EE 472 – Embedded Systems

Dr. Shwetak Patel
Assistant Professor
Computer Science & Engineering
Electrical Engineering

Class logistics

- TTH 12:30am-2:20pm
- EE 054, Lab: EE 345
- Class website: http://abstract.cs.washington.edu/~shwetak/classes/ee472

Class logistics

- Contact Information:
  - shwetak@cs.washington.edu
  - www.shwetak.com
  - Office: CSE 540
- Grad TA:
  - Morgan Redfield
- Volunteers:
  - Andrew Karpenko
  - Tom Sommerville

Syllabus
My Research and Background

- Human-Computer Interaction (HCI)
  - Applications and technology to improve peoples lives

- Ubiquitous Computing (ubicomp)
  - Making computing seamless
  - New sensing techniques
  - Embedded Systems

- User Interface Technologies
  - Novel interaction techniques

Index Cards

- Name
- Major, year in program
- Specialization (areas of interest)
- Programming experience
  - What classes, what languages, comfort level
  - What’s your interest in embedded systems

Introduction to Embedded Systems

- An embedded system is a microcomputer system comprising hardware and software
  - Designed and optimized to solve specific problem very efficiently

Various Platforms

- Microprocessor (ex. Arm 7 proc)
- Microcomputer (ex. Make controller)
- Microcontroller (ex. MSP 430, Pic)
Platforms

Examples

- Cars contain as many as 100 embedded microprocessors and microcontrollers
  - Engine ignition, transmission, anti-lock brakes, safety
- Consumer electronics
  - Mobile phones, routers, mouse, keyboard, appliances

Need for Embedded Systems

- Microcontrollers.com: “In the aggregate, PC microprocessors are responsible for less than 1% of all processors sold. Embedded processors outsell PC processors by more than 99%.”
- Huge demand for embedded engineers!
**Time**

- Many embedded systems are considered to be real time systems
- Must respond within constrained time interval to external or internal events

**Real Time System**

- **Soft real time system**
  - If time constraint not met performance degraded
- **Hard real time system**
  - If time constraint not met system considered to have failed
  - Failure may be considered to be catastrophic
  - Use real time operating system (RTOS)

**Programming**

C Compilation Process

- Source -> Preprocessor -> Compiler -> Assembler -> Linker -> Loader
Example Program in C

```c
#include <stdio.h> // this imports a definition file (a header).

int main() // note the return type of main: 0 means ok
{
    int x=5;
    printf("local = %d", x);
    return 0; // return success
}
```

C types (java vs c)

<table>
<thead>
<tr>
<th>Type</th>
<th>Java Bits</th>
<th>C Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>8 bits</td>
<td>8 bits</td>
</tr>
<tr>
<td>short</td>
<td>16 bits</td>
<td>16 bits</td>
</tr>
<tr>
<td>int</td>
<td>32 bits</td>
<td>16, 32, or 64 bits</td>
</tr>
<tr>
<td>long</td>
<td>64 bits</td>
<td>32 or 64 bits</td>
</tr>
<tr>
<td>float</td>
<td>32 bits</td>
<td>32 bits</td>
</tr>
<tr>
<td>double</td>
<td>64 bits</td>
<td>64 bits</td>
</tr>
<tr>
<td>boolean</td>
<td>1 bit</td>
<td>1 bit or int</td>
</tr>
<tr>
<td>byte</td>
<td>8 bits</td>
<td>use char</td>
</tr>
<tr>
<td>long long</td>
<td>N/A</td>
<td>64 bits</td>
</tr>
<tr>
<td>long double</td>
<td>N/A</td>
<td>80, 96 or 128 bits</td>
</tr>
</tbody>
</table>

Questions?