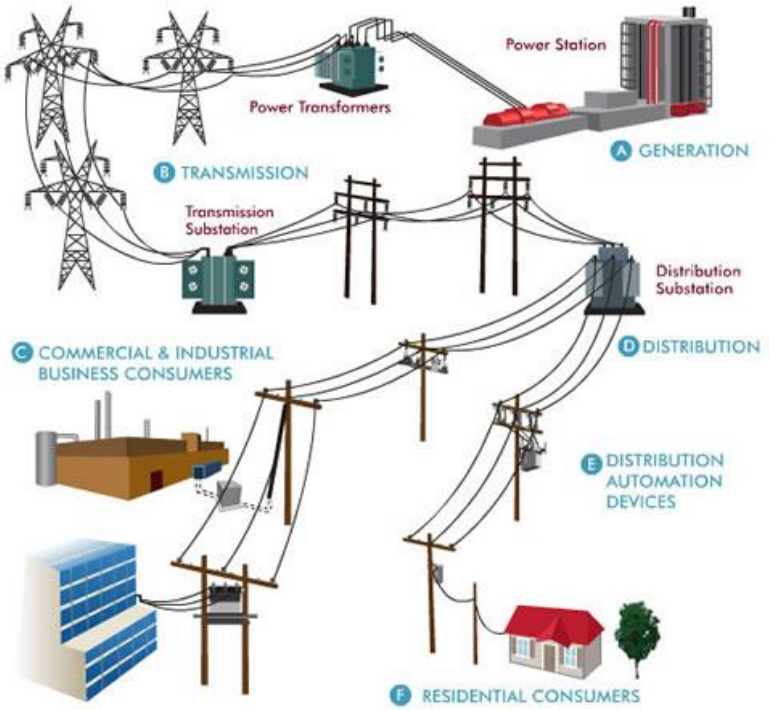
- A product, e.g., video decoder, may be implemented using pure hardware or microprocessor + embedded software
  - A chip may be implemented using pure logic gates or a microprocessor + peripheral logic + software
- An embedded system may or may not have OS
  - Simple systems may be implemented by a single program that runs continuously
  - Systems that need to control and respond to many activities may require an OS for management

# Embedded = Smart

- Computers embedded into objects
  - Augment objects with programmatic control, communication, sensing, and actuation

- Let the world know you:
  - Make physical objects/phenomena accessible to digital world
- Let you know the world:
  - Give intelligence/life to physical objects so that they can sense/react
  - Put a “robot” inside everything!

# Future Embedded Systems




Smart Grid

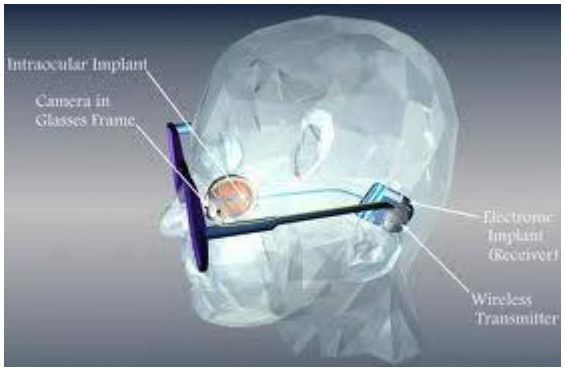
Smart City

(Source: oncor.com, Prof. L.G. Chen)

# Future Embedded Systems

Smart Glasses

Unmanned Cars



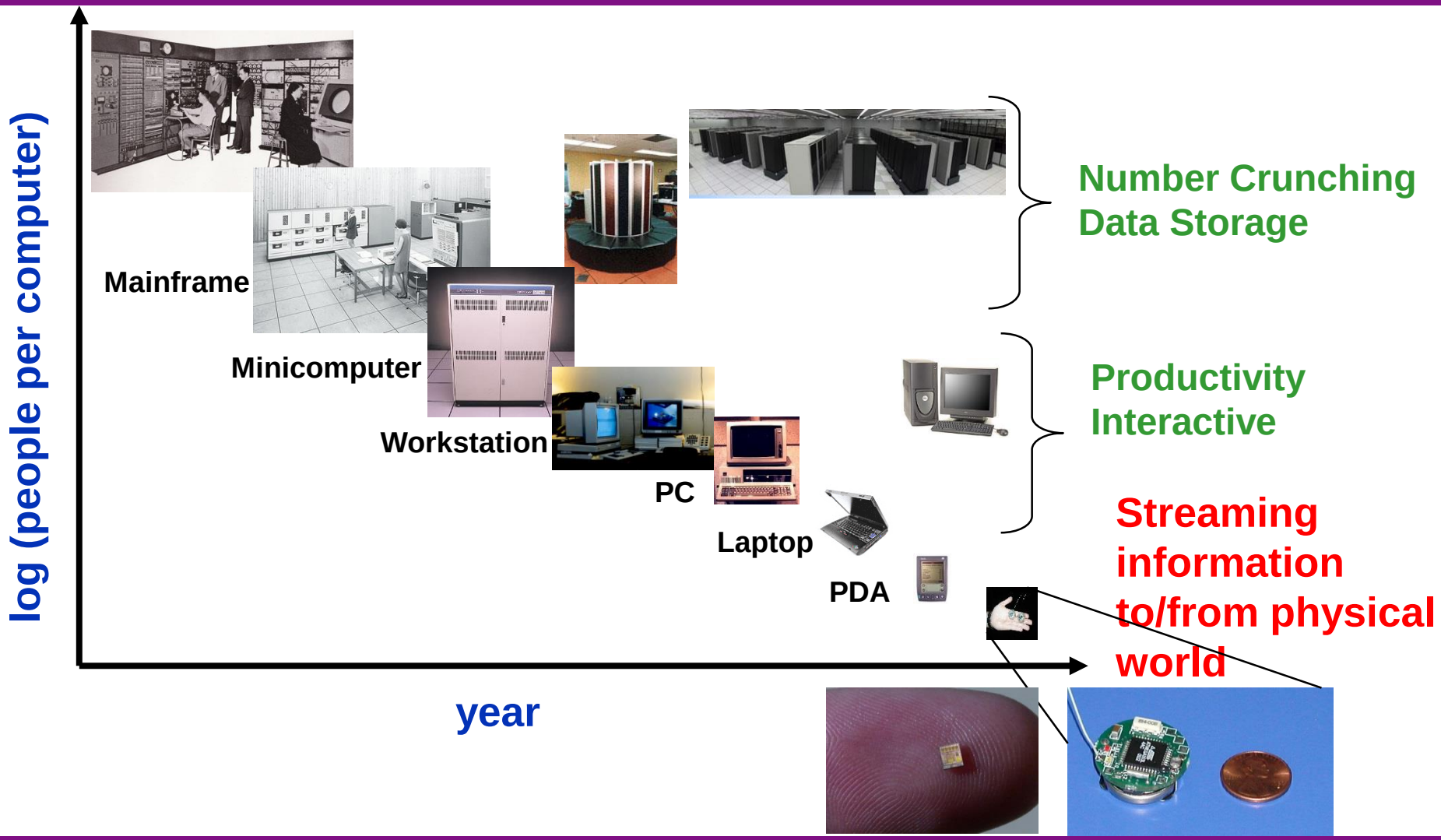

Retinal Implant

# Implications

資工系的人有飯吃了！

- Infinite opportunities and innovations
- Integration: must know application domain, packaging, A/D, sensor/actuator, power, …
- Innovation and execution

# A New Paradigm of Computing



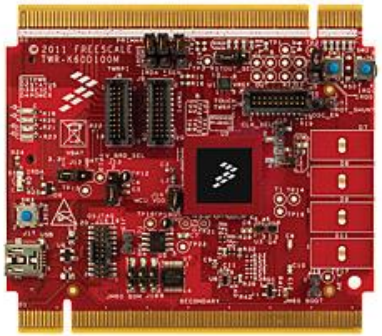
# About This Course

- Principles behind the design of the course:
  - Build the course around labs: Use labs to carry out the course contents. Labs are to develop a simple embedded system, from I/O device to system
  - Cover basic concepts in embedded system development: interrupt, clocking, I/O, real-time system, OS service, development tools
  - Expose to assembly programming: MSP430 and ARM architecture and programming
  - Term project development: innovation, development process, communication and team work, learning-by-doing

# Lab Platforms

- TI MSP430 LaunchPad

- Freescale Kinetis Tower: TWR-K60D100M-KIT



# Labs (subject to change)

- LaunchPad:
  - Registers/addressing mode/IO
  - Stacks
  - Timer and watchdog
  - Serial communication interface, UART
- Kinetis Tower
  - Environment and cross-compiler
  - Bootloader
  - Non-OS embedded application
  - Real-time OS
  - Device driver
- Kinetis Tower integrated with LaunchPad

# Lectures

- Discuss concepts related to the labs
  - May lack of a systematic discussion of embedded systems and a comprehensive coverage of all important concepts

# What Will and Will Not Learn?

- Will learn:
  - Basic concepts of embedded systems
  - Hands-on development of a system and know what's behind
  - Innovation development, presentation, team work
- Will not learn (but important):
  - Software engineering for embedded systems
  - System evaluation, optimization

# Course Information

- Instructor: Prof. Chung-Ta King (金仲達教授)
  - Office: Delta 640 Phone: x42804
  - email: king@cs.nthu.edu.tw
- Teaching assistants: 李荏敏、柯安琪、廖柏皓、張子逸、廖毓強、鄭又仁
  - Office: CSEE 734 Phone: x33553
- Class time:
  - Tuesday 15:30 - 17:20
  - Thursday 15:30 - 16:20
- Classroom: Delta 105
- http://www.cs.nthu.edu.tw/~king/courses/cs4101.html

# Expected Workload

- Labs:
  - Run on Tuesday in PC room
  - At least 2 basic assignments to be completed in class plus 1 advanced assignment for bonus
- Term project:
  - Proposal, progress report, final demonstration, project report
- Grade breakdown
  - Assignments and Labs 70%
  - Term project 30%