

# What is an Embedded System?

What is an embedded system?

- An embedded system is *not* a microprocessor used in a “traditional” computing application
  - desktop
  - laptop
  - workstation

An embedded system is a microprocessor used as a *component* in another piece of technology

- cell phone
- digital camera
- portable digital assistant
- household appliance
- automobile antilock brake system
- . . . (how many can you name)?

Are embedded systems common? According to Tennenhouse [4], in the year 2000

- 150 million microprocessors used in “traditional” computers
- 8 billion microprocessors used in embedded applications!!

# Where are Embedded Systems?

Many articles emphasize the growing ubiquity of embedded microprocessors, including *networked* embedded processors. A few examples:

- The New York Times: “Honey, I Programmed the Blanket” [2]
- Communications of the ACM: “Embedding the Internet” [1]
- any issue of Embedded Systems Programming Magazine

Embedded systems in Michigan:

- automobiles (Ford, GM, and suppliers)
  - antilock braking systems
  - engine controls
  - active suspension
  - . . .
- manufacturing
- appliances (Whirlpool)
  - washers, dryers
  - toasters
  - . . .

# General Types of Embedded Systems

According to Koopman [3], there are 4 types of embedded systems

- General
  - similar to traditional computer systems, in a smaller package
  - PDA's
  - portable games
- Communications
  - cell phones
- Signal Processing
  - video and audio
- Control
  - real time feedback control
  - automotive
  - aerospace
  - appliances

# Characteristics of Embedded Systems

Embedded system applications are distinguished from traditional applications by many factors. A partial list includes

- cost
  - hardware (recurring engineering cost, pennies crucial in consumer market)
  - software (nonrecurring engineering cost, but may limit time-to-market!)
- Limits not usually present in desktop world:
  - memory
  - development environment
  - power consumption
  - operator interface
- reliability
- safety
- real time critical
- processor speed not as important as guaranteed response time
- interface to environment through sensors and actuators
- “hybrid” behavior – interaction of continuous dynamics with logic and finite state machines

Not all these issues are present in every embedded system...

# Skills Needed for Embedded Applications

An embedded system application involves a diverse set of skills that extend across traditional disciplinary boundaries, including

- computer hardware
- software
- algorithms
- interface electronics
- application domain

⇒ How to make engineering tradeoffs that extend across these boundaries?

# Embedded Systems and You

As engineers, it is very likely that you will

- design algorithms (control, signal processing) that will be implemented on embedded microprocessors
- design microprocessors to be used in embedded applications
- design software (e.g, RTOS) for the embedded market
- work in application fields that involve an embedded microprocessor
- develop sensors/actuators (e.g., MEMS devices) that may be used in embedded systems

⇒ It is *certain* that you will encounter embedded systems in all aspects of your daily life!

## References

- [1] D. Estrin, R. Govindan, and John Heideman. Embedding the internet. *Communications of the ACM*, 43(5):39–41, May 2000.
- [2] K. Hafner. Honey, I programmed the blanket. *The New York Times*, May 1999.
- [3] P. Koopman. Embedded systems in the real world. Carnegie Mellon University, April 1998.
- [4] D. Tennenhouse. Proactive computing. *Communications of the ACM*, 43(5):43–50, May 2000.