CSE 466: Software for Embedded Systems

Autumn, 2003

Instructor: Bruce Hemingway
Office: CSE 464
Office Hours: MW 11:30 A.M.-12:30 P.M.
e-mail: bruceh@cs.washington.edu
Phone: (206) 543-6274

Teaching Associate: Waylon Brunette
Teaching Assistants: William Welbourne and Tom Anderl

Class Meeting Times and Location:
• Lectures: MWF 9:30-10:20 A.M. LOWE 102
  Lab: Tues. Section 1, 2:30- 5:20 P.M. CSE 003
  Thurs. Section 2, 2:30- 5:20 P.M. CSE 003

What is an Embedded System?

- It's not a desktop system
  - Fixed or semi-fixed functionality (not user programmable)
  - Lacks some or all traditional human interfaces: screen, keyboard, pointing device, audio
  - May have stringent real-time requirements (Hard and Soft)
  - Usually has sensors and actuators for interface to physical world
- It may:
  - replace discrete logic circuits
  - provide feature implementation path
  - Make maintenance easier
  - Protect intellectual property
  - Improve mechanical performance
  - Replace analog circuits

What is an Embedded System

- Figures of Merit for embedded systems
  - Reliability – it can never crash
  - Safety – involves things that move and can harm/kill a person
  - Power Consumption – may run on limited power supply. Want slowest possible clock, least amount of memory. You will always be resource constrained!
  - Cost – Engineering Cost, Mfg Cost, Schedule tradeoffs
  - Product life cycle issues: maintainability, upgradeablility, serviceability
  - Performance

“To Have and Have Not” ...

- We don’t have
  - User Interface
  - Dynamic Linking and Loading
  - Virtual Memory, Protection Modes
  - Disk
  - Processes
- Instead we have
  - Real Time Kernel (very small OS) (If we’re lucky)
  - Tasks (threads)
  - Task communication primitives
  - ADC
  - Timers
  - Event Capture
  - PWM
**An Example: Temp Controller w/ AVR MCU**

- **Task: Tachometer (external interrupt)**
  - `now = getTime();`
  - `period = then - now; //overflow?`
  - `then = now;`
  - `return;`

- **Task: TempControl (periodic, soft constraint)**
  - `if (Temp > setpoint) Thi++;`
  - `if (Temp < setpoint) Thi--;`
  - `if (period<min || period>max) PD0 = 1;`

- **Task: FanPWM (periodic, hard constraint)**
  - `count++;`
  - `if (count == 0) PD6 = 1;`
  - `if (count > Thi) PD6 = 0;`
  - `return;`

- **Task: Main**
  - `Thi = 0;`
  - `setup timer for 1ms interrupt;`
  - `setup timer for 10ms interrupt;`
  - `while (1) ;`

---

**Resource Analysis of Temp Controller**

<table>
<thead>
<tr>
<th>Task</th>
<th>Instructions/Sec</th>
<th>RAM</th>
<th>ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tach</td>
<td></td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>FanPWM</td>
<td></td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>TempControl</td>
<td></td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

**Total Instructions/Sec = 8260**, at 8MIPS, that’s .1% utilization

**Class and Lab Policies**

- **Lecture**
  - See Syllabus and Schedule. Generally coordinated with design problems
  - Mondays– this week’s lab assignment
  - Wednesdays– background and some theory
  - Fridays– discuss lab and more background for next lab

- **Lab**
  - Implementation of the design, as specified in class
  - Lab reports due prior to start of next lab section (2:30pm)

- **Exams**
  - Two, based on lecture, lab, and reading

- **No Final MAYBE– Final Group Project Participation Required**

- **Reading and Source Material assigned as needed**

---

**Capacity**

- **Assume:**
  - 8 MHz processor @ one instruction/cycle
  - Assume fan runs between 30Hz and 60Hz
  - Assume 256ms period on speed control PWM, with 1ms resolution.

- **What percent of the available cycles are used for the temperature controller?**
  - \([\text{total instructions in one second}] / (8 \text{m I/sec})\)

- **How much RAM do you need?**

- **How much ROM?**
**Business Matters**

- Lecture slides will be handed out and online after class.
- Go to the 466 schedule link for links to lecture slides, labs, etc.
- If you have a home PC, get and use the tools!
- The documents:
  - Atmel CD-Rom Data Books
  - ATmega16 Datasheet—on CD, on web, in course pak
  - Prototyping with the Design Kit on web
  - HWLab web page docs
- “Lab equipment required for the duration of a course or project must be first checked out from the Lab Manager and secured with a deposit check of $200 made payable to "University of Washington" (note that this check will not be cashed but will be returned to the student upon the return of all checked-out equipment in good condition.)” from lab policy...
- Sign up for CSE466 mailing list

---

**Grading**

- Lab reports: Demo required, sometimes Report, sometimes hand-in code
- Ratios:
  - Lab: 50%
  - Exams total: 40%
  - Class and Lab Participation: 10%

---

**CSE466 Syllabus-1**

- The course will focus on software issues in embedded systems including use of an advanced 8-bit microcontroller and its development environment, interrupt programming and management, and peripheral interfacing and drivers.
- Laboratory assignments will use prototyping boards, LEDs, audio transducers, A/D converters, pulse-width modulators, wireless communications, Berkeley Motes and TinyOS.
- **Required Readings:** We’ll be using the Atmel ATmega16 microprocessor extensively, and you will refer to the datasheet, which is found at: [http://www.atmel.com/mote/acspar/doc2466.pdf](http://www.atmel.com/mote/acspar/doc2466.pdf). It is 323 pages.
- Atmel has made available CD-ROMs of their data and application notes, which will be available for each student at the first lecture. The CD-ROM includes copies of the assembly tools as well.
- Hardbound copies of the datasheet as a coursepak are available at the Communication Copy Center in the Communications bldg, Rm B-042, cost $21.45.

---

**CSE466 Syllabus-2**

- Introduction: What is an Embedded System.
- AVR Development Tools
- Reading the AVR datasheet
- The Rule of (Ohm’s) Law
- Timers, Interrupts, A/D converters
- Interrupt-driven Task Structures
- Pulse Width Modulation & DACs
- Table-driven Wave Synthesis
- Event-driven OS: programming- TinyOS
- Wireless networking- using Motes
- Debugging tools: Logic analyzer
- Safety, Ethics, and Societal Impact
- Design Trade-offs: Memory, Speed, Power, Cost
- Serial Interfaces: SPI, I2C, USB
**Family of Motes**

<table>
<thead>
<tr>
<th>Mote Type</th>
<th>Wet</th>
<th>Rover</th>
<th>Mica</th>
<th>Mica2</th>
<th>Mica2Dot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller</td>
<td>AT04L5655S2</td>
<td>ATmega163</td>
<td>ATmega128</td>
<td>ATmega128</td>
<td>ATmega128</td>
</tr>
<tr>
<td>CPU Clock (MHz)</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>7.3728</td>
<td>4</td>
</tr>
<tr>
<td>Program Memory (kB)</td>
<td>8</td>
<td>16</td>
<td>128</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>Ram (kB)</td>
<td>0.5</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>UARTS</td>
<td>1</td>
<td>1</td>
<td>2 (one 1-meth)</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>SPI</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>EEPROM</td>
<td>Software</td>
<td>Software</td>
<td>Software</td>
<td>Hardware</td>
<td>Hardware</td>
</tr>
</tbody>
</table>

**Radio Communication**

- **Radio**
  - 802.15.4 (single freq)
  - 868/916 MHz (multiple channels)
- **Frequenzy**
  - 916 MHz
- **Radio speed (kbps)**
  - DSSS
- **Transmit Power Control**
  - Programmable via radio registers
- **Encoding**
  - Manchester (hardware)
  - Packet (software)

---

**Lab 1**

- Compile the code and download
- Blink the 7-segment display