Introduction to Embedded Systems

Jin-Soo Kim (jinsookim@skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Embedded Systems Everywhere
What are Embedded Systems?
Definition

- **Embedded System (ES):** any device that includes a programmable computer but is not itself a general-purpose computer.

  - Take advantage of application characteristics to optimize the design
Embedding a Computer

- CPU
- input
- mem
- output
- analog
- embedded computer
Where are the CPUs?

- Estimated 98% of 8 billion CPUs produced in 2000 used for embedded applications
- Smartphone shipments (101M) surpass PCs (2010Q4)

Source: DARPA/Intel (Tennenhouse)
Embedded Processors

- **Microcontroller (μC or MCU)**
  - A small computer on a single IC containing a processor core, memory, and I/O peripherals

- **Microprocessor**
  - A general-purpose CPU in a single chip

- **SoC (System-on-a-Chip)**
  - More integration than MCU
  - Mostly, require external memory
Early History (1)

- MIT Whirlwind computer (Late 1940’s)
  - Originally designed to control a flight simulator for training bomber crews
  - The first computer that operated in real time
  - 5000 vacuum tubes
Early History (2)

- Intel 4004 (1971)
  - The first microprocessor (4-bit)
  - Originally designed for use in a calculator
  - The first complete CPU on one chip
  - The first commercially available microprocessor
  - 2300 transistors @ 108KHz
Early History (3)

- Automobiles used microprocessor-based engine controllers starting in 1970’s
  - Control fuel/air mixture, engine timing, etc.
  - Multiple modes of operation: warm-up, cruise, hill climbing, etc.
  - Provides lower emissions, better fuel efficiency
Keyboard
Mouse
Hard Disk Drive
Digital Still Camera

Canon EOS3 uses three microprocessors for auto-focus, etc. (1998)
iPhone 5S

- Apple M7 Co-processor
- Apple A7 Application Processor & 1GB LPDDR3 DRAM
- Qualcomm MDM9615M LTE Modem
- Qualcomm WTR1605L LTE/HSPA+/CDMA2K/TDSCDMA/EDGE/GPS Transceiver
- Apple 338S1216 Power Management IC
- SK Hynix 16GB NAND Flash
- Qualcomm PM8018 RF Power Management IC
- Broadcom BCM5976 Touchscreen Controller
- TI 343S0645 Touchscreen Interface
- TriQuint TQM6M6224 Dual-band PA duplexer

Apple A7 Application Processor & 1GB LPDDR3 DRAM

Qualcomm MDM9615M LTE Modem

TriQuint TQM6M6224 Dual-band PA duplexer
Digital TV

Programmable CPUs + hardwired logic for video/audio decode, etc.
Automobile

- A high-end automobile
  - > 100 microprocessors
  - 4-bit microcontroller checks seat belt
  - Microcontrollers run dashboard devices
  - 16/32-bit microprocessor controls engine
Fitbit Flex

STMicroelectronics 32L151C6
Ultra Low Power ARM Cortex M3
Microcontroller

Charger IC: TI BQ24040

Accelerometer IC?

Nordic Semiconductor nRF8001
Bluetooth Low Energy IC

Source: https://www.ifixit.com/Teardown/Fitbit+Flex+Teardown/16050
DJI Phantom 2 Drone

**Phantom 2 Vision Plus**
- Maximum flight speed: 33.5 m.p.h.
- Maximum ascent speed: 13.4 m.p.h.
- Body width: 11.4 in.
- Weight: 2.7 lbs.

**PROPELLERS**
To stabilize motion, two propellers spin clockwise (black hub), and two spin counterclockwise (gray).

**GPS RECEIVER**
On the underside of the shell, it determines the position and height of the quadcopter.

**FLIGHT CONTROLLER**
Acts as the brains of the Phantom when it is in the air. Contains a gyroscope and an accelerometer.

**COMPASS** (not shown)
Gathers geomagnetic information that helps the GPS calculate the drone's position and height.

**LANDING GEAR**

**3-AXIS GIMBAL**

**CAMERA**

**VIDEO TRANSMITTER**
Sends an HD video signal to the remote control and app.

**PROPELLER MOTOR**

**ELECTRONIC SPEED CONTROLLER**
One for each motor. Controls the speed and direction of how the propeller spins. Also controls the onboard LED lights.

Solid State Drive (SSD)
Want More?

- **You name it!**

<table>
<thead>
<tr>
<th>Category</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anti-lock brakes</td>
<td>Modems</td>
</tr>
<tr>
<td>Auto-focus cameras</td>
<td>MPEG decoders</td>
</tr>
<tr>
<td>Automatic teller machines</td>
<td>Network cards</td>
</tr>
<tr>
<td>Automatic toll systems</td>
<td>Network switches/routers</td>
</tr>
<tr>
<td>Automatic transmission</td>
<td>On-board navigation</td>
</tr>
<tr>
<td>Avionic systems</td>
<td>Pagers</td>
</tr>
<tr>
<td>Battery chargers</td>
<td>Photocopiers</td>
</tr>
<tr>
<td>Camcorders</td>
<td>Point-of-sale systems</td>
</tr>
<tr>
<td>Cell phones</td>
<td>Portable video games</td>
</tr>
<tr>
<td>Cell-phone base stations</td>
<td>Printers</td>
</tr>
<tr>
<td>Cordless phones</td>
<td>Satellite phones</td>
</tr>
<tr>
<td>Cruise control</td>
<td>Scanners</td>
</tr>
<tr>
<td>Curbside check-in systems</td>
<td>Smart ovens/dishwashers</td>
</tr>
<tr>
<td>Digital cameras</td>
<td>Speech recognizers</td>
</tr>
<tr>
<td>Disk drives</td>
<td>Stereo systems</td>
</tr>
<tr>
<td>Electronic card readers</td>
<td>Teleconferencing systems</td>
</tr>
<tr>
<td>Electronic instruments</td>
<td>Televisions</td>
</tr>
<tr>
<td>Electronic toys/games</td>
<td>Temperature controllers</td>
</tr>
<tr>
<td>Factory control</td>
<td>Theft tracking systems</td>
</tr>
<tr>
<td>Fax machines</td>
<td>TV set-top boxes</td>
</tr>
<tr>
<td>Fingerprint identifiers</td>
<td>VCR’s, DVD players</td>
</tr>
<tr>
<td>Home security systems</td>
<td>Video game consoles</td>
</tr>
<tr>
<td>Life-support systems</td>
<td>Video phones</td>
</tr>
<tr>
<td>Medical testing systems</td>
<td>Washers and dryers</td>
</tr>
</tbody>
</table>

And the list goes on and on …
Challenges
ES Characteristics (1)

- Single-functioned
  - Executes a single program, repeatedly

- Sophisticated functionality
  - Often have to run sophisticated algorithms or multiple algorithms
    - Cell phone, laser printer, digital TV, etc.
  - Often provide sophisticated user interfaces
ES Characteristics (2)

- Reactive and real-time operation:
  Must finish operations by deadlines
  - Continually reacts to changes in the systems environment
  - **Hard real-time**: missing deadline causes failure
  - **Soft real-time**: missing deadline results in degraded performance
  - Many systems are multi-rate:
    Must handle operations at widely varying rate
ES Characteristics (3)

- Low cost
  - Manufacturing cost
    - The monetary cost of manufacturing each copy
  - NRE (Non-Recurring Engineering) cost
    - The one-time monetary cost of designing the system
  - Many embedded systems are mass-market items that must have low manufacturing cost
  - Limited memory, microprocessor power, etc.
ES Characteristics (4)

- **Low power**
  - Power consumption is critical in battery-powered devices
  - Excessive power consumption increases system cost even in wall-powered devices
ES Characteristics (5)

- Designed to tight deadlines by small teams
  - Often designed by a small team of designers
  - Often must meet tight deadlines
    - 6-month time-to-market is common
    - Can’t miss back-to-school window for calculator
  - Many design alternatives
  - Hard to develop and debug
Challenges in ES Design (I)

▪ How much hardware do we need?
  • Powerful CPU? Big memory?

▪ How do we meet our deadlines?
  • Faster hardware or cleverer software?

▪ How do we minimize power?
  • Turn off unnecessary logic? Reduce memory accesses?

▪ How do we design for upgradeability?
  • e.g. Evolution Kit in Samsung Digital TVs

▪ Time-to-market?
Challenges in ES Design (2)

▪ Does it really work?
  • Is the specification correct?
  • Does the implementation meet the spec?
  • How do we test for real-time characteristics?
  • How do we test on real data?
  • Does it work reliably?

▪ How do we work on the system?
  • Limited observability and controllability
  • Restricted development environments
Challenges in ES Design (3)

- Optimizing design metrics
  - Improving one may worsen others
  - Expertise with both software and hardware is needed to optimize design metrics
  - A designer must be comfortable with various technologies in order to choose the best for a given application and constraints