Chapter 1
Introduction

Hsung-Pin Chang

Department of Computer Science

National Chung Hsing University

Outline

- 1.1 What Is an Embedded System?
- 1.2 Characteristics of Embedded System
- 1.3 Examples of Embedded Systems
- 1.4 Challenges in Embedded System Design
- 1.5 Embedded System Design Process
- Appendix: Software Storage and Upgradeability in Embedded Systems

1.1 What Is an Embedded System?

- *Embedded systems* are computing systems with tightly coupled hardware and software integration, that are *designed to perform a dedicated function*.

- ***Definition:*** *Embedded system*
  - Any device that includes a programmable computer but is not itself a general-purpose computer.

1.2 Characteristics of Embedded System (1/4)

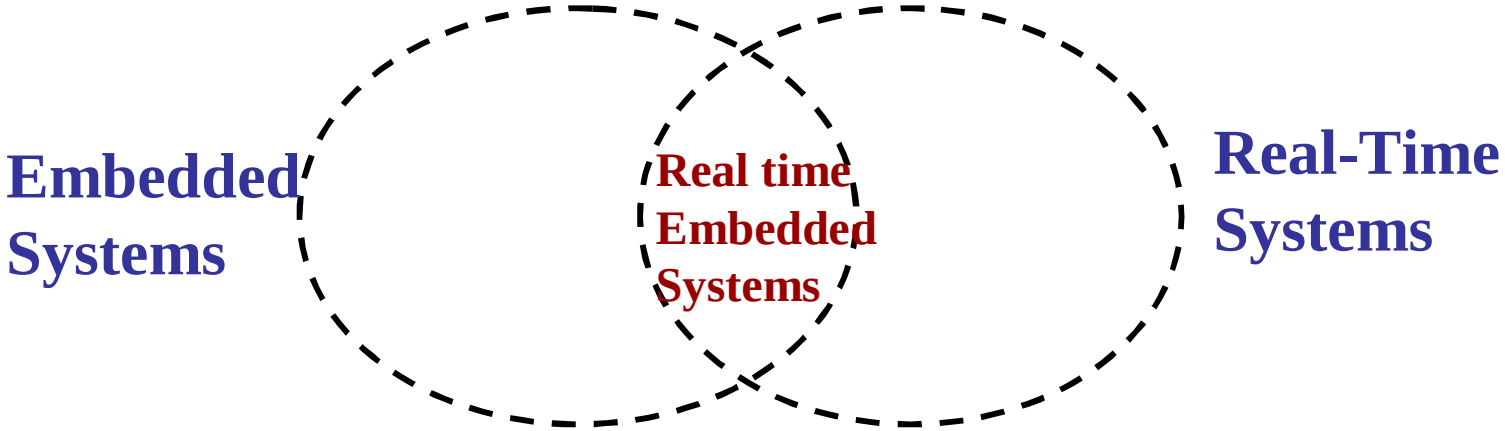
- *Object codes are stored in ROM/Flash*
  - Usually do not equal with a hard disk
  - ROM and RAM are used to store the executable code and temporary storage for runtime data manipulation
- *User interface*
  - Contain some special type of inputs and outputs
  - Input: sensors, probes, communication signals, or control knobs and buttons.
  - Output: displays, communication signals

1.2 Characteristics of Embedded System (2/4)

- *Tightly-constrained*
  - Low cost, low power, small size, fast, etc.
- *Manufacturing cost*
  - Mass-market items must have low manufacturing costs: limited memory, microprocessor power, the type of I/O devices
- *Power*
  - Critical in battery-powered devices.
  - Excessive power consumption increases system cost even in wall-powered devices

1.2 Characteristics of Embedded System (3/4)

- *Real-time*
  - Some embedded systems have real-time constraints
    - Hard-real-time: life-critical system
    - Soft-real-time: GPS navigation system



1.2 Characteristics of Embedded System (4/4)

- *Require specialized tools*
  - ICE
  - FPGA
  - Logic analyzer
- *Require specialized design processes*
  - Hardware/software partition
  - Hardware/software co-design
  - Cross-platform development
- *Designed by small teams on tight deadlines*
  - 6-month market window in common
  - International competitive environment

1.3 Examples of Embedded Systems (1/2)

- Embedded system are ubiquitous
  - Industrial automation
  - Defense
  - Transportation
  - Aerospace
- For example
  - Personal digital assistant (PDA).
  - NASA's Mars Path Finder
  - Lockheed Martin's missile guidance system
  - Automobile: engine, brakes, etc.
  - PC keyboard (scans keys).

1.3 Examples of Embedded Systems (2/2)

- In the home environment
  - Security systems
  - Cable and satellite boxes for televisions
  - Home theater systems
  - Telephone answering machines
- In the work environment
  - Network end-points: printers, cable modems, routers
  - Backbone gigabit switches
- In leisure activities
  - In-car navigation systems
  - Portable music player
  - Web tablet

Examples of Embedded Systems

Source: Qing Li and Caroline Yao, “real-time concepts for embedded systems”

Automotive Embedded Systems

- Today's high-end automobile may have 100 microprocessors:
  - 4-bit microcontroller checks seat belt
  - The ignition and braking systems
  - 16/32-bit microprocessor controls engine.

BMW 850i Brake and Stability Control System

- Anti-lock brake system (ABS):
  - Pumps brakes to reduce skidding
- Automatic stability control (ASC+T):
  - Controls engine to improve stability
- ABS and ASC+T communicate.
  - ABS was introduced first
  - ASC+T needed to interface to existing ABS module.

Mars Path Finder

- The Mars Pathfinder was actually two embedded systems
  - A landing craft: a 32-bit processor and 128 MB of RAM
  - A rover: only an 8-bit processor and 512KB.
- Design Key Point: ***Reliability***
  - All of the potential failure points are enhanced by adding redundant circuitry or extra functionality
    - For example: an extra processor, special memory diagnostics, a hardware timer to reset the system if the software got stuck, and so on.

1.4 Challenges in Embedded System Design (1/2)

- *How much hardware do we need?*
  - How big is the CPU? Memory? Peripheral devices
- *How do we meet our deadlines?*
  - Faster hardware or cleverer software?
  - Select a faster CPU may also miss deadline since memory system is the bottleneck
- *How do we minimize power?*
  - Turn off unnecessary logic? Reduce memory accesses? Run at slower clock?

1.4 Challenges in Embedded System Design (2/2)

- *How do we design for upgradeability?*
- *Does it really work?*
  - Reliability is always important in selling products
- *Complex testing*
- *Limited observability and controllability*
  - Usually do not have keyboards and screens
- *Restricted development platform*
  - Much more limited than in PCs

1.5 The Embedded System Design Process

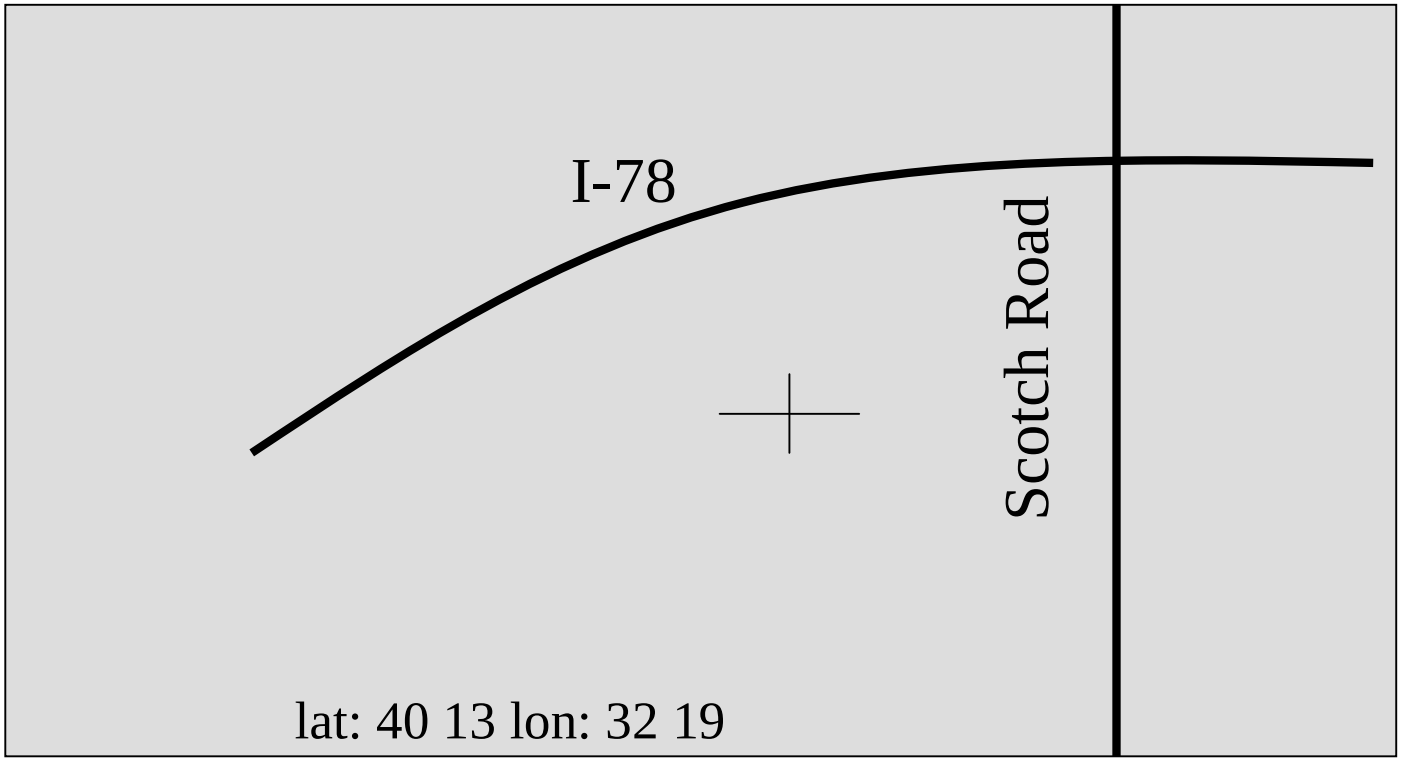
- 1.5.1 Requirements
- 1.5.2 Specification
- 1.5.3 Architecture Design
- 1.5.4 Designing Hardware and Software Components
- 1.5.5 System Integration
- 1.5.6 Issues

1.5.1 Requirements

- Functional requirements
  - Basic functions of a system
  - However, not sufficient
- Non-functional requirements
  - Performance
  - Cost
    - Manufacturing cost: the cost of components and assembly
    - Nonrecurring engineering (NRE): the personnel and other costs of designing the system
  - Physical size and weight
  - Power consumption

Example: GPS Moving Map Requirements

- Moving map obtains position from GPS, paints map from local database.



Initial Requirements List

- Functionality
  - For automotive use.
  - Show major roads and landmarks.
- User interface
  - At least 400 x 600 pixel screen.
  - Three buttons max.
  - Pop-up menu
- Cost:
  - $500 street price

Initial Requirements List

- Performance
  - Map should scroll smoothly
  - Display take no more than 1 sec upon power-up
  - Verify and display the current map within 15 seconds.
- Physical size/weight
  - Should fit in hand.
- Power consumption
  - Should run for 8 hours on four AA batteries.

Requirement Chart

Name	GPS Moving Map
Purpose	Consumer-grade
Inputs	Power button, two control buttons
Outputs	Back-lit LCD 400x600
Functions	5-receiver GPS; three resolution; display current lat/lon
Performance	Updates screen within 0.25 sec of movement
Manufacturing cost	$100 cost-of-goods-sold
Power	100 mW
Physical size/weight	No more than 2” x 6”, 12 ounces

1.5.2 Specification

- A more precise description of the system:
  - Should not imply a particular architecture;
  - Provides input to the architecture design process.

- Must be carefully written to
  - Reflect the customer's requirements
  - Be clearly followed during design

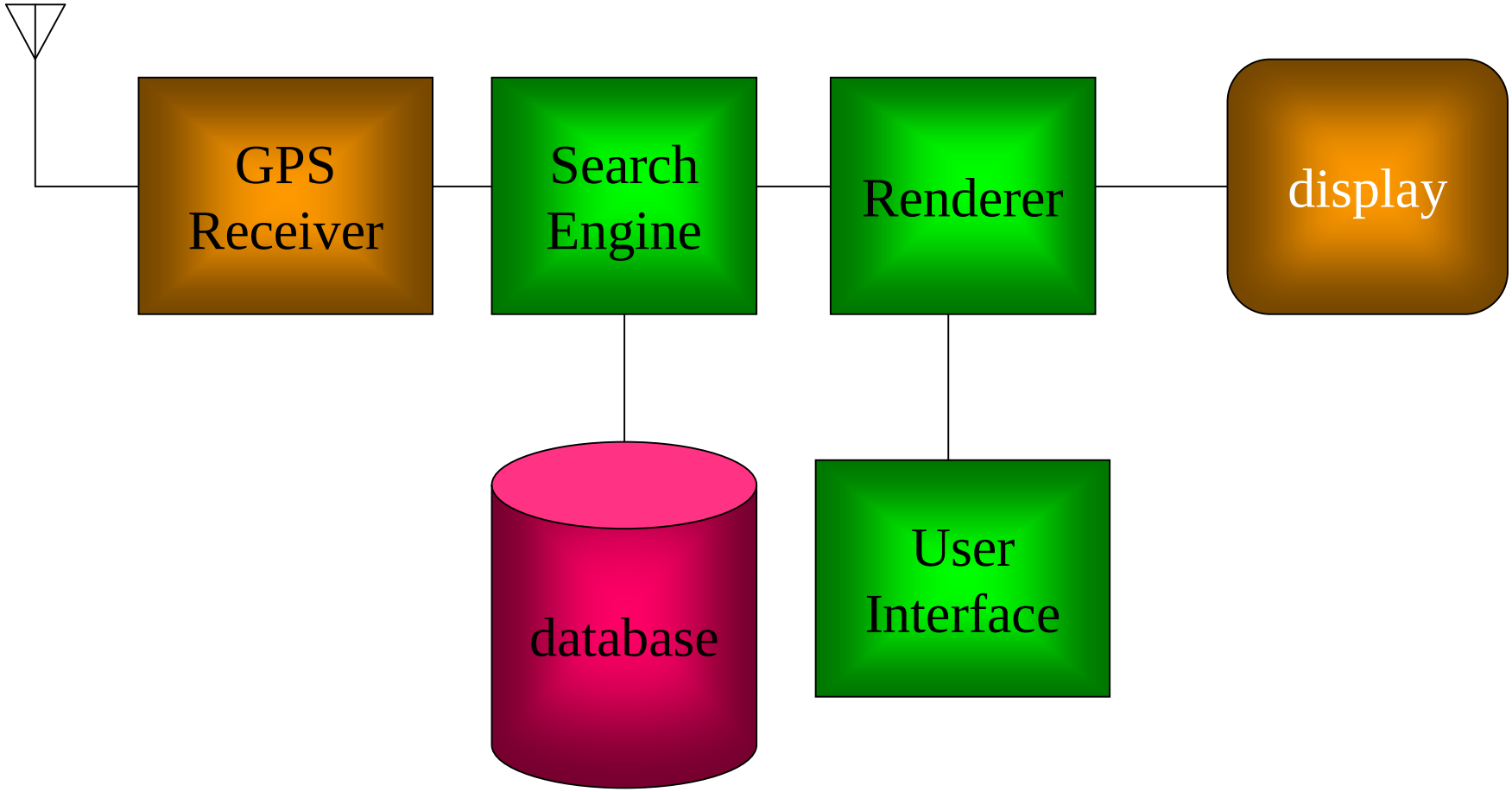
Example: GPS Specification

- A GPS specification should include several components
  - What is received from GPS
  - Map data
  - User interface
  - Operations required to satisfy user requests;
  - Background operations needed to keep the system running
    - Such as operating the GPS receiver

1.5.3 Architecture Design

- What major components go satisfying the specification?
- Hardware components:
  - CPUs, peripherals, etc.
- Software components:
  - Major programs and their operations.

Example: GPS Moving Map Block Diagram



1.5.4 Designing Hardware and Software Components (1/3)

- Must spend time *architecting the system* before you start coding.

- Some components are
  - *Ready-made*
    - CPU, memory chip
  - Some can be *modified from existing designs*
  - Others must be *designed from scratch*.

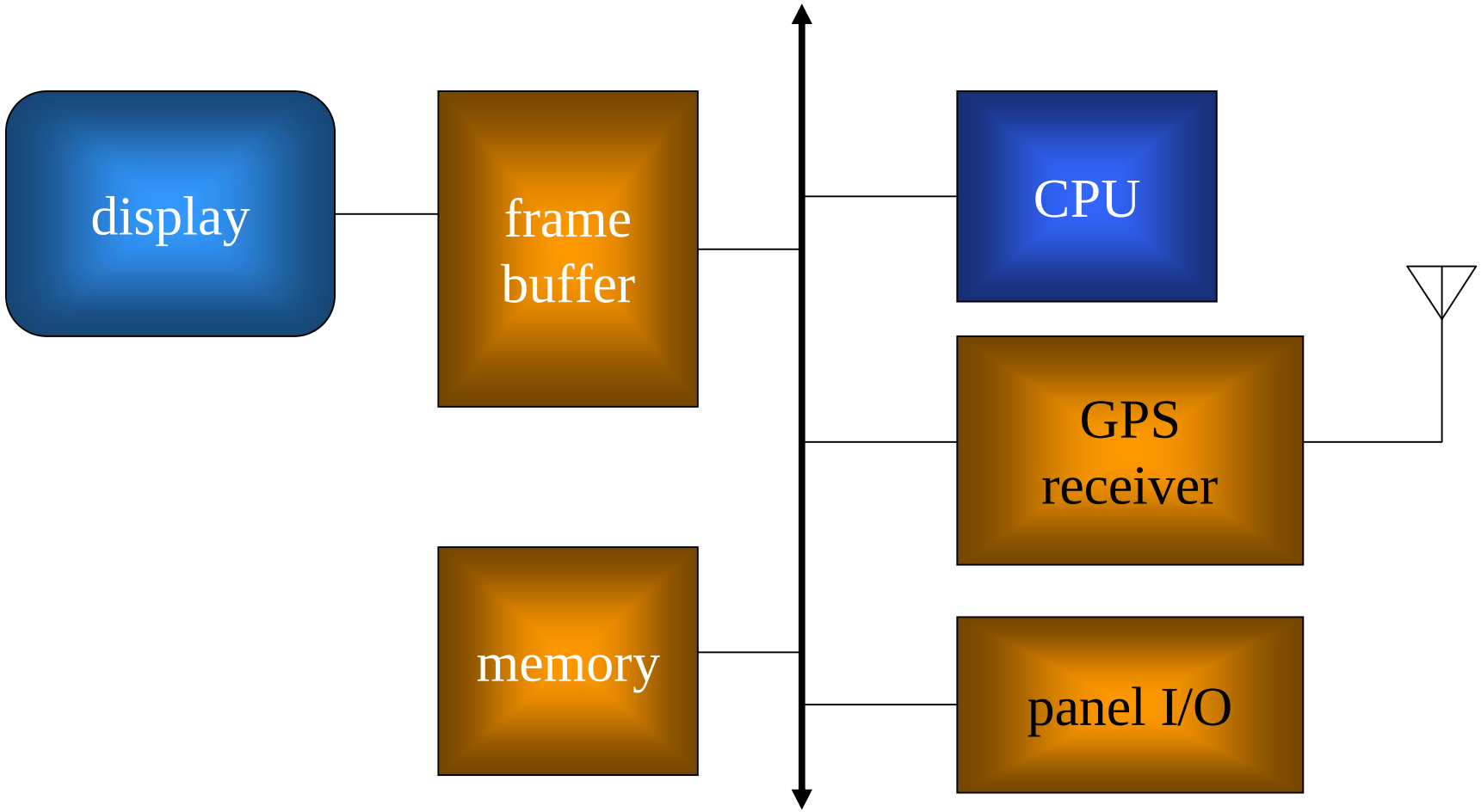
1.5.4 Designing Hardware and Software Components (2/3)

- Both the hardware and the software for an embedded system are developed in parallel
  - Software can take advantage of special hardware features to *gain performance*
  - Hardware can simplify design if functionality can be achieved in software to *reduce hardware complexity and cost*

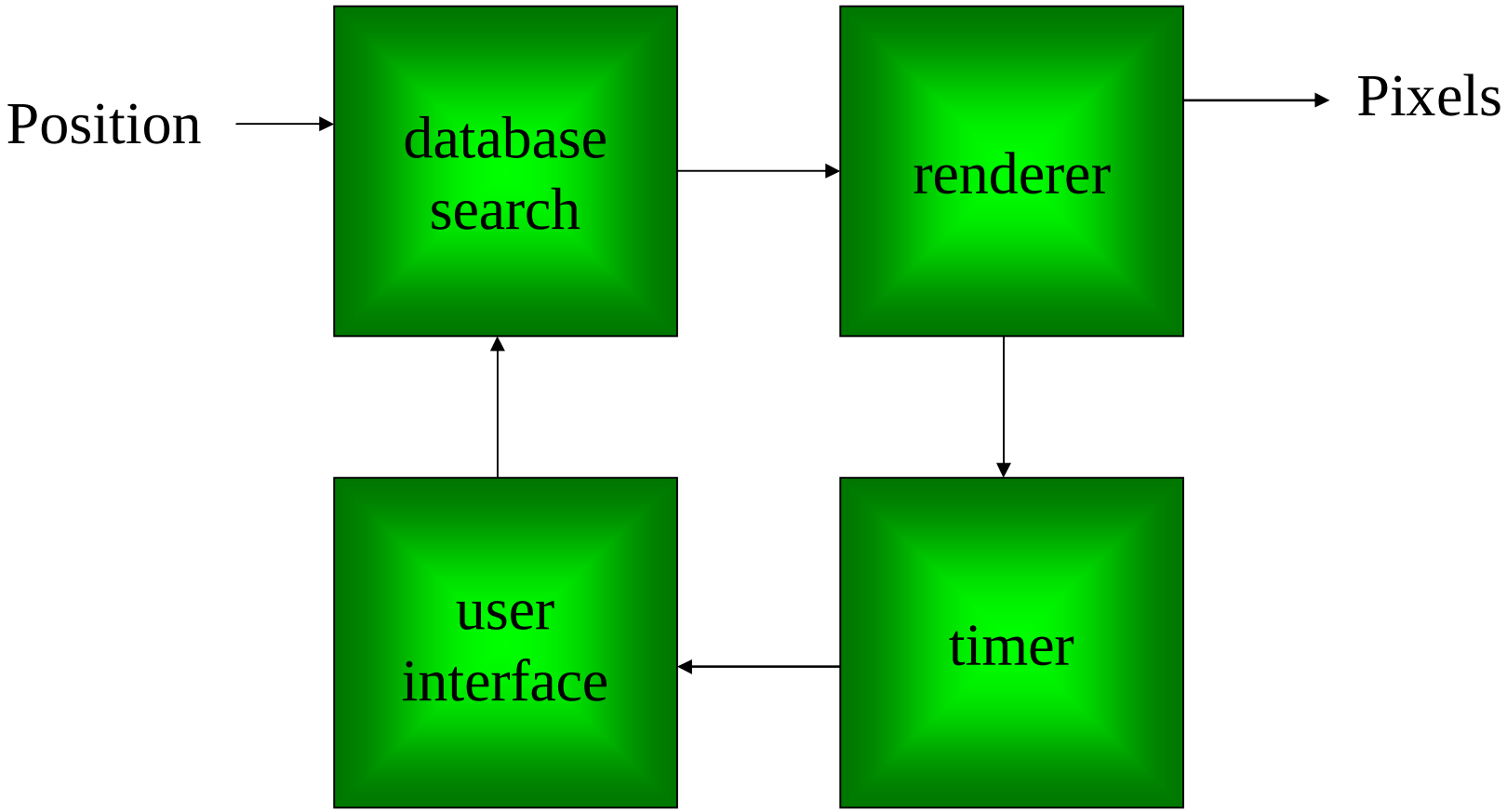
1.5.4 Designing Hardware and Software Components (3/3)

- Software development in embedded system: *cross-platform development*
  - Software, including system and application software, is developed on one platform but runs on another
  - *Platform*: a combination of hardware, operating system, and software development tools
  - *Host system*: the system on which the embedded software is developed
  - *Target system*: the embedded system under development
- *Cross-compiler*: the main tool in cross-platform development
  - A compiler that runs on one type of processor architecture but produces object code for a different type of processor architecture

GPS Moving Map Hardware Architecture



GPS Moving Map Software Architecture



1.5.6 System Integration

- Put together the components.
  - Many bugs appear only at this stage.

- Solutions
  - Have a plan for integrating components to uncover bugs quickly
  - Test as much functionality as early as possible.

Appendix: Software Storage and Upgradeability in Embedded Systems (1/2)

- Code for embedded system is commonly stored in ROM and NVRAM memory devices
- Depending on the memory devices used, upgrading an embedded system may need to
  - Build new PROM
  - Deploy special equipment and/or a special method to reprogram the EPROM
  - Reprogram the flash memory

Appendix: Software Storage and Upgradeability in Embedded Systems (2/2)

- Software storage device has an impact on software development
  - To reprogram an EPROM when small changes are made is tedious and time-consuming during software development
    - Must remove the EPROM device from its socket
- Software storage device also has an impact on maintenance
  - Must replace PROM and EPROM chips when upgrading an embedded system
  - However, flash memory or EEPROM can be upgraded dynamically without the need for chip replacement

Read Only Memory (ROM) (1/2)

- **Mask Programmed ROM**
  - Memory content is programmed during manufacturing process
  - Programmed once and cannot be changed
- **Field Programmed ROM (PROM)**
  - Memory content cannot change once programmed
- **Erasable Programmable ROM (EPROM)**
  - Can be custom-programmed, erased, and reprogramming as often as required
  - To reprogram: remove from its hosing unit and use a EPROM programmer

Read Only Memory (ROM) (2/2)

- **Electrically Erasable Programmable ROM (EEPROM or E^2PROM)**
  - Memory content of a single byte can be selectively erased and reprogrammed
  - Can be reprogrammed while staying in the device
    - Without a special programmer
- **Flash Memory**
  - A variation of EEPROM
  - But allow for block-level programmability that is much faster than EEPROM

Random Access Memory (RAM)

- **Dynamic RAM (DRAM)**
  - Require periodic refreshing to retain its content
- **Static RAM (SRAM)**
  - Faster than DRAM because it does not require periodic refreshing
- **Non-Volatile RAM (NVRAM)**
  - A special type of SRAM that has backup battery power
  - Or a combination of SRAM and EEPROM

Reference

- Qing Li and Caroline Yao, “Real-Time Concepts for Embedded Systems”, CMP Books, ISBN: 1-57820-124-1, 2003
- Wayne Wolf, “Computers as Components: Principles of Embedded Computing System Design,” Morgan Kaufman Publishers, 2001
- Silberschatz, Galvin, and Gagne, “Operating System Concepts”, John Wiley, 2002.
- Sixto Ortiz Jr., “Embedded OSs Gain the Inside Track,” IEEE Computer Magazine, November 2001, pp. 14~16