1. INTRODUCTION

Embedded Software Design

熊博安

國立中正大學資訊工程研究所

pahsiung@cs.ccu.edu.tw

Textbooks: Embedded Software Primer, David E. Simon, Addison Wesley
Programming Embedded Systems with C and GNU Development Tools, Barr, Massa, O'Reilly

What is an embedded system?

- Combination of computer hardware, software, and some mechanical parts,
- Designed to perform a specific function.
- Every household has one!
- Very few people realize that a processor and software are involved in the preparation of their **lunch** or **dinner**!!!
- General-purpose computer is not designed to perform a specific function. It is a blank slate!

What is an embedded system?

- Installed in a larger system
- Eg: cars and trucks contain many embedded systems
  - anti-lock brake controller
  - vehicle emission monitor and controller
  - dashboard information display
- Existence of processor & software should be completely unnoticed by a device user
- Eg: microwave oven, VCR, alarm clock

History and Future

- When did embedded systems first appear?
- Not before 1971! Why?
- Intel designed the world's first microprocessor, the 4004, in 1971!
- 4004 was designed for use in a line of business calculators produced by a Japanese company Busicom.
- In 1969, Busicom asked Intel to design a set of IC --- one for each of their new calculator models
- The 4004 was Intel's response.

History and Future

- The microprocessor was an overnight success!
- Increased use in the next decade:
  - unmanned space probes
  - computerized traffic lights
  - aircraft flight control systems
- In 1980's, embedded systems quietly rode the waves of microcomputer age.

History and Future

Electronic devices in

- kitchen
  - bread machines
  - food processors
  - microwave ovens
- living rooms
  - TV
  - stereos
  - remote controls
- offices
  - fax machines
  - pagers
  - laser printers
  - cash registers
  - credit card readers

History and Future

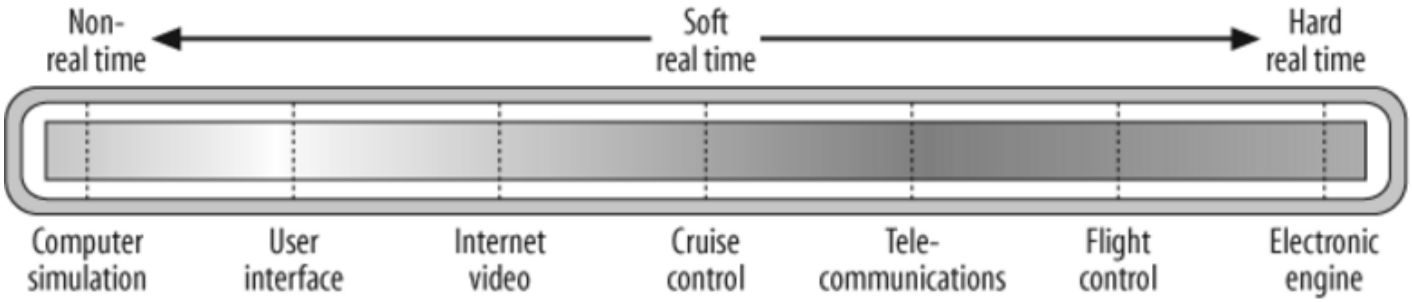
Embedded systems will continue to increase

- light switches and thermostats controlled by a central computer
- intelligent air-bag systems that don't inflate when children or small adults are present
- Personal Digital Assistants (PDA)
- digital cameras
- dashboard navigation systems

Real-Time Systems

- Most embedded systems are real-time
- Has timing constraints
- Make certain calculations or decisions in a timely manner
- A missed deadline is as bad as a wrong answer
- Consequences of a missed deadline:
  - severe → hard real-time
  - acceptable → soft real-time

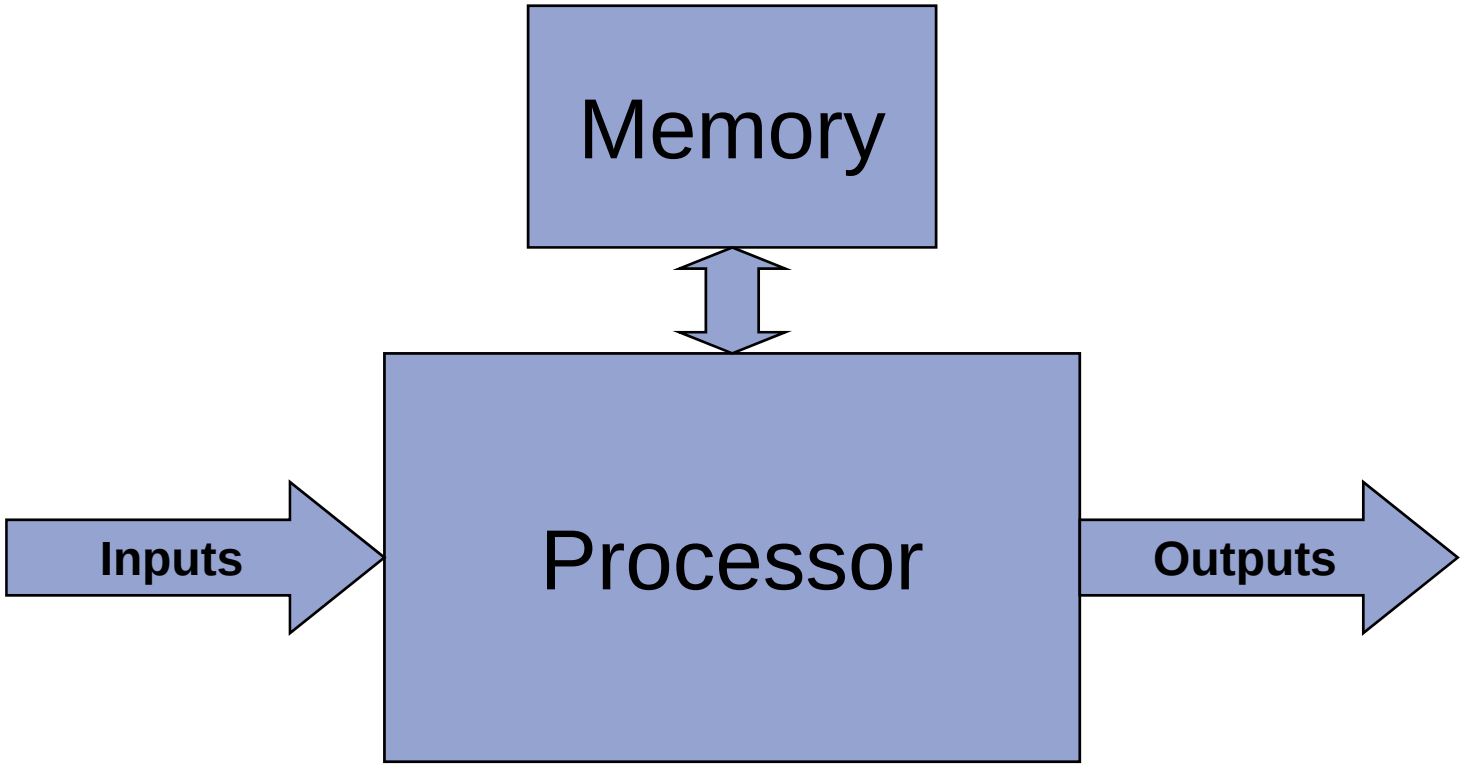
Real-Time Systems



Embedded System Variation

- Besides CPU and software, what else is common among embedded systems?
- Memory storage: ROM, RAM, …
- Input: knobs, buttons, probes, sensors, communication signals, …
- Output: human-readable display, microwave radiation, communication signals, changes to physical world
- Outputs = functions (inputs, elapsed time, current temperature, etc.)

Generic Embedded System



Common design requirements

- Production Cost
- Processing power
- Memory
- Development cost
- Number of units
- Expected lifetime
- Reliability

Common Design Requirements

Criterion	Low	Medium	High
Processor	4 or 8-bit	16-bit	32 or 64-bit
Memory	< 16 KB	64 KB~1 MB	> 1 MB
Development Cost	<$100,000	$100,000 ~ $1,000,000	> $1,000,000
Production Cost	<$10	$10 ~ $1,000	> $1,000
# Units	< 100	100~10,000	> 10,000
Power Consumption	> 10 mW/MIPS	1~10 mW/MIPS	< 1 mW/MIPS
Expected Lifetime	Days, weeks, months	Years	Decades
Reliability	May fail	Reliable	Fail-proof

Examples of Embedded Systems

- Digital Watch
- Telegraph
- Cordless Bar-Code Scanner
- Laser Printer
- Video Game Player
- Underground Tank Monitor
- Mars Explorer
- Nuclear Reactor Monitor

Digital Watch

- Function:
  - display date/time
  - measure event length to the nearest 1/100 s
- Simple tasks
- Small processing power or memory
- Then, why use a processor?
- Ans: to support a range of models & features from a single hardware

Digital Watch (contd)

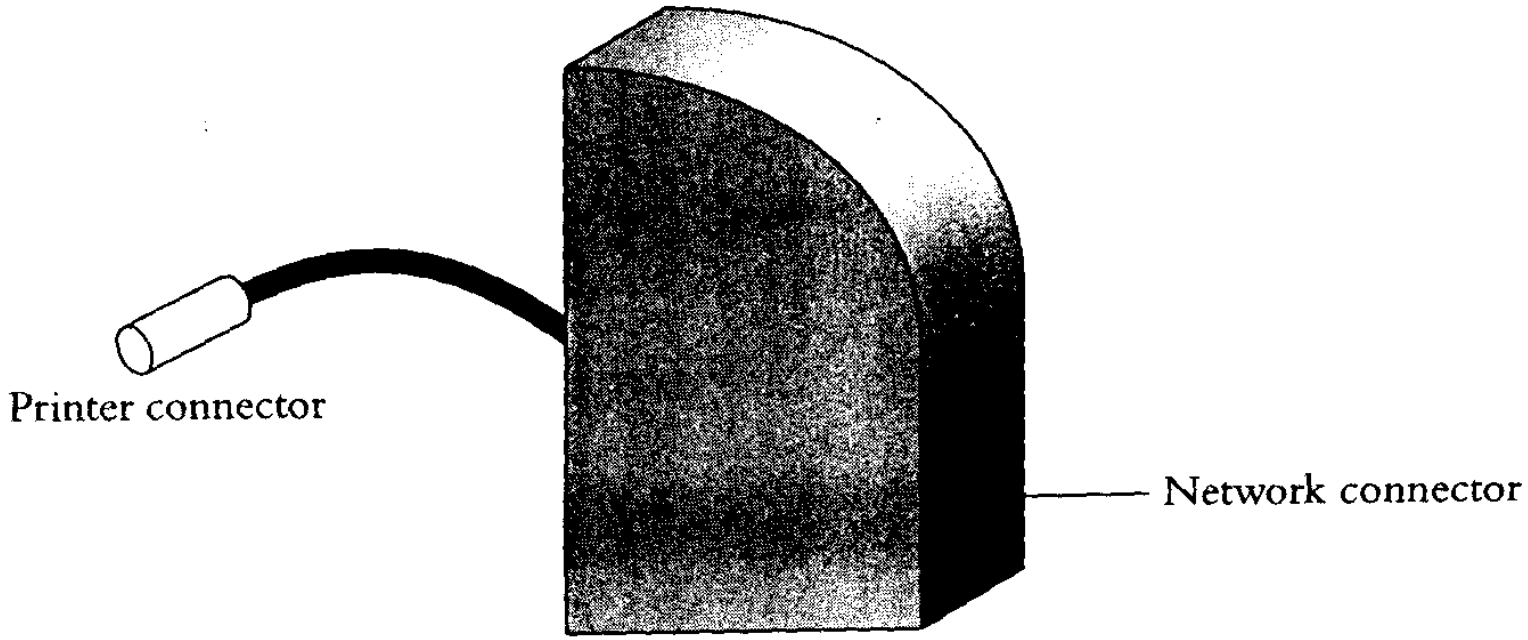
- Simple, inexpensive 8-bit processor
- On-chip ROM
- Only registers, no RAM
- Inputs: buttons
- Outputs: LCD and speaker
- Requirements:
  - High reliability
  - Low production cost

Telegraph

- Connects a printer to a network
- Printer has a high-speed serial port
- Telegraph description:
  - Little plastic box
  - 2 to 3 inches on a side
  - ½ inch thick
  - pigtail cable connects to printer serial port
  - network connector

Telegraph (sketch)

Figure 1.1 Telegraph



Telegraph (functions)

- Receive data from network
- Copy data to serial port of printer
- Sort unordered data packets and provide a clean data stream to printer
- Feed printer one print job at a time and hold off all other computers
- Network printer must provide status information to any requesting computer on network, even if it is busy printing

Telegraph (functions)

- Work with several types of printers without user configuration
- Respond rapidly to certain events: various kinds of network frames to which Telegraph must send response within 200 microseconds
- Must keep trace of time. If a computer crashes, must give up on that print job after 2 minutes and print from another computer. Otherwise, printer will be unavailable.

Telegraph Development Challenges

- Throughput
- Response
- Testability
- Debugability
- Reliability
- Memory Space
- Program Installation

Telegraph: Throughput

- Printer can print only as fast as Telegraph provides data to it
- Must not be a bottleneck between computer and printer
- throughput = run faster
- solution: clever programming
  - better searching and sorting
  - better numerical algorithms
  - data structures faster to parse

Telegraph: Response

- Response to frames within 200 microseconds
- Response is a common problem in embedded systems
- Tradeoff between
  - Throughput, and
  - Response

Telegraph: Testability

- Not easy to determine if it works
- Lot of software deals with uncommon events
- Embedded systems must deal with ANYTHING without human intervention
- Eg: lots of code deals with the problem of network data loss
- However, data does not get lost often, especially in a perfect, new lab
- Hard to test those lines of code

Telegraph: Debugability

- What if testing uncovers a bug?
- Telegraph has no screen, no keyboard, no speaker, not even little lights!
- No cute icons or message boxes!
- It just stops working!
- A bug in network software?
- A bug in software for tracking printing job?
- A bug in software for printer status reporting?

Telegraph: Reliability

- It is not allowed to crash!
- Customers may have tolerance for crash/reboot of PC, but nobody has patience for little plastic boxes that CRASH!
- Must function without human intervention

Telegraph: Memory Space

- 32 KB memory for program
- 32 KB memory for data
- How to make software fit into the available space?
- A necessary skill for embedded-system software engineers!

Telegraph: Program Installation

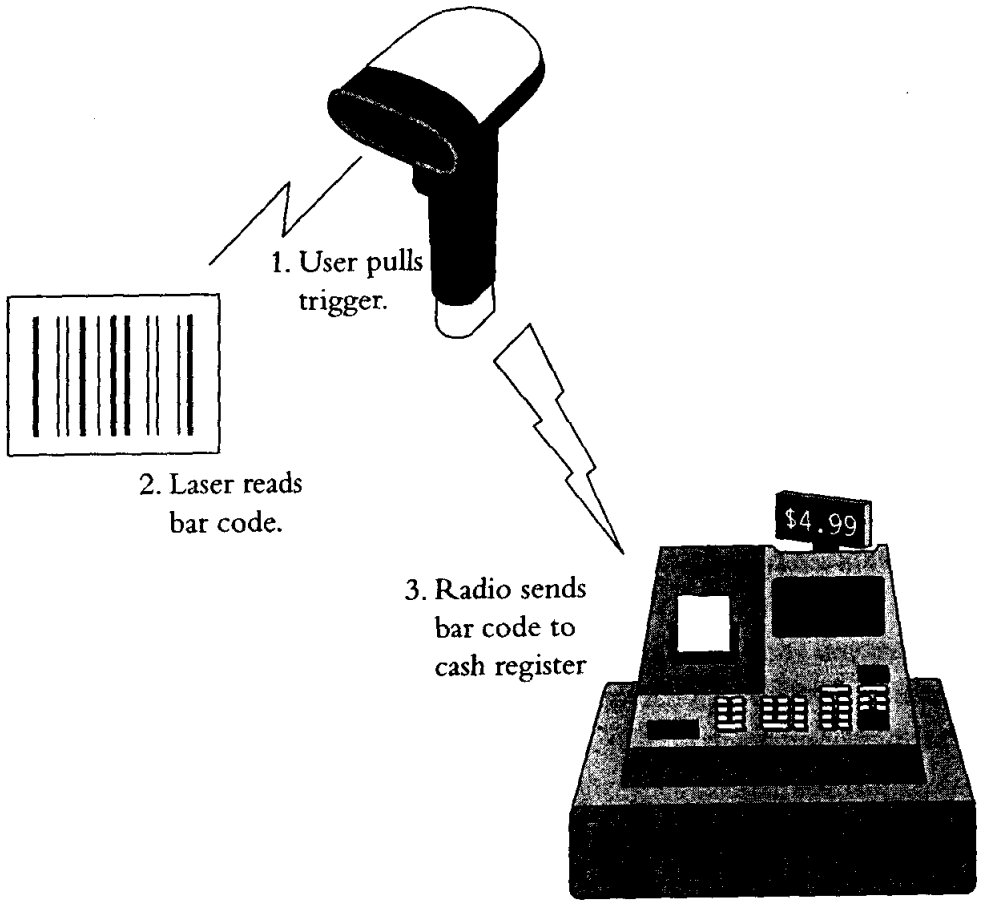
- The software in Telegraph did not get there because someone clicked a mouse on an icon!
- How to install software into embedded systems?
- What tools are necessary?

Cordless Bar-Code Scanner

- User pulls trigger
- Cordless Bar-Code Scanner activates laser to read bar code
- Sends bar code across a radio link to cash register

Cordless Bar-Code Scanner

Figure 1.2 Cordless Bar-Code Scanner



Cordless Bar-Code Scanner

- How different is its design from telegraph?
- Mostly same problems as telegraph
- No problem of throughput:
  - little data in a bar code
  - user can’t pull the trigger that fast
- One problem the telegraph does not have: Power Consumption

Cordless Bar-Code Scanner: Power Consumption

- Cordless → power source = battery
- Handheld → limited weight of battery (for comfortable use)
- How long must battery last?
- Forever!!! (Infeasible)
- Next best answer:
  - Last for an 8-hour shift
  - Recharge in holster at night

Cordless Bar-Code Scanner: Power Consumption

- 8-hours also not feasible!
- How to run laser, microprocessor, memory, and radio for 8 hours on battery?
- Solution: Use software to turn off hardware that are not needed at any given time, including processor!

Laser Printer

- High processing power
- Microprocessor responsible for
  - getting data from printer ports
  - sensing user button press on control panel
  - presenting messages to user on control panel
  - sensing paper jams
  - recovering from paper jams
  - noticing printer is out of paper
  - etc.

Laser Printer: Processor Hogs

- Print job:
  - text on a slanted line
  - unusual font
  - screwball size
- Figure out where the black dots go on a page!
- Users expect quick response when they push buttons, no concern of
  - trigonometric function value computations
  - where serifs of a rotated letter should go?

Video Game Player

- Some features are more powerful than PC
- High processing power
- Low production cost
- Companies don’t care how much it costs to develop the system
- but, production cost must be low (~US$100)!
- Even encourage engineers to design custom processors at hundreds of thousands dollars
- Highly specialized processor!

Video Game Player (contd)

- Production cost is crucial
- Tricks to shift costs around
- Move as much memory and other peripheral electronics as possible
  - **off** of the main circuit board and
  - **onto** the game cartridges!
- Powerful 64-bit CPU + few MB memory
- Enough to bootstrap the machine to a state from which additional memory on the game cartridges can be accessed

Underground Tank Monitor

- Watches gasoline levels in the underground tanks at a gas station
- Detect leaks before gas station turns into a toxic waste dump by mistake
- Set off a loud alarm if a leak is discovered
- System description
  - 16 buttons
  - 20-character Liquid Crystal Display (LCD)
  - Thermal printer

Underground Tank Monitor (contd)

- How much gasoline is in a tank?
- Read the level of two floats in the tank
  - level of gasoline
  - level of water at the bottom of tank
- Read temperature at various levels in tank (gasoline expands & contracts considerably with temperature changes)
- No false alarms (gasoline cooled off, contracted, float lowered → alarm?)

Underground Tank Monitor: Cost

- buys one only because government agency tells the gas station owner he has to!
- thus, as inexpensive as possible!
- Extremely inexpensive microcontroller: add 8-bit numbers
- Microprocessor will be very busy just calculating how much gasoline there is really down there → processor hog

Mars Explorer

- In 1976, two unmanned spacecraft arrived on the planet Mars.
- to collect samples of Martian surface, analyze chemical makeup, transmit results back to earth
- PC rebooted everyday
- BUT, 2 computers survived a journey of 34 million miles and functioned correctly for 5 years!!! RELIABILITY!!!

Mars Explorer

- NASA launched the Pathfinder
- Primary goal: getting to Mars on a budget
- Two embedded systems
  - a landing craft: 32-bit CPU, 128 MB RAM
  - a rover: 8-bit CPU, 512 KB RAM
- Low production cost

Mars Explorer

- What if a memory chip failed?
- Or, software bugs caused a crash?
- Fault tolerance:
  - Redundant circuitry
  - Extra functionality
  - Extra processor
  - Special memory diagnostics
  - Hardware timer to reset system if software got stuck

Nuclear Reactor Monitor

- Must do many things
- Only thing of interest to us:
  - two temperatures must be always equal
- If not, a malfunction!
- Consequence: disastrous!!!

Why C for embedded software?

- a “very-level” high-level language
- compact, efficient code for almost all processors
- direct hardware control, without losing the benefits of a high-level language
- appropriate for both 8-bit and 64-bit processors
- for systems with bytes, KB, MB of memory
- for design teams of 1, 12, or more people

Other embedded languages

- assembly language
- C++
- Ada

Assembly Language

- complete control of CPU and hardware
- high software development costs
- lack of code portability
- lack of skilled assembly programmers
- used as an adjunct to high-level languages, for small pieces of code that must be
  - extremely efficient or
  - ultra-compact, or
  - cannot be written in any other way.

C++

- better data abstraction
- reduce efficiency of executable programs
- more popular with large development teams, where the benefits to developers outweighs the loss of program efficiency

Ada

- Object-Oriented
- substantially different from C++
- designed by US Department of Defense
- for mission-critical military software development
- twice accepted as international standard (Ada83, Ada95)
- Not popular outside of defense and aerospace industries

Typical Hardware

- Microprocessor: execute code
- Memory: different memories for program & data
- Embedded systems do not have the following:
  - a keyboard
  - a screen
  - a disk drive
  - CD, speakers, microphones, diskettes, modems
- Embedded systems have: serial port, network interface, sensors, actuators, etc.

Microprocessors in embedded systems

Table 1.1 Microprocessors Used in Embedded Systems

Processor	Bus Width	Largest External Memory	Internal Peripherals	Speed (MIPS)
Zilog Z8 family	8	None on some models; 64 KB on others	2 timers	1
Intel 8051 family	8	64 KB program + 64 KB data	3 timers + 1 serial port	1
Zilog Z80 family	8	64 KB; 1 MB, sort of	Various	2
Intel 80188	8	1 MB	3 timers + 2 DMA channels	2
Intel 80386 family	16	64 MB	3 timers + 2 DMA channels + various others	5
Motorola 68000 family	32	4 GB	Varying	10
Motorola PowerPC family	32	64 MB	Many	75

Programming Embedded Systems in C and C++ 2nd Ed. (Design Platform)

- Arcom VIPER-Lite board
- Intel's 32-bit XScale ARM PXA255 processor
- 64MB RAM
- 16 MB ROM
- inputs
- outputs
- peripheral components

Arcom VIPER-Lite Board